

Isrep17 Documentation

AdventureWorks

Server	Isrep17
Author	author
Created	28 kwietnia 2017 15:48:55
File Path	C:\Users\Magda\Documents\Dokumentacja\Isrep17_documentation.pdf

Table of Contents

Table of Contents.....	2
Isrep17	6
User databases.....	8
AdventureWorks Database	9
Tables	12
[dbo].[AWBuildVersion].....	15
[dbo].[DatabaseLog]	17
[dbo].[ErrorLog].....	19
[HumanResources].[Department]	22
[HumanResources].[Employee]	24
[HumanResources].[EmployeeDepartmentHistory]	31
[HumanResources].[EmployeePayHistory].....	35
[HumanResources].[JobCandidate].....	38
[HumanResources].[Shift].....	41
[Person].[Address]	43
[Person].[AddressType]	47
[Person].[BusinessEntity].....	50
[Person].[BusinessEntityAddress]	52
[Person].[BusinessEntityContact]	56
[Person].[ContactType]	60
[Person].[CountryRegion]	62
[Person].[EmailAddress]	64
[Person].[Password].....	67
[Person].[Person].....	70
[Person].[PersonPhone]	77
[Person].[PhoneNumberType]	80
[Person].[StateProvince].....	82
[Production].[BillOfMaterials]	86
[Production].[Culture]	91
[Production].[Document]	93
[Production].[Illustration]	98
[Production].[Location]	100
[Production].[Product]	103
[Production].[ProductCategory]	111
[Production].[ProductCostHistory]	114
[Production].[ProductDescription]	117
[Production].[ProductDocument].....	119
[Production].[ProductInventory]	121
[Production].[ProductListPriceHistory]	125
[Production].[ProductModel]	128

	[Production].[ProductModelIllustration]	131
	[Production].[ProductModelProductDescriptionCulture]	133
	[Production].[ProductPhoto]	136
	[Production].[ProductProductPhoto]	138
	[Production].[ProductReview]	141
	[Production].[ProductSubcategory]	145
	[Production].[ScrapReason]	148
	[Production].[TransactionHistory]	150
	[Production].[TransactionHistoryArchive]	154
	[Production].[UnitMeasure]	157
	[Production].[WorkOrder]	159
	[Production].[WorkOrderRouting]	165
	[Purchasing].[ProductVendor]	170
	[Purchasing].[PurchaseOrderDetail]	175
	[Purchasing].[PurchaseOrderHeader]	182
	[Purchasing].[ShipMethod]	189
	[Purchasing].[Vendor]	192
	[Sales].[CountryRegionCurrency]	197
	[Sales].[CreditCard]	200
	[Sales].[Currency]	202
	[Sales].[CurrencyRate]	204
	[Sales].[Customer]	207
	[Sales].[PersonCreditCard]	211
	[Sales].[SalesOrderDetail]	213
	[Sales].[SalesOrderHeader]	220
	[Sales].[SalesOrderHeaderSalesReason]	231
	[Sales].[SalesPerson]	233
	[Sales].[SalesPersonQuotaHistory]	238
	[Sales].[SalesReason]	241
	[Sales].[SalesTaxRate]	243
	[Sales].[SalesTerritory]	246
	[Sales].[SalesTerritoryHistory]	251
	[Sales].[ShoppingCartItem]	254
	[Sales].[SpecialOffer]	257
	[Sales].[SpecialOfferProduct]	261
	[Sales].[Store]	264
	Views	268
	[HumanResources].[vEmployee]	269
	[HumanResources].[vEmployeeDepartment]	271
	[HumanResources].[vEmployeeDepartmentHistory]	273
	[HumanResources].[vJobCandidate]	275

	[HumanResources].[vJobCandidateEducation]	277
	[HumanResources].[vJobCandidateEmployment]	279
	[Person].[vAdditionalContactInfo]	281
	[Person].[vStateProvinceCountryRegion]	284
	[Production].[vProductAndDescription]	286
	[Production].[vProductModelCatalogDescription]	288
	[Production].[vProductModelInstructions]	292
	[Purchasing].[vVendorWithAddresses]	294
	[Purchasing].[vVendorWithContacts]	296
	[Sales].[vIndividualCustomer]	298
	[Sales].[vPersonDemographics]	301
	[Sales].[vSalesPerson]	303
	[Sales].[vSalesPersonSalesByFiscalYears]	306
	[Sales].[vStoreWithAddresses]	308
	[Sales].[vStoreWithContacts]	310
	[Sales].[vStoreWithDemographics]	312
	Stored Procedures	314
	[dbo].[uspGetBillOfMaterials]	315
	[dbo].[uspGetEmployeeManagers]	317
	[dbo].[uspGetManagerEmployees]	319
	[dbo].[uspGetWhereUsedProductID]	321
	[dbo].[uspLogError]	323
	[dbo].[uspPrintError]	326
	[dbo].[uspSearchCandidateResumes]	327
	[HumanResources].[uspUpdateEmployeeHireInfo]	329
	[HumanResources].[uspUpdateEmployeeLogin]	332
	[HumanResources].[uspUpdateEmployeePersonalInfo]	334
	Table-valued Functions	336
	[dbo].[ufnGetContactInformation]	337
	Scalar-valued Functions	340
	[dbo].[ufnGetAccountingEndDate]	341
	[dbo].[ufnGetAccountingStartDate]	342
	[dbo].[ufnGetDocumentStatusText]	343
	[dbo].[ufnGetProductDealerPrice]	344
	[dbo].[ufnGetProductListPrice]	346
	[dbo].[ufnGetProductStandardCost]	348
	[dbo].[ufnGetPurchaseOrderStatusText]	350
	[dbo].[ufnGetSalesOrderStatusText]	351
	[dbo].[ufnGetStock]	353
	[dbo].[ufnLeadingZeros]	355
	Database Triggers	357

	ddlDatabaseTriggerLog	358
	User-Defined Data Types	360
	[dbo].[AccountNumber]	361
	[dbo].[Flag]	362
	[dbo].[Name]	363
	[dbo].[NameStyle]	365
	[dbo].[OrderNumber]	366
	[dbo].[Phone]	367
	XML Schema Collections	368
	[HumanResources].[HRResumeSchemaCollection]	369
	[Person].[AdditionalContactInfoSchemaCollection]	373
	[Person].[IndividualSurveySchemaCollection]	376
	[Production].[ManuInstructionsSchemaCollection]	378
	[Production].[ProductDescriptionSchemaCollection]	380
	[Sales].[StoreSurveySchemaCollection]	383
	Full Text Catalogs	385
	AW2008FullTextCatalog	386
	Users	387
	piotrkononow	388
	Database Roles	389
	db_accessadmin	389
	db_backupoperator	389
	db_datareader	390
	db_datawriter	390
	db_ddladmin	390
	db_denydatareader	391
	db_denydatawriter	391
	db_owner	391
	db_securityadmin	392
	public	392
	Schemas	393
	HumanResources	394
	Person	395
	Production	396
	Purchasing	398
	Sales	399

Isrep17

Databases (1)

-  AdventureWorks

Server Properties

Property	Value
Product	Microsoft SQL Server
Version	11.0.2100.60
Language	Angielski (Stany Zjednoczone)
Platform	NT INTEL X86
Edition	Express Edition
Processors	1
OS Version	6.1 (7601)
Physical Memory	4096
Is Clustered	False
Root Directory	c:\Program Files (x86)\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL
Collation	Polish_CI_AS

Server Settings

Property	Value
Default data file path	c:\Program Files (x86)\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\
Default backup file path	c:\Program Files (x86)\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Backup
Default log file path	c:\Program Files (x86)\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\
Recovery Interval (minutes)	0
Default index fill factor	0
Default backup media retention	0

Advanced Server Settings

Property	Value
Locks	0
Nested triggers enabled	True
Allow triggers to fire others	True
Default language	English
Network packet size	4096
Default fulltext language LCID	1033
Two-digit year cutoff	2049

Remote login timeout	10
Cursor threshold	-1
Max text replication size	65536
Parallelism cost threshold	5
Scan for startup procs	False
Transform noise words	False
Blocked process threshold	0
Filestream access level	False
Optimize for ad hoc workloads	False

 **User databases**

Databases (1)

-  AdventureWorks

AdventureWorks Database

MS_Description

AdventureWorks 2012 Sample OLTP Database

Database Properties

Property	Value
SQL Server Version	SQL Server 2012
Compatibility Level	SQL Server 2012
Last backup time	01/20/2017
Last log backup time	-
Creation date	Jun 29 2014
Users	5
Database Encryption Enabled	False
Database Encryption Algorithm	None
Database size	207.00 MB
Unallocated space	14.08 MB

Database Options

Property	Value
Compatibility Level	110
Database collation	SQL_Latin1_General_CP1_CI_AS
Restrict access	MULTI_USER
Is read-only	False
Auto close	True
Auto shrink	False
Database status	ONLINE
In standby	False
Cleanly shutdown	False
Supplemental logging enabled	False
Snapshot isolation state	OFF
Read committed snapshot on	False
Recovery model	SIMPLE
Page verify option	CHECKSUM
Auto create statistics	True
Auto update statistics	True
Auto update statistics asynchronously	False
ANSI NULL default	False
ANSI NULL enabled	True
ANSI padding enabled	True

ANSI warnings enabled	True
Arithmetic abort enabled	True
Concatenating NULL yields NULL	True
Numeric roundabort enabled	False
Quoted Identifier On	True
Recursive triggers enabled	False
Close cursors on commit	False
Local cursors by default	False
Fulltext enabled	True
Trustworthy	False
Database chaining	False
Forced parameterization	False
Master key encrypted by server	False
Published	False
Subscribed	False
Merge published	False
Is distribution database	False
Sync with backup	False
Service broker GUID	79fc82e3-43cd-4897-9a80-9b057b738673
Service broker enabled	False
Log reuse wait	CHECKPOINT
Date correlation	False
CDC enabled	False
Encrypted	False
Honor broker priority	False
Default language	English
Default fulltext language LCID	1033
Nested triggers enabled	True
Transform noise words	False
Two-digit year cutoff	2049
Containment	NONE
Target recovery time	0
Database owner	sa

Files

Name	Type	Size	Maxsize	Autogrowth	File Name
AdventureWorks2012_Data	Data	205,00 MB	unlimited	16,00 MB	c:\Program Files (x86)\Microsoft SQL Server\MSSQL11.MS SQLSERVER\MSSQL\DATA\Adventure-Works.mdf
AdventureWorks2012_Log	Log	2,00 MB	unlimited	10,00 percent	c:\Program Files (x86)\Microsoft SQL Server\MSSQL11.MS SQLSERVER\MSSQL\DATA\Adventure-

					Works_log.ldf
--	--	--	--	--	---------------

Tables

Objects

Name
dbo.AWBuildVersion <i>Current version number of the AdventureWorks 2012 sample database.</i>
dbo.DatabaseLog <i>Audit table tracking all DDL changes made to the AdventureWorks database. Data is captured by the database trigger ddlDatabaseTriggerLog.</i>
dbo.ErrorLog <i>Audit table tracking errors in the the AdventureWorks database that are caught by the CATCH block of a TRY...CATCH construct. Data is inserted by stored procedure dbo.uspLogError when it is executed from inside the CATCH block of a TRY...CATCH construct.</i>
HumanResources.Department <i>Lookup table containing the departments within the Adventure Works Cycles company.</i>
HumanResources.Employee <i>Employee information such as salary, department, and title.</i>
HumanResources.EmployeeDepartmentHistory <i>Employee department transfers.</i>
HumanResources.EmployeePayHistory <i>Employee pay history.</i>
HumanResources.JobCandidate <i>Résumés submitted to Human Resources by job applicants.</i>
HumanResources.Shift <i>Work shift lookup table.</i>
Person.Address <i>Street address information for customers, employees, and vendors.</i>
Person.AddressType <i>Types of addresses stored in the Address table.</i>
Person.BusinessEntity <i>Source of the ID that connects vendors, customers, and employees with address and contact information.</i>
Person.BusinessEntityAddress <i>Cross-reference table mapping customers, vendors, and employees to their addresses.</i>
Person.BusinessEntityContact <i>Cross-reference table mapping stores, vendors, and employees to people</i>
Person.ContactType <i>Lookup table containing the types of business entity contacts.</i>
Person.CountryRegion <i>Lookup table containing the ISO standard codes for countries and regions.</i>
Person.EmailAddress <i>Where to send a person email.</i>
Person.Password <i>One way hashed authentication information</i>
Person.Person <i>Human beings involved with AdventureWorks: employees, customer contacts, and vendor contacts.</i>
Person.PersonPhone <i>Telephone number and type of a person.</i>
Person.PhoneNumberType <i>Type of phone number of a person.</i>
Person.StateProvince <i>State and province lookup table.</i>

Production.BillOfMaterials <i>Items required to make bicycles and bicycle subassemblies. It identifies the heirarchical relationship between a parent product and its components.</i>
Production.Culture <i>Lookup table containing the languages in which some AdventureWorks data is stored.</i>
Production.Document <i>Product maintenance documents.</i>
Production.Illustration <i>Bicycle assembly diagrams.</i>
Production.Location <i>Product inventory and manufacturing locations.</i>
Production.Product <i>Products sold or used in the manufacturing of sold products.</i>
Production.ProductCategory <i>High-level product categorization.</i>
Production.ProductCostHistory <i>Changes in the cost of a product over time.</i>
Production.ProductDescription <i>Product descriptions in several languages.</i>
Production.ProductDocument <i>Cross-reference table mapping products to related product documents.</i>
Production.ProductInventory <i>Product inventory information.</i>
Production.ProductListPriceHistory <i>Changes in the list price of a product over time.</i>
Production.ProductModel <i>Product model classification.</i>
Production.ProductModelIllustration <i>Cross-reference table mapping product models and illustrations.</i>
Production.ProductModelProductDescriptionCulture <i>Cross-reference table mapping product descriptions and the language the description is written in.</i>
Production.ProductPhoto <i>Product images.</i>
Production.ProductProductPhoto <i>Cross-reference table mapping products and product photos.</i>
Production.ProductReview <i>Customer reviews of products they have purchased.</i>
Production.ProductSubcategory <i>Product subcategories. See ProductCategory table.</i>
Production.ScrapReason <i>Manufacturing failure reasons lookup table.</i>
Production.TransactionHistory <i>Record of each purchase order, sales order, or work order transaction year to date.</i>
Production.TransactionHistoryArchive <i>Transactions for previous years.</i>
Production.UnitMeasure <i>Unit of measure lookup table.</i>
Production.WorkOrder <i>Manufacturing work orders.</i>
Production.WorkOrderRouting <i>Work order details.</i>
Purchasing.ProductVendor <i>Cross-reference table mapping vendors with the products they supply.</i>
Purchasing.PurchaseOrderDetail <i>Individual products associated with a specific purchase order. See PurchaseOrderHeader.</i>

Purchasing.PurchaseOrderHeader <i>General purchase order information. See PurchaseOrderDetail.</i>
Purchasing.ShipMethod <i>Shipping company lookup table.</i>
Purchasing.Vendor <i>Companies from whom Adventure Works Cycles purchases parts or other goods.</i>
Sales.CountryRegionCurrency <i>Cross-reference table mapping ISO currency codes to a country or region.</i>
Sales.CreditCard <i>Customer credit card information.</i>
Sales.Currency <i>Lookup table containing standard ISO currencies.</i>
Sales.CurrencyRate <i>Currency exchange rates.</i>
Sales.Customer <i>Current customer information. Also see the Person and Store tables.</i>
Sales.PersonCreditCard <i>Cross-reference table mapping people to their credit card information in the CreditCard table.</i>
Sales.SalesOrderDetail <i>Individual products associated with a specific sales order. See SalesOrderHeader.</i>
Sales.SalesOrderHeader <i>General sales order information.</i>
Sales.SalesOrderHeaderSalesReason <i>Cross-reference table mapping sales orders to sales reason codes.</i>
Sales.SalesPerson <i>Sales representative current information.</i>
Sales.SalesPersonQuotaHistory <i>Sales performance tracking.</i>
Sales.SalesReason <i>Lookup table of customer purchase reasons.</i>
Sales.SalesTaxRate <i>Tax rate lookup table.</i>
Sales.SalesTerritory <i>Sales territory lookup table.</i>
Sales.SalesTerritoryHistory <i>Sales representative transfers to other sales territories.</i>
Sales.ShoppingCartItem <i>Contains online customer orders until the order is submitted or cancelled.</i>
Sales.SpecialOffer <i>Sale discounts lookup table.</i>
Sales.SpecialOfferProduct <i>Cross-reference table mapping products to special offer discounts.</i>
Sales.Store <i>Customers (resellers) of Adventure Works products.</i>

 [dbo].[AWBuildVersion]

MS_Description

Current version number of the AdventureWorks 2012 sample database.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	1
Created	13:14:19 14 marca 2012
Last Modified	13:14:41 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	SystemInformationID <i>Primary key for AWBuildVersion records.</i>	tinyint	1	False	1 - 1	
	Database Version <i>Version number of the database in 9.yy.mm.dd.00 format.</i>	nvarchar(25)	50	False		
	VersionDate <i>Date and time the record was last updated.</i>	datetime	8	False		
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_AWBuildVersion_SystemInformationID <i>Primary key (clustered) constraint</i>	SystemInformationID	True

SQL Script

```
CREATE TABLE [dbo].[AWBuildVersion]
(
[SystemInformationID] [tinyint] NOT NULL IDENTITY(1, 1),
[Database Version] [nvarchar] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[VersionDate] [datetime] NOT NULL,
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_AWBuildVersion_ModifiedDate]
DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[AWBuildVersion] ADD CONSTRAINT [PK_AWBuildVersion_System-
```

```
InformationID] PRIMARY KEY CLUSTERED ([SystemInformationID]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Current version number of the
AdventureWorks 2012 sample database.', 'SCHEMA', N'dbo', 'TABLE', N'AWBuild-
Version', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Veersion number of the database in
9.yy.mm.dd.00 format.', 'SCHEMA', N'dbo', 'TABLE', N'AWBuildVersion', 'COLUMN',
N'Database Version'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'dbo', 'TABLE', N'AWBuildVersion', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for AWBuildVersion
records.', 'SCHEMA', N'dbo', 'TABLE', N'AWBuildVersion', 'COLUMN', N'System-
InformationID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'dbo', 'TABLE', N'AWBuildVersion', 'COLUMN', N'VeersionDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'dbo', 'TABLE', N'AWBuildVersion', 'CONSTRAINT', N'DF_AWBuild-
Version_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'dbo', 'TABLE', N'AWBuildVersion', 'CONSTRAINT', N'PK_-
AWBuildVersion_SystemInformationID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'dbo', 'TABLE', N'AWBuildVersion', 'INDEX',
N'PK_AWBuildVersion_SystemInformationID'
GO
```

 [dbo].[DatabaseLog]

MS_Description

Audit table tracking all DDL changes made to the AdventureWorks database. Data is captured by the database trigger ddlDatabaseTriggerLog.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Heap	True
Row Count (~)	1597
Created	13:14:18 14 marca 2012
Last Modified	13:14:41 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity
	DatabaseLogID <i>Primary key for DatabaseLog records.</i>	int	4	False	1 - 1
	PostTime <i>The date and time the DDL change occurred.</i>	datetime	8	False	
	DatabaseUser <i>The user who implemented the DDL change.</i>	[sys].[sysname]	256	False	
	Event <i>The type of DDL statement that was executed.</i>	[sys].[sysname]	256	False	
	Schema <i>The schema to which the changed object belongs.</i>	[sys].[sysname]	256	True	
	Object <i>The object that was changed by the DDL statement.</i>	[sys].[sysname]	256	True	
	TSQL <i>The exact Transact-SQL statement that was executed.</i>	nvarchar(max)	max	False	
	XmlEvent <i>The raw XML data generated by database trigger.</i>	xml	max	False	

Indexes

Key	Name	Key Columns	Unique
	PK_DatabaseLog_DatabaseLogID <i>Primary key (nonclustered) constraint</i>	DatabaseLogID	True

SQL Script

```
CREATE TABLE [dbo].[DatabaseLog]
(
    [DatabaseLogID] [int] NOT NULL IDENTITY(1, 1),
    [PostTime] [datetime] NOT NULL,
    [DatabaseUser] [sys].[sysname] NOT NULL,
    [Event] [sys].[sysname] NOT NULL,
    [Schema] [sys].[sysname] NULL,
    [Object] [sys].[sysname] NULL,
    [TSQL] [nvarchar] (max) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [XmlEvent] [xml] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
ALTER TABLE [dbo].[DatabaseLog] ADD CONSTRAINT [PK_DatabaseLog_DatabaseLogID]
PRIMARY KEY NONCLUSTERED ([DatabaseLogID]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Audit table tracking all DDL
changes made to the AdventureWorks database. Data is captured by the database
trigger ddlDatabaseTriggerLog.', 'SCHEMA', N'dbo', 'TABLE', N'DatabaseLog', NULL,
NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for DatabaseLog
records.', 'SCHEMA', N'dbo', 'TABLE', N'DatabaseLog', 'COLUMN', N'DatabaseLogID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'The user who implemented the DDL
change.', 'SCHEMA', N'dbo', 'TABLE', N'DatabaseLog', 'COLUMN', N'DatabaseUser'
GO
EXEC sp_addextendedproperty N'MS_Description', N'The type of DDL statement that was
executed.', 'SCHEMA', N'dbo', 'TABLE', N'DatabaseLog', 'COLUMN', N'Event'
GO
EXEC sp_addextendedproperty N'MS_Description', N'The object that was changed by the
DDL statement.', 'SCHEMA', N'dbo', 'TABLE', N'DatabaseLog', 'COLUMN', N'Object'
GO
EXEC sp_addextendedproperty N'MS_Description', N'The date and time the DDL change
occurred.', 'SCHEMA', N'dbo', 'TABLE', N'DatabaseLog', 'COLUMN', N'PostTime'
GO
EXEC sp_addextendedproperty N'MS_Description', N'The schema to which the changed
object belongs.', 'SCHEMA', N'dbo', 'TABLE', N'DatabaseLog', 'COLUMN', N'Schema'
GO
EXEC sp_addextendedproperty N'MS_Description', N'The exact Transact-SQL statement
that was executed.', 'SCHEMA', N'dbo', 'TABLE', N'DatabaseLog', 'COLUMN', N'TSQL'
GO
EXEC sp_addextendedproperty N'MS_Description', N'The raw XML data generated by
database trigger.', 'SCHEMA', N'dbo', 'TABLE', N'DatabaseLog', 'COLUMN', N'XmlEvent'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (nonclustered)
constraint', 'SCHEMA', N'dbo', 'TABLE', N'DatabaseLog', 'CONSTRAINT', N'PK_Database-
Log_DatabaseLogID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index created by a
primary key constraint.', 'SCHEMA', N'dbo', 'TABLE', N'DatabaseLog', 'INDEX', N'PK_-
DatabaseLog_DatabaseLogID'
GO
```

 [dbo].[ErrorLog]

MS_Description

Audit table tracking errors in the the AdventureWorks database that are caught by the CATCH block of a TRY...CATCH construct. Data is inserted by stored procedure dbo.uspLogError when it is executed from inside the CATCH block of a TRY...CATCH construct.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	0
Created	13:14:18 14 marca 2012
Last Modified	13:14:18 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	ErrorLogID <i>Primary key for ErrorLog records.</i>	int	4	False	1 - 1	
	ErrorTime <i>The date and time at which the error occurred.</i>	datetime	8	False		(getdate())
	UserName <i>The user who executed the batch in which the error occurred.</i>	[sys].[sysname]	256	False		
	ErrorNumber <i>The error number of the error that occurred.</i>	int	4	False		
	ErrorSeverity <i>The severity of the error that occurred.</i>	int	4	True		
	ErrorState <i>The state number of the error that occurred.</i>	int	4	True		
	ErrorProcedure <i>The name of the stored procedure or trigger where the error occurred.</i>	nvarchar(126)	252	True		
	ErrorLine <i>The line number at which the error occurred.</i>	int	4	True		
	ErrorMessage <i>The message text of the error that occurred.</i>	nvarchar(4000)	8000	False		

Indexes

Key	Name	Key Columns	Unique
-----	------	-------------	--------

 PK_ErrorLog_ErrorLogID <i>Primary key (clustered) constraint</i>	ErrorLogID	True
--	------------	------

SQL Script

```

CREATE TABLE [dbo].[ErrorLog]
(
    [ErrorLogID] [int] NOT NULL IDENTITY(1, 1),
    [ErrorTime] [datetime] NOT NULL CONSTRAINT [DF_ErrorLog_ErrorTime] DEFAULT
    (getdate()),
    [UserName] [sys].[sysname] NOT NULL,
    [ErrorNumber] [int] NOT NULL,
    [ErrorSeverity] [int] NULL,
    [ErrorState] [int] NULL,
    [ErrorProcedure] [nvarchar] (126) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [ErrorLine] [int] NULL,
    [ErrorMessage] [nvarchar] (4000) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[ErrorLog] ADD CONSTRAINT [PK_ErrorLog_ErrorLogID] PRIMARY KEY
CLUSTERED ([ErrorLogID]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Audit table tracking errors in the
the AdventureWorks database that are caught by the CATCH block of a TRY...CATCH
construct. Data is inserted by stored procedure dbo.uspLogError when it is executed
from inside the CATCH block of a TRY...CATCH construct.', 'SCHEMA', N'dbo', 'TABLE',
N'ErrorLog', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'The line number at which the error
occurred.', 'SCHEMA', N'dbo', 'TABLE', N'ErrorLog', 'COLUMN', N'ErrorLine'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for ErrorLog records.',
'SCHEMA', N'dbo', 'TABLE', N'ErrorLog', 'COLUMN', N'ErrorLogID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'The message text of the error that
occurred.', 'SCHEMA', N'dbo', 'TABLE', N'ErrorLog', 'COLUMN', N'ErrorMessage'
GO
EXEC sp_addextendedproperty N'MS_Description', N'The error number of the error that
occurred.', 'SCHEMA', N'dbo', 'TABLE', N'ErrorLog', 'COLUMN', N'ErrorNumber'
GO
EXEC sp_addextendedproperty N'MS_Description', N'The name of the stored procedure or
trigger where the error occurred.', 'SCHEMA', N'dbo', 'TABLE', N'ErrorLog',
'COLUMN', N'ErrorProcedure'
GO
EXEC sp_addextendedproperty N'MS_Description', N'The severity of the error that
occurred.', 'SCHEMA', N'dbo', 'TABLE', N'ErrorLog', 'COLUMN', N'ErrorSeverity'
GO
EXEC sp_addextendedproperty N'MS_Description', N'The state number of the error that
occurred.', 'SCHEMA', N'dbo', 'TABLE', N'ErrorLog', 'COLUMN', N'ErrorState'
GO
EXEC sp_addextendedproperty N'MS_Description', N'The date and time at which the
error occurred.', 'SCHEMA', N'dbo', 'TABLE', N'ErrorLog', 'COLUMN', N'ErrorTime'
GO
EXEC sp_addextendedproperty N'MS_Description', N'The user who executed the batch in
which the error occurred.', 'SCHEMA', N'dbo', 'TABLE', N'ErrorLog', 'COLUMN', N'User-
Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'dbo', 'TABLE', N'ErrorLog', 'CONSTRAINT', N'DF_ErrorLog_
ErrorTime'

```

```
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'dbo', 'TABLE', N'ErrorLog', 'CONSTRAINT', N'PK_ErrorLog_
ErrorLogID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'dbo', 'TABLE', N'ErrorLog', 'INDEX', N'PK_
ErrorLog_ErrorLogID'
GO
```

Used By

[dbo].[uspLogError]

[HumanResources].[Department]**MS_Description**

Lookup table containing the departments within the Adventure Works Cycles company.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	16
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	DepartmentID <i>Primary key for Department records.</i>	smallint	2	False	1 - 1	
	Name <i>Name of the department.</i>	[dbo].[Name]	100	False		
	GroupName <i>Name of the group to which the department belongs.</i>	[dbo].[Name]	100	False		
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_Department_DepartmentID <i>Primary key (clustered) constraint</i>	DepartmentID	True
	AK_Department_Name <i>Unique nonclustered index.</i>	Name	True

SQL Script

```
CREATE TABLE [HumanResources].[Department]
(
  [DepartmentID] [smallint] NOT NULL IDENTITY(1, 1),
  [Name] [dbo].[Name] NOT NULL,
  [GroupName] [dbo].[Name] NOT NULL,
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_Department_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
```

```

GO
ALTER TABLE [HumanResources].[Department] ADD CONSTRAINT [PK_Department_Department-
ID] PRIMARY KEY CLUSTERED ([DepartmentID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_Department_Name] ON [Human-
Resources].[Department] ([Name]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Lookup table containing the
departments within the Adventure Works Cycles company.', 'SCHEMA', N'Human-
Resources', 'TABLE', N'Department', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Department
records.', 'SCHEMA', N'HumanResources', 'TABLE', N'Department', 'COLUMN',
N'DepartmentID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Name of the group to which the
department belongs.', 'SCHEMA', N'HumanResources', 'TABLE', N'Department', 'COLUMN',
N'GroupName'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'HumanResources', 'TABLE', N'Department', 'COLUMN', N'Modified-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Name of the department.', 'SCHEMA',
N'HumanResources', 'TABLE', N'Department', 'COLUMN', N'Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'HumanResources', 'TABLE', N'Department', 'CONSTRAINT', N'DF_-
Department_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'HumanResources', 'TABLE', N'Department', 'CONSTRAINT',
N'PK_Department_DepartmentID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SCHEMA', N'HumanResources', 'TABLE', N'Department', 'INDEX', N'AK_Department_Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'HumanResources', 'TABLE', N'Department',
'INDEX', N'PK_Department_DepartmentID'
GO

```

Uses

[dbo].[Name]
HumanResources

Used By

[HumanResources].[EmployeeDepartmentHistory]
[HumanResources].[vEmployeeDepartment]
[HumanResources].[vEmployeeDepartmentHistory]

 **[HumanResources].[Employee]****MS_Description**

Employee information such as salary, department, and title.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	290
Created	15:26:58 22 marca 2017
Last Modified	15:27:05 22 marca 2017

Columns

Key	Name	Data Type	Computed	Max Length (Bytes)	Allow Nulls	Default
	BusinessEntityID <i>Primary key for Employee records. Foreign key to BusinessEntity.BusinessEntity-ID.</i>	int		4	False	
	NationalIDNumber <i>Unique national identification number such as a social security number.</i>	nvarchar(15)		30	False	
	LoginID <i>Network login.</i>	nvarchar(256)		512	False	
 (2)	OrganizationNode <i>Where the employee is located in corporate hierarchy.</i>	hierarchyid		892	True	
	OrganizationLevel <i>The depth of the employee in the corporate hierarchy.</i>	smallint	True	2	True	
	JobTitle <i>Work title such as Buyer or Sales Representative.</i>	nvarchar(50)		100	False	
	BirthDate <i>Date of birth.</i>	date		3	False	
	MaritalStatus <i>M = Married, S = Single</i>	nchar(1)		2	False	
	Gender <i>M = Male, F = Female</i>	nchar(1)		2	False	
	HireDate <i>Employee hired on this date.</i>	date		3	False	
	SalariedFlag <i>Job classification. 0 = Hourly, not exempt from collective bargaining. 1 = Salaried, exempt from collective bargaining.</i>	[dbo].[Flag]		1	False	((1))
	VacationHours <i>Number of available vacation</i>	smallint		2	False	((0))

	hours.					
	SickLeaveHours <i>Number of available sick leave hours.</i>	smallint		2	False	((0))
	CurrentFlag <i>0 = Inactive, 1 = Active</i>	[dbo].[Flag]		1	False	((1))
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier		16	False	(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime		8	False	(getdate())

Computed columns

Name	Column definition
OrganizationLevel	([OrganizationNode].[GetLevel]())

Indexes

Key	Name	Key Columns	Unique
	PK_Employee_BusinessEntityID <i>Clustered index created by a primary key constraint.</i>	BusinessEntity-ID	True
	AK_Employee_LoginID <i>Unique nonclustered index.</i>	LoginID	True
	AK_Employee_NationalIDNumber <i>Unique nonclustered index.</i>	National-IDNumber	True
	AK_Employee_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True
	IX_Employee_OrganizationLevel_OrganizationNode <i>Unique nonclustered index.</i>	Organization-Level, Organization-Node	
	IX_Employee_OrganizationNode <i>Unique nonclustered index.</i>	Organization-Node	

Triggers

Name	ANSI Nulls On	Quoted Identifier On	On	Not For Replication
dEmployee <i>INSTEAD OF DELETE trigger which keeps Employees from being deleted.</i>	True	True	Instead Of Delete	True

Check Constraints

Name	On Column	Constraint
CK_Employee_BirthDate <i>Check constraint [BirthDate] >= '1930-01-01' AND [BirthDate] <= dateadd(year,(-18),GETDATE())</i>	BirthDate	([BirthDate]>='1930-01-01' AND [BirthDate]<=dateadd

		(year,(-18),getdate()))
CK_Employee_HireDate <i>Check constraint [HireDate] >= '1996-07-01' AND [HireDate] <= dateadd(day,(1),GETDATE())</i>	HireDate	(([HireDate]>='1996-07-01' AND [HireDate]<=dateadd(day,(1),getdate())))
CK_Employee_SickLeaveHours <i>Check constraint [SickLeaveHours] >= (0) AND [SickLeaveHours] <= (120)</i>	SickLeaveHours	(([SickLeaveHours]>=(0) AND [SickLeaveHours]<=(120))
CK_Employee_VacationHours <i>Check constraint [VacationHours] >= (-40) AND [VacationHours] <= (240)</i>	VacationHours	(([VacationHours]>=(-40)) AND [VacationHours]<=(240))
CK_Employee_Gender <i>Check constraint [Gender]='F' OR [Gender]='m' OR [Gender]='F' OR [Gender]='M'</i>	Gender	(upper([Gender])='F' OR upper([Gender])='M')
CK_Employee_MaritalStatus <i>Check constraint [MaritalStatus]='s' OR [MaritalStatus]='m' OR [MaritalStatus]='S' OR [MaritalStatus]='M'</i>	MaritalStatus	(upper([MaritalStatus])='S' OR upper([MaritalStatus])='M')

Foreign Keys

Name	Columns
FK_Employee_Person_BusinessEntityID <i>Foreign key constraint referencing Person.BusinessEntityID.</i>	BusinessEntityID->[Person].[Person].[BusinessEntityID]

SQL Script

```

CREATE TABLE [HumanResources].[Employee]
(
    [BusinessEntityID] [int] NOT NULL,
    [NationalIDNumber] [nvarchar] (15) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [LoginID] [nvarchar] (256) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [OrganizationNode] [sys].[hierarchyid] NULL,
    [OrganizationLevel] AS ([OrganizationNode].[GetLevel]()),
    [JobTitle] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [BirthDate] [date] NOT NULL,
    [MaritalStatus] [nchar] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [Gender] [nchar] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [HireDate] [date] NOT NULL,
    [SalariedFlag] [dbo].[Flag] NOT NULL CONSTRAINT [DF_Employee_SalariedFlag] DEFAULT (1),
    [VacationHours] [smallint] NOT NULL CONSTRAINT [DF_Employee_VacationHours] DEFAULT (0),
    [SickLeaveHours] [smallint] NOT NULL CONSTRAINT [DF_Employee_SickLeaveHours] DEFAULT (0),
    [CurrentFlag] [dbo].[Flag] NOT NULL CONSTRAINT [DF_Employee_CurrentFlag] DEFAULT (1),
    [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_Employee_rowguid] DEFAULT (newid()),
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_Employee_ModifiedDate] DEFAULT (getdate())
)
    
```

```

) ON [PRIMARY]
GO
CREATE TRIGGER [HumanResources].[dEmployee] ON [HumanResources].[Employee]
INSTEAD OF DELETE NOT FOR REPLICATION AS
BEGIN
    DECLARE @Count int;

    SET @Count = @@ROWCOUNT;
    IF @Count = 0
        RETURN;

    SET NOCOUNT ON;

    BEGIN
        RAISERROR
            (N'Employees cannot be deleted. They can only be marked as not
current.', -- Message
            10, -- Severity.
            1); -- State.

        -- Rollback any active or uncommittable transactions
        IF @@TRANCOUNT > 0
            BEGIN
                ROLLBACK TRANSACTION;
            END
    END;
END;
GO
ALTER TABLE [HumanResources].[Employee] ADD CONSTRAINT [CK_Employee_BirthDate] CHECK
((([BirthDate]>='1930-01-01' AND [BirthDate]<=dateadd(year,(-18),getdate()))))
GO
ALTER TABLE [HumanResources].[Employee] ADD CONSTRAINT [CK_Employee_HireDate] CHECK
((([HireDate]>='1996-07-01' AND [HireDate]<=dateadd(day,(1),getdate()))))
GO
ALTER TABLE [HumanResources].[Employee] ADD CONSTRAINT [CK_Employee_SickLeaveHours]
CHECK (([SickLeaveHours]>=(0) AND [SickLeaveHours]<=(120)))
GO
ALTER TABLE [HumanResources].[Employee] ADD CONSTRAINT [CK_Employee_VacationHours]
CHECK (([VacationHours]>=(-40) AND [VacationHours]<=(240)))
GO
ALTER TABLE [HumanResources].[Employee] ADD CONSTRAINT [CK_Employee_Gender] CHECK
((upper([Gender])='F' OR upper([Gender])='M'))
GO
ALTER TABLE [HumanResources].[Employee] ADD CONSTRAINT [CK_Employee_MaritalStatus]
CHECK ((upper([MaritalStatus])='S' OR upper([MaritalStatus])='M'))
GO
ALTER TABLE [HumanResources].[Employee] ADD CONSTRAINT [PK_Employee_BusinessEntity-
ID] PRIMARY KEY CLUSTERED ([BusinessEntityID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_Employee_LoginID] ON [Human-
Resources].[Employee] ([LoginID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_Employee_NationalIDNumber] ON [Human-
Resources].[Employee] ([NationalIDNumber]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_Employee_OrganizationLevel_OrganizationNode] ON [Human-
Resources].[Employee] ([OrganizationLevel],[OrganizationNode]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_Employee_OrganizationNode] ON [Human-
Resources].[Employee] ([OrganizationNode]) ON [PRIMARY]

```

```

GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_Employee_rowguid] ON [Human-
Resources].[Employee] ([rowguid]) ON [PRIMARY]
GO
ALTER TABLE [HumanResources].[Employee] ADD CONSTRAINT [FK_Employee_Person_Business-
EntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [Person].[Person] ([Business-
EntityID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Employee information such as
salary, department, and title.', 'SCHEMA', N'HumanResources', 'TABLE', N'Employee',
NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date of birth.', 'SCHEMA', N'Human-
Resources', 'TABLE', N'Employee', 'COLUMN', N'BirthDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Employee records.
Foreign key to BusinessEntity.BusinessEntityID.', 'SCHEMA', N'HumanResources',
'TABLE', N'Employee', 'COLUMN', N'BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'0 = Inactive, 1 = Active',
'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'COLUMN', N'CurrentFlag'
GO
EXEC sp_addextendedproperty N'MS_Description', N'M = Male, F = Female', 'SCHEMA',
N'HumanResources', 'TABLE', N'Employee', 'COLUMN', N'Gender'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Employee hired on this date.',
'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'COLUMN', N'HireDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Work title such as Buyer or Sales
Representative.', 'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'COLUMN', N'Job-
Title'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Network login.', 'SCHEMA', N'Human-
Resources', 'TABLE', N'Employee', 'COLUMN', N>LoginID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'M = Married, S = Single', 'SCHEMA',
N'HumanResources', 'TABLE', N'Employee', 'COLUMN', N'MaritalStatus'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'COLUMN', N'Modified-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique national identification
number such as a social security number.', 'SCHEMA', N'HumanResources', 'TABLE',
N'Employee', 'COLUMN', N'NationalIDNumber'
GO
EXEC sp_addextendedproperty N'MS_Description', N'The depth of the employee in the
corporate hierarchy.', 'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'COLUMN',
N'OrganizationLevel'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Where the employee is located in
corporate hierarchy.', 'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'COLUMN',
N'OrganizationNode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'HumanResources', 'TABLE', N'Employee', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Job classification. 0 = Hourly, not
exempt from collective bargaining. 1 = Salaried, exempt from collective
bargaining.', 'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'COLUMN',
N'SalariedFlag'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Number of available sick leave

```

```

hours.', 'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'COLUMN', N'SickLeave-
Hours'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Number of available vacation
hours.', 'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'COLUMN', N'Vacation-
Hours'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [BirthDate] >=
''1930-01-01'' AND [BirthDate] <= dateadd(year, (-18), GETDATE())', 'SCHEMA', N'Human-
Resources', 'TABLE', N'Employee', 'CONSTRAINT', N'CK_Employee_BirthDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Gender]=''f'' OR
[Gender]=''m'' OR [Gender]=''F'' OR [Gender]=''M''', 'SCHEMA', N'HumanResources',
'TABLE', N'Employee', 'CONSTRAINT', N'CK_Employee_Gender'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [HireDate] >=
''1996-07-01'' AND [HireDate] <= dateadd(day, (1), GETDATE())', 'SCHEMA', N'Human-
Resources', 'TABLE', N'Employee', 'CONSTRAINT', N'CK_Employee_HireDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Marital-
Status]=''s'' OR [MaritalStatus]=''m'' OR [MaritalStatus]=''S'' OR [Marital-
Status]=''M''', 'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'CONSTRAINT',
N'CK_Employee_MaritalStatus'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [SickLeaveHours]
>= (0) AND [SickLeaveHours] <= (120)', 'SCHEMA', N'HumanResources', 'TABLE',
N'Employee', 'CONSTRAINT', N'CK_Employee_SickLeaveHours'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [VacationHours] >=
(-40) AND [VacationHours] <= (240)', 'SCHEMA', N'HumanResources', 'TABLE',
N'Employee', 'CONSTRAINT', N'CK_Employee_VacationHours'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Person.BusinessEntityID.', 'SCHEMA', N'HumanResources', 'TABLE', N'Employee',
'CONSTRAINT', N'FK_Employee_Person_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'HumanResources', 'TABLE', N'Employee',
'CONSTRAINT', N'PK_Employee_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'INDEX', N'AK_Employee_LoginID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'INDEX', N'AK_Employee_National-
IDNumber'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'HumanResources', 'TABLE', N'Employee',
'INDEX', N'AK_Employee_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'INDEX', N'IX_Employee_-
OrganizationLevel_OrganizationNode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'INDEX', N'IX_Employee_-
OrganizationNode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'INSTEAD OF DELETE trigger which
keeps Employees from being deleted.', 'SCHEMA', N'HumanResources', 'TABLE',
N'Employee', 'TRIGGER', N'dEmployee'
GO

```

Uses

[Person].[Person]
[dbo].[Flag]
HumanResources

Used By

[HumanResources].[EmployeeDepartmentHistory]
[HumanResources].[EmployeePayHistory]
[HumanResources].[JobCandidate]
[Purchasing].[PurchaseOrderHeader]
[Sales].[SalesPerson]
[HumanResources].[vEmployee]
[HumanResources].[vEmployeeDepartment]
[HumanResources].[vEmployeeDepartmentHistory]
[Sales].[vSalesPerson]
[Sales].[vSalesPersonSalesByFiscalYears]
[dbo].[uspGetEmployeeManagers]
[dbo].[uspGetManagerEmployees]
[HumanResources].[uspUpdateEmployeeHireInfo]
[HumanResources].[uspUpdateEmployeeLogin]
[HumanResources].[uspUpdateEmployeePersonalInfo]
[dbo].[ufnGetContactInformation]

[HumanResources].[EmployeeDepartmentHistory]

MS_Description

Employee department transfers.

Properties

Property	Value
Row Count (~)	296
Created	13:14:19 14 marca 2012
Last Modified	15:26:58 22 marca 2017

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	BusinessEntityID <i>Employee identification number. Foreign key to Employee.BusinessEntityID.</i>	int	4	False	
	DepartmentID <i>Department in which the employee worked including currently. Foreign key to Department.DepartmentID.</i>	smallint	2	False	
	ShiftID <i>Identifies which 8-hour shift the employee works. Foreign key to Shift.ShiftID.</i>	tinyint	1	False	
	StartDate <i>Date the employee started work in the department.</i>	date	3	False	
	EndDate <i>Date the employee left the department. NULL = Current department.</i>	date	3	True	
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_EmployeeDepartmentHistory_BusinessEntityID_StartDate_DepartmentID <i>Primary key (clustered) constraint</i>	BusinessEntityID, StartDate, DepartmentID, ShiftID	True
	IX_EmployeeDepartmentHistory_DepartmentID <i>Nonclustered index.</i>	DepartmentID	
	IX_EmployeeDepartmentHistory_ShiftID <i>Nonclustered index.</i>	ShiftID	

Check Constraints

Name	Constraint
CK_EmployeeDepartmentHistory_EndDate <i>Check constraint [EndDate] >= [StartDate] OR [EndDate] IS NULL</i>	(([EndDate]>=[StartDate] OR [EndDate] IS NULL)

Foreign Keys

Name	Columns
FK_EmployeeDepartmentHistory_Department_DepartmentID <i>Foreign key constraint referencing Department.DepartmentID.</i>	DepartmentID->[HumanResources].[Department].[DepartmentID]
FK_EmployeeDepartmentHistory_Employee_BusinessEntityID <i>Foreign key constraint referencing Employee.EmployeeID.</i>	BusinessEntityID->[HumanResources].[Employee].[BusinessEntityID]
FK_EmployeeDepartmentHistory_Shift_ShiftID <i>Foreign key constraint referencing Shift.ShiftID</i>	ShiftID->[HumanResources].[Shift].[ShiftID]

SQL Script

```
CREATE TABLE [HumanResources].[EmployeeDepartmentHistory]
(
    [BusinessEntityID] [int] NOT NULL,
    [DepartmentID] [smallint] NOT NULL,
    [ShiftID] [tinyint] NOT NULL,
    [StartDate] [date] NOT NULL,
    [EndDate] [date] NULL,
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_EmployeeDepartmentHistory_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [HumanResources].[EmployeeDepartmentHistory] ADD CONSTRAINT [CK_EmployeeDepartmentHistory_EndDate] CHECK (([EndDate]>=[StartDate] OR [EndDate] IS NULL))
GO
ALTER TABLE [HumanResources].[EmployeeDepartmentHistory] ADD CONSTRAINT [PK_EmployeeDepartmentHistory_BusinessEntityID_StartDate_DepartmentID] PRIMARY KEY CLUSTERED ([BusinessEntityID], [StartDate], [DepartmentID], [ShiftID]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_EmployeeDepartmentHistory_DepartmentID] ON [HumanResources].[EmployeeDepartmentHistory] ([DepartmentID]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_EmployeeDepartmentHistory_ShiftID] ON [HumanResources].[EmployeeDepartmentHistory] ([ShiftID]) ON [PRIMARY]
GO
ALTER TABLE [HumanResources].[EmployeeDepartmentHistory] ADD CONSTRAINT [FK_EmployeeDepartmentHistory_Department_DepartmentID] FOREIGN KEY ([DepartmentID]) REFERENCES [HumanResources].[Department] ([DepartmentID])
GO
ALTER TABLE [HumanResources].[EmployeeDepartmentHistory] ADD CONSTRAINT [FK_EmployeeDepartmentHistory_Employee_BusinessEntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [HumanResources].[Employee] ([BusinessEntityID])
GO
ALTER TABLE [HumanResources].[EmployeeDepartmentHistory] ADD CONSTRAINT [FK_EmployeeDepartmentHistory_Shift_ShiftID] FOREIGN KEY ([ShiftID]) REFERENCES [HumanResources].[Shift] ([ShiftID])
```

```
GO
EXEC sp_addextendedproperty N'MS_Description', N'Employee department transfers.',
'SCHEMA', N'HumanResources', 'TABLE', N'EmployeeDepartmentHistory', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Employee identification number.
Foreign key to Employee.BusinessEntityID.', 'SCHEMA', N'HumanResources', 'TABLE',
N'EmployeeDepartmentHistory', 'COLUMN', N'BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Department in which the employee
worked including currently. Foreign key to Department.DepartmentID.', 'SCHEMA',
N'HumanResources', 'TABLE', N'EmployeeDepartmentHistory', 'COLUMN', N'DepartmentID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date the employee left the
department. NULL = Current department.', 'SCHEMA', N'HumanResources', 'TABLE',
N'EmployeeDepartmentHistory', 'COLUMN', N'EndDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeeDepartmentHistory',
'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Identifies which 8-hour shift the
employee works. Foreign key to Shift.Shift.ID.', 'SCHEMA', N'HumanResources',
'TABLE', N'EmployeeDepartmentHistory', 'COLUMN', N'ShiftID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date the employee started work in
the department.', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeeDepartment-
History', 'COLUMN', N'StartDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [EndDate] >=
[StartDate] OR [EndDate] IS NUL', 'SCHEMA', N'HumanResources', 'TABLE', N'Employee-
DepartmentHistory', 'CONSTRAINT', N'CK_EmployeeDepartmentHistory_EndDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeeDepartmentHistory',
'CONSTRAINT', N'DF_EmployeeDepartmentHistory_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Department.DepartmentID.', 'SCHEMA', N'HumanResources', 'TABLE', N'Employee-
DepartmentHistory', 'CONSTRAINT', N'FK_EmployeeDepartmentHistory_Department_-
DepartmentID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Employee.EmployeeID.', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeeDepartment-
History', 'CONSTRAINT', N'FK_EmployeeDepartmentHistory_Employee_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Shift.ShiftID', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeeDepartmentHistory',
'CONSTRAINT', N'FK_EmployeeDepartmentHistory_Shift_ShiftID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeeDepartmentHistory',
'CONSTRAINT', N'PK_EmployeeDepartmentHistory_BusinessEntityID_StartDate_Departme-
ntID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'HumanResources', 'TABLE', N'EmployeeDepartmentHistory', 'INDEX', N'IX_Employee-
DepartmentHistory_DepartmentID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'HumanResources', 'TABLE', N'EmployeeDepartmentHistory', 'INDEX', N'IX_Employee-
DepartmentHistory_ShiftID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeeDepartment-
History', 'INDEX', N'PK_EmployeeDepartmentHistory_BusinessEntityID_StartDate -
```

DepartmentID'

GO

Uses

[HumanResources].[Department]

[HumanResources].[Employee]

[HumanResources].[Shift]

HumanResources

Used By

[HumanResources].[vEmployeeDepartment]

[HumanResources].[vEmployeeDepartmentHistory]

 **[HumanResources].[EmployeePayHistory]**

MS_Description

Employee pay history.

Properties

Property	Value
Row Count (~)	316
Created	13:14:19 14 marca 2012
Last Modified	15:26:58 22 marca 2017

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	BusinessEntityID <i>Employee identification number. Foreign key to Employee.BusinessEntityID.</i>	int	4	False	
	RateChangeDate <i>Date the change in pay is effective</i>	datetime	8	False	
	Rate <i>Salary hourly rate.</i>	money	8	False	
	PayFrequency <i>1 = Salary received monthly, 2 = Salary received biweekly</i>	tinyint	1	False	
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_EmployeePayHistory_BusinessEntityID_RateChangeDate <i>Primary key (clustered) constraint</i>	BusinessEntityID, RateChangeDate	True

Check Constraints

Name	On Column	Constraint
CK_EmployeePayHistory_PayFrequency <i>Check constraint [PayFrequency]=(3) OR [PayFrequency]=(2) OR [PayFrequency]=(1)</i>	PayFrequency	((Pay-Frequency)=(2) OR [Pay-Frequency]=(1))
CK_EmployeePayHistory_Rate <i>Check constraint [Rate] >= (6.50) AND [Rate] <= (200.00)</i>	Rate	(([Rate]>=(6.50) AND [Rate]<=(200.00))

Foreign Keys

Name	Columns
FK_EmployeePayHistory_Employee_BusinessEntityID <i>Foreign key constraint referencing Employee.EmployeeID.</i>	BusinessEntityID->[HumanResources].[Employee].[BusinessEntityID]

SQL Script

```

CREATE TABLE [HumanResources].[EmployeePayHistory]
(
  [BusinessEntityID] [int] NOT NULL,
  [RateChangeDate] [datetime] NOT NULL,
  [Rate] [money] NOT NULL,
  [PayFrequency] [tinyint] NOT NULL,
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_EmployeePayHistory_ModifiedDate]
  DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [HumanResources].[EmployeePayHistory] ADD CONSTRAINT [CK_EmployeePay-
History_PayFrequency] CHECK (([PayFrequency]=(2) OR [PayFrequency]=(1)))
GO
ALTER TABLE [HumanResources].[EmployeePayHistory] ADD CONSTRAINT [CK_EmployeePay-
History_Rate] CHECK (([Rate]>=(6.50) AND [Rate]<=(200.00)))
GO
ALTER TABLE [HumanResources].[EmployeePayHistory] ADD CONSTRAINT [PK_EmployeePay-
History_BusinessEntityID_RateChangeDate] PRIMARY KEY CLUSTERED ([BusinessEntityID],
[RateChangeDate]) ON [PRIMARY]
GO
ALTER TABLE [HumanResources].[EmployeePayHistory] ADD CONSTRAINT [FK_EmployeePay-
History_Employee_BusinessEntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES
[HumanResources].[Employee] ([BusinessEntityID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Employee pay history.', 'SCHEMA',
N'HumanResources', 'TABLE', N'EmployeePayHistory', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Employee identification number.
Foreign key to Employee.BusinessEntityID.', 'SCHEMA', N'HumanResources', 'TABLE',
N'EmployeePayHistory', 'COLUMN', N'BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeePayHistory', 'COLUMN',
N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'1 = Salary received monthly, 2 =
Salary received biweekly', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeePay-
History', 'COLUMN', N'PayFrequency'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Salary hourly rate.', 'SCHEMA',
N'HumanResources', 'TABLE', N'EmployeePayHistory', 'COLUMN', N'Rate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date the change in pay is
effective', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeePayHistory', 'COLUMN',
N'RateChangeDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [PayFrequency]=(3)
OR [PayFrequency]=(2) OR [PayFrequency]=(1)', 'SCHEMA', N'HumanResources', 'TABLE',
N'EmployeePayHistory', 'CONSTRAINT', N'CK_EmployeePayHistory_PayFrequency'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Rate] >= (6.50)
AND [Rate] <= (200.00)', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeePay-

```

```
History', 'CONSTRAINT', N'CK_EmployeePayHistory_Rate'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of  
GETDATE()', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeePayHistory',  
'CONSTRAINT', N'DF_EmployeePayHistory_ModifiedDate'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing  
Employee.EmployeeID.', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeePayHistory',  
'CONSTRAINT', N'FK_EmployeePayHistory_Employee_BusinessEntityID'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)  
constraint', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeePayHistory',  
'CONSTRAINT', N'PK_EmployeePayHistory_BusinessEntityID_RateChangeDate'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a  
primary key constraint.', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeePay-  
History', 'INDEX', N'PK_EmployeePayHistory_BusinessEntityID_RateChangeDate'  
GO
```

Uses

[HumanResources].[Employee]
HumanResources

Used By

[HumanResources].[uspUpdateEmployeeHireInfo]

[HumanResources].[JobCandidate]

MS_Description

Résumés submitted to Human Resources by job applicants.

Properties

Property	Value
Full Text Catalog	AW2008FullTextCatalog
Full Text Key Index	PK_JobCandidate_JobCandidateID
Row Count (~)	13
Created	13:14:19 14 marca 2012
Last Modified	15:26:58 22 marca 2017

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Full Text Indexed	Language	Identity	Default
	JobCandidateID <i>Primary key for Job-Candidate records.</i>	int	4	False			1 - 1	
	BusinessEntityID <i>Employee identification number if applicant was hired. Foreign key to Employee.Business-EntityID.</i>	int	4	True				
	Resume <i>Résumé in XML format.</i>	xml([Human-Resources].[HRResumeSchema-Collection])	max	True	True	1033		
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False				(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_JobCandidate_JobCandidateID <i>Primary key (clustered) constraint</i>	JobCandidateID	True
	IX_JobCandidate_BusinessEntityID <i>Nonclustered index.</i>	BusinessEntityID	

Foreign Keys

Name	Columns
------	---------

FK_JobCandidate_Employee_BusinessEntityID
Foreign key constraint referencing Employee.EmployeeID.

BusinessEntityID->[HumanResources].[Employee].[BusinessEntityID]

SQL Script

```
CREATE TABLE [HumanResources].[JobCandidate]
(
    [JobCandidateID] [int] NOT NULL IDENTITY(1, 1),
    [BusinessEntityID] [int] NULL,
    [Resume] [xml] (CONTENT [HumanResources].[HRResumeSchemaCollection]) NULL,
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_JobCandidate_ModifiedDate] DEFAULT
    (getdate())
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
ALTER TABLE [HumanResources].[JobCandidate] ADD CONSTRAINT [PK_JobCandidate_JobCandidateID] PRIMARY KEY CLUSTERED ([JobCandidateID]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_JobCandidate_BusinessEntityID] ON [HumanResources].[JobCandidate] ([BusinessEntityID]) ON [PRIMARY]
GO
ALTER TABLE [HumanResources].[JobCandidate] ADD CONSTRAINT [FK_JobCandidate_Employee_BusinessEntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [HumanResources].[Employee] ([BusinessEntityID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Resumés submitted to Human Resources by job applicants.', 'SCHEMA', N'HumanResources', 'TABLE', N'JobCandidate', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Employee identification number if applicant was hired. Foreign key to Employee.BusinessEntityID.', 'SCHEMA', N'HumanResources', 'TABLE', N'JobCandidate', 'COLUMN', N'BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for JobCandidate records.', 'SCHEMA', N'HumanResources', 'TABLE', N'JobCandidate', 'COLUMN', N'JobCandidateID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last updated.', 'SCHEMA', N'HumanResources', 'TABLE', N'JobCandidate', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Resumé in XML format.', 'SCHEMA', N'HumanResources', 'TABLE', N'JobCandidate', 'COLUMN', N'Resume'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'HumanResources', 'TABLE', N'JobCandidate', 'CONSTRAINT', N'DF_JobCandidate_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing Employee.EmployeeID.', 'SCHEMA', N'HumanResources', 'TABLE', N'JobCandidate', 'CONSTRAINT', N'FK_JobCandidate_Employee_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint', 'SCHEMA', N'HumanResources', 'TABLE', N'JobCandidate', 'CONSTRAINT', N'PK_JobCandidate_JobCandidateID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA', N'HumanResources', 'TABLE', N'JobCandidate', 'INDEX', N'IX_JobCandidate_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'HumanResources', 'TABLE', N'JobCandidate',
```

```
'INDEX', N'PK_JobCandidate_JobCandidateID'  
GO  
CREATE FULLTEXT INDEX ON [HumanResources].[JobCandidate] KEY INDEX [PK_JobCandidate_  
JobCandidateID] ON [AW2008FullTextCatalog]  
GO  
ALTER FULLTEXT INDEX ON [HumanResources].[JobCandidate] ADD ([Resume] LANGUAGE 1033)  
GO
```

Uses

[HumanResources].[Employee]
HumanResources
[HumanResources].[HRResumeSchemaCollection]

Used By

[HumanResources].[vJobCandidate]
[HumanResources].[vJobCandidateEducation]
[HumanResources].[vJobCandidateEmployment]
[dbo].[uspSearchCandidateResumes]
AW2008FullTextCatalog

 **[HumanResources].[Shift]****MS_Description**

Work shift lookup table.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	3
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	ShiftID <i>Primary key for Shift records.</i>	tinyint	1	False	1 - 1	
	Name <i>Shift description.</i>	[dbo].[Name]	100	False		
	StartTime <i>Shift start time.</i>	time	5	False		
	EndTime <i>Shift end time.</i>	time	5	False		
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_Shift_ShiftID <i>Primary key (clustered) constraint</i>	ShiftID	True
	AK_Shift_Name <i>Unique nonclustered index.</i>	Name	True
	AK_Shift_StartTime_EndTime <i>Unique nonclustered index.</i>	StartTime, EndTime	True

SQL Script

```
CREATE TABLE [HumanResources].[Shift]
(
  [ShiftID] [tinyint] NOT NULL IDENTITY(1, 1),
  [Name] [dbo].[Name] NOT NULL,
  [StartTime] [time] NOT NULL,
  [EndTime] [time] NOT NULL,
```

```

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_Shift_ModifiedDate] DEFAULT
(GETDATE())
) ON [PRIMARY]
GO
ALTER TABLE [HumanResources].[Shift] ADD CONSTRAINT [PK_Shift_ShiftID] PRIMARY KEY
CLUSTERED ([ShiftID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_Shift_Name] ON [HumanResources].[Shift]
([Name]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_Shift_StartTime_EndTime] ON [Human-
Resources].[Shift] ([StartTime], [EndTime]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Work shift lookup table.',
'SHEMA', N'HumanResources', 'TABLE', N'Shift', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Shift end time.', 'SCHEMA', N'Human-
Resources', 'TABLE', N'Shift', 'COLUMN', N'EndTime'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'HumanResources', 'TABLE', N'Shift', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Shift description.', 'SCHEMA',
N'HumanResources', 'TABLE', N'Shift', 'COLUMN', N'Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Shift records.',
'SHEMA', N'HumanResources', 'TABLE', N'Shift', 'COLUMN', N'ShiftID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Shift start time.', 'SCHEMA',
N'HumanResources', 'TABLE', N'Shift', 'COLUMN', N'StartTime'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'HumanResources', 'TABLE', N'Shift', 'CONSTRAINT', N'DF_-
Shift_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'HumanResources', 'TABLE', N'Shift', 'CONSTRAINT', N'PK_-
Shift_ShiftID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SHEMA', N'HumanResources', 'TABLE', N'Shift', 'INDEX', N'AK_Shift_Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SHEMA', N'HumanResources', 'TABLE', N'Shift', 'INDEX', N'AK_Shift_StartTime_End-
Time'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'HumanResources', 'TABLE', N'Shift', 'INDEX',
N'PK_Shift_ShiftID'
GO

```

Uses

[dbo].[Name]
HumanResources

Used By

[HumanResources].[EmployeeDepartmentHistory]
[HumanResources].[vEmployeeDepartmentHistory]

 **[Person].[Address]**

MS_Description

Street address information for customers, employees, and vendors.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	19614
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Identity Replication	Default
	AddressID <i>Primary key for Address records.</i>	int	4	False	1 - 1	False	
	AddressLine1 <i>First street address line.</i>	nvarchar(60)	120	False			
	AddressLine2 <i>Second street address line.</i>	nvarchar(60)	120	True			
	City <i>Name of the city.</i>	nvarchar(30)	60	False			
	StateProvinceID <i>Unique identification number for the state or province. Foreign key to StateProvince table.</i>	int	4	False			
	PostalCode <i>Postal code for the street address.</i>	nvarchar(15)	30	False			
	SpatialLocation <i>Latitude and longitude of this address.</i>	geography	max	True			
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False			(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False			(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_Address_AddressID <i>Primary key (clustered) constraint</i>	AddressID	True
	AK_Address_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True
	IX_Address_AddressLine1_AddressLine2_City_StateProvinceID_PostalCode <i>Nonclustered index.</i>	AddressLine1, AddressLine2, City, StateProvinceID, PostalCode	True
	IX_Address_StateProvinceID <i>Nonclustered index.</i>	StateProvinceID	

Foreign Keys

Name	Columns
FK_Address_StateProvince_StateProvinceID <i>Foreign key constraint referencing StateProvince.StateProvinceID.</i>	StateProvinceID->[Person].[StateProvince].[StateProvinceID]

SQL Script

```
CREATE TABLE [Person].[Address]
(
  [AddressID] [int] NOT NULL IDENTITY(1, 1) NOT FOR REPLICATION,
  [AddressLine1] [nvarchar] (60) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
  [AddressLine2] [nvarchar] (60) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
  [City] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
  [StateProvinceID] [int] NOT NULL,
  [PostalCode] [nvarchar] (15) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
  [SpatialLocation] [sys].[geography] NULL,
  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_Address_rowguid]
  DEFAULT (newid()),
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_Address_ModifiedDate] DEFAULT
  (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Person].[Address] ADD CONSTRAINT [PK_Address_AddressID] PRIMARY KEY
  CLUSTERED ([AddressID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [IX_Address_AddressLine1_AddressLine2_City_State-
  ProvinceID_PostalCode] ON [Person].[Address] ([AddressLine1], [AddressLine2],
  [City], [StateProvinceID], [PostalCode]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_Address_rowguid] ON [Person].[Address]
  ([rowguid]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_Address_StateProvinceID] ON [Person].[Address] ([State-
  ProvinceID]) ON [PRIMARY]
GO
ALTER TABLE [Person].[Address] ADD CONSTRAINT [FK_Address_StateProvince_State-
  ProvinceID] FOREIGN KEY ([StateProvinceID]) REFERENCES [Person].[StateProvince]
  ([StateProvinceID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Street address information for
```

```

customers, employees, and vendors.', 'SCHEMA', N'Person', 'TABLE', N'Address', NULL,
NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Address records.',
'SCHEMA', N'Person', 'TABLE', N'Address', 'COLUMN', N'AddressID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'First street address line.',
'SCHEMA', N'Person', 'TABLE', N'Address', 'COLUMN', N'AddressLine1'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Second street address line.',
'SCHEMA', N'Person', 'TABLE', N'Address', 'COLUMN', N'AddressLine2'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Name of the city.', 'SCHEMA',
N'Person', 'TABLE', N'Address', 'COLUMN', N'City'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Person', 'TABLE', N'Address', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Postal code for the street
address.', 'SCHEMA', N'Person', 'TABLE', N'Address', 'COLUMN', N'PostalCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Person', 'TABLE', N'Address', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Latitude and longitude of this
address.', 'SCHEMA', N'Person', 'TABLE', N'Address', 'COLUMN', N'SpatialLocation'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique identification number for
the state or province. Foreign key to StateProvince table.', 'SCHEMA', N'Person',
'TABLE', N'Address', 'COLUMN', N'StateProvinceID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'Address', 'CONSTRAINT', N'DF_Address_-
ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Person', 'TABLE', N'Address', 'CONSTRAINT', N'DF_-
Address_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
StateProvince.StateProvinceID.', 'SCHEMA', N'Person', 'TABLE', N'Address',
'CONSTRAINT', N'FK_Address_StateProvince_StateProvinceID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Person', 'TABLE', N'Address', 'CONSTRAINT', N'PK_Address_-
AddressID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Person', 'TABLE', N'Address', 'INDEX',
N'AK_Address_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Person', 'TABLE', N'Address', 'INDEX', N'IX_Address_AddressLine1_AddressLine2_-
City_StateProvinceID_PostalCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Person', 'TABLE', N'Address', 'INDEX', N'IX_Address_StateProvinceID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'Address', 'INDEX', N'PK_-
Address_AddressID'
GO

```

Uses

[Person].[StateProvince]
Person

Used By

[Person].[BusinessEntityAddress]
[Sales].[SalesOrderHeader]
[HumanResources].[vEmployee]
[Purchasing].[vVendorWithAddresses]
[Sales].[vIndividualCustomer]
[Sales].[vSalesPerson]
[Sales].[vStoreWithAddresses]

 **[Person].[AddressType]****MS_Description**

Types of addresses stored in the Address table.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	6
Created	13:14:19 14 marca 2012
Last Modified	13:14:53 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	AddressTypeID <i>Primary key for AddressType records.</i>	int	4	False	1 - 1	
	Name <i>Address type description. For example, Billing, Home, or Shipping.</i>	[dbo].[Name]	100	False		
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False		(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_AddressType_AddressTypeID <i>Primary key (clustered) constraint</i>	AddressType-ID	True
	AK_AddressType_Name <i>Unique nonclustered index.</i>	Name	True
	AK_AddressType_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True

SQL Script

```
CREATE TABLE [Person].[AddressType]
(
  [AddressTypeID] [int] NOT NULL IDENTITY(1, 1),
```

```

[Name] [dbo].[Name] NOT NULL,
[rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_AddressType_rowguid]
DEFAULT (newid()),
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_AddressType_ModifiedDate] DEFAULT
(getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Person].[AddressType] ADD CONSTRAINT [PK_AddressType_AddressTypeID]
PRIMARY KEY CLUSTERED ([AddressTypeID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_AddressType_Name] ON [Person].[AddressType]
([Name]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_AddressType_rowguid] ON [Person].[AddressType]
([rowguid]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Types of addresses stored in the
Address table.', 'SCHEMA', N'Person', 'TABLE', N'AddressType', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for AddressType
records.', 'SCHEMA', N'Person', 'TABLE', N'AddressType', 'COLUMN', N'AddressTypeID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Person', 'TABLE', N'AddressType', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Address type description. For
example, Billing, Home, or Shipping.', 'SCHEMA', N'Person', 'TABLE', N'AddressType',
'COLUMN', N'Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Person', 'TABLE', N'AddressType', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'AddressType', 'CONSTRAINT', N'DF_Address-
Type_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Person', 'TABLE', N'AddressType', 'CONSTRAINT', N'DF_Address-
Type_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Person', 'TABLE', N'AddressType', 'CONSTRAINT', N'PK_-
AddressType_AddressTypeID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SCHEMA', N'Person', 'TABLE', N'AddressType', 'INDEX', N'AK_AddressType_Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Person', 'TABLE', N'AddressType',
'INDEX', N'AK_AddressType_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'AddressType', 'INDEX',
N'PK_AddressType_AddressTypeID'
GO

```

Uses

[dbo].[Name]

Person

Used By

[Person].[BusinessEntityAddress]

[Purchasing].[vVendorWithAddresses]

[Sales].[vIndividualCustomer]

[Sales].[vStoreWithAddresses]

 **[Person].[BusinessEntity]****MS_Description**

Source of the ID that connects vendors, customers, and employees with address and contact information.

Properties

Property	Value
Row Count (~)	20777
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Identity Replication	Default
	BusinessEntityID <i>Primary key for all customers, vendors, and employees.</i>	int	4	False	1 - 1	False	
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False			(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False			(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_BusinessEntity_BusinessEntityID <i>Primary key (clustered) constraint</i>	BusinessEntityID	True
	AK_BusinessEntity_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True

SQL Script

```
CREATE TABLE [Person].[BusinessEntity]
(
  [BusinessEntityID] [int] NOT NULL IDENTITY(1, 1) NOT FOR REPLICATION,
  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_Business-
Entity_rowguid] DEFAULT (newid()),
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_BusinessEntity_ModifiedDate]
DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Person].[BusinessEntity] ADD CONSTRAINT [PK_BusinessEntity_Business-
```

```

EntityID] PRIMARY KEY CLUSTERED ([BusinessEntityID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_BusinessEntity_rowguid] ON [Person].[Business-
Entity] ([rowguid]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Source of the ID that connects
vendors, customers, and employees with address and contact information.', 'SCHEMA',
N'Person', 'TABLE', N'BusinessEntity', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for all customers,
vendors, and employees.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntity', 'COLUMN',
N'BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntity', 'COLUMN', N'Modified-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Person', 'TABLE', N'BusinessEntity', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntity', 'CONSTRAINT', N'DF_-
BusinessEntity_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntity', 'CONSTRAINT', N'DF_-
BusinessEntity_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntity', 'CONSTRAINT', N'PK_-
BusinessEntity_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntity',
'INDEX', N'AK_BusinessEntity_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntity', 'INDEX',
N'PK_BusinessEntity_BusinessEntityID'
GO

```

Uses

Person

Used By

[Person].[BusinessEntityAddress]

[Person].[BusinessEntityContact]

[Person].[Person]

[Purchasing].[Vendor]

[Sales].[Store]

[Person].[BusinessEntityAddress]

MS_Description

Cross-reference table mapping customers, vendors, and employees to their addresses.

Properties

Property	Value
Row Count (~)	19614
Created	13:14:19 14 marca 2012
Last Modified	13:14:53 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	BusinessEntityID <i>Primary key. Foreign key to Business-Entity.BusinessEntityID.</i>	int	4	False	
	AddressID <i>Primary key. Foreign key to Address.AddressID.</i>	int	4	False	
	AddressTypeID <i>Primary key. Foreign key to Address-Type.AddressTypeID.</i>	int	4	False	
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False	(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_BusinessEntityAddress_BusinessEntityID_AddressID_AddressTypeID <i>Primary key (clustered) constraint</i>	BusinessEntity-ID, AddressID, AddressType-ID	True
	AK_BusinessEntityAddress_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True
	IX_BusinessEntityAddress_AddressID <i>Nonclustered index.</i>	AddressID	
	IX_BusinessEntityAddress_AddressTypeID <i>Nonclustered index.</i>	AddressType-ID	

Foreign Keys

Name	Columns
FK_BusinessEntityAddress_Address_AddressID <i>Foreign key constraint referencing Address.AddressID.</i>	AddressID->[Person].[Address].[AddressID]
FK_BusinessEntityAddress_AddressType_AddressTypeID <i>Foreign key constraint referencing AddressType.AddressTypeID.</i>	AddressTypeID->[Person].[AddressType].[AddressTypeID]
FK_BusinessEntityAddress_BusinessEntity_BusinessEntityID <i>Foreign key constraint referencing BusinessEntity.BusinessEntityID.</i>	BusinessEntityID->[Person].[BusinessEntity].[BusinessEntityID]

SQL Script

```

CREATE TABLE [Person].[BusinessEntityAddress]
(
    [BusinessEntityID] [int] NOT NULL,
    [AddressID] [int] NOT NULL,
    [AddressTypeID] [int] NOT NULL,
    [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_BusinessEntityAddress_rowguid] DEFAULT (newid()),
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_BusinessEntityAddress_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Person].[BusinessEntityAddress] ADD CONSTRAINT [PK_BusinessEntityAddress_BusinessEntityID_AddressID_AddressTypeID] PRIMARY KEY CLUSTERED ([BusinessEntityID], [AddressID], [AddressTypeID]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_BusinessEntityAddress_AddressID] ON [Person].[BusinessEntityAddress] ([AddressID]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_BusinessEntityAddress_AddressTypeID] ON [Person].[BusinessEntityAddress] ([AddressTypeID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_BusinessEntityAddress_rowguid] ON [Person].[BusinessEntityAddress] ([rowguid]) ON [PRIMARY]
GO
ALTER TABLE [Person].[BusinessEntityAddress] ADD CONSTRAINT [FK_BusinessEntityAddress_Address_AddressID] FOREIGN KEY ([AddressID]) REFERENCES [Person].[Address] ([AddressID])
GO
ALTER TABLE [Person].[BusinessEntityAddress] ADD CONSTRAINT [FK_BusinessEntityAddress_AddressType_AddressTypeID] FOREIGN KEY ([AddressTypeID]) REFERENCES [Person].[AddressType] ([AddressTypeID])
GO
ALTER TABLE [Person].[BusinessEntityAddress] ADD CONSTRAINT [FK_BusinessEntityAddress_BusinessEntity_BusinessEntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [Person].[BusinessEntity] ([BusinessEntityID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Cross-reference table mapping customers, vendors, and employees to their addresses.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityAddress', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Foreign key to Address.AddressID.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityAddress', 'COLUMN', N'AddressID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Foreign key to Address-

```

```

Type.AddressTypeID.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityAddress',
'COLUMN', N'AddressTypeID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Foreign key to
BusinessEntity.BusinessEntityID.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntity-
Address', 'COLUMN', N'BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityAddress', 'COLUMN',
N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Person', 'TABLE', N'BusinessEntityAddress', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityAddress', 'CONSTRAINT',
N'DF_BusinessEntityAddress_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityAddress', 'CONSTRAINT',
N'DF_BusinessEntityAddress_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Address.AddressID.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityAddress',
'CONSTRAINT', N'FK_BusinessEntityAddress_Address_AddressID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
AddressType.AddressTypeID.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityAddress',
'CONSTRAINT', N'FK_BusinessEntityAddress_AddressType_AddressTypeID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
BusinessEntity.BusinessEntityID.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntity-
Address', 'CONSTRAINT', N'FK_BusinessEntityAddress_BusinessEntity_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityAddress', 'CONSTRAINT',
N'PK_BusinessEntityAddress_BusinessEntityID_AddressID_AddressTypeID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntity-
Address', 'INDEX', N'AK_BusinessEntityAddress_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Person', 'TABLE', N'BusinessEntityAddress', 'INDEX', N'IX_BusinessEntityAddress_-
AddressID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Person', 'TABLE', N'BusinessEntityAddress', 'INDEX', N'IX_BusinessEntityAddress_-
AddressTypeID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityAddress',
'INDEX', N'PK_BusinessEntityAddress_BusinessEntityID_AddressID_AddressTypeID'
GO

```

Uses

[Person].[Address]
 [Person].[AddressType]
 [Person].[BusinessEntity]
 Person

Used By

[HumanResources].[vEmployee]
[Purchasing].[vVendorWithAddresses]
[Sales].[vIndividualCustomer]
[Sales].[vSalesPerson]
[Sales].[vStoreWithAddresses]

[Person].[BusinessEntityContact]

MS_Description

Cross-reference table mapping stores, vendors, and employees to people

Properties

Property	Value
Row Count (~)	909
Created	13:14:19 14 marca 2012
Last Modified	13:14:53 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	BusinessEntityID <i>Primary key. Foreign key to Business-Entity.BusinessEntityID.</i>	int	4	False	
	PersonID <i>Primary key. Foreign key to Person.BusinessEntityID.</i>	int	4	False	
	ContactTypeID <i>Primary key. Foreign key to Contact-Type.ContactTypeID.</i>	int	4	False	
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False	(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_BusinessEntityContact_BusinessEntityID_PersonID_ContactTypeID <i>Primary key (clustered) constraint</i>	BusinessEntity-ID, PersonID, ContactTypeID	True
	AK_BusinessEntityContact_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True
	IX_BusinessEntityContact_ContactTypeID <i>Nonclustered index.</i>	ContactTypeID	
	IX_BusinessEntityContact_PersonID <i>Nonclustered index.</i>	PersonID	

Foreign Keys

Name	Columns
FK_BusinessEntityContact_BusinessEntity_BusinessEntityID <i>Foreign key constraint referencing BusinessEntity.BusinessEntityID.</i>	BusinessEntityID->[Person].[BusinessEntity].[BusinessEntityID]
FK_BusinessEntityContact_ContactType_ContactTypeID <i>Foreign key constraint referencing ContactType.ContactTypeID.</i>	ContactTypeID->[Person].[ContactType].[ContactTypeID]
FK_BusinessEntityContact_Person_PersonID <i>Foreign key constraint referencing Person.BusinessEntityID.</i>	PersonID->[Person].[Person].[BusinessEntityID]

SQL Script

```

CREATE TABLE [Person].[BusinessEntityContact]
(
    [BusinessEntityID] [int] NOT NULL,
    [PersonID] [int] NOT NULL,
    [ContactTypeID] [int] NOT NULL,
    [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_BusinessEntity-Contact_rowguid] DEFAULT (newid()),
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_BusinessEntityContact_Modified-Date] DEFAULT (getdate())
) ON [PRIMARY]
GO

ALTER TABLE [Person].[BusinessEntityContact] ADD CONSTRAINT [PK_BusinessEntity-Contact_BusinessEntityID_PersonID_ContactTypeID] PRIMARY KEY CLUSTERED ([BusinessEntityID], [PersonID], [ContactTypeID]) ON [PRIMARY]
GO

CREATE NONCLUSTERED INDEX [IX_BusinessEntityContact_ContactTypeID] ON [Person].[BusinessEntityContact] ([ContactTypeID]) ON [PRIMARY]
GO

CREATE NONCLUSTERED INDEX [IX_BusinessEntityContact_PersonID] ON [Person].[BusinessEntityContact] ([PersonID]) ON [PRIMARY]
GO

CREATE UNIQUE NONCLUSTERED INDEX [AK_BusinessEntityContact_rowguid] ON [Person].[BusinessEntityContact] ([rowguid]) ON [PRIMARY]
GO

ALTER TABLE [Person].[BusinessEntityContact] ADD CONSTRAINT [FK_BusinessEntity-Contact_BusinessEntity_BusinessEntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [Person].[BusinessEntity] ([BusinessEntityID])
GO

ALTER TABLE [Person].[BusinessEntityContact] ADD CONSTRAINT [FK_BusinessEntity-Contact_ContactType_ContactTypeID] FOREIGN KEY ([ContactTypeID]) REFERENCES [Person].[ContactType] ([ContactTypeID])
GO

ALTER TABLE [Person].[BusinessEntityContact] ADD CONSTRAINT [FK_BusinessEntity-Contact_Person_PersonID] FOREIGN KEY ([PersonID]) REFERENCES [Person].[Person] ([BusinessEntityID])
GO

EXEC sp_addextendedproperty N'MS_Description', N'Cross-reference table mapping stores, vendors, and employees to people', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityContact', NULL, NULL
GO

EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Foreign key to BusinessEntity.BusinessEntityID.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityContact', 'COLUMN', N'BusinessEntityID'
GO

```

```

EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Foreign key to
ContactType.ContactTypeID.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityContact',
'COLUMN', N'ContactTypeID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityContact', 'COLUMN',
N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Foreign key to
Person.BusinessEntityID.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityContact',
'COLUMN', N'PersonID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Person', 'TABLE', N'BusinessEntityContact', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityContact', 'CONSTRAINT',
N'DF_BusinessEntityContact_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityContact', 'CONSTRAINT',
N'DF_BusinessEntityContact_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
BusinessEntity.BusinessEntityID.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntity-
Contact', 'CONSTRAINT', N'FK_BusinessEntityContact_BusinessEntity_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
ContactType.ContactTypeID.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityContact',
'CONSTRAINT', N'FK_BusinessEntityContact_ContactType_ContactTypeID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Person.BusinessEntityID.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityContact',
'CONSTRAINT', N'FK_BusinessEntityContact_Person_PersonID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityContact', 'CONSTRAINT',
N'PK_BusinessEntityContact_BusinessEntityID_PersonID_ContactTypeID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntity-
Contact', 'INDEX', N'AK_BusinessEntityContact_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Person', 'TABLE', N'BusinessEntityContact', 'INDEX', N'IX_BusinessEntityContact_-
ContactTypeID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Person', 'TABLE', N'BusinessEntityContact', 'INDEX', N'IX_BusinessEntityContact_-
PersonID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityContact',
'INDEX', N'PK_BusinessEntityContact_BusinessEntityID_PersonID_ContactTypeID'
GO

```

Uses

[Person].[BusinessEntity]

[Person].[ContactType]

[Person].[Person]

Person

Used By

[Purchasing].[vVendorWithContacts]

[Sales].[vStoreWithContacts]

[dbo].[ufnGetContactInformation]

 **[Person].[ContactType]****MS_Description**

Lookup table containing the types of business entity contacts.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	20
Created	13:14:19 14 marca 2012
Last Modified	13:14:53 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	ContactTypeID <i>Primary key for ContactType records.</i>	int	4	False	1 - 1	
	Name <i>Contact type description.</i>	[dbo].[Name]	100	False		
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_ContactType_ContactTypeID <i>Primary key (clustered) constraint</i>	ContactTypeID	True
	AK_ContactType_Name <i>Unique nonclustered index.</i>	Name	True

SQL Script

```
CREATE TABLE [Person].[ContactType]
(
    [ContactTypeID] [int] NOT NULL IDENTITY(1, 1),
    [Name] [dbo].[Name] NOT NULL,
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_ContactType_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Person].[ContactType] ADD CONSTRAINT [PK_ContactType_ContactTypeID]
PRIMARY KEY CLUSTERED ([ContactTypeID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_ContactType_Name] ON [Person].[ContactType]
```

```
([Name]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Lookup table containing the types
of business entity contacts.', 'SCHEMA', N'Person', 'TABLE', N'ContactType', NULL,
NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for ContactType
records.', 'SCHEMA', N'Person', 'TABLE', N'ContactType', 'COLUMN', N'ContactTypeID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Person', 'TABLE', N'ContactType', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Contact type description.',
'SCHEMA', N'Person', 'TABLE', N'ContactType', 'COLUMN', N'Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'ContactType', 'CONSTRAINT', N'DF_Contact-
Type_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Person', 'TABLE', N'ContactType', 'CONSTRAINT', N'PK_-
ContactType_ContactTypeID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SCHEMA', N'Person', 'TABLE', N'ContactType', 'INDEX', N'AK_ContactType_Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'ContactType', 'INDEX',
N'PK_ContactType_ContactTypeID'
GO
```

Uses

[dbo].[Name]

Person

Used By

[Person].[BusinessEntityContact]

[Purchasing].[vVendorWithContacts]

[Sales].[vStoreWithContacts]

[dbo].[ufnGetContactInformation]

 **[Person].[CountryRegion]****MS_Description**

Lookup table containing the ISO standard codes for countries and regions.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	238
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	CountryRegionCode <i>ISO standard code for countries and regions.</i>	nvarchar(3)	6	False	
	Name <i>Country or region name.</i>	[dbo].[Name]	100	False	
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_CountryRegion_CountryRegionCode <i>Primary key (clustered) constraint</i>	CountryRegionCode	True
	AK_CountryRegion_Name <i>Unique nonclustered index.</i>	Name	True

SQL Script

```
CREATE TABLE [Person].[CountryRegion]
(
  [CountryRegionCode] [nvarchar] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
  [Name] [dbo].[Name] NOT NULL,
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_CountryRegion_ModifiedDate]
  DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Person].[CountryRegion] ADD CONSTRAINT [PK_CountryRegion_CountryRegion-
Code] PRIMARY KEY CLUSTERED ([CountryRegionCode]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_CountryRegion_Name] ON [Person].[CountryRegion]
```

```
([Name]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Lookup table containing the ISO standard codes for countries and regions.', 'SCHEMA', N'Person', 'TABLE', N'CountryRegion', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'ISO standard code for countries and regions.', 'SCHEMA', N'Person', 'TABLE', N'CountryRegion', 'COLUMN', N'CountryRegionCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last updated.', 'SCHEMA', N'Person', 'TABLE', N'CountryRegion', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Country or region name.', 'SCHEMA', N'Person', 'TABLE', N'CountryRegion', 'COLUMN', N'Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'CountryRegion', 'CONSTRAINT', N'DF_CountryRegion_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Person', 'TABLE', N'CountryRegion', 'CONSTRAINT', N'PK_CountryRegion_CountryRegionCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.', 'SCHEMA', N'Person', 'TABLE', N'CountryRegion', 'INDEX', N'AK_CountryRegion_Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'CountryRegion', 'INDEX', N'PK_CountryRegion_CountryRegionCode'
GO
```

Uses

[dbo].[Name]
Person

Used By

[Person].[StateProvince]
[Sales].[CountryRegionCurrency]
[Sales].[SalesTerritory]
[HumanResources].[vEmployee]
[Person].[vStateProvinceCountryRegion]
[Purchasing].[vVendorWithAddresses]
[Sales].[vIndividualCustomer]
[Sales].[vSalesPerson]
[Sales].[vStoreWithAddresses]

 **[Person].[EmailAddress]**

MS_Description

Where to send a person email.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	19972
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	BusinessEntityID <i>Primary key. Person associated with this email address. Foreign key to Person.BusinessEntityID</i>	int	4	False		
	EmailAddressID <i>Primary key. ID of this email address.</i>	int	4	False	1 - 1	
	EmailAddress <i>E-mail address for the person.</i>	nvarchar(50)	100	True		
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False		(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_EmailAddress_BusinessEntityID_EmailAddressID <i>Primary key (clustered) constraint</i>	BusinessEntityID, EmailAddressID	True
	IX_EmailAddress_EmailAddress <i>Nonclustered index.</i>	EmailAddress	

Foreign Keys

Name	Columns
FK_EmailAddress_Person_BusinessEntityID <i>Foreign key constraint referencing Person.Business-</i>	BusinessEntityID->[Person].[Person].[BusinessEntityID]

EntityID.	
-----------	--

SQL Script

```

CREATE TABLE [Person].[EmailAddress]
(
    [BusinessEntityID] [int] NOT NULL,
    [EmailAddressID] [int] NOT NULL IDENTITY(1, 1),
    [EmailAddress] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_Email-
Address_rowguid] DEFAULT (newid()),
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_EmailAddress_ModifiedDate] DEFAULT
(getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Person].[EmailAddress] ADD CONSTRAINT [PK_EmailAddress_BusinessEntity-
ID_EmailAddressID] PRIMARY KEY CLUSTERED ([BusinessEntityID], [EmailAddressID]) ON
[PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_EmailAddress_EmailAddress] ON [Person].[EmailAddress]
([EmailAddress]) ON [PRIMARY]
GO
ALTER TABLE [Person].[EmailAddress] ADD CONSTRAINT [FK_EmailAddress_Person_Business-
EntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [Person].[Person] ([Business-
EntityID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Where to send a person email.',
'SHEMA', N'Person', 'TABLE', N'EmailAddress', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Person associated with
this email address. Foreign key to Person.BusinessEntityID', 'SCHEMA', N'Person',
'TABLE', N'EmailAddress', 'COLUMN', N'BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'E-mail address for the person.',
'SHEMA', N'Person', 'TABLE', N'EmailAddress', 'COLUMN', N'EmailAddress'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. ID of this email
address.', 'SCHEMA', N'Person', 'TABLE', N'EmailAddress', 'COLUMN', N'EmailAddres-
sID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Person', 'TABLE', N'EmailAddress', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Person', 'TABLE', N'EmailAddress', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'EmailAddress', 'CONSTRAINT', N'DF_Email-
Address_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Person', 'TABLE', N'EmailAddress', 'CONSTRAINT', N'DF_Email-
Address_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Person.BusinessEntityID.', 'SCHEMA', N'Person', 'TABLE', N'EmailAddress',
'CONSTRAINT', N'FK_EmailAddress_Person_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Person', 'TABLE', N'EmailAddress', 'CONSTRAINT', N'PK_Email-
Address_BusinessEntityID_EmailAddressID'

```

```
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Person', 'TABLE', N'EmailAddress', 'INDEX', N'IX_EmailAddress_EmailAddress'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'EmailAddress', 'INDEX',
N'PK_EmailAddress_BusinessEntityID_EmailAddressID'
GO
```

Uses

[Person].[Person]
Person

Used By

[HumanResources].[vEmployee]
[Purchasing].[vVendorWithContacts]
[Sales].[vIndividualCustomer]
[Sales].[vSalesPerson]
[Sales].[vStoreWithContacts]

 **[Person].[Password]**
MS_Description

One way hashed authentication information

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	19972
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	BusinessEntityID	int	4	False	
	PasswordHash <i>Password for the e-mail account.</i>	varchar(128)	128	False	
	PasswordSalt <i>Random value concatenated with the password string before the password is hashed.</i>	varchar(10)	10	False	
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False	(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_Password_BusinessEntityID <i>Primary key (clustered) constraint</i>	BusinessEntityID	True

Foreign Keys

Name	Columns
FK_Password_Person_BusinessEntityID <i>Foreign key constraint referencing Person.BusinessEntityID.</i>	BusinessEntityID->[Person].[Person].[BusinessEntityID]

SQL Script

```

CREATE TABLE [Person].[Password]
(
    [BusinessEntityID] [int] NOT NULL,
    [PasswordHash] [varchar] (128) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [PasswordSalt] [varchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_Password_rowguid]
    DEFAULT (newid()),
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_Password_ModifiedDate] DEFAULT
    (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Person].[Password] ADD CONSTRAINT [PK_Password_BusinessEntityID]
PRIMARY KEY CLUSTERED ([BusinessEntityID]) ON [PRIMARY]
GO
ALTER TABLE [Person].[Password] ADD CONSTRAINT [FK_Password_Person_BusinessEntityID]
FOREIGN KEY ([BusinessEntityID]) REFERENCES [Person].[Person] ([BusinessEntityID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'One way hashed authentication
information', 'SCHEMA', N'Person', 'TABLE', N'Password', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Person', 'TABLE', N'Password', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Password for the e-mail account.',
'SCHEMA', N'Person', 'TABLE', N'Password', 'COLUMN', N'PasswordHash'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Random value concatenated with the
password string before the password is hashed.', 'SCHEMA', N'Person', 'TABLE',
N'Password', 'COLUMN', N'PasswordSalt'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Person', 'TABLE', N'Password', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'Password', 'CONSTRAINT', N'DF_Password_
ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Person', 'TABLE', N'Password', 'CONSTRAINT', N'DF_
Password_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Person.BusinessEntityID.', 'SCHEMA', N'Person', 'TABLE', N'Password', 'CONSTRAINT',
N'FK_Password_Person_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Person', 'TABLE', N'Password', 'CONSTRAINT', N'PK_Password_
BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'Password', 'INDEX', N'PK_
Password_BusinessEntityID'
GO

```

Uses

[Person].[Person]

Person

 **[Person].[Person]**
MS_Description

Human beings involved with AdventureWorks: employees, customer contacts, and vendor contacts.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	19972
Created	13:14:19 14 marca 2012
Last Modified	15:26:58 22 marca 2017

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	BusinessEntityID <i>Primary key for Person records.</i>	int	4	False	
	PersonType <i>Primary type of person: SC = Store Contact, IN = Individual (retail) customer, SP = Sales person, EM = Employee (non-sales), VC = Vendor contact, GC = General contact</i>	nchar(2)	4	False	
	NameStyle <i>0 = The data in FirstName and LastName are stored in western style (first name, last name) order. 1 = Eastern style (last name, first name) order.</i>	[dbo].[NameStyle]	1	False	((0))
	Title <i>A courtesy title. For example, Mr. or Ms.</i>	nvarchar(8)	16	True	
	FirstName <i>First name of the person.</i>	[dbo].[Name]	100	False	
	MiddleName <i>Middle name or middle initial of the person.</i>	[dbo].[Name]	100	True	
	LastName <i>Last name of the person.</i>	[dbo].[Name]	100	False	
	Suffix <i>Surname suffix. For example, Sr. or Jr.</i>	nvarchar(10)	20	True	
	EmailPromotion <i>0 = Contact does not wish to receive e-mail promotions, 1 = Contact does wish to receive e-mail promotions from AdventureWorks, 2 = Contact does wish to receive e-mail promotions from AdventureWorks and selected partners.</i>	int	4	False	((0))
	AdditionalContactInfo <i>Additional contact information about the person stored in xml format.</i>	xml([Person].[AdditionalContactInfoSchema-Collection])	max	True	

 (4)	Demographics <i>Personal information such as hobbies, and income collected from online shoppers. Used for sales analysis.</i>	xml([Person].[Individual-SurveySchemaCollection])	max	True	
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False	(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Type	Unique	XML Type
	PK_Person_BusinessEntityID <i>Primary key (clustered) constraint</i>	Business-EntityID		True	
	AK_Person_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid		True	
	IX_Person_LastName_FirstName_MiddleName	LastName, FirstName, MiddleName			
	PXML_Person_AddContact <i>Primary XML index.</i>	Additional-ContactInfo	xml		Primary
	PXML_Person_Demographics <i>Primary XML index.</i>	Demographics	xml		Primary
	XMLPATH_Person_Demographics <i>Secondary XML index for path.</i>	Demographics	xml		Secondary
	XMLPROPERTY_Person_Demographics <i>Secondary XML index for property.</i>	Demographics	xml		Secondary
	XMLVALUE_Person_Demographics <i>Secondary XML index for value.</i>	Demographics	xml		Secondary

Triggers

Name	ANSI Nulls On	Quoted Identifier On	On	Not For Replication
iuPerson <i>AFTER INSERT, UPDATE trigger inserting Individual only if the Customer does not exist in the Store table and setting the ModifiedDate column in the Person table to the current date.</i>	True	True	After Insert Update	True

Check Constraints

Name	On Column	Constraint
CK_Person_EmailPromotion <i>Check constraint [EmailPromotion] >= (0) AND [EmailPromotion] <= (2)</i>	EmailPromotion	(([Email-Promotion]>=(0) AND [Email-Promotion]<=(2))
CK_Person_PersonType <i>Check constraint [PersonType] is one of SC, VC, IN, EM or SP.</i>	PersonType	(([PersonType] IS NULL OR

		upper([PersonType])='GC' OR upper([PersonType])='SP' OR upper([PersonType])='EM' OR upper([PersonType])='IN' OR upper([PersonType])='VC' OR upper([PersonType])='SC')
--	--	--

Foreign Keys

Name	Columns
FK_Person_BusinessEntity_BusinessEntityID <i>Foreign key constraint referencing BusinessEntity.BusinessEntityID.</i>	BusinessEntityID->[Person].[BusinessEntity].[BusinessEntityID]

SQL Script

```
CREATE TABLE [Person].[Person]
(
    [BusinessEntityID] [int] NOT NULL,
    [PersonType] [nchar] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [NameStyle] [dbo].[NameStyle] NOT NULL CONSTRAINT [DF_Person_NameStyle] DEFAULT ((0)),
    [Title] [nvarchar] (8) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [FirstName] [dbo].[Name] NOT NULL,
    [MiddleName] [dbo].[Name] NULL,
    [LastName] [dbo].[Name] NOT NULL,
    [Suffix] [nvarchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [EmailPromotion] [int] NOT NULL CONSTRAINT [DF_Person_EmailPromotion] DEFAULT ((0)),
    [AdditionalContactInfo] [xml] (CONTENT [Person].[AdditionalContactInfoSchemaCollection]) NULL,
    [Demographics] [xml] (CONTENT [Person].[IndividualSurveySchemaCollection]) NULL,
    [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_Person_rowguid] DEFAULT (newid()),
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_Person_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TRIGGER [Person].[iuPerson] ON [Person].[Person]
AFTER INSERT, UPDATE NOT FOR REPLICATION AS
BEGIN
    DECLARE @Count int;

    SET @Count = @@ROWCOUNT;
    IF @Count = 0
        RETURN;

    SET NOCOUNT ON;

    IF UPDATE([BusinessEntityID]) OR UPDATE([Demographics])
    BEGIN
        UPDATE [Person].[Person]
```

```

        SET [Person].[Person].[Demographics] = N'<IndividualSurvey
xmlns="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Individual-
Survey">
    <TotalPurchaseYTD>0.00</TotalPurchaseYTD>
</IndividualSurvey>'
FROM inserted
WHERE [Person].[Person].[BusinessEntityID] = inserted.[BusinessEntityID]
    AND inserted.[Demographics] IS NULL;

UPDATE [Person].[Person]
SET [Demographics].modify(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey";
insert <TotalPurchaseYTD>0.00</TotalPurchaseYTD>
as first
into (/IndividualSurvey)[1]')
FROM inserted
WHERE [Person].[Person].[BusinessEntityID] = inserted.[BusinessEntityID]
    AND inserted.[Demographics] IS NOT NULL
    AND inserted.[Demographics].exist(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-
works/IndividualSurvey";
/IndividualSurvey/TotalPurchaseYTD') <> 1;

END;
END;
GO
ALTER TABLE [Person].[Person] ADD CONSTRAINT [CK_Person_EmailPromotion] CHECK
(([EmailPromotion]>=(0) AND [EmailPromotion]<=(2)))
GO
ALTER TABLE [Person].[Person] ADD CONSTRAINT [CK_Person_PersonType] CHECK (([Person-
Type] IS NULL OR upper([PersonType])='GC' OR upper([PersonType])='SP' OR
upper([PersonType])='EM' OR upper([PersonType])='IN' OR upper([PersonType])='VC' OR
upper([PersonType])='SC'))
GO
ALTER TABLE [Person].[Person] ADD CONSTRAINT [PK_Person_BusinessEntityID] PRIMARY
KEY CLUSTERED ([BusinessEntityID]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_Person_LastName_FirstName_MiddleName] ON
[Person].[Person] ([LastName], [FirstName], [MiddleName]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_Person_rowguid] ON [Person].[Person]
([rowguid]) ON [PRIMARY]
GO
CREATE PRIMARY XML INDEX [PXML_Person_AddContact]
ON [Person].[Person] ([AdditionalContactInfo])
GO
CREATE PRIMARY XML INDEX [PXML_Person_Demographics]
ON [Person].[Person] ([Demographics])
GO
CREATE XML INDEX [XMLPATH_Person_Demographics]
ON [Person].[Person] ([Demographics])
USING XML INDEX [PXML_Person_Demographics]
FOR PATH
GO
CREATE XML INDEX [XMLPROPERTY_Person_Demographics]
ON [Person].[Person] ([Demographics])
USING XML INDEX [PXML_Person_Demographics]
FOR PROPERTY
GO
CREATE XML INDEX [XMLVALUE_Person_Demographics]
ON [Person].[Person] ([Demographics])

```

```

USING XML INDEX [PXML_Person_Demographics]
FOR VALUE
GO
ALTER TABLE [Person].[Person] ADD CONSTRAINT [FK_Person_BusinessEntity_Business-
EntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [Person].[BusinessEntity]
([BusinessEntityID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Human beings involved with
AdventureWorks: employees, customer contacts, and vendor contacts.', 'SCHEMA',
N'Person', 'TABLE', N'Person', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Additional contact information
about the person stored in xml format. ', 'SCHEMA', N'Person', 'TABLE', N'Person',
'COLUMN', N'AdditionalContactInfo'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Person records.',
'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Personal information such as
hobbies, and income collected from online shoppers. Used for sales analysis.',
'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'Demographics'
GO
EXEC sp_addextendedproperty N'MS_Description', N'0 = Contact does not wish to
receive e-mail promotions, 1 = Contact does wish to receive e-mail promotions from
AdventureWorks, 2 = Contact does wish to receive e-mail promotions from Adventure-
Works and selected partners. ', 'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN',
N'EmailPromotion'
GO
EXEC sp_addextendedproperty N'MS_Description', N'First name of the person.',
'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'FirstName'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Last name of the person.',
'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'LastName'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Middle name or middle initial of
the person.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'MiddleName'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'0 = The data in FirstName and Last-
Name are stored in western style (first name, last name) order. 1 = Eastern style
(last name, first name) order.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN',
N'NameStyle'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary type of person: SC = Store
Contact, IN = Individual (retail) customer, SP = Sales person, EM = Employee (non-
sales), VC = Vendor contact, GC = General contact', 'SCHEMA', N'Person', 'TABLE',
N'Person', 'COLUMN', N'PersonType'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Person', 'TABLE', N'Person', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Surname suffix. For example, Sr. or
Jr.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'Suffix'
GO
EXEC sp_addextendedproperty N'MS_Description', N'A courtesy title. For example, Mr.
or Ms.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'Title'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [EmailPromotion]
>= (0) AND [EmailPromotion] <= (2)', 'SCHEMA', N'Person', 'TABLE', N'Person',
'CONSTRAINT', N'CK_Person_EmailPromotion'
GO

```

```

EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [PersonType] is
one of SC, VC, IN, EM or SP.', 'SCHEMA', N'Person', 'TABLE', N'Person',
'CONSTRAINT', N'CK_Person_PersonType'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0',
'SCHEMA', N'Person', 'TABLE', N'Person', 'CONSTRAINT', N'DF_Person_EmailPromotion'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'Person', 'CONSTRAINT', N'DF_Person_
ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0',
'SCHEMA', N'Person', 'TABLE', N'Person', 'CONSTRAINT', N'DF_Person_NameStyle'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Person', 'TABLE', N'Person', 'CONSTRAINT', N'DF_
Person_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
BusinessEntity.BusinessEntityID.', 'SCHEMA', N'Person', 'TABLE', N'Person',
'CONSTRAINT', N'FK_Person_BusinessEntity_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Person', 'TABLE', N'Person', 'CONSTRAINT', N'PK_Person_
BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'INDEX',
N'AK_Person_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'INDEX', N'PK_
Person_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary XML index.', 'SCHEMA',
N'Person', 'TABLE', N'Person', 'INDEX', N'PXML_Person_AddContact'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary XML index.', 'SCHEMA',
N'Person', 'TABLE', N'Person', 'INDEX', N'PXML_Person_Demographics'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Secondary XML index for path.',
'SCHEMA', N'Person', 'TABLE', N'Person', 'INDEX', N'XMLPATH_Person_Demographics'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Secondary XML index for property.',
'SCHEMA', N'Person', 'TABLE', N'Person', 'INDEX', N'XMLPROPERTY_Person_Demographics'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Secondary XML index for value.',
'SCHEMA', N'Person', 'TABLE', N'Person', 'INDEX', N'XMLVALUE_Person_Demographics'
GO
EXEC sp_addextendedproperty N'MS_Description', N'AFTER INSERT, UPDATE trigger
inserting Individual only if the Customer does not exist in the Store table and
setting the ModifiedDate column in the Person table to the current date.', 'SCHEMA',
N'Person', 'TABLE', N'Person', 'TRIGGER', N'iuPerson'
GO

```

Uses

[Person].[BusinessEntity]

[dbo].[Name]

[dbo].[NameStyle]

Person

[Person].[AdditionalContactInfoSchemaCollection]

[Person].[IndividualSurveySchemaCollection]

Used By

[HumanResources].[Employee]

[Person].[BusinessEntityContact]

[Person].[EmailAddress]

[Person].[Password]

[Person].[PersonPhone]

[Sales].[Customer]

[Sales].[PersonCreditCard]

[HumanResources].[vEmployee]

[HumanResources].[vEmployeeDepartment]

[HumanResources].[vEmployeeDepartmentHistory]

[Person].[vAdditionalContactInfo]

[Purchasing].[vVendorWithContacts]

[Sales].[vIndividualCustomer]

[Sales].[vPersonDemographics]

[Sales].[vSalesPerson]

[Sales].[vSalesPersonSalesByFiscalYears]

[Sales].[vStoreWithContacts]

[dbo].[uspGetEmployeeManagers]

[dbo].[uspGetManagerEmployees]

[dbo].[ufnGetContactInformation]

 **[Person].[PersonPhone]**
MS_Description

Telephone number and type of a person.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	19972
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
 	BusinessEntityID <i>Business entity identification number. Foreign key to Person.BusinessEntityID.</i>	int	4	False	
	PhoneNumber <i>Telephone number identification number.</i>	[dbo].[Phone]	50	False	
 	PhoneNumberTypeID <i>Kind of phone number. Foreign key to PhoneNumberType.PhoneNumberTypeID.</i>	int	4	False	
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_PersonPhone_BusinessEntityID_PhoneNumber_PhoneNumberTypeID <i>Primary key (clustered) constraint</i>	BusinessEntityID, PhoneNumber, PhoneNumberTypeID	True
	IX_PersonPhone_PhoneNumber <i>Nonclustered index.</i>	PhoneNumber	

Foreign Keys

Name	Columns
FK_PersonPhone_Person_BusinessEntityID <i>Foreign key constraint referencing Person.BusinessEntityID.</i>	BusinessEntityID->[Person].[Person].[BusinessEntityID]

FK_PersonPhone_PhoneNumberType_PhoneNumberTypeID <i>Foreign key constraint referencing PhoneNumberType.PhoneNumberTypeID.</i>	PhoneNumberTypeID->[Person].[PhoneNumberType].[PhoneNumberTypeID]
--	---

SQL Script

```

CREATE TABLE [Person].[PersonPhone]
(
    [BusinessEntityID] [int] NOT NULL,
    [PhoneNumber] [dbo].[Phone] NOT NULL,
    [PhoneNumberTypeID] [int] NOT NULL,
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_PersonPhone_ModifiedDate] DEFAULT
    (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Person].[PersonPhone] ADD CONSTRAINT [PK_PersonPhone_BusinessEntityID_
PhoneNumber_PhoneNumberTypeID] PRIMARY KEY CLUSTERED ([BusinessEntityID], [Phone-
Number], [PhoneNumberTypeID]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_PersonPhone_PhoneNumber] ON [Person].[PersonPhone]
([PhoneNumber]) ON [PRIMARY]
GO
ALTER TABLE [Person].[PersonPhone] ADD CONSTRAINT [FK_PersonPhone_Person_Business-
EntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [Person].[Person] ([Business-
EntityID])
GO
ALTER TABLE [Person].[PersonPhone] ADD CONSTRAINT [FK_PersonPhone_PhoneNumberType_
PhoneNumberTypeID] FOREIGN KEY ([PhoneNumberTypeID]) REFERENCES [Person].[Phone-
NumberType] ([PhoneNumberTypeID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Telephone number and type of a
person.', 'SCHEMA', N'Person', 'TABLE', N'PersonPhone', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Business entity identification
number. Foreign key to Person.BusinessEntityID.', 'SCHEMA', N'Person', 'TABLE',
N'PersonPhone', 'COLUMN', N'BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Person', 'TABLE', N'PersonPhone', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Telephone number identification
number.', 'SCHEMA', N'Person', 'TABLE', N'PersonPhone', 'COLUMN', N'PhoneNumber'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Kind of phone number. Foreign key
to PhoneNumberType.PhoneNumberTypeID.', 'SCHEMA', N'Person', 'TABLE', N'Person-
Phone', 'COLUMN', N'PhoneNumberTypeID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'PersonPhone', 'CONSTRAINT', N'DF_Person-
Phone_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Person.BusinessEntityID.', 'SCHEMA', N'Person', 'TABLE', N'PersonPhone',
'CONSTRAINT', N'FK_PersonPhone_Person_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
PhoneNumberType.PhoneNumberTypeID.', 'SCHEMA', N'Person', 'TABLE', N'PersonPhone',
'CONSTRAINT', N'FK_PersonPhone_PhoneNumberType_PhoneNumberTypeID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Person', 'TABLE', N'PersonPhone', 'CONSTRAINT', N'PK_Person-

```

```
Phone_BusinessEntityID_PhoneNumber_PhoneNumberTypeID'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',  
N'Person', 'TABLE', N'PersonPhone', 'INDEX', N'IX_PersonPhone_PhoneNumber'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a  
primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'PersonPhone', 'INDEX',  
N'PK_PersonPhone_BusinessEntityID_PhoneNumber_PhoneNumberTypeID'  
GO
```

Uses

[Person].[Person]
[Person].[PhoneNumberType]
[dbo].[Phone]
Person

Used By

[HumanResources].[vEmployee]
[Purchasing].[vVendorWithContacts]
[Sales].[vIndividualCustomer]
[Sales].[vSalesPerson]
[Sales].[vStoreWithContacts]

[Person].[PhoneNumberType]**MS_Description**

Type of phone number of a person.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	3
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	PhoneNumberTypeID <i>Primary key for telephone number type records.</i>	int	4	False	1 - 1	
	Name <i>Name of the telephone number type</i>	[dbo].[Name]	100	False		
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_PhoneNumberType_PhoneNumberTypeID <i>Primary key (clustered) constraint</i>	PhoneNumberTypeID	True

SQL Script

```
CREATE TABLE [Person].[PhoneNumberType]
(
  [PhoneNumberTypeID] [int] NOT NULL IDENTITY(1, 1),
  [Name] [dbo].[Name] NOT NULL,
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_PhoneNumberType_ModifiedDate]
  DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Person].[PhoneNumberType] ADD CONSTRAINT [PK_PhoneNumberType_Phone-
NumberTypeID] PRIMARY KEY CLUSTERED ([PhoneNumberTypeID]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Type of phone number of a person.',
'SHEMA', N'Person', 'TABLE', N'PhoneNumberType', NULL, NULL
```

```
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Person', 'TABLE', N'PhoneNumberType', 'COLUMN', N'Modified-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Name of the telephone number type',
'SCHEMA', N'Person', 'TABLE', N'PhoneNumberType', 'COLUMN', N'Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for telephone number
type records.', 'SCHEMA', N'Person', 'TABLE', N'PhoneNumberType', 'COLUMN', N'Phone-
NumberTypeID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'PhoneNumberType', 'CONSTRAINT', N'DF_-
PhoneNumberType_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Person', 'TABLE', N'PhoneNumberType', 'CONSTRAINT', N'PK_-
PhoneNumberType_PhoneNumberTypeID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'PhoneNumberType', 'INDEX',
N'PK_PhoneNumberType_PhoneNumberTypeID'
GO
```

Uses

[dbo].[Name]
Person

Used By

[Person].[PersonPhone]
[HumanResources].[vEmployee]
[Purchasing].[vVendorWithContacts]
[Sales].[vIndividualCustomer]
[Sales].[vSalesPerson]
[Sales].[vStoreWithContacts]

 **[Person].[StateProvince]**
MS_Description

State and province lookup table.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	181
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	StateProvinceID <i>Primary key for StateProvince records.</i>	int	4	False	1 - 1	
	StateProvinceCode <i>ISO standard state or province code.</i>	nchar(3)	6	False		
	CountryRegionCode <i>ISO standard country or region code. Foreign key to Country-Region.CountryRegionCode.</i>	nvarchar(3)	6	False		
	IsOnlyStateProvinceFlag <i>0 = StateProvinceCode exists. 1 = StateProvinceCode unavailable, using Country-RegionCode.</i>	[dbo].[Flag]	1	False		((1))
	Name <i>State or province description.</i>	[dbo].[Name]	100	False		
	TerritoryID <i>ID of the territory in which the state or province is located. Foreign key to Sales-Territory.SalesTerritoryID.</i>	int	4	False		
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False		(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
-----	------	-------------	--------

	PK_StateProvince_StateProvinceID <i>Primary key (clustered) constraint</i>	StateProvince- ID	True
	AK_StateProvince_Name <i>Unique nonclustered index.</i>	Name	True
	AK_StateProvince_StateProvinceCode_CountryRegionCode <i>Unique nonclustered index.</i>	StateProvince- Code, Country- RegionCode	True
	AK_StateProvince_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True

Foreign Keys

Name	Columns
FK_StateProvince_CountryRegion_CountryRegionCode <i>Foreign key constraint referencing Country-Region.CountryRegionCode.</i>	CountryRegionCode->[Person].[Country-Region].[CountryRegionCode]
FK_StateProvince_SalesTerritory_TerritoryID <i>Foreign key constraint referencing Sales-Territory.TerritoryID.</i>	TerritoryID->[Sales].[SalesTerritory].[TerritoryID]

SQL Script

```

CREATE TABLE [Person].[StateProvince]
(
    [StateProvinceID] [int] NOT NULL IDENTITY(1, 1),
    [StateProvinceCode] [nchar] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [CountryRegionCode] [nvarchar] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [IsOnlyStateProvinceFlag] [dbo].[Flag] NOT NULL CONSTRAINT [DF_StateProvince_IsOnlyStateProvinceFlag] DEFAULT ((1)),
    [Name] [dbo].[Name] NOT NULL,
    [TerritoryID] [int] NOT NULL,
    [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_StateProvince_rowguid] DEFAULT (newid()),
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_StateProvince_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Person].[StateProvince] ADD CONSTRAINT [PK_StateProvince_StateProvinceID] PRIMARY KEY CLUSTERED ([StateProvinceID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_StateProvince_Name] ON [Person].[StateProvince] ([Name]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_StateProvince_rowguid] ON [Person].[StateProvince] ([rowguid]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_StateProvince_StateProvinceCode_CountryRegionCode] ON [Person].[StateProvince] ([StateProvinceCode], [CountryRegionCode]) ON [PRIMARY]
GO
ALTER TABLE [Person].[StateProvince] ADD CONSTRAINT [FK_StateProvince_CountryRegion_CountryRegionCode] FOREIGN KEY ([CountryRegionCode]) REFERENCES [Person].[CountryRegion] ([CountryRegionCode])
GO
ALTER TABLE [Person].[StateProvince] ADD CONSTRAINT [FK_StateProvince_SalesTerritory_TerritoryID] FOREIGN KEY ([TerritoryID]) REFERENCES [Sales].[SalesTerritory] ([TerritoryID])

```

```

GO
EXEC sp_addextendedproperty N'MS_Description', N'State and province lookup table.',
'SHEMA', N'Person', 'TABLE', N'StateProvince', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'ISO standard country or region
code. Foreign key to CountryRegion.CountryRegionCode.', 'SCHEMA', N'Person',
'TABLE', N'StateProvince', 'COLUMN', N'CountryRegionCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'0 = StateProvinceCode exists. 1 =
StateProvinceCode unavailable, using CountryRegionCode.', 'SCHEMA', N'Person',
'TABLE', N'StateProvince', 'COLUMN', N'IsOnlyStateProvinceFlag'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Person', 'TABLE', N'StateProvince', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'State or province description.',
'SHEMA', N'Person', 'TABLE', N'StateProvince', 'COLUMN', N'Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Person', 'TABLE', N'StateProvince', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ISO standard state or province
code.', 'SCHEMA', N'Person', 'TABLE', N'StateProvince', 'COLUMN', N'StateProvince-
Code'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for StateProvince
records.', 'SCHEMA', N'Person', 'TABLE', N'StateProvince', 'COLUMN', N'StateProvince-
ID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ID of the territory in which the
state or province is located. Foreign key to SalesTerritory.SalesTerritoryID.',
'SHEMA', N'Person', 'TABLE', N'StateProvince', 'COLUMN', N'TerritoryID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 1
(TRUE)', 'SCHEMA', N'Person', 'TABLE', N'StateProvince', 'CONSTRAINT', N'DF_State-
Province_IsOnlyStateProvinceFlag'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'StateProvince', 'CONSTRAINT', N'DF_State-
Province_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Person', 'TABLE', N'StateProvince', 'CONSTRAINT', N'DF_State-
Province_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
CountryRegion.CountryRegionCode.', 'SCHEMA', N'Person', 'TABLE', N'StateProvince',
'CONSTRAINT', N'FK_StateProvince_CountryRegion_CountryRegionCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
SalesTerritory.TerritoryID.', 'SCHEMA', N'Person', 'TABLE', N'StateProvince',
'CONSTRAINT', N'FK_StateProvince_SalesTerritory_TerritoryID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Person', 'TABLE', N'StateProvince', 'CONSTRAINT', N'PK_-
StateProvince_StateProvinceID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SHEMA', N'Person', 'TABLE', N'StateProvince', 'INDEX', N'AK_StateProvince_Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Person', 'TABLE', N'StateProvince',
'INDEX', N'AK_StateProvince_rowguid'

```

```
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SHEMA', N'Person', 'TABLE', N'StateProvince', 'INDEX', N'AK_StateProvince_State-
ProvinceCode_CountryRegionCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'StateProvince', 'INDEX',
N'PK_StateProvince_StateProvinceID'
GO
```

Uses

[Person].[CountryRegion]
[Sales].[SalesTerritory]
[dbo].[Flag]
[dbo].[Name]
Person

Used By

[Person].[Address]
[Sales].[SalesTaxRate]
[HumanResources].[vEmployee]
[Person].[vStateProvinceCountryRegion]
[Purchasing].[vVendorWithAddresses]
[Sales].[vIndividualCustomer]
[Sales].[vSalesPerson]
[Sales].[vStoreWithAddresses]

 **[Production].[BillOfMaterials]**

MS_Description

Items required to make bicycles and bicycle subassemblies. It identifies the heirarchical relationship between a parent product and its components.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	2679
Created	13:14:19 14 marca 2012
Last Modified	13:14:53 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	BillOfMaterialsID <i>Primary key for BillOfMaterials records.</i>	int	4	False	1 - 1	
	ProductAssemblyID <i>Parent product identification number. Foreign key to Product.ProductID.</i>	int	4	True		
	ComponentID <i>Component identification number. Foreign key to Product.ProductID.</i>	int	4	False		
	StartDate <i>Date the component started being used in the assembly item.</i>	datetime	8	False		(getdate())
	EndDate <i>Date the component stopped being used in the assembly item.</i>	datetime	8	True		
	UnitMeasureCode <i>Standard code identifying the unit of measure for the quantity.</i>	nchar(3)	6	False		
	BOMLevel <i>Indicates the depth the component is from its parent (AssemblyID).</i>	smallint	2	False		
	PerAssemblyQty <i>Quantity of the component needed to create the assembly.</i>	decimal(8,2)	5	False		((1.00))
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_BillOfMaterials_BillOfMaterialsID <i>Primary key (clustered) constraint</i>	BillOfMaterials-ID	True
	AK_BillOfMaterials_ProductAssemblyID_ComponentID_StartDate <i>Clustered index.</i>	Product-AssemblyID, ComponentID, StartDate	True
	IX_BillOfMaterials_UnitMeasureCode <i>Nonclustered index.</i>	UnitMeasure-Code	

Check Constraints

Name	On Column	Constraint
CK_BillOfMaterials_EndDate <i>Check constraint [EndDate] > [StartDate] OR [EndDate] IS NULL</i>		([EndDate]>[StartDate] OR [EndDate] IS NULL)
CK_BillOfMaterials_PerAssemblyQty <i>Check constraint [PerAssemblyQty] >= (1.00)</i>	PerAssembly-Qty	([PerAssembly-Qty]>=(1.00))
CK_BillOfMaterials_BOMLevel <i>Check constraint [ProductAssemblyID] IS NULL AND [BOMLevel] = (0) AND [PerAssemblyQty] = (1) OR [ProductAssemblyID] IS NOT NULL AND [BOMLevel] >= (1)</i>		([Product-AssemblyID] IS NULL AND [BOMLevel]=0) AND [Per-Assembly-Qty]=(1.00) OR [Product-AssemblyID] IS NOT NULL AND [BOMLevel]>=(1))
CK_BillOfMaterials_ProductAssemblyID <i>Check constraint [ProductAssemblyID] <> [ComponentID]</i>		([Product-AssemblyID]<>[ComponentID])

Foreign Keys

Name	Columns
FK_BillOfMaterials_Product_ComponentID <i>Foreign key constraint referencing Product.ComponentID.</i>	ComponentID->[Production].[Product].[ProductID]
FK_BillOfMaterials_Product_ProductAssemblyID <i>Foreign key constraint referencing Product.ProductAssemblyID.</i>	ProductAssemblyID->[Production].[Product].[ProductID]
FK_BillOfMaterials_UnitMeasure_UnitMeasureCode <i>Foreign key constraint referencing UnitMeasure.UnitMeasureCode.</i>	UnitMeasureCode->[Production].[UnitMeasure].[UnitMeasureCode]

SQL Script

```
CREATE TABLE [Production].[BillOfMaterials]
(
  [BillOfMaterialsID] [int] NOT NULL IDENTITY(1, 1),
```

```

[ProductAssemblyID] [int] NULL,
[ComponentID] [int] NOT NULL,
[StartDate] [datetime] NOT NULL CONSTRAINT [DF_BillOfMaterials_StartDate] DEFAULT
(getdate()),
[EndDate] [datetime] NULL,
[UnitMeasureCode] [nchar] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[BOMLevel] [smallint] NOT NULL,
[PerAssemblyQty] [decimal] (8, 2) NOT NULL CONSTRAINT [DF_BillOfMaterials_Per-
AssemblyQty] DEFAULT ((1.00)),
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_BillOfMaterials_ModifiedDate]
DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Production].[BillOfMaterials] ADD CONSTRAINT [CK_BillOfMaterials_End-
Date] CHECK (([EndDate]>[StartDate] OR [EndDate] IS NULL))
GO
ALTER TABLE [Production].[BillOfMaterials] ADD CONSTRAINT [CK_BillOfMaterials_Per-
AssemblyQty] CHECK (([PerAssemblyQty]>=(1.00)))
GO
ALTER TABLE [Production].[BillOfMaterials] ADD CONSTRAINT [CK_BillOfMaterials_-
BOMLevel] CHECK (([ProductAssemblyID] IS NULL AND [BOMLevel]=0) AND [PerAssembly-
Qty]=1.00) OR [ProductAssemblyID] IS NOT NULL AND [BOMLevel]>=(1))
GO
ALTER TABLE [Production].[BillOfMaterials] ADD CONSTRAINT [CK_BillOfMaterials_-
ProductAssemblyID] CHECK (([ProductAssemblyID]<>[ComponentID]))
GO
ALTER TABLE [Production].[BillOfMaterials] ADD CONSTRAINT [PK_BillOfMaterials_BillOf-
MaterialsID] PRIMARY KEY NONCLUSTERED ([BillOfMaterialsID]) ON [PRIMARY]
GO
CREATE UNIQUE CLUSTERED INDEX [AK_BillOfMaterials_ProductAssemblyID_ComponentID_-
StartDate] ON [Production].[BillOfMaterials] ([ProductAssemblyID], [ComponentID],
[StartDate]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_BillOfMaterials_UnitMeasureCode] ON [Production].[Bill-
OfMaterials] ([UnitMeasureCode]) ON [PRIMARY]
GO
ALTER TABLE [Production].[BillOfMaterials] ADD CONSTRAINT [FK_BillOfMaterials_-
Product_ComponentID] FOREIGN KEY ([ComponentID]) REFERENCES [Production].[Product]
([ProductID])
GO
ALTER TABLE [Production].[BillOfMaterials] ADD CONSTRAINT [FK_BillOfMaterials_-
Product_ProductAssemblyID] FOREIGN KEY ([ProductAssemblyID]) REFERENCES
[Production].[Product] ([ProductID])
GO
ALTER TABLE [Production].[BillOfMaterials] ADD CONSTRAINT [FK_BillOfMaterials_Unit-
Measure_UnitMeasureCode] FOREIGN KEY ([UnitMeasureCode]) REFERENCES
[Production].[UnitMeasure] ([UnitMeasureCode])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Items required to make bicycles and
bicycle subassemblies. It identifies the heirarchical relationship between a parent
product and its components.', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials',
NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for BillOfMaterials
records.', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials', 'COLUMN', N'BillOf-
MaterialsID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Indicates the depth the component
is from its parent (AssemblyID).', 'SCHEMA', N'Production', 'TABLE', N'BillOf-
Materials', 'COLUMN', N'BOMLevel'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Component identification number.
Foreign key to Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'BillOf-
Materials', 'COLUMN', N'ComponentID'

```

```

GO
EXEC sp_addextendedproperty N'MS_Description', N'Date the component stopped being
used in the assembly item.', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials',
'COLUMN', N'EndDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials', 'COLUMN',
N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Quantity of the component needed to
create the assembly.', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials',
'COLUMN', N'PerAssemblyQty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Parent product identification
number. Foreign key to Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'Bill-
OfMaterials', 'COLUMN', N'ProductAssemblyID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date the component started being
used in the assembly item.', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials',
'COLUMN', N'StartDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Standard code identifying the unit
of measure for the quantity.', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials',
'COLUMN', N'UnitMeasureCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [ProductAssembly-
ID] IS NULL AND [BOMLevel] = (0) AND [PerAssemblyQty] = (1) OR [ProductAssemblyID]
IS NOT NULL AND [BOMLevel] >= (1)', 'SCHEMA', N'Production', 'TABLE', N'BillOf-
Materials', 'CONSTRAINT', N'CK_BillofMaterials_BOMLevel'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [EndDate] > [Start-
Date] OR [EndDate] IS NULL', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials',
'CONSTRAINT', N'CK_BillofMaterials_EndDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [PerAssemblyQty]
>= (1.00)', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials', 'CONSTRAINT',
N'CK_BillofMaterials_PerAssemblyQty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [ProductAssembly-
ID] <> [ComponentID]', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials',
'CONSTRAINT', N'CK_BillofMaterials_ProductAssemblyID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials', 'CONSTRAINT',
N'DF_BillofMaterials_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 1.0',
'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials', 'CONSTRAINT', N'DF_Billof-
Materials_PerAssemblyQty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials', 'CONSTRAINT',
N'DF_BillofMaterials_StartDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Product.ComponentID.', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials',
'CONSTRAINT', N'FK_BillofMaterials_Product_ComponentID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Product.ProductAssemblyID.', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials',
'CONSTRAINT', N'FK_BillofMaterials_Product_ProductAssemblyID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
UnitMeasure.UnitMeasureCode.', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials',
'CONSTRAINT', N'FK_BillofMaterials_UnitMeasure_UnitMeasureCode'

```

```
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials', 'CONSTRAINT',
N'PK_BillOfMaterials_BillOfMaterialsID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index.', 'SCHEMA',
N'Production', 'TABLE', N'BillOfMaterials', 'INDEX', N'AK_BillOfMaterials_Product-
AssemblyID_ComponentID_StartDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'BillOfMaterials', 'INDEX', N'IX_BillOfMaterials_Unit-
MeasureCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index created by a
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials',
'INDEX', N'PK_BillOfMaterials_BillOfMaterialsID'
GO
```

Uses

[Production].[Product]
[Production].[UnitMeasure]
Production

Used By

[dbo].[uspGetBillOfMaterials]
[dbo].[uspGetWhereUsedProductID]

[Production].[Culture]**MS_Description**

Lookup table containing the languages in which some AdventureWorks data is stored.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	8
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	CultureID <i>Primary key for Culture records.</i>	nchar(6)	12	False	
	Name <i>Culture description.</i>	[dbo].[Name]	100	False	
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_Culture_CultureID <i>Primary key (clustered) constraint</i>	CultureID	True
	AK_Culture_Name <i>Unique nonclustered index.</i>	Name	True

SQL Script

```
CREATE TABLE [Production].[Culture]
(
  [CultureID] [nchar] (6) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
  [Name] [dbo].[Name] NOT NULL,
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_Culture_ModifiedDate] DEFAULT
  (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Production].[Culture] ADD CONSTRAINT [PK_Culture_CultureID] PRIMARY KEY
CLUSTERED ([CultureID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_Culture_Name] ON [Production].[Culture]
([Name]) ON [PRIMARY]
```

```
GO
EXEC sp_addextendedproperty N'MS_Description', N'Lookup table containing the
languages in which some AdventureWorks data is stored.', 'SCHEMA', N'Production',
'TABLE', N'Culture', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Culture records.',
'SCHEMA', N'Production', 'TABLE', N'Culture', 'COLUMN', N'CultureID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'Culture', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Culture description.', 'SCHEMA',
N'Production', 'TABLE', N'Culture', 'COLUMN', N'Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Culture', 'CONSTRAINT', N'DF_
Culture_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Production', 'TABLE', N'Culture', 'CONSTRAINT', N'PK_
Culture_CultureID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SCHEMA', N'Production', 'TABLE', N'Culture', 'INDEX', N'AK_Culture_Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Culture', 'INDEX',
N'PK_Culture_CultureID'
GO
```

Uses

[dbo].[Name]
Production

Used By

[Production].[ProductModelProductDescriptionCulture]

 **[Production].[Document]**
MS_Description

Product maintenance documents.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Full Text Catalog	AW2008FullTextCatalog
Full Text Key Index	PK_Document_DocumentNode
Row Count (~)	13
Created	13:14:19 14 marca 2012
Last Modified	15:26:31 22 marca 2017

Columns

Key	Name	Data Type	Computed	Max Length (Bytes)	Allow Nulls	Full Text Indexed	Language	Default
	DocumentNode <i>Primary key for Document records.</i>	hierarchyid		892	False			
	DocumentLevel <i>Depth in the document hierarchy.</i>	smallint	True	2	True			
	Title <i>Title of the document.</i>	nvarchar(50)		100	False			
	Owner <i>Employee who controls the document. Foreign key to Employee.Business-EntityID</i>	int		4	False			
	FolderFlag <i>0 = This is a folder, 1 = This is a document.</i>	bit		1	False			((0))
	FileName <i>File name of the document</i>	nvarchar(400)		800	False			
	FileExtension <i>File extension indicating the document type. For example, .doc or .txt.</i>	nvarchar(8)		16	False			
	Revision <i>Revision number of the document.</i>	nchar(5)		10	False			
	ChangeNumber <i>Engineering change approval number.</i>	int		4	False			((0))

	Status <i>1 = Pending approval, 2 = Approved, 3 = Obsolete</i>	tinyint		1	False			
	DocumentSummary <i>Document abstract.</i>	nvarchar(max)		max	True	True	1033	
	Document <i>Complete document.</i>	varbinary(max)		max	True	True	1033	
 (2)	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Required for File- Stream.</i>	uniqueidentifier		16	False			(newid ())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime		8	False			(getda te())

Computed columns

Name	Column definition
DocumentLevel	([DocumentNode].[GetLevel]())

Indexes

Key	Name	Key Columns	Unique
	PK_Document_DocumentNode <i>Primary key (clustered) constraint</i>	DocumentNode	True
	AK_Document_DocumentLevel_DocumentNode <i>Unique nonclustered index.</i>	DocumentLevel, DocumentNode	True
	AK_Document_rowguid <i>Unique nonclustered index. Used to support FileStream.</i>	rowguid	True
	UQ__Document__F73921F793071A63	rowguid	True
	IX_Document_FileName_Revision <i>Unique nonclustered index.</i>	FileName, Revision	

Check Constraints

Name	On Column	Constraint
CK_Document_Status <i>Check constraint [Status] BETWEEN (1) AND (3)</i>	Status	([Status]>=(1) AND [Status]<=(3))

SQL Script

```
CREATE TABLE [Production].[Document]
(
  [DocumentNode] [sys].[hierarchyid] NOT NULL,
  [DocumentLevel] AS ([DocumentNode].[GetLevel]()),
  [Title] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
  [Owner] [int] NOT NULL,
  [FolderFlag] [bit] NOT NULL CONSTRAINT [DF_Document_FolderFlag] DEFAULT ((0)),
```

```

[FileName] [nvarchar] (400) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[FileExtension] [nvarchar] (8) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[Revision] [nchar] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[ChangeNumber] [int] NOT NULL CONSTRAINT [DF_Document_ChangeNumber] DEFAULT ((0)),
[Status] [tinyint] NOT NULL,
[DocumentSummary] [nvarchar] (max) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[Document] [varbinary] (max) NULL,
[rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_Document_rowguid]
DEFAULT (newid()),
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_Document_ModifiedDate] DEFAULT
(getdate())
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
ALTER TABLE [Production].[Document] ADD CONSTRAINT [CK_Document_Status] CHECK
([Status]>=(1) AND [Status]<=(3))
GO
ALTER TABLE [Production].[Document] ADD CONSTRAINT [PK_Document_DocumentNode]
PRIMARY KEY CLUSTERED ([DocumentNode]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_Document_DocumentLevel_DocumentNode] ON
[Production].[Document] ([DocumentLevel], [DocumentNode]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_Document_FileName_Revision] ON [Production].[Document]
([FileName], [Revision]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_Document_rowguid] ON [Production].[Document]
([rowguid]) ON [PRIMARY]
GO
ALTER TABLE [Production].[Document] ADD CONSTRAINT [UQ__Document__F73921F793071A63]
UNIQUE NONCLUSTERED ([rowguid]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product maintenance documents.',
'SHEMA', N'Production', 'TABLE', N'Document', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Engineering change approval
number.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'ChangeNumber'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Complete document.', 'SCHEMA',
N'Production', 'TABLE', N'Document', 'COLUMN', N'Document'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Depth in the document hierarchy.',
'SHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'DocumentLevel'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Document records.',
'SHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'DocumentNode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Document abstract.', 'SCHEMA',
N'Production', 'TABLE', N'Document', 'COLUMN', N'DocumentSummary'
GO
EXEC sp_addextendedproperty N'MS_Description', N'File extension indicating the
document type. For example, .doc or .txt.', 'SCHEMA', N'Production', 'TABLE',
N'Document', 'COLUMN', N'FileExtension'
GO
EXEC sp_addextendedproperty N'MS_Description', N'File name of the document',
'SHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'FileName'
GO
EXEC sp_addextendedproperty N'MS_Description', N'0 = This is a folder, 1 = This is a
document.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'FolderFlag'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'ModifiedDate'
GO

```

```

EXEC sp_addextendedproperty N'MS_Description', N'Employee who controls the document. Foreign key to Employee.BusinessEntityID', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'Owner'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Revision number of the document.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'Revision'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely identifying the record. Required for FileStream.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'1 = Pending approval, 2 = Approved, 3 = Obsolete', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'Status'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Title of the document.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'Title'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Status] BETWEEN (1) AND (3)', 'SCHEMA', N'Production', 'TABLE', N'Document', 'CONSTRAINT', N'CK_Document_Status'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0', 'SCHEMA', N'Production', 'TABLE', N'Document', 'CONSTRAINT', N'DF_Document_ChangeNumber'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Document', 'CONSTRAINT', N'DF_Document_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Production', 'TABLE', N'Document', 'CONSTRAINT', N'DF_Document_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Document', 'CONSTRAINT', N'PK_Document_DocumentNode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'INDEX', N'AK_Document_DocumentLevel_DocumentNode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to support FileStream.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'INDEX', N'AK_Document_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'INDEX', N'IX_Document_FileName_Revision'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'INDEX', N'PK_Document_DocumentNode'
GO
CREATE FULLTEXT INDEX ON [Production].[Document] KEY INDEX [PK_Document_DocumentNode] ON [AW2008FullTextCatalog]
GO
ALTER FULLTEXT INDEX ON [Production].[Document] ADD ([DocumentSummary] LANGUAGE 1033)
GO
ALTER FULLTEXT INDEX ON [Production].[Document] ADD ([Document] TYPE COLUMN [FileExtension] LANGUAGE 1033)
GO

```

Uses

Production

Used By

[Production].[ProductDocument]
AW2008FullTextCatalog

 **[Production].[Illustration]****MS_Description**

Bicycle assembly diagrams.

Properties

Property	Value
Row Count (~)	5
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	IllustrationID <i>Primary key for Illustration records.</i>	int	4	False	1 - 1	
	Diagram <i>Illustrations used in manufacturing instructions. Stored as XML.</i>	xml	max	True		
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_Illustration_IllustrationID <i>Primary key (clustered) constraint</i>	IllustrationID	True

SQL Script

```
CREATE TABLE [Production].[Illustration]
(
  [IllustrationID] [int] NOT NULL IDENTITY(1, 1),
  [Diagram] [xml] NULL,
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_Illustration_ModifiedDate] DEFAULT
(getdate())
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
ALTER TABLE [Production].[Illustration] ADD CONSTRAINT [PK_Illustration_Illustration-
ID] PRIMARY KEY CLUSTERED ([IllustrationID]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Bicycle assembly diagrams.',
'SHEMA', N'Production', 'TABLE', N'Illustration', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Illustrations used in manufacturing
instructions. Stored as XML.', 'SCHEMA', N'Production', 'TABLE', N'Illustration',
```

```
'COLUMN', N'Diagram'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Illustration  
records.', 'SCHEMA', N'Production', 'TABLE', N'Illustration', 'COLUMN',  
N'IllustrationID'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last  
updated.', 'SCHEMA', N'Production', 'TABLE', N'Illustration', 'COLUMN', N'Modified-  
Date'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of  
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Illustration', 'CONSTRAINT', N'DF_-  
Illustration_ModifiedDate'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)  
constraint', 'SCHEMA', N'Production', 'TABLE', N'Illustration', 'CONSTRAINT', N'PK_-  
Illustration_IllustrationID'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a  
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Illustration',  
'INDEX', N'PK_Illustration_IllustrationID'  
GO
```

Uses

Production

Used By

[Production].[ProductModelIllustration]

[Production].[Location]**MS_Description**

Product inventory and manufacturing locations.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	14
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	LocationID <i>Primary key for Location records.</i>	smallint	2	False	1 - 1	
	Name <i>Location description.</i>	[dbo].[Name]	100	False		
	CostRate <i>Standard hourly cost of the manufacturing location.</i>	smallmoney	4	False		((0.00))
	Availability <i>Work capacity (in hours) of the manufacturing location.</i>	decimal(8,2)	5	False		((0.00))
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_Location_LocationID <i>Primary key (clustered) constraint</i>	LocationID	True
	AK_Location_Name <i>Unique nonclustered index.</i>	Name	True

Check Constraints

Name	On Column	Constraint
CK_Location_Availability <i>Check constraint [Availability] >= (0.00)</i>	Availability	([Availability]>=(0.00))
CK_Location_CostRate <i>Check constraint [CostRate] >= (0.00)</i>	CostRate	([CostRate]>=(0.00))

SQL Script

```

CREATE TABLE [Production].[Location]
(
[LocationID] [smallint] NOT NULL IDENTITY(1, 1),
[Name] [dbo].[Name] NOT NULL,
[CostRate] [smallmoney] NOT NULL CONSTRAINT [DF_Location_CostRate] DEFAULT ((0.00)),
[Availability] [decimal] (8, 2) NOT NULL CONSTRAINT [DF_Location_Availability]
DEFAULT ((0.00)),
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_Location_ModifiedDate] DEFAULT
(getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Production].[Location] ADD CONSTRAINT [CK_Location_Availability] CHECK
([Availability]>=(0.00))
GO
ALTER TABLE [Production].[Location] ADD CONSTRAINT [CK_Location_CostRate] CHECK
([CostRate]>=(0.00))
GO
ALTER TABLE [Production].[Location] ADD CONSTRAINT [PK_Location_LocationID] PRIMARY
KEY CLUSTERED ([LocationID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_Location_Name] ON [Production].[Location]
([Name]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product inventory and manufacturing
locations.', 'SCHEMA', N'Production', 'TABLE', N'Location', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Work capacity (in hours) of the
manufacturing location.', 'SCHEMA', N'Production', 'TABLE', N'Location', 'COLUMN',
N'Availability'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Standard hourly cost of the
manufacturing location.', 'SCHEMA', N'Production', 'TABLE', N'Location', 'COLUMN',
N'CostRate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Location records.',
'SCHEMA', N'Production', 'TABLE', N'Location', 'COLUMN', N'LocationID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'Location', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Location description.', 'SCHEMA',
N'Production', 'TABLE', N'Location', 'COLUMN', N'Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Availability] >=
(0.00)', 'SCHEMA', N'Production', 'TABLE', N'Location', 'CONSTRAINT', N'CK_Location_
Availability'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [CostRate] >=
(0.00)', 'SCHEMA', N'Production', 'TABLE', N'Location', 'CONSTRAINT', N'CK_Location_
CostRate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.00',
'SCHEMA', N'Production', 'TABLE', N'Location', 'CONSTRAINT', N'DF_Location_
Availability'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Production', 'TABLE', N'Location', 'CONSTRAINT', N'DF_Location_CostRate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Location', 'CONSTRAINT', N'DF_

```

```
Location_ModifiedDate'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)  
constraint', 'SCHEMA', N'Production', 'TABLE', N'Location', 'CONSTRAINT', N'PK_  
Location_LocationID'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',  
'SCHEMA', N'Production', 'TABLE', N'Location', 'INDEX', N'AK_Location_Name'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a  
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Location', 'INDEX',  
N'PK_Location_LocationID'  
GO
```

Uses

[dbo].[Name]
Production

Used By

[Production].[ProductInventory]
[Production].[WorkOrderRouting]

 **[Production].[Product]****MS_Description**

Products sold or used in the manufacturing of sold products.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	504
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	ProductID <i>Primary key for Product records.</i>	int	4	False	1 - 1	
	Name <i>Name of the product.</i>	[dbo].[Name]	100	False		
	ProductNumber <i>Unique product identification number.</i>	nvarchar(25)	50	False		
	MakeFlag <i>0 = Product is purchased, 1 = Product is manufactured in-house.</i>	[dbo].[Flag]	1	False		((1))
	FinishedGoodsFlag <i>0 = Product is not a salable item. 1 = Product is salable.</i>	[dbo].[Flag]	1	False		((1))
	Color <i>Product color.</i>	nvarchar(15)	30	True		
	SafetyStockLevel <i>Minimum inventory quantity.</i>	smallint	2	False		
	ReorderPoint <i>Inventory level that triggers a purchase order or work order.</i>	smallint	2	False		
	StandardCost <i>Standard cost of the product.</i>	money	8	False		
	ListPrice <i>Selling price.</i>	money	8	False		
	Size <i>Product size.</i>	nvarchar(5)	10	True		
	SizeUnitMeasureCode <i>Unit of measure for Size column.</i>	nchar(3)	6	True		
	WeightUnitMeasureCode <i>Unit of measure for Weight column.</i>	nchar(3)	6	True		

	Weight <i>Product weight.</i>	decimal(8,2)	5	True		
	DaysToManufacture <i>Number of days required to manufacture the product.</i>	int	4	False		
	ProductLine <i>R = Road, M = Mountain, T = Touring, S = Standard</i>	nchar(2)	4	True		
	Class <i>H = High, M = Medium, L = Low</i>	nchar(2)	4	True		
	Style <i>W = Womens, M = Mens, U = Universal</i>	nchar(2)	4	True		
	ProductSubcategoryID <i>Product is a member of this product subcategory. Foreign key to ProductSubCategory.ProductSubCategoryID.</i>	int	4	True		
	ProductModelID <i>Product is a member of this product model. Foreign key to ProductModel.ProductModelID.</i>	int	4	True		
	SellStartDate <i>Date the product was available for sale.</i>	datetime	8	False		
	SellEndDate <i>Date the product was no longer available for sale.</i>	datetime	8	True		
	DiscontinuedDate <i>Date the product was discontinued.</i>	datetime	8	True		
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False		(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_Product_ProductID <i>Primary key (clustered) constraint</i>	ProductID	True
	AK_Product_Name <i>Unique nonclustered index.</i>	Name	True
	AK_Product_ProductNumber <i>Unique nonclustered index.</i>	Product-Number	True
	AK_Product_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True

Check Constraints

Name	On Column	Constraint
CK_Product_DaysToManufacture	DaysTo-	((DaysTo-

Check constraint [DaysToManufacture] >= (0)	Manufacture	Manufacture]>=(0))
CK_Product_ListPrice Check constraint [ListPrice] >= (0.00)	ListPrice	([ListPrice]>=(0.00))
CK_Product_ReorderPoint Check constraint [ReorderPoint] > (0)	ReorderPoint	([ReorderPoint]>(0))
CK_Product_SafetyStockLevel Check constraint [SafetyStockLevel] > (0)	SafetyStockLevel	([SafetyStockLevel]>(0))
CK_Product_SellEndDate Check constraint [SellEndDate] >= [SellStartDate] OR [SellEndDate] IS NULL		([SellEndDate]>=[SellStartDate] OR [SellEndDate] IS NULL)
CK_Product_StandardCost Check constraint [StandardCost] > (0)	StandardCost	([StandardCost]>(0.00))
CK_Product_Weight Check constraint [Weight] > (0.00)	Weight	([Weight]>(0.00))
CK_Product_Class Check constraint [Class]='h' OR [Class]='m' OR [Class]='l' OR [Class]='H' OR [Class]='M' OR [Class]='L' OR [Class] IS NULL	Class	(upper([Class])='H' OR upper([Class])='M' OR upper([Class])='L' OR [Class] IS NULL)
CK_Product_ProductLine Check constraint [ProductLine]='r' OR [ProductLine]='m' OR [ProductLine]='t' OR [ProductLine]='s' OR [ProductLine]='R' OR [ProductLine]='M' OR [ProductLine]='T' OR [ProductLine]='S' OR [ProductLine] IS NULL	ProductLine	(upper([ProductLine])='R' OR upper([ProductLine])='M' OR upper([ProductLine])='T' OR upper([ProductLine])='S' OR [ProductLine] IS NULL)
CK_Product_Style Check constraint [Style]='u' OR [Style]='m' OR [Style]='w' OR [Style]='U' OR [Style]='M' OR [Style]='W' OR [Style] IS NULL	Style	(upper([Style])='U' OR upper([Style])='M' OR upper([Style])='W' OR [Style] IS NULL)

Foreign Keys

Name	Columns
FK_Product_ProductModel_ProductModelID Foreign key constraint referencing ProductModel.ProductModelID.	ProductModelID->[Production].[ProductModel].[ProductModelID]
FK_Product_ProductSubcategory_ProductSubcategoryID Foreign key constraint referencing ProductSubcategory.ProductSubcategoryID.	ProductSubcategoryID->[Production].[ProductSubcategory].[ProductSubcategoryID]
FK_Product_UnitMeasure_SizeUnitMeasureCode Foreign key constraint referencing UnitMeasure.UnitMeasureCode.	SizeUnitMeasureCode->[Production].[UnitMeasure].[UnitMeasureCode]
FK_Product_UnitMeasure_WeightUnitMeasureCode Foreign key constraint referencing UnitMeasure.UnitMeasureCode.	WeightUnitMeasureCode->[Production].[UnitMeasure].[UnitMeasureCode]

SQL Script

```

CREATE TABLE [Production].[Product]
(
  [ProductID] [int] NOT NULL IDENTITY(1, 1),
  [Name] [dbo].[Name] NOT NULL,
  [ProductNumber] [nvarchar] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
  [MakeFlag] [dbo].[Flag] NOT NULL CONSTRAINT [DF_Product_MakeFlag] DEFAULT ((1)),
  [FinishedGoodsFlag] [dbo].[Flag] NOT NULL CONSTRAINT [DF_Product_FinishedGoodsFlag]
  DEFAULT ((1)),
  [Color] [nvarchar] (15) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
  [SafetyStockLevel] [smallint] NOT NULL,
  [ReorderPoint] [smallint] NOT NULL,
  [StandardCost] [money] NOT NULL,
  [ListPrice] [money] NOT NULL,
  [Size] [nvarchar] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
  [SizeUnitMeasureCode] [nchar] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
  [WeightUnitMeasureCode] [nchar] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
  [Weight] [decimal] (8, 2) NULL,
  [DaysToManufacture] [int] NOT NULL,
  [ProductLine] [nchar] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
  [Class] [nchar] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
  [Style] [nchar] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
  [ProductSubcategoryID] [int] NULL,
  [ProductModelID] [int] NULL,
  [SellStartDate] [datetime] NOT NULL,
  [SellEndDate] [datetime] NULL,
  [DiscontinuedDate] [datetime] NULL,
  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_Product_rowguid]
  DEFAULT (newid()),
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_Product_ModifiedDate] DEFAULT
  (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK_Product_DaysToManufacture]
CHECK (([DaysToManufacture]>=(0)))
GO
ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK_Product_ListPrice] CHECK
((([ListPrice]>=(0.00)))
GO
ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK_Product_ReorderPoint] CHECK
((([ReorderPoint]>(0)))
GO
ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK_Product_SafetyStockLevel]
CHECK (([SafetyStockLevel]>(0)))
GO
ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK_Product_SellEndDate] CHECK
((([SellEndDate]>=[SellStartDate] OR [SellEndDate] IS NULL))
GO
ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK_Product_StandardCost] CHECK
((([StandardCost]>=(0.00)))
GO
ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK_Product_Weight] CHECK
((([Weight]>(0.00)))
GO
ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK_Product_Class] CHECK
((upper([Class])='H' OR upper([Class])='M' OR upper([Class])='L' OR [Class] IS
NULL))
GO
ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK_Product_ProductLine] CHECK

```

```

((upper([ProductLine])='R' OR upper([ProductLine])='M' OR upper([ProductLine])='T'
OR upper([ProductLine])='S' OR [ProductLine] IS NULL))
GO
ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK_Product_Style] CHECK
((upper([Style])='U' OR upper([Style])='M' OR upper([Style])='W' OR [Style] IS
NULL))
GO
ALTER TABLE [Production].[Product] ADD CONSTRAINT [PK_Product_ProductID] PRIMARY KEY
CLUSTERED ([ProductID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_Product_Name] ON [Production].[Product]
([Name]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_Product_ProductNumber] ON
[Production].[Product] ([ProductNumber]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_Product_rowguid] ON [Production].[Product]
([rowguid]) ON [PRIMARY]
GO
ALTER TABLE [Production].[Product] ADD CONSTRAINT [FK_Product_ProductModel_Product-
ModelID] FOREIGN KEY ([ProductModelID]) REFERENCES [Production].[ProductModel]
([ProductModelID])
GO
ALTER TABLE [Production].[Product] ADD CONSTRAINT [FK_Product_ProductSubcategory_-
ProductSubcategoryID] FOREIGN KEY ([ProductSubcategoryID]) REFERENCES
[Production].[ProductSubcategory] ([ProductSubcategoryID])
GO
ALTER TABLE [Production].[Product] ADD CONSTRAINT [FK_Product_UnitMeasure_SizeUnit-
MeasureCode] FOREIGN KEY ([SizeUnitMeasureCode]) REFERENCES [Production].[Unit-
Measure] ([UnitMeasureCode])
GO
ALTER TABLE [Production].[Product] ADD CONSTRAINT [FK_Product_UnitMeasure_WeightUnit-
MeasureCode] FOREIGN KEY ([WeightUnitMeasureCode]) REFERENCES [Production].[Unit-
Measure] ([UnitMeasureCode])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Products sold or used in the
manufacturing of sold products.', 'SCHEMA', N'Production', 'TABLE', N'Product', NULL,
NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'H = High, M = Medium, L = Low',
'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Class'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product color.', 'SCHEMA',
N'Production', 'TABLE', N'Product', 'COLUMN', N'Color'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Number of days required to
manufacture the product.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN',
N'DaysToManufacture'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date the product was
discontinued.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN',
N'DiscontinuedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'0 = Product is not a salable item.
1 = Product is salable.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN',
N'FinishedGoodsFlag'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Selling price.', 'SCHEMA',
N'Production', 'TABLE', N'Product', 'COLUMN', N'ListPrice'
GO
EXEC sp_addextendedproperty N'MS_Description', N'0 = Product is purchased, 1 =
Product is manufactured in-house.', 'SCHEMA', N'Production', 'TABLE', N'Product',
'COLUMN', N'MakeFlag'
GO

```

```

EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Name of the product.', 'SCHEMA',
N'Production', 'TABLE', N'Product', 'COLUMN', N'Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Product records.',
'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'R = Road, M = Mountain, T =
Touring, S = Standard', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN',
N'ProductLine'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product is a member of this product
model. Foreign key to ProductModel.ProductModelID.', 'SCHEMA', N'Production',
'TABLE', N'Product', 'COLUMN', N'ProductModelID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique product identification
number.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'ProductNumber'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product is a member of this product
subcategory. Foreign key to ProductSubCategory.ProductSubCategoryID.', 'SCHEMA',
N'Production', 'TABLE', N'Product', 'COLUMN', N'ProductSubCategoryID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Inventory level that triggers a
purchase order or work order.', 'SCHEMA', N'Production', 'TABLE', N'Product',
'COLUMN', N'ReorderPoint'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Production', 'TABLE', N'Product', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Minimum inventory quantity.',
'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'SafetyStockLevel'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date the product was no longer
available for sale.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Sell-
EndDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date the product was available for
sale.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'SellStartDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product size.', 'SCHEMA',
N'Production', 'TABLE', N'Product', 'COLUMN', N'Size'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unit of measure for Size column.',
'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'SizeUnitMeasureCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Standard cost of the product.',
'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'StandardCost'
GO
EXEC sp_addextendedproperty N'MS_Description', N'W = Womens, M = Mens, U =
Universal', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Style'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product weight.', 'SCHEMA',
N'Production', 'TABLE', N'Product', 'COLUMN', N'Weight'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unit of measure for Weight
column.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'WeightUnit-
MeasureCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Class]=''h'' OR
[Class]=''m'' OR [Class]=''l'' OR [Class]=''H'' OR [Class]=''M'' OR [Class]=''L'' OR
[Class] IS NULL', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK_

```

```

Product_Class'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [DaysTo-
Manufacture] >= (0)', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT',
N'CK_Product_DaysToManufacture'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [ListPrice] >=
(0.00)', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK_Product_-
ListPrice'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Product-
Line]='r' OR [ProductLine]='m' OR [ProductLine]='t' OR [ProductLine]='s' OR
[ProductLine]='R' OR [ProductLine]='M' OR [ProductLine]='T' OR [Product-
Line]='S' OR [ProductLine] IS NULL', 'SCHEMA', N'Production', 'TABLE', N'Product',
'CONSTRAINT', N'CK_Product_ProductLine'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [ReorderPoint] >
(0)', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK_Product_-
ReorderPoint'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [SafetyStockLevel]
> (0)', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK_Product_-
SafetyStockLevel'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [SellEndDate] >=
[SellStartDate] OR [SellEndDate] IS NULL', 'SCHEMA', N'Production', 'TABLE',
N'Product', 'CONSTRAINT', N'CK_Product_SellEndDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [SafetyStockLevel]
> (0)', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK_Product_-
StandardCost'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Style]='u' OR
[Style]='m' OR [Style]='w' OR [Style]='U' OR [Style]='M' OR [Style]='W' OR
[Style] IS NULL', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK_-
Product_Style'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Weight] >
(0.00)', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK_Product_-
Weight'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 1',
'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'DF_Product_Finished-
GoodsFlag'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 1',
'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'DF_Product_MakeFlag'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'DF_-
Product_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'DF_-
Product_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
ProductModel.ProductModelID.', 'SCHEMA', N'Production', 'TABLE', N'Product',
'CONSTRAINT', N'FK_Product_ProductModel_ProductModelID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
ProductSubcategory.ProductSubcategoryID.', 'SCHEMA', N'Production', 'TABLE',
N'Product', 'CONSTRAINT', N'FK_Product_ProductSubcategory_ProductSubcategoryID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing

```

```

UnitMeasure.UnitMeasureCode.', 'SCHEMA', N'Production', 'TABLE', N'Product',
'CONSTRAINT', N'FK_Product_UnitMeasure_SizeUnitMeasureCode'

GO

EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
UnitMeasure.UnitMeasureCode.', 'SCHEMA', N'Production', 'TABLE', N'Product',
'CONSTRAINT', N'FK_Product_UnitMeasure_WeightUnitMeasureCode'

GO

EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'PK_
Product_ProductID'

GO

EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SCHEMA', N'Production', 'TABLE', N'Product', 'INDEX', N'AK_Product_Name'

GO

EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SCHEMA', N'Production', 'TABLE', N'Product', 'INDEX', N'AK_Product_ProductNumber'

GO

EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Production', 'TABLE', N'Product',
'INDEX', N'AK_Product_rowguid'

GO

EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'INDEX',
N'PK_Product_ProductID'

GO

```

Uses

[Production].[ProductModel]
 [Production].[ProductSubcategory]
 [Production].[UnitMeasure]
 [dbo].[Flag]
 [dbo].[Name]
 Production

Used By

[Production].[BillOfMaterials]
 [Production].[ProductCostHistory]
 [Production].[ProductDocument]
 [Production].[ProductInventory]
 [Production].[ProductListPriceHistory]
 [Production].[ProductProductPhoto]
 [Production].[ProductReview]
 [Production].[TransactionHistory]
 [Production].[WorkOrder]
 [Purchasing].[ProductVendor]
 [Purchasing].[PurchaseOrderDetail]
 [Sales].[ShoppingCartItem]
 [Sales].[SpecialOfferProduct]
 [Production].[vProductAndDescription]
 [dbo].[uspGetBillOfMaterials]
 [dbo].[uspGetWhereUsedProductID]
 [dbo].[ufnGetProductDealerPrice]
 [dbo].[ufnGetProductListPrice]
 [dbo].[ufnGetProductStandardCost]

[Production].[ProductCategory]**MS_Description**

High-level product categorization.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	4
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	ProductCategoryID <i>Primary key for ProductCategory records.</i>	int	4	False	1 - 1	
	Name <i>Category description.</i>	[dbo].[Name]	100	False		
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False		(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_ProductCategory_ProductCategoryID <i>Primary key (clustered) constraint</i>	Product-CategoryID	True
	AK_ProductCategory_Name <i>Unique nonclustered index.</i>	Name	True
	AK_ProductCategory_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True

SQL Script

```
CREATE TABLE [Production].[ProductCategory]
(
  [ProductCategoryID] [int] NOT NULL IDENTITY(1, 1),
  [Name] [dbo].[Name] NOT NULL,
  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_Product-
```

```

Category_rowguid] DEFAULT (newid()),
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_ProductCategory_ModifiedDate]
DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Production].[ProductCategory] ADD CONSTRAINT [PK_ProductCategory_
ProductCategoryID] PRIMARY KEY CLUSTERED ([ProductCategoryID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_ProductCategory_Name] ON [Production].[Product-
Category] ([Name]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_ProductCategory_rowguid] ON
[Production].[ProductCategory] ([rowguid]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'High-level product
categorization.', 'SCHEMA', N'Production', 'TABLE', N'ProductCategory', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductCategory', 'COLUMN',
N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Category description.', 'SCHEMA',
N'Production', 'TABLE', N'ProductCategory', 'COLUMN', N'Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for ProductCategory
records.', 'SCHEMA', N'Production', 'TABLE', N'ProductCategory', 'COLUMN', N'Product-
CategoryID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Production', 'TABLE', N'ProductCategory', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductCategory', 'CONSTRAINT',
N'DF_ProductCategory_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Production', 'TABLE', N'ProductCategory', 'CONSTRAINT',
N'DF_ProductCategory_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Production', 'TABLE', N'ProductCategory', 'CONSTRAINT',
N'PK_ProductCategory_ProductCategoryID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SCHEMA', N'Production', 'TABLE', N'ProductCategory', 'INDEX', N'AK_ProductCategory_
Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Production', 'TABLE', N'ProductCategory',
'INDEX', N'AK_ProductCategory_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'ProductCategory',
'INDEX', N'PK_ProductCategory_ProductCategoryID'
GO

```

Uses

[dbo].[Name]

Production

Used By

[Production].[ProductSubcategory]

[Production].[ProductCostHistory]

MS_Description

Changes in the cost of a product over time.

Properties

Property	Value
Row Count (~)	395
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	ProductID <i>Product identification number. Foreign key to Product.ProductID</i>	int	4	False	
	StartDate <i>Product cost start date.</i>	datetime	8	False	
	EndDate <i>Product cost end date.</i>	datetime	8	True	
	StandardCost <i>Standard cost of the product.</i>	money	8	False	
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_ProductCostHistory_ProductID_StartDate <i>Primary key (clustered) constraint</i>	ProductID, StartDate	True

Check Constraints

Name	On Column	Constraint
CK_ProductCostHistory_EndDate <i>Check constraint [EndDate] >= [StartDate] OR [EndDate] IS NULL</i>		(([End-Date]>=[Start-Date] OR [End-Date] IS NULL)
CK_ProductCostHistory_StandardCost <i>Check constraint [StandardCost] >= (0.00)</i>	StandardCost	(([Standard-Cost]>=(0.00))

Foreign Keys

Name	Columns
FK_ProductCostHistory_Product_ProductID <i>Foreign key constraint referencing Product.ProductID.</i>	ProductID->[Production].[Product].[ProductID]

SQL Script

```

CREATE TABLE [Production].[ProductCostHistory]
(
    [ProductID] [int] NOT NULL,
    [StartDate] [datetime] NOT NULL,
    [EndDate] [datetime] NULL,
    [StandardCost] [money] NOT NULL,
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_ProductCostHistory_ModifiedDate]
    DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Production].[ProductCostHistory] ADD CONSTRAINT [CK_ProductCostHistory_
-EndDate] CHECK (([EndDate]>=[StartDate] OR [EndDate] IS NULL))
GO
ALTER TABLE [Production].[ProductCostHistory] ADD CONSTRAINT [CK_ProductCostHistory_
-StandardCost] CHECK (([StandardCost]>=(0.00)))
GO
ALTER TABLE [Production].[ProductCostHistory] ADD CONSTRAINT [PK_ProductCostHistory_
-ProductID_StartDate] PRIMARY KEY CLUSTERED ([ProductID], [StartDate]) ON [PRIMARY]
GO
ALTER TABLE [Production].[ProductCostHistory] ADD CONSTRAINT [FK_ProductCostHistory_
-Product_ProductID] FOREIGN KEY ([ProductID]) REFERENCES [Production].[Product]
([ProductID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Changes in the cost of a product
over time.', 'SCHEMA', N'Production', 'TABLE', N'ProductCostHistory', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product cost end date.', 'SCHEMA',
N'Production', 'TABLE', N'ProductCostHistory', 'COLUMN', N'EndDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductCostHistory', 'COLUMN',
N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product identification number.
Foreign key to Product.ProductID', 'SCHEMA', N'Production', 'TABLE', N'ProductCost-
History', 'COLUMN', N'ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Standard cost of the product.',
'SCHEMA', N'Production', 'TABLE', N'ProductCostHistory', 'COLUMN', N'StandardCost'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product cost start date.',
'SCHEMA', N'Production', 'TABLE', N'ProductCostHistory', 'COLUMN', N'StartDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [EndDate] >=
[StartDate] OR [EndDate] IS NULL', 'SCHEMA', N'Production', 'TABLE', N'ProductCost-
History', 'CONSTRAINT', N'CK_ProductCostHistory_EndDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [StandardCost] >=
(0.00)', 'SCHEMA', N'Production', 'TABLE', N'ProductCostHistory', 'CONSTRAINT',
N'CK_ProductCostHistory_StandardCost'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of

```

```
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductCostHistory', 'CONSTRAINT',  
N'DF_ProductCostHistory_ModifiedDate'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing  
Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'ProductCostHistory',  
'CONSTRAINT', N'FK_ProductCostHistory_Product_ProductID'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)  
constraint', 'SCHEMA', N'Production', 'TABLE', N'ProductCostHistory', 'CONSTRAINT',  
N'PK_ProductCostHistory_ProductID_StartDate'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a  
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'ProductCostHistory',  
'INDEX', N'PK_ProductCostHistory_ProductID_StartDate'  
GO
```

Uses

[Production].[Product]
Production

Used By

[dbo].[ufnGetProductStandardCost]

[Production].[ProductDescription]**MS_Description**

Product descriptions in several languages.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	762
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	ProductDescriptionID <i>Primary key for Product-Description records.</i>	int	4	False	1 - 1	
	Description <i>Description of the product.</i>	nvarchar(400)	800	False		
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False		(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_ProductDescription_ProductDescriptionID <i>Primary key (clustered) constraint</i>	Product-DescriptionID	True
	AK_ProductDescription_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True

SQL Script

```
CREATE TABLE [Production].[ProductDescription]
(
  [ProductDescriptionID] [int] NOT NULL IDENTITY(1, 1),
  [Description] [nvarchar] (400) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_Product-Description_rowguid] DEFAULT (newid()),
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_ProductDescription_ModifiedDate]
```

```

DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Production].[ProductDescription] ADD CONSTRAINT [PK_ProductDescription_
ProductDescriptionID] PRIMARY KEY CLUSTERED ([ProductDescriptionID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_ProductDescription_rowguid] ON
[Production].[ProductDescription] ((rowguid)) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product descriptions in several
languages.', 'SCHEMA', N'Production', 'TABLE', N'ProductDescription', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Description of the product.',
'SCHEMA', N'Production', 'TABLE', N'ProductDescription', 'COLUMN', N'Description'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductDescription', 'COLUMN',
N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for ProductDescription
records.', 'SCHEMA', N'Production', 'TABLE', N'ProductDescription', 'COLUMN',
N'ProductDescriptionID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Production', 'TABLE', N'ProductDescription', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductDescription', 'CONSTRAINT',
N'DF_ProductDescription_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Production', 'TABLE', N'ProductDescription', 'CONSTRAINT',
N'DF_ProductDescription_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Production', 'TABLE', N'ProductDescription', 'CONSTRAINT',
N'PK_ProductDescription_ProductDescriptionID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Production', 'TABLE', N'Product-
Description', 'INDEX', N'AK_ProductDescription_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'ProductDescription',
'INDEX', N'PK_ProductDescription_ProductDescriptionID'
GO

```

Uses

Production

Used By

[Production].[ProductModelProductDescriptionCulture]

[Production].[vProductAndDescription]

[Production].[ProductDocument]**MS_Description**

Cross-reference table mapping products to related product documents.

Properties

Property	Value
Row Count (~)	32
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	ProductID <i>Product identification number. Foreign key to Product.ProductID.</i>	int	4	False	
	DocumentNode <i>Document identification number. Foreign key to Document.DocumentNode.</i>	hierarchyid	892	False	
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_ProductDocument_ProductID_DocumentNode <i>Primary key (clustered) constraint</i>	ProductID, Document-Node	True

Foreign Keys

Name	Columns
FK_ProductDocument_Document_DocumentNode <i>Foreign key constraint referencing Document.DocumentNode.</i>	DocumentNode->[Production].[Document].[Document-Node]
FK_ProductDocument_Product_ProductID <i>Foreign key constraint referencing Product.ProductID.</i>	ProductID->[Production].[Product].[ProductID]

SQL Script

```
CREATE TABLE [Production].[ProductDocument]
(
  [ProductID] [int] NOT NULL,
  [DocumentNode] [sys].[hierarchyid] NOT NULL,
```

```

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_ProductDocument_ModifiedDate]
DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Production].[ProductDocument] ADD CONSTRAINT [PK_ProductDocument_
ProductID_DocumentNode] PRIMARY KEY CLUSTERED ([ProductID], [DocumentNode]) ON
[PRIMARY]
GO
ALTER TABLE [Production].[ProductDocument] ADD CONSTRAINT [FK_ProductDocument_
Document_DocumentNode] FOREIGN KEY ([DocumentNode]) REFERENCES
[Production].[Document] ([DocumentNode])
GO
ALTER TABLE [Production].[ProductDocument] ADD CONSTRAINT [FK_ProductDocument_
Product_ProductID] FOREIGN KEY ([ProductID]) REFERENCES [Production].[Product]
([ProductID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Cross-reference table mapping
products to related product documents.', 'SCHEMA', N'Production', 'TABLE', N'Product-
Document', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Document identification number.
Foreign key to Document.DocumentNode.', 'SCHEMA', N'Production', 'TABLE', N'Product-
Document', 'COLUMN', N'DocumentNode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductDocument', 'COLUMN',
N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product identification number.
Foreign key to Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'Product-
Document', 'COLUMN', N'ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductDocument', 'CONSTRAINT',
N'DF_ProductDocument_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Document.DocumentNode.', 'SCHEMA', N'Production', 'TABLE', N'ProductDocument',
'CONSTRAINT', N'FK_ProductDocument_Document_DocumentNode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'ProductDocument',
'CONSTRAINT', N'FK_ProductDocument_Product_ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Production', 'TABLE', N'ProductDocument', 'CONSTRAINT',
N'PK_ProductDocument_ProductID_DocumentNode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'ProductDocument',
'INDEX', N'PK_ProductDocument_ProductID_DocumentNode'
GO

```

Uses

[Production].[Document]
[Production].[Product]
Production

 **[Production].[ProductInventory]****MS_Description**

Product inventory information.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	1069
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	ProductID <i>Product identification number. Foreign key to Product.ProductID.</i>	int	4	False	
	LocationID <i>Inventory location identification number. Foreign key to Location.LocationID.</i>	smallint	2	False	
	Shelf <i>Storage compartment within an inventory location.</i>	nvarchar(10)	20	False	
	Bin <i>Storage container on a shelf in an inventory location.</i>	tinyint	1	False	
	Quantity <i>Quantity of products in the inventory location.</i>	smallint	2	False	((0))
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False	(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_ProductInventory_ProductID_LocationID <i>Primary key (clustered) constraint</i>	ProductID, LocationID	True

Check Constraints

Name	On Column	Constraint
CK_ProductInventory_Bin <i>Check constraint [Bin] BETWEEN (0) AND (100)</i>	Bin	([Bin]>=(0) AND [Bin]<=(100))
CK_ProductInventory_Shelf <i>Check constraint [Shelf] like '[A-Za-z]' OR [Shelf]='N/A'</i>	Shelf	([Shelf] like '[A-Za-z]' OR [Shelf]='N/A')

Foreign Keys

Name	Columns
FK_ProductInventory_Location_LocationID <i>Foreign key constraint referencing Location.LocationID.</i>	LocationID->[Production].[Location].[LocationID]
FK_ProductInventory_Product_ProductID <i>Foreign key constraint referencing Product.ProductID.</i>	ProductID->[Production].[Product].[ProductID]

SQL Script

```

CREATE TABLE [Production].[ProductInventory]
(
    [ProductID] [int] NOT NULL,
    [LocationID] [smallint] NOT NULL,
    [Shelf] [nvarchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [Bin] [tinyint] NOT NULL,
    [Quantity] [smallint] NOT NULL CONSTRAINT [DF_ProductInventory_Quantity] DEFAULT ((0)),
    [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_ProductInventory_rowguid] DEFAULT (newid()),
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_ProductInventory_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Production].[ProductInventory] ADD CONSTRAINT [CK_ProductInventory_Bin] CHECK (([Bin]>=(0) AND [Bin]<=(100)))
GO
ALTER TABLE [Production].[ProductInventory] ADD CONSTRAINT [CK_ProductInventory_Shelf] CHECK (([Shelf] like '[A-Za-z]' OR [Shelf]='N/A'))
GO
ALTER TABLE [Production].[ProductInventory] ADD CONSTRAINT [PK_ProductInventory_ProductID_LocationID] PRIMARY KEY CLUSTERED ([ProductID], [LocationID]) ON [PRIMARY]
GO
ALTER TABLE [Production].[ProductInventory] ADD CONSTRAINT [FK_ProductInventory_Location_LocationID] FOREIGN KEY ([LocationID]) REFERENCES [Production].[Location] ([LocationID])
GO
ALTER TABLE [Production].[ProductInventory] ADD CONSTRAINT [FK_ProductInventory_Product_ProductID] FOREIGN KEY ([ProductID]) REFERENCES [Production].[Product] ([ProductID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product inventory information.', 'SCHEMA', N'Production', 'TABLE', N'ProductInventory', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Storage container on a shelf in an inventory location.', 'SCHEMA', N'Production', 'TABLE', N'ProductInventory', 'COLUMN', N'Bin'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Inventory location identification

```

```

number. Foreign key to Location.LocationID.', 'SCHEMA', N'Production', 'TABLE',
N'ProductInventory', 'COLUMN', N'LocationID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductInventory', 'COLUMN',
N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product identification number.
Foreign key to Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'Product-
Inventory', 'COLUMN', N'ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Quantity of products in the
inventory location.', 'SCHEMA', N'Production', 'TABLE', N'ProductInventory',
'COLUMN', N'Quantity'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Production', 'TABLE', N'ProductInventory', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Storage compartment within an
inventory location.', 'SCHEMA', N'Production', 'TABLE', N'ProductInventory',
'COLUMN', N'Shelf'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Bin] BETWEEN (0)
AND (100)', 'SCHEMA', N'Production', 'TABLE', N'ProductInventory', 'CONSTRAINT',
N'CK_ProductInventory_Bin'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Shelf] like ''[A-
Za-z]'' OR [Shelf]=''N/A''', 'SCHEMA', N'Production', 'TABLE', N'ProductInventory',
'CONSTRAINT', N'CK_ProductInventory_Shelf'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductInventory', 'CONSTRAINT',
N'DF_ProductInventory_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0',
'SCHEMA', N'Production', 'TABLE', N'ProductInventory', 'CONSTRAINT', N'DF_Product-
Inventory_Quantity'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Production', 'TABLE', N'ProductInventory', 'CONSTRAINT', N'DF_-
ProductInventory_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Location.LocationID.', 'SCHEMA', N'Production', 'TABLE', N'ProductInventory',
'CONSTRAINT', N'FK_ProductInventory_Location_LocationID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'ProductInventory',
'CONSTRAINT', N'FK_ProductInventory_Product_ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Production', 'TABLE', N'ProductInventory', 'CONSTRAINT',
N'PK_ProductInventory_ProductID_LocationID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'ProductInventory',
'INDEX', N'PK_ProductInventory_ProductID_LocationID'
GO

```

Uses

[Production].[Location]
[Production].[Product]
Production

Used By

[dbo].[ufnGetStock]

 **[Production].[ProductListPriceHistory]****MS_Description**

Changes in the list price of a product over time.

Properties

Property	Value
Row Count (~)	395
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	ProductID <i>Product identification number. Foreign key to Product.ProductID</i>	int	4	False	
	StartDate <i>List price start date.</i>	datetime	8	False	
	EndDate <i>List price end date</i>	datetime	8	True	
	ListPrice <i>Product list price.</i>	money	8	False	
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_ProductListPriceHistory_ProductID_StartDate <i>Primary key (clustered) constraint</i>	ProductID, StartDate	True

Check Constraints

Name	On Column	Constraint
CK_ProductListPriceHistory_EndDate <i>Check constraint [EndDate] >= [StartDate] OR [EndDate] IS NULL</i>		(([EndDate]>=[Start-Date] OR [EndDate] IS NULL)
CK_ProductListPriceHistory_ListPrice <i>Check constraint [ListPrice] > (0.00)</i>	ListPrice	(([ListPrice]>(0.00))

Foreign Keys

Name	Columns
------	---------

FK_ProductListPriceHistory_Product_ProductID <i>Foreign key constraint referencing Product.ProductID.</i>	ProductID->[Production].[Product].[ProductID]
--	---

SQL Script

```

CREATE TABLE [Production].[ProductListPriceHistory]
(
    [ProductID] [int] NOT NULL,
    [StartDate] [datetime] NOT NULL,
    [EndDate] [datetime] NULL,
    [ListPrice] [money] NOT NULL,
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_ProductListPriceHistory_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Production].[ProductListPriceHistory] ADD CONSTRAINT [CK_ProductListPriceHistory_EndDate] CHECK (([EndDate]>=[StartDate] OR [EndDate] IS NULL))
GO
ALTER TABLE [Production].[ProductListPriceHistory] ADD CONSTRAINT [CK_ProductListPriceHistory_ListPrice] CHECK (([ListPrice]>(0.00)))
GO
ALTER TABLE [Production].[ProductListPriceHistory] ADD CONSTRAINT [PK_ProductListPriceHistory_ProductID_StartDate] PRIMARY KEY CLUSTERED ([ProductID], [StartDate]) ON [PRIMARY]
GO
ALTER TABLE [Production].[ProductListPriceHistory] ADD CONSTRAINT [FK_ProductListPriceHistory_Product_ProductID] FOREIGN KEY ([ProductID]) REFERENCES [Production].[Product] ([ProductID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Changes in the list price of a product over time.', 'SCHEMA', N'Production', 'TABLE', N'ProductListPriceHistory', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'List price end date', 'SCHEMA', N'Production', 'TABLE', N'ProductListPriceHistory', 'COLUMN', N'EndDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product list price.', 'SCHEMA', N'Production', 'TABLE', N'ProductListPriceHistory', 'COLUMN', N'ListPrice'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductListPriceHistory', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product identification number. Foreign key to Product.ProductID', 'SCHEMA', N'Production', 'TABLE', N'ProductListPriceHistory', 'COLUMN', N'ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'List price start date.', 'SCHEMA', N'Production', 'TABLE', N'ProductListPriceHistory', 'COLUMN', N'StartDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [EndDate] >= [StartDate] OR [EndDate] IS NULL', 'SCHEMA', N'Production', 'TABLE', N'ProductListPriceHistory', 'CONSTRAINT', N'CK_ProductListPriceHistory_EndDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [ListPrice] > (0.00)', 'SCHEMA', N'Production', 'TABLE', N'ProductListPriceHistory', 'CONSTRAINT', N'CK_ProductListPriceHistory_ListPrice'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductListPriceHistory', 'CONSTRAINT', N'DF_ProductListPriceHistory_ModifiedDate'

```

```
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'ProductListPriceHistory',
'CONSTRAINT', N'FK_ProductListPriceHistory_Product_ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Production', 'TABLE', N'ProductListPriceHistory',
'CONSTRAINT', N'PK_ProductListPriceHistory_ProductID_StartDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'ProductListPrice-
History', 'INDEX', N'PK_ProductListPriceHistory_ProductID_StartDate'
GO
```

Uses

[Production].[Product]
Production

Used By

[dbo].[ufnGetProductDealerPrice]
[dbo].[ufnGetProductListPrice]

 **[Production].[ProductModel]**
MS_Description

Product model classification.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	128
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	ProductModelID <i>Primary key for ProductModel records.</i>	int	4	False	1 - 1	
	Name <i>Product model description.</i>	[dbo].[Name]	100	False		
	CatalogDescription <i>Detailed product catalog information in xml format.</i>	xml([Production].[Product-DescriptionSchema-Collection])	max	True		
	Instructions <i>Manufacturing instructions in xml format.</i>	xml([Production].[Manu-InstructionsSchema-Collection])	max	True		
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False		(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Type	Unique	XML Type
	PK_ProductModel_ProductModelID <i>Primary key (clustered) constraint</i>	Product-ModelID		True	
	AK_ProductModel_Name <i>Unique nonclustered index.</i>	Name		True	
	AK_ProductModel_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid		True	
	PXML_ProductModel_CatalogDescription <i>Primary XML index.</i>	Catalog-Description	xml		Primary

PXML_ProductModel_Instructions <i>Primary XML index.</i>	Instructions	xml		Primary
---	--------------	-----	--	---------

SQL Script

```

CREATE TABLE [Production].[ProductModel]
(
  [ProductModelID] [int] NOT NULL IDENTITY(1, 1),
  [Name] [dbo].[Name] NOT NULL,
  [CatalogDescription] [xml] (CONTENT [Production].[ProductDescriptionSchema-
Collection]) NULL,
  [Instructions] [xml] (CONTENT [Production].[ManuInstructionsSchemaCollection]) NULL,
  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_Product-
Model_rowguid] DEFAULT (newid()),
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_ProductModel_ModifiedDate] DEFAULT
(getdate())
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
ALTER TABLE [Production].[ProductModel] ADD CONSTRAINT [PK_ProductModel_ProductModel-
ID] PRIMARY KEY CLUSTERED ([ProductModelID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_ProductModel_Name] ON [Production].[Product-
Model] ([Name]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_ProductModel_rowguid] ON [Production].[Product-
Model] ([rowguid]) ON [PRIMARY]
GO
CREATE PRIMARY XML INDEX [PXML_ProductModel_CatalogDescription]
ON [Production].[ProductModel] ([CatalogDescription])
GO
CREATE PRIMARY XML INDEX [PXML_ProductModel_Instructions]
ON [Production].[ProductModel] ([Instructions])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product model classification.',
'SHEMA', N'Production', 'TABLE', N'ProductModel', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Detailed product catalog
information in xml format.', 'SCHEMA', N'Production', 'TABLE', N'ProductModel',
'COLUMN', N'CatalogDescription'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Manufacturing instructions in xml
format.', 'SCHEMA', N'Production', 'TABLE', N'ProductModel', 'COLUMN',
N'Instructions'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductModel', 'COLUMN', N'Modified-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product model description.',
'SHEMA', N'Production', 'TABLE', N'ProductModel', 'COLUMN', N'Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for ProductModel
records.', 'SCHEMA', N'Production', 'TABLE', N'ProductModel', 'COLUMN', N'Product-
ModelID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Production', 'TABLE', N'ProductModel', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductModel', 'CONSTRAINT', N'DF_-

```

```

ProductModel_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Production', 'TABLE', N'ProductModel', 'CONSTRAINT', N'DF_-
ProductModel_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Production', 'TABLE', N'ProductModel', 'CONSTRAINT', N'PK_-
ProductModel_ProductModelID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SCHEMA', N'Production', 'TABLE', N'ProductModel', 'INDEX', N'AK_ProductModel_Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Production', 'TABLE', N'ProductModel',
'INDEX', N'AK_ProductModel_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'ProductModel',
'INDEX', N'PK_ProductModel_ProductModelID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary XML index.', 'SCHEMA',
N'Production', 'TABLE', N'ProductModel', 'INDEX', N'PXML_ProductModel_Catalog-
Description'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary XML index.', 'SCHEMA',
N'Production', 'TABLE', N'ProductModel', 'INDEX', N'PXML_ProductModel_Instructions'
GO

```

Uses

[dbo].[Name]

Production

[Production].[ManuInstructionsSchemaCollection]

[Production].[ProductDescriptionSchemaCollection]

Used By

[Production].[Product]

[Production].[ProductModelIllustration]

[Production].[ProductModelProductDescriptionCulture]

[Production].[vProductAndDescription]

[Production].[vProductModelCatalogDescription]

[Production].[vProductModelInstructions]

 **[Production].[ProductModelIllustration]****MS_Description**

Cross-reference table mapping product models and illustrations.

Properties

Property	Value
Row Count (~)	7
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	ProductModelID <i>Primary key. Foreign key to Product-Model.ProductModelID.</i>	int	4	False	
	IllustrationID <i>Primary key. Foreign key to Illustration.IllustrationID.</i>	int	4	False	
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_ProductModelIllustration_ProductModelID_IllustrationID <i>Primary key (clustered) constraint</i>	ProductModelID, IllustrationID	True

Foreign Keys

Name	Columns
FK_ProductModelIllustration_Illustration_IllustrationID <i>Foreign key constraint referencing Illustration.IllustrationID.</i>	IllustrationID->[Production].[Illustration].[IllustrationID]
FK_ProductModelIllustration_ProductModel_ProductModelID <i>Foreign key constraint referencing Product-Model.ProductModelID.</i>	ProductModelID->[Production].[ProductModel].[ProductModelID]

SQL Script

```
CREATE TABLE [Production].[ProductModelIllustration]
(
  [ProductModelID] [int] NOT NULL,
```

```

[IllustrationID] [int] NOT NULL,
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_ProductModelIllustration_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Production].[ProductModelIllustration] ADD CONSTRAINT [PK_ProductModelIllustration_ProductModelID_IllustrationID] PRIMARY KEY CLUSTERED ([ProductModelID], [IllustrationID]) ON [PRIMARY]
GO
ALTER TABLE [Production].[ProductModelIllustration] ADD CONSTRAINT [FK_ProductModelIllustration_Illustration_IllustrationID] FOREIGN KEY ([IllustrationID]) REFERENCES [Production].[Illustration] ([IllustrationID])
GO
ALTER TABLE [Production].[ProductModelIllustration] ADD CONSTRAINT [FK_ProductModelIllustration_ProductModel_ProductModelID] FOREIGN KEY ([ProductModelID]) REFERENCES [Production].[ProductModel] ([ProductModelID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Cross-reference table mapping product models and illustrations.', 'SCHEMA', N'Production', 'TABLE', N'ProductModelIllustration', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Foreign key to Illustration.IllustrationID.', 'SCHEMA', N'Production', 'TABLE', N'ProductModelIllustration', 'COLUMN', N'IllustrationID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductModelIllustration', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Foreign key to ProductModel.ProductModelID.', 'SCHEMA', N'Production', 'TABLE', N'ProductModelIllustration', 'COLUMN', N'ProductModelID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductModelIllustration', 'CONSTRAINT', N'DF_ProductModelIllustration_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing Illustration.IllustrationID.', 'SCHEMA', N'Production', 'TABLE', N'ProductModelIllustration', 'CONSTRAINT', N'FK_ProductModelIllustration_Illustration_IllustrationID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing ProductModel.ProductModelID.', 'SCHEMA', N'Production', 'TABLE', N'ProductModelIllustration', 'CONSTRAINT', N'FK_ProductModelIllustration_ProductModel_ProductModelID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'ProductModelIllustration', 'CONSTRAINT', N'PK_ProductModelIllustration_ProductModelID_IllustrationID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'ProductModelIllustration', 'INDEX', N'PK_ProductModelIllustration_ProductModelID_IllustrationID'
GO

```

Uses

[Production].[Illustration]
[Production].[ProductModel]
Production

[Production].[ProductModelProductDescriptionCulture]

MS_Description

Cross-reference table mapping product descriptions and the language the description is written in.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	762
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
 	ProductModelID <i>Primary key. Foreign key to Product-Model.ProductModelID.</i>	int	4	False	
 	ProductDescriptionID <i>Primary key. Foreign key to Product-Description.ProductDescriptionID.</i>	int	4	False	
	CultureID <i>Culture identification number. Foreign key to Culture.CultureID.</i>	nchar(6)	12	False	
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
 	PK_ProductModelProductDescriptionCulture_ProductModelID_Product-DescriptionID_CultureID <i>Primary key (clustered) constraint</i>	ProductModelID, Product-DescriptionID, CultureID	True

Foreign Keys

Name	Columns
FK_ProductModelProductDescriptionCulture_CultureID <i>Foreign key constraint referencing Culture.CultureID.</i>	CultureID->[Production].[Culture].[CultureID]
FK_ProductModelProductDescriptionCulture_Product-DescriptionID <i>Foreign key constraint referencing Product-Description.ProductDescriptionID.</i>	ProductDescriptionID->[Production].[Product-Description].[ProductDescriptionID]
FK_ProductModelProductDescriptionCulture_Product-	ProductModelID->[Production].[ProductModel].[Product-

Model_ProductModelID <i>Foreign key constraint referencing Product-Model.ProductModelID.</i>	ModelID]
---	----------

SQL Script

```

CREATE TABLE [Production].[ProductModelProductDescriptionCulture]
(
    [ProductModelID] [int] NOT NULL,
    [ProductDescriptionID] [int] NOT NULL,
    [CultureID] [nchar] (6) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_ProductModelProductDescription-
Culture_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Production].[ProductModelProductDescriptionCulture] ADD CONSTRAINT [PK_
ProductModelProductDescriptionCulture_ProductModelID_ProductDescriptionID_CultureID]
PRIMARY KEY CLUSTERED ([ProductModelID], [ProductDescriptionID], [CultureID]) ON
[PRIMARY]
GO
ALTER TABLE [Production].[ProductModelProductDescriptionCulture] ADD CONSTRAINT [FK_
ProductModelProductDescriptionCulture_Culture_CultureID] FOREIGN KEY ([CultureID])
REFERENCES [Production].[Culture] ([CultureID])
GO
ALTER TABLE [Production].[ProductModelProductDescriptionCulture] ADD CONSTRAINT [FK_
ProductModelProductDescriptionCulture_ProductDescription_ProductDescriptionID]
FOREIGN KEY ([ProductDescriptionID]) REFERENCES [Production].[ProductDescription]
([ProductDescriptionID])
GO
ALTER TABLE [Production].[ProductModelProductDescriptionCulture] ADD CONSTRAINT [FK_
ProductModelProductDescriptionCulture_ProductModel_ProductModelID] FOREIGN KEY
([ProductModelID]) REFERENCES [Production].[ProductModel] ([ProductModelID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Cross-reference table mapping
product descriptions and the language the description is written in.', 'SCHEMA',
N'Production', 'TABLE', N'ProductModelProductDescriptionCulture', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Culture identification number.
Foreign key to Culture.CultureID.', 'SCHEMA', N'Production', 'TABLE', N'ProductModel-
ProductDescriptionCulture', 'COLUMN', N'CultureID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductModelProductDescription-
Culture', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Foreign key to Product-
Description.ProductDescriptionID.', 'SCHEMA', N'Production', 'TABLE', N'ProductModel-
ProductDescriptionCulture', 'COLUMN', N'ProductDescriptionID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Foreign key to Product-
Model.ProductModelID.', 'SCHEMA', N'Production', 'TABLE', N'ProductModelProduct-
DescriptionCulture', 'COLUMN', N'ProductModelID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductModelProductDescription-
Culture', 'CONSTRAINT', N'DF_ProductModelProductDescriptionCulture_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Culture.CultureID.', 'SCHEMA', N'Production', 'TABLE', N'ProductModelProduct-
DescriptionCulture', 'CONSTRAINT', N'FK_ProductModelProductDescriptionCulture_
Culture_CultureID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing

```

```
ProductDescription.ProductDescriptionID.', 'SCHEMA', N'Production', 'TABLE',
N'ProductModelProductDescriptionCulture', 'CONSTRAINT', N'FK_ProductModelProduct-
DescriptionCulture_ProductDescription_ProductDescriptionID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
ProductModel.ProductModelID.', 'SCHEMA', N'Production', 'TABLE', N'ProductModel-
ProductDescriptionCulture', 'CONSTRAINT', N'FK_ProductModelProductDescription-
Culture_ProductModel_ProductModelID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Production', 'TABLE', N'ProductModelProductDescription-
Culture', 'CONSTRAINT', N'PK_ProductModelProductDescriptionCulture_ProductModelID_-
ProductDescriptionID_CultureID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'ProductModelProduct-
DescriptionCulture', 'INDEX', N'PK_ProductModelProductDescriptionCulture_Product-
ModelID_ProductDescriptionID_CultureID'
GO
```

Uses

[Production].[Culture]
[Production].[ProductDescription]
[Production].[ProductModel]
Production

Used By

[Production].[vProductAndDescription]

 **[Production].[ProductPhoto]****MS_Description**

Product images.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	101
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	ProductPhotoID <i>Primary key for ProductPhoto records.</i>	int	4	False	1 - 1	
	ThumbNailPhoto <i>Small image of the product.</i>	varbinary(max)	max	True		
	ThumbNailPhotoFileName <i>Small image file name.</i>	nvarchar(50)	100	True		
	LargePhoto <i>Large image of the product.</i>	varbinary(max)	max	True		
	LargePhotoFileName <i>Large image file name.</i>	nvarchar(50)	100	True		
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_ProductPhoto_ProductPhotoID <i>Primary key (clustered) constraint</i>	ProductPhotoID	True

SQL Script

```
CREATE TABLE [Production].[ProductPhoto]
(
  [ProductPhotoID] [int] NOT NULL IDENTITY(1, 1),
  [ThumbNailPhoto] [varbinary] (max) NULL,
  [ThumbNailPhotoFileName] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
  [LargePhoto] [varbinary] (max) NULL,
  [LargePhotoFileName] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
```

```

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_ProductPhoto_ModifiedDate] DEFAULT
(getdate())
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
ALTER TABLE [Production].[ProductPhoto] ADD CONSTRAINT [PK_ProductPhoto_ProductPhoto-
ID] PRIMARY KEY CLUSTERED ([ProductPhotoID]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product images.', 'SCHEMA',
N'Production', 'TABLE', N'ProductPhoto', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Large image of the product.',
'SCHEMA', N'Production', 'TABLE', N'ProductPhoto', 'COLUMN', N'LargePhoto'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Large image file name.', 'SCHEMA',
N'Production', 'TABLE', N'ProductPhoto', 'COLUMN', N'LargePhotoFileName'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductPhoto', 'COLUMN', N'Modified-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for ProductPhoto
records.', 'SCHEMA', N'Production', 'TABLE', N'ProductPhoto', 'COLUMN', N'Product-
PhotoID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Small image of the product.',
'SCHEMA', N'Production', 'TABLE', N'ProductPhoto', 'COLUMN', N'ThumbNailPhoto'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Small image file name.', 'SCHEMA',
N'Production', 'TABLE', N'ProductPhoto', 'COLUMN', N'ThumbNailPhotoFileName'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductPhoto', 'CONSTRAINT', N'DF_-
ProductPhoto_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Production', 'TABLE', N'ProductPhoto', 'CONSTRAINT', N'PK_-
ProductPhoto_ProductPhotoID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'ProductPhoto',
'INDEX', N'PK_ProductPhoto_ProductPhotoID'
GO

```

Uses

Production

Used By

[Production].[ProductProductPhoto]

[Production].[ProductProductPhoto]

MS_Description

Cross-reference table mapping products and product photos.

Properties

Property	Value
Heap	True
Row Count (~)	504
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
 	ProductID <i>Product identification number. Foreign key to Product.ProductID.</i>	int	4	False	
 	ProductPhotoID <i>Product photo identification number. Foreign key to ProductPhoto.ProductPhotoID.</i>	int	4	False	
	Primary <i>0 = Photo is not the principal image. 1 = Photo is the principal image.</i>	[dbo].[Flag]	1	False	((0))
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_ProductProductPhoto_ProductID_ProductPhotoID <i>Primary key (clustered) constraint</i>	ProductID, ProductPhotoID	True

Foreign Keys

Name	Columns
FK_ProductProductPhoto_Product_ProductID <i>Foreign key constraint referencing Product.ProductID.</i>	ProductID->[Production].[Product].[ProductID]
FK_ProductProductPhoto_ProductPhoto_ProductPhotoID <i>Foreign key constraint referencing ProductPhoto.ProductPhotoID.</i>	ProductPhotoID->[Production].[ProductPhoto].[ProductPhotoID]

SQL Script

```

CREATE TABLE [Production].[ProductProductPhoto]
(
    [ProductID] [int] NOT NULL,
    [ProductPhotoID] [int] NOT NULL,
    [Primary] [dbo].[Flag] NOT NULL CONSTRAINT [DF_ProductProductPhoto_Primary] DEFAULT
    ((0)),
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_ProductProductPhoto_ModifiedDate]
    DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Production].[ProductProductPhoto] ADD CONSTRAINT [PK_ProductProduct-
Photo_ProductID_ProductPhotoID] PRIMARY KEY NONCLUSTERED ([ProductID], [Product-
PhotoID]) ON [PRIMARY]
GO
ALTER TABLE [Production].[ProductProductPhoto] ADD CONSTRAINT [FK_ProductProduct-
Photo_Product_ProductID] FOREIGN KEY ([ProductID]) REFERENCES [Production].[Product]
([ProductID])
GO
ALTER TABLE [Production].[ProductProductPhoto] ADD CONSTRAINT [FK_ProductProduct-
Photo_ProductPhoto_ProductPhotoID] FOREIGN KEY ([ProductPhotoID]) REFERENCES
[Production].[ProductPhoto] ([ProductPhotoID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Cross-reference table mapping
products and product photos.', 'SCHEMA', N'Production', 'TABLE', N'ProductProduct-
Photo', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductProductPhoto', 'COLUMN',
N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'0 = Photo is not the principal
image. 1 = Photo is the principal image.', 'SCHEMA', N'Production', 'TABLE',
N'ProductProductPhoto', 'COLUMN', N'Primary'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product identification number.
Foreign key to Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'Product-
ProductPhoto', 'COLUMN', N'ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product photo identification
number. Foreign key to ProductPhoto.ProductPhotoID.', 'SCHEMA', N'Production',
'TABLE', N'ProductProductPhoto', 'COLUMN', N'ProductPhotoID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductProductPhoto', 'CONSTRAINT',
N'DF_ProductProductPhoto_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0
(FALSE)', 'SCHEMA', N'Production', 'TABLE', N'ProductProductPhoto', 'CONSTRAINT',
N'DF_ProductProductPhoto_Primary'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'ProductProductPhoto',
'CONSTRAINT', N'FK_ProductProductPhoto_Product_ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
ProductPhoto.ProductPhotoID.', 'SCHEMA', N'Production', 'TABLE', N'ProductProduct-
Photo', 'CONSTRAINT', N'FK_ProductProductPhoto_ProductPhoto_ProductPhotoID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Production', 'TABLE', N'ProductProductPhoto', 'CONSTRAINT',
N'PK_ProductProductPhoto_ProductID_ProductPhotoID'
GO

```

```
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index created by a  
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'ProductProductPhoto',  
'INDEX', N'PK_ProductProductPhoto_ProductID_ProductPhotoID'  
GO
```

Uses

[Production].[Product]

[Production].[ProductPhoto]

[dbo].[Flag]

Production

 **[Production].[ProductReview]**
MS_Description

Customer reviews of products they have purchased.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Full Text Catalog	AW2008FullTextCatalog
Full Text Key Index	PK_ProductReview_ProductReviewID
Row Count (~)	4
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Full Text Indexed	Language	Identity	Default
	ProductReviewID <i>Primary key for Product-Review records.</i>	int	4	False			1 - 1	
	ProductID <i>Product identification number. Foreign key to Product.ProductID.</i>	int	4	False				
	ReviewerName <i>Name of the reviewer.</i>	[dbo].[Name]	100	False				
	ReviewDate <i>Date review was submitted.</i>	datetime	8	False				(getdate())
	EmailAddress <i>Reviewer's e-mail address.</i>	nvarchar(50)	100	False				
	Rating <i>Product rating given by the reviewer. Scale is 1 to 5 with 5 as the highest rating.</i>	int	4	False				
	Comments <i>Reviewer's comments</i>	nvarchar(3850)	7700	True	True	1033		
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False				(getdate())

Indexes

Key	Name	Key Columns	Included Columns	Unique
	PK_ProductReview_ProductReviewID <i>Primary key (clustered) constraint</i>	ProductReviewID		True
	IX_ProductReview_ProductID_Name <i>Nonclustered index.</i>	ProductID, ReviewerName	Comments	

Check Constraints

Name	On Column	Constraint
CK_ProductReview_Rating <i>Check constraint [Rating] BETWEEN (1) AND (5)</i>	Rating	(([Rating]>=(1) AND [Rating]<=(5))

Foreign Keys

Name	Columns
FK_ProductReview_Product_ProductID <i>Foreign key constraint referencing Product.ProductID.</i>	ProductID->[Production].[Product].[ProductID]

SQL Script

```
CREATE TABLE [Production].[ProductReview]
(
    [ProductReviewID] [int] NOT NULL IDENTITY(1, 1),
    [ProductID] [int] NOT NULL,
    [ReviewerName] [dbo].[Name] NOT NULL,
    [ReviewDate] [datetime] NOT NULL CONSTRAINT [DF_ProductReview_ReviewDate] DEFAULT (getdate()),
    [EmailAddress] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [Rating] [int] NOT NULL,
    [Comments] [nvarchar] (3850) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_ProductReview_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Production].[ProductReview] ADD CONSTRAINT [CK_ProductReview_Rating] CHECK (([Rating]>=(1) AND [Rating]<=(5)))
GO
ALTER TABLE [Production].[ProductReview] ADD CONSTRAINT [PK_ProductReview_ProductReviewID] PRIMARY KEY CLUSTERED ([ProductReviewID]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_ProductReview_ProductID_Name] ON [Production].[ProductReview] ([ProductID], [ReviewerName]) INCLUDE ([Comments]) ON [PRIMARY]
GO
ALTER TABLE [Production].[ProductReview] ADD CONSTRAINT [FK_ProductReview_Product_ProductID] FOREIGN KEY ([ProductID]) REFERENCES [Production].[Product] ([ProductID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Customer reviews of products they have purchased.', 'SCHEMA', N'Production', 'TABLE', N'ProductReview', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Reviewer's comments', 'SCHEMA', N'Production', 'TABLE', N'ProductReview', 'COLUMN', N'Comments'
GO
```

```

EXEC sp_addextendedproperty N'MS_Description', N'Reviewer's e-mail address.',
'SHEMA', N'Production', 'TABLE', N'ProductReview', 'COLUMN', N'EmailAddress'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductReview', 'COLUMN', N'Modified-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product identification number.
Foreign key to Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'Product-
Review', 'COLUMN', N'ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for ProductReview
records.', 'SCHEMA', N'Production', 'TABLE', N'ProductReview', 'COLUMN', N'Product-
ReviewID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product rating given by the
reviewer. Scale is 1 to 5 with 5 as the highest rating.', 'SCHEMA', N'Production',
'TABLE', N'ProductReview', 'COLUMN', N'Rating'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date review was submitted.',
'SHEMA', N'Production', 'TABLE', N'ProductReview', 'COLUMN', N'ReviewDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Name of the reviewer.', 'SCHEMA',
N'Production', 'TABLE', N'ProductReview', 'COLUMN', N'ReviewerName'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Rating] BETWEEN
(1) AND (5)', 'SCHEMA', N'Production', 'TABLE', N'ProductReview', 'CONSTRAINT',
N'CK_ProductReview_Rating'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductReview', 'CONSTRAINT', N'DF_-
ProductReview_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductReview', 'CONSTRAINT', N'DF_-
ProductReview_ReviewDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'ProductReview',
'CONSTRAINT', N'FK_ProductReview_Product_ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Production', 'TABLE', N'ProductReview', 'CONSTRAINT', N'PK_-
ProductReview_ProductReviewID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'ProductReview', 'INDEX', N'IX_ProductReview_ProductID_-
Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'ProductReview',
'INDEX', N'PK_ProductReview_ProductReviewID'
GO
CREATE FULLTEXT INDEX ON [Production].[ProductReview] KEY INDEX [PK_ProductReview_-
ProductReviewID] ON [AW2008FullTextCatalog]
GO
ALTER FULLTEXT INDEX ON [Production].[ProductReview] ADD ([Comments] LANGUAGE 1033)
GO

```

Uses

[Production].[Product]

[dbo].[Name]
Production

Used By

AW2008FullTextCatalog

[Production].[ProductSubcategory]**MS_Description**

Product subcategories. See ProductCategory table.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	37
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	ProductSubcategoryID <i>Primary key for Product-Subcategory records.</i>	int	4	False	1 - 1	
	ProductCategoryID <i>Product category identification number. Foreign key to Product-Category.ProductCategoryID.</i>	int	4	False		
	Name <i>Subcategory description.</i>	[dbo].[Name]	100	False		
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False		(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_ProductSubcategory_ProductSubcategoryID <i>Primary key (clustered) constraint</i>	Product-SubcategoryID	True
	AK_ProductSubcategory_Name <i>Unique nonclustered index.</i>	Name	True
	AK_ProductSubcategory_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True

Foreign Keys

Name	Columns
------	---------

FK_ProductSubcategory_ProductCategory_ProductCategoryID <i>Foreign key constraint referencing ProductCategory.ProductCategoryID.</i>	ProductCategoryID->[Production].[ProductCategory].[ProductCategoryID]
--	---

SQL Script

```

CREATE TABLE [Production].[ProductSubcategory]
(
  [ProductSubcategoryID] [int] NOT NULL IDENTITY(1, 1),
  [ProductCategoryID] [int] NOT NULL,
  [Name] [dbo].[Name] NOT NULL,
  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_ProductSubcategory_rowguid] DEFAULT (newid()),
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_ProductSubcategory_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Production].[ProductSubcategory] ADD CONSTRAINT [PK_ProductSubcategory_ProductSubcategoryID] PRIMARY KEY CLUSTERED ([ProductSubcategoryID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_ProductSubcategory_Name] ON [Production].[ProductSubcategory] ([Name]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_ProductSubcategory_rowguid] ON [Production].[ProductSubcategory] ([rowguid]) ON [PRIMARY]
GO
ALTER TABLE [Production].[ProductSubcategory] ADD CONSTRAINT [FK_ProductSubcategory_ProductCategory_ProductCategoryID] FOREIGN KEY ([ProductCategoryID]) REFERENCES [Production].[ProductCategory] ([ProductCategoryID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product subcategories. See ProductCategory table.', 'SCHEMA', N'Production', 'TABLE', N'ProductSubcategory', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductSubcategory', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Subcategory description.', 'SCHEMA', N'Production', 'TABLE', N'ProductSubcategory', 'COLUMN', N'Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product category identification number. Foreign key to ProductCategory.ProductCategoryID.', 'SCHEMA', N'Production', 'TABLE', N'ProductSubcategory', 'COLUMN', N'ProductCategoryID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for ProductSubcategory records.', 'SCHEMA', N'Production', 'TABLE', N'ProductSubcategory', 'COLUMN', N'ProductSubcategoryID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Production', 'TABLE', N'ProductSubcategory', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductSubcategory', 'CONSTRAINT', N'DF_ProductSubcategory_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Production', 'TABLE', N'ProductSubcategory', 'CONSTRAINT', N'DF_ProductSubcategory_rowguid'
GO

```

```
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing ProductCategory.ProductCategoryID.', 'SCHEMA', N'Production', 'TABLE', N'ProductSubcategory', 'CONSTRAINT', N'FK_ProductSubcategory_ProductCategory_ProductCategoryID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint.', 'SCHEMA', N'Production', 'TABLE', N'ProductSubcategory', 'CONSTRAINT', N'PK_ProductSubcategory_ProductSubcategoryID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'ProductSubcategory', 'INDEX', N'AK_ProductSubcategory_Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Production', 'TABLE', N'ProductSubcategory', 'INDEX', N'AK_ProductSubcategory_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'ProductSubcategory', 'INDEX', N'PK_ProductSubcategory_ProductSubcategoryID'
GO
```

Uses

[Production].[ProductCategory]
[dbo].[Name]
Production

Used By

[Production].[Product]

 **[Production].[ScrapReason]****MS_Description**

Manufacturing failure reasons lookup table.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	16
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	ScrapReasonID <i>Primary key for ScrapReason records.</i>	smallint	2	False	1 - 1	
	Name <i>Failure description.</i>	[dbo].[Name]	100	False		
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_ScrapReason_ScrapReasonID <i>Primary key (clustered) constraint</i>	ScrapReasonID	True
	AK_ScrapReason_Name <i>Unique nonclustered index.</i>	Name	True

SQL Script

```
CREATE TABLE [Production].[ScrapReason]
(
  [ScrapReasonID] [smallint] NOT NULL IDENTITY(1, 1),
  [Name] [dbo].[Name] NOT NULL,
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_ScrapReason_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Production].[ScrapReason] ADD CONSTRAINT [PK_ScrapReason_ScrapReasonID] PRIMARY KEY CLUSTERED ([ScrapReasonID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_ScrapReason_Name] ON [Production].[ScrapReason]
```

```
([Name]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Manufacturing failure reasons
lookup table.', 'SCHEMA', N'Production', 'TABLE', N'ScrapReason', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'ScrapReason', 'COLUMN', N'Modified-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Failure description.', 'SCHEMA',
N'Production', 'TABLE', N'ScrapReason', 'COLUMN', N'Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for ScrapReason
records.', 'SCHEMA', N'Production', 'TABLE', N'ScrapReason', 'COLUMN', N'ScrapReason-
ID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ScrapReason', 'CONSTRAINT', N'DF_-
ScrapReason_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Production', 'TABLE', N'ScrapReason', 'CONSTRAINT', N'PK_-
ScrapReason_ScrapReasonID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SCHEMA', N'Production', 'TABLE', N'ScrapReason', 'INDEX', N'AK_ScrapReason_Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'ScrapReason', 'INDEX',
N'PK_ScrapReason_ScrapReasonID'
GO
```

Uses

[dbo].[Name]
Production

Used By

[Production].[WorkOrder]

[Production].[TransactionHistory]**MS_Description**

Record of each purchase order, sales order, or work order transaction year to date.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	113443
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	TransactionID <i>Primary key for Transaction-History records.</i>	int	4	False	100000 - 1	
	ProductID <i>Product identification number. Foreign key to Product.Product-ID.</i>	int	4	False		
	ReferenceOrderID <i>Purchase order, sales order, or work order identification number.</i>	int	4	False		
	ReferenceOrderLineID <i>Line number associated with the purchase order, sales order, or work order.</i>	int	4	False		((0))
	TransactionDate <i>Date and time of the transaction.</i>	datetime	8	False		(getdate())
	TransactionType <i>W = WorkOrder, S = Sales-Order, P = PurchaseOrder</i>	nchar(1)	2	False		
	Quantity <i>Product quantity.</i>	int	4	False		
	ActualCost <i>Product cost.</i>	money	8	False		
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_TransactionHistory_TransactionID <i>Primary key (clustered) constraint</i>	TransactionID	True

IX_TransactionHistory_ProductID <i>Nonclustered index.</i>	ProductID	
IX_TransactionHistory_ReferenceOrderID_ReferenceOrderLineID <i>Nonclustered index.</i>	Reference-OrderID, Reference-OrderLineID	

Check Constraints

Name	On Column	Constraint
CK_TransactionHistory_TransactionType <i>Check constraint [TransactionType]='p' OR [TransactionType]='s' OR [TransactionType]='w' OR [TransactionType]='P' OR [TransactionType]='S' OR [TransactionType]='W'</i>	Transaction-Type	(upper([TransactionType])='P' OR upper([TransactionType])='S' OR upper([TransactionType])='W')

Foreign Keys

Name	Columns
FK_TransactionHistory_Product_ProductID <i>Foreign key constraint referencing Product.ProductID.</i>	ProductID->[Production].[Product].[ProductID]

SQL Script

```
CREATE TABLE [Production].[TransactionHistory]
(
  [TransactionID] [int] NOT NULL IDENTITY(100000, 1),
  [ProductID] [int] NOT NULL,
  [ReferenceOrderID] [int] NOT NULL,
  [ReferenceOrderLineID] [int] NOT NULL CONSTRAINT [DF_TransactionHistory_ReferenceOrderLineID] DEFAULT ((0)),
  [TransactionDate] [datetime] NOT NULL CONSTRAINT [DF_TransactionHistory_TransactionDate] DEFAULT (getdate()),
  [TransactionType] [nchar] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
  [Quantity] [int] NOT NULL,
  [ActualCost] [money] NOT NULL,
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_TransactionHistory_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Production].[TransactionHistory] ADD CONSTRAINT [CK_TransactionHistory_TransactionType] CHECK ((upper([TransactionType])='P' OR upper([TransactionType])='S' OR upper([TransactionType])='W'))
GO
ALTER TABLE [Production].[TransactionHistory] ADD CONSTRAINT [PK_TransactionHistory_TransactionID] PRIMARY KEY CLUSTERED ([TransactionID]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_TransactionHistory_ProductID] ON [Production].[TransactionHistory] ([ProductID]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_TransactionHistory_ReferenceOrderID_ReferenceOrderLineID] ON [Production].[TransactionHistory] ([ReferenceOrderID], [ReferenceOrderLineID]) ON [PRIMARY]
GO
```

```

ALTER TABLE [Production].[TransactionHistory] ADD CONSTRAINT [FK_TransactionHistory-
Product_ProductID] FOREIGN KEY ([ProductID]) REFERENCES [Production].[Product]
([ProductID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Record of each purchase order,
sales order, or work order transaction year to date.', 'SCHEMA', N'Production',
'TABLE', N'TransactionHistory', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product cost.', 'SCHEMA',
N'Production', 'TABLE', N'TransactionHistory', 'COLUMN', N'ActualCost'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistory', 'COLUMN',
N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product identification number.
Foreign key to Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'Transaction-
History', 'COLUMN', N'ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product quantity.', 'SCHEMA',
N'Production', 'TABLE', N'TransactionHistory', 'COLUMN', N'Quantity'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Purchase order, sales order, or
work order identification number.', 'SCHEMA', N'Production', 'TABLE', N'Transaction-
History', 'COLUMN', N'ReferenceOrderID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Line number associated with the
purchase order, sales order, or work order.', 'SCHEMA', N'Production', 'TABLE',
N'TransactionHistory', 'COLUMN', N'ReferenceOrderLineID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time of the transaction.',
'SCHEMA', N'Production', 'TABLE', N'TransactionHistory', 'COLUMN', N'Transaction-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for TransactionHistory
records.', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistory', 'COLUMN',
N'TransactionID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'W = WorkOrder, S = SalesOrder, P =
PurchaseOrder', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistory', 'COLUMN',
N'TransactionType'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Transaction-
Type]=''p'' OR [TransactionType]=''s'' OR [TransactionType]=''w'' OR [Transaction-
Type]=''P'' OR [TransactionType]=''S'' OR [TransactionType]=''W''', 'SCHEMA',
N'Production', 'TABLE', N'TransactionHistory', 'CONSTRAINT', N'CK_Transaction-
History_TransactionType'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistory', 'CONSTRAINT',
N'DF_TransactionHistory_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0',
'SCHEMA', N'Production', 'TABLE', N'TransactionHistory', 'CONSTRAINT', N'DF_-
TransactionHistory_ReferenceOrderLineID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistory', 'CONSTRAINT',
N'DF_TransactionHistory_TransactionDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistory',
'CONSTRAINT', N'FK_TransactionHistory_Product_ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)

```

```
constraint', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistory', 'CONSTRAINT',
N'PK_TransactionHistory_TransactionID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'TransactionHistory', 'INDEX', N'IX_TransactionHistory_
ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'TransactionHistory', 'INDEX', N'IX_TransactionHistory_
ReferenceOrderID_ReferenceOrderLineID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistory',
'INDEX', N'PK_TransactionHistory_TransactionID'
GO
```

Uses

[Production].[Product]
Production

[Production].[TransactionHistoryArchive]

MS_Description

Transactions for previous years.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	89253
Created	13:14:19 14 marca 2012
Last Modified	13:14:47 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	TransactionID <i>Primary key for TransactionHistoryArchive records.</i>	int	4	False	
	ProductID <i>Product identification number. Foreign key to Product.ProductID.</i>	int	4	False	
	ReferenceOrderID <i>Purchase order, sales order, or work order identification number.</i>	int	4	False	
	ReferenceOrderLineID <i>Line number associated with the purchase order, sales order, or work order.</i>	int	4	False	((0))
	TransactionDate <i>Date and time of the transaction.</i>	datetime	8	False	(getdate())
	TransactionType <i>W = Work Order, S = Sales Order, P = Purchase Order</i>	nchar(1)	2	False	
	Quantity <i>Product quantity.</i>	int	4	False	
	ActualCost <i>Product cost.</i>	money	8	False	
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_TransactionHistoryArchive_TransactionID <i>Primary key (clustered) constraint</i>	TransactionID	True
	IX_TransactionHistoryArchive_ProductID <i>Nonclustered index.</i>	ProductID	
	IX_TransactionHistoryArchive_ReferenceOrderID_ReferenceOrderLineID	Reference-OrderID,	

	<i>Nonclustered index.</i>	Reference-OrderLineID	
--	----------------------------	-----------------------	--

Check Constraints

Name	On Column	Constraint
CK_TransactionHistoryArchive_TransactionType Check constraint [TransactionType]='P' OR [TransactionType]='S' OR [TransactionType]='W' OR [TransactionType]='P' OR [TransactionType]='S' OR [TransactionType]='W'	Transaction-Type	(upper([TransactionType])='P' OR upper([TransactionType])='S' OR upper([TransactionType])='W')

SQL Script

```

CREATE TABLE [Production].[TransactionHistoryArchive]
(
  [TransactionID] [int] NOT NULL,
  [ProductID] [int] NOT NULL,
  [ReferenceOrderID] [int] NOT NULL,
  [ReferenceOrderLineID] [int] NOT NULL CONSTRAINT [DF_TransactionHistoryArchive_
ReferenceOrderLineID] DEFAULT ((0)),
  [TransactionDate] [datetime] NOT NULL CONSTRAINT [DF_TransactionHistoryArchive_
TransactionDate] DEFAULT (getdate()),
  [TransactionType] [nchar] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
  [Quantity] [int] NOT NULL,
  [ActualCost] [money] NOT NULL,
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_TransactionHistoryArchive_Modified-
Date] DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Production].[TransactionHistoryArchive] ADD CONSTRAINT [CK_Transaction-
HistoryArchive_TransactionType] CHECK ((upper([TransactionType])='P' OR
upper([TransactionType])='S' OR upper([TransactionType])='W'))
GO
ALTER TABLE [Production].[TransactionHistoryArchive] ADD CONSTRAINT [PK_Transaction-
HistoryArchive_TransactionID] PRIMARY KEY CLUSTERED ([TransactionID]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_TransactionHistoryArchive_ProductID] ON
[Production].[TransactionHistoryArchive] ([ProductID]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_TransactionHistoryArchive_ReferenceOrderID_Reference-
OrderLineID] ON [Production].[TransactionHistoryArchive] ([ReferenceOrderID],
[ReferenceOrderLineID]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Transactions for previous years.',
'SHEMA', N'Production', 'TABLE', N'TransactionHistoryArchive', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product cost.', 'SCHEMA',
N'Production', 'TABLE', N'TransactionHistoryArchive', 'COLUMN', N'ActualCost'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistoryArchive', 'COLUMN',
N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product identification number.
Foreign key to Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'Transaction-
HistoryArchive', 'COLUMN', N'ProductID'

```

```

GO
EXEC sp_addextendedproperty N'MS_Description', N'Product quantity.', 'SCHEMA',
N'Production', 'TABLE', N'TransactionHistoryArchive', 'COLUMN', N'Quantity'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Purchase order, sales order, or
work order identification number.', 'SCHEMA', N'Production', 'TABLE', N'Transaction-
HistoryArchive', 'COLUMN', N'ReferenceOrderID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Line number associated with the
purchase order, sales order, or work order.', 'SCHEMA', N'Production', 'TABLE',
N'TransactionHistoryArchive', 'COLUMN', N'ReferenceOrderLineID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time of the transaction.',
'SCHEMA', N'Production', 'TABLE', N'TransactionHistoryArchive', 'COLUMN',
N'TransactionDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for TransactionHistory-
Archive records.', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistoryArchive',
'COLUMN', N'TransactionID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'W = Work Order, S = Sales Order, P
= Purchase Order', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistoryArchive',
'COLUMN', N'TransactionType'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Transaction-
Type]='p' OR [TransactionType]='s' OR [TransactionType]='w' OR [Transaction-
Type]='P' OR [TransactionType]='S' OR [TransactionType]='W'', 'SCHEMA',
N'Production', 'TABLE', N'TransactionHistoryArchive', 'CONSTRAINT', N'CK_Transaction-
HistoryArchive_TransactionType'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistoryArchive',
'CONSTRAINT', N'DF_TransactionHistoryArchive_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0',
'SCHEMA', N'Production', 'TABLE', N'TransactionHistoryArchive', 'CONSTRAINT', N'DF_-
TransactionHistoryArchive_ReferenceOrderLineID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistoryArchive',
'CONSTRAINT', N'DF_TransactionHistoryArchive_TransactionDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistoryArchive',
'CONSTRAINT', N'PK_TransactionHistoryArchive_TransactionID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'TransactionHistoryArchive', 'INDEX', N'IX_Transaction-
HistoryArchive_ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'TransactionHistoryArchive', 'INDEX', N'IX_Transaction-
HistoryArchive_ReferenceOrderID_ReferenceOrderLineID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistory-
Archive', 'INDEX', N'PK_TransactionHistoryArchive_TransactionID'
GO

```

Uses

Production

[Production].[UnitMeasure]**MS_Description**

Unit of measure lookup table.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	38
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	UnitMeasureCode <i>Primary key.</i>	nchar(3)	6	False	
	Name <i>Unit of measure description.</i>	[dbo].[Name]	100	False	
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_UnitMeasure_UnitMeasureCode <i>Primary key (clustered) constraint</i>	UnitMeasureCode	True
	AK_UnitMeasure_Name <i>Unique nonclustered index.</i>	Name	True

SQL Script

```
CREATE TABLE [Production].[UnitMeasure]
(
  [UnitMeasureCode] [nchar] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
  [Name] [dbo].[Name] NOT NULL,
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_UnitMeasure_ModifiedDate] DEFAULT
  (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Production].[UnitMeasure] ADD CONSTRAINT [PK_UnitMeasure_UnitMeasure-
Code] PRIMARY KEY CLUSTERED ([UnitMeasureCode]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_UnitMeasure_Name] ON [Production].[UnitMeasure]
([Name]) ON [PRIMARY]
```

```
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unit of measure lookup table.',
'SHEMA', N'Production', 'TABLE', N'UnitMeasure', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'UnitMeasure', 'COLUMN', N'Modified-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unit of measure description.',
'SHEMA', N'Production', 'TABLE', N'UnitMeasure', 'COLUMN', N'Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key.', 'SCHEMA',
N'Production', 'TABLE', N'UnitMeasure', 'COLUMN', N'UnitMeasureCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'UnitMeasure', 'CONSTRAINT', N'DF_-
UnitMeasure_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Production', 'TABLE', N'UnitMeasure', 'CONSTRAINT', N'PK_-
UnitMeasure_UnitMeasureCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SHEMA', N'Production', 'TABLE', N'UnitMeasure', 'INDEX', N'AK_UnitMeasure_Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'UnitMeasure', 'INDEX',
N'PK_UnitMeasure_UnitMeasureCode'
GO
```

Uses

[dbo].[Name]

Production

Used By

[Production].[BillOfMaterials]

[Production].[Product]

[Purchasing].[ProductVendor]

 **[Production].[WorkOrder]****MS_Description**

Manufacturing work orders.

Properties

Property	Value
Row Count (~)	72591
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Computed	Max Length (Bytes)	Allow Nulls	Identity	Default
	WorkOrderID <i>Primary key for WorkOrder records.</i>	int		4	False	1 - 1	
	ProductID <i>Product identification number. Foreign key to Product.ProductID.</i>	int		4	False		
	OrderQty <i>Product quantity to build.</i>	int		4	False		
	StockedQty <i>Quantity built and put in inventory.</i>	int	True	4	False		
	ScrappedQty <i>Quantity that failed inspection.</i>	smallint		2	False		
	StartDate <i>Work order start date.</i>	datetime		8	False		
	EndDate <i>Work order end date.</i>	datetime		8	True		
	DueDate <i>Work order due date.</i>	datetime		8	False		
	ScrapReasonID <i>Reason for inspection failure.</i>	smallint		2	True		
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime		8	False		(getdate())

Computed columns

Name	Column definition
StockedQty	(isnull([OrderQty]-[ScrappedQty],(0)))

Indexes

Key	Name	Key Columns	Unique
	PK_WorkOrder_WorkOrderID <i>Primary key (clustered) constraint</i>	WorkOrderID	True
	IX_WorkOrder_ProductID <i>Nonclustered index.</i>	ProductID	
	IX_WorkOrder_ScrapReasonID <i>Nonclustered index.</i>	ScrapReasonID	

Triggers

Name	ANSI Nulls On	Quoted Identifier On	On
iWorkOrder <i>AFTER INSERT trigger that inserts a row in the TransactionHistory table.</i>	True	True	After Insert
uWorkOrder <i>AFTER UPDATE trigger that inserts a row in the TransactionHistory table, updates ModifiedDate in the WorkOrder table.</i>	True	True	After Update

Check Constraints

Name	On Column	Constraint
CK_WorkOrder_EndDate <i>Check constraint [EndDate] >= [StartDate] OR [EndDate] IS NULL</i>		([EndDate]>=[StartDate] OR [EndDate] IS NULL)
CK_WorkOrder_OrderQty <i>Check constraint [OrderQty] > (0)</i>	OrderQty	([OrderQty]>(0))
CK_WorkOrder_ScrappedQty <i>Check constraint [ScrappedQty] >= (0)</i>	ScrappedQty	([ScrappedQty]>=(0))

Foreign Keys

Name	Columns
FK_WorkOrder_Product_ProductID <i>Foreign key constraint referencing Product.ProductID.</i>	ProductID->[Production].[Product].[ProductID]
FK_WorkOrder_ScrapReason_ScrapReasonID <i>Foreign key constraint referencing ScrapReason.ScrapReasonID.</i>	ScrapReasonID->[Production].[ScrapReason].[ScrapReasonID]

SQL Script

```
CREATE TABLE [Production].[WorkOrder]
(
  [WorkOrderID] [int] NOT NULL IDENTITY(1, 1),
  [ProductID] [int] NOT NULL,
  [OrderQty] [int] NOT NULL,
  [StockedQty] AS (isnull([OrderQty]-[ScrappedQty], (0))),
  [ScrappedQty] [smallint] NOT NULL,
  [StartDate] [datetime] NOT NULL,
```

```

[EndDate] [datetime] NULL,
[DueDate] [datetime] NOT NULL,
[ScrapReasonID] [smallint] NULL,
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_WorkOrder_ModifiedDate] DEFAULT
(getdate())
) ON [PRIMARY]
GO

CREATE TRIGGER [Production].[iWorkOrder] ON [Production].[WorkOrder]
AFTER INSERT AS
BEGIN
    DECLARE @Count int;

    SET @Count = @@ROWCOUNT;
    IF @Count = 0
        RETURN;

    SET NOCOUNT ON;

    BEGIN TRY
        INSERT INTO [Production].[TransactionHistory] (
            [ProductID]
            , [ReferenceOrderID]
            , [TransactionType]
            , [TransactionDate]
            , [Quantity]
            , [ActualCost])
        SELECT
            inserted.[ProductID]
            , inserted.[WorkOrderID]
            , 'W'
            , GETDATE()
            , inserted.[OrderQty]
            , 0
        FROM inserted;
    END TRY
    BEGIN CATCH
        EXECUTE [dbo].[uspPrintError];

        -- Rollback any active or uncommittable transactions before
        -- inserting information in the ErrorLog
        IF @@TRANCOUNT > 0
            BEGIN
                ROLLBACK TRANSACTION;
            END

        EXECUTE [dbo].[uspLogError];
    END CATCH;
END;
GO

CREATE TRIGGER [Production].[uWorkOrder] ON [Production].[WorkOrder]
AFTER UPDATE AS
BEGIN
    DECLARE @Count int;

    SET @Count = @@ROWCOUNT;

```

```

IF @Count = 0
    RETURN;

SET NOCOUNT ON;

BEGIN TRY
    IF UPDATE([ProductID]) OR UPDATE([OrderQty])
    BEGIN
        INSERT INTO [Production].[TransactionHistory] (
            [ProductID]
            , [ReferenceOrderID]
            , [TransactionType]
            , [TransactionDate]
            , [Quantity])
        SELECT
            inserted.[ProductID]
            , inserted.[WorkOrderID]
            , 'W'
            , GETDATE()
            , inserted.[OrderQty]
        FROM inserted;
    END;
END TRY
BEGIN CATCH
    EXECUTE [dbo].[uspPrintError];

    -- Rollback any active or uncommittable transactions before
    -- inserting information in the ErrorLog
    IF @@TRANCOUNT > 0
    BEGIN
        ROLLBACK TRANSACTION;
    END

    EXECUTE [dbo].[uspLogError];
END CATCH;
END;
GO
ALTER TABLE [Production].[WorkOrder] ADD CONSTRAINT [CK_WorkOrder_EndDate] CHECK
((([EndDate]>=[StartDate] OR [EndDate] IS NULL))
GO
ALTER TABLE [Production].[WorkOrder] ADD CONSTRAINT [CK_WorkOrder_OrderQty] CHECK
((([OrderQty]>(0))
GO
ALTER TABLE [Production].[WorkOrder] ADD CONSTRAINT [CK_WorkOrder_ScrappedQty] CHECK
((([ScrappedQty]>=(0))
GO
ALTER TABLE [Production].[WorkOrder] ADD CONSTRAINT [PK_WorkOrder_WorkOrderID]
PRIMARY KEY CLUSTERED ([WorkOrderID]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_WorkOrder_ProductID] ON [Production].[WorkOrder]
([ProductID]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_WorkOrder_ScrapReasonID] ON [Production].[WorkOrder]
([ScrapReasonID]) ON [PRIMARY]
GO
ALTER TABLE [Production].[WorkOrder] ADD CONSTRAINT [FK_WorkOrder_Product_ProductID]
FOREIGN KEY ([ProductID]) REFERENCES [Production].[Product] ([ProductID])
GO
ALTER TABLE [Production].[WorkOrder] ADD CONSTRAINT [FK_WorkOrder_ScrapReason_Scrap-

```

```

ReasonID] FOREIGN KEY ([ScrapReasonID]) REFERENCES [Production].[ScrapReason]
([ScrapReasonID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Manufacturing work orders.',
'SCHEMA', N'Production', 'TABLE', N'WorkOrder', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Work order due date.', 'SCHEMA',
N'Production', 'TABLE', N'WorkOrder', 'COLUMN', N'DueDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Work order end date.', 'SCHEMA',
N'Production', 'TABLE', N'WorkOrder', 'COLUMN', N'EndDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product quantity to build.',
'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'COLUMN', N'OrderQty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product identification number.
Foreign key to Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'WorkOrder',
'COLUMN', N'ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Quantity that failed inspection.',
'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'COLUMN', N'ScrappedQty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Reason for inspection failure.',
'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'COLUMN', N'ScrapReasonID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Work order start date.', 'SCHEMA',
N'Production', 'TABLE', N'WorkOrder', 'COLUMN', N'StartDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Quantity built and put in
inventory.', 'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'COLUMN', N'StockedQty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for WorkOrder
records.', 'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'COLUMN', N'WorkOrderID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [EndDate] >=
[StartDate] OR [EndDate] IS NULL', 'SCHEMA', N'Production', 'TABLE', N'WorkOrder',
'CONSTRAINT', N'CK_WorkOrder_EndDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [OrderQty] > (0)',
'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'CONSTRAINT', N'CK_WorkOrder_Order-
Qty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [ScrappedQty] >=
(0)', 'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'CONSTRAINT', N'CK_WorkOrder_-
ScrappedQty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'CONSTRAINT', N'DF_Work-
Order_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'CONSTRAINT',
N'FK_WorkOrder_Product_ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
ScrapReason.ScrapReasonID.', 'SCHEMA', N'Production', 'TABLE', N'WorkOrder',
'CONSTRAINT', N'FK_WorkOrder_ScrapReason_ScrapReasonID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'CONSTRAINT', N'PK_Work-
Order_WorkOrderID'

```

```
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'WorkOrder', 'INDEX', N'IX_WorkOrder_ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'WorkOrder', 'INDEX', N'IX_WorkOrder_ScrapReasonID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'INDEX',
N'PK_WorkOrder_WorkOrderID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'AFTER INSERT trigger that inserts a
row in the TransactionHistory table.', 'SCHEMA', N'Production', 'TABLE', N'Work-
Order', 'TRIGGER', N'iWorkOrder'
GO
EXEC sp_addextendedproperty N'MS_Description', N'AFTER UPDATE trigger that inserts a
row in the TransactionHistory table, updates ModifiedDate in the WorkOrder table.',
'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'TRIGGER', N'uWorkOrder'
GO
```

Uses

[Production].[Product]
[Production].[ScrapReason]
Production

Used By

[Production].[WorkOrderRouting]

 **[Production].[WorkOrderRouting]**
MS_Description

Work order details.

Properties

Property	Value
Row Count (~)	67131
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	WorkOrderID <i>Primary key. Foreign key to Work-Order.WorkOrderID.</i>	int	4	False	
	ProductID <i>Primary key. Foreign key to Product.ProductID.</i>	int	4	False	
	OperationSequence <i>Primary key. Indicates the manufacturing process sequence.</i>	smallint	2	False	
	LocationID <i>Manufacturing location where the part is processed. Foreign key to Location.LocationID.</i>	smallint	2	False	
	ScheduledStartDate <i>Planned manufacturing start date.</i>	datetime	8	False	
	ScheduledEndDate <i>Planned manufacturing end date.</i>	datetime	8	False	
	ActualStartDate <i>Actual start date.</i>	datetime	8	True	
	ActualEndDate <i>Actual end date.</i>	datetime	8	True	
	ActualResourceHrs <i>Number of manufacturing hours used.</i>	decimal(9,4)	5	True	
	PlannedCost <i>Estimated manufacturing cost.</i>	money	8	False	
	ActualCost <i>Actual manufacturing cost.</i>	money	8	True	
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
-----	------	-------------	--------

	PK_WorkOrderRouting_WorkOrderID_ProductID_OperationSequence <i>Primary key (clustered) constraint</i>	WorkOrderID, ProductID, Operation- Sequence	True
	IX_WorkOrderRouting_ProductID <i>Nonclustered index.</i>	ProductID	

Check Constraints

Name	On Column	Constraint
CK_WorkOrderRouting_ActualCost <i>Check constraint [ActualCost] > (0.00)</i>	ActualCost	((Actual-Cost)>(0.00))
CK_WorkOrderRouting_ActualEndDate <i>Check constraint [ActualEndDate] >= [ActualStartDate] OR [ActualEndDate] IS NULL OR [ActualStartDate] IS NULL</i>		(([ActualEnd-Date]>=[Actual-StartDate] OR [ActualEndDate] IS NULL OR [ActualStartDate] IS NULL))
CK_WorkOrderRouting_ActualResourceHrs <i>Check constraint [ActualResourceHrs] >= (0.0000)</i>	ActualResourceHrs	((Actual-Resource-Hrs)>=(0.0000))
CK_WorkOrderRouting_PlannedCost <i>Check constraint [PlannedCost] > (0.00)</i>	PlannedCost	((Planned-Cost)>(0.00))
CK_WorkOrderRouting_ScheduledEndDate <i>Check constraint [ScheduledEndDate] >= [ScheduledStartDate]</i>		(([ScheduledEnd-Date]>=[Schedu ledStartDate])

Foreign Keys

Name	Columns
FK_WorkOrderRouting_Location_LocationID <i>Foreign key constraint referencing Location.LocationID.</i>	LocationID->[Production].[Location].[LocationID]
FK_WorkOrderRouting_WorkOrder_WorkOrderID <i>Foreign key constraint referencing WorkOrder.WorkOrderID.</i>	WorkOrderID->[Production].[WorkOrder].[WorkOrderID]

SQL Script

```
CREATE TABLE [Production].[WorkOrderRouting]
(
    [WorkOrderID] [int] NOT NULL,
    [ProductID] [int] NOT NULL,
    [OperationSequence] [smallint] NOT NULL,
    [LocationID] [smallint] NOT NULL,
    [ScheduledStartDate] [datetime] NOT NULL,
    [ScheduledEndDate] [datetime] NOT NULL,
    [ActualStartDate] [datetime] NULL,
    [ActualEndDate] [datetime] NULL,
    [ActualResourceHrs] [decimal] (9, 4) NULL,
    [PlannedCost] [money] NOT NULL,
    [ActualCost] [money] NULL,
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_WorkOrderRouting_ModifiedDate]
    DEFAULT (getdate())
) ON [PRIMARY]
```

```

GO
ALTER TABLE [Production].[WorkOrderRouting] ADD CONSTRAINT [CK_WorkOrderRouting_
ActualCost] CHECK (([ActualCost]>(0.00)))
GO
ALTER TABLE [Production].[WorkOrderRouting] ADD CONSTRAINT [CK_WorkOrderRouting_
ActualEndDate] CHECK (([ActualEndDate]>=[ActualStartDate] OR [ActualEndDate] IS NULL
OR [ActualStartDate] IS NULL))
GO
ALTER TABLE [Production].[WorkOrderRouting] ADD CONSTRAINT [CK_WorkOrderRouting_
ActualResourceHrs] CHECK (([ActualResourceHrs]>=(0.0000)))
GO
ALTER TABLE [Production].[WorkOrderRouting] ADD CONSTRAINT [CK_WorkOrderRouting_
PlannedCost] CHECK (([PlannedCost]>(0.00)))
GO
ALTER TABLE [Production].[WorkOrderRouting] ADD CONSTRAINT [CK_WorkOrderRouting_
ScheduledEndDate] CHECK (([ScheduledEndDate]>=[ScheduledStartDate]))
GO
ALTER TABLE [Production].[WorkOrderRouting] ADD CONSTRAINT [PK_WorkOrderRouting_Work
OrderID_ProductID_OperationSequence] PRIMARY KEY CLUSTERED ([WorkOrderID], [Product
ID], [OperationSequence]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_WorkOrderRouting_ProductID] ON [Production].[WorkOrder
Routing] ([ProductID]) ON [PRIMARY]
GO
ALTER TABLE [Production].[WorkOrderRouting] ADD CONSTRAINT [FK_WorkOrderRouting_
Location_LocationID] FOREIGN KEY ([LocationID]) REFERENCES [Production].[Location]
([LocationID])
GO
ALTER TABLE [Production].[WorkOrderRouting] ADD CONSTRAINT [FK_WorkOrderRouting_Work
Order_WorkOrderID] FOREIGN KEY ([WorkOrderID]) REFERENCES [Production].[WorkOrder]
([WorkOrderID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Work order details.', 'SCHEMA',
N'Production', 'TABLE', N'WorkOrderRouting', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Actual manufacturing cost.',
'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting', 'COLUMN', N'ActualCost'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Actual end date.', 'SCHEMA',
N'Production', 'TABLE', N'WorkOrderRouting', 'COLUMN', N'ActualEndDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Number of manufacturing hours
used.', 'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting', 'COLUMN', N'Actual-
ResourceHrs'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Actual start date.', 'SCHEMA',
N'Production', 'TABLE', N'WorkOrderRouting', 'COLUMN', N'ActualStartDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Manufacturing location where the
part is processed. Foreign key to Location.LocationID.', 'SCHEMA', N'Production',
'TABLE', N'WorkOrderRouting', 'COLUMN', N'LocationID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting', 'COLUMN',
N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Indicates the
manufacturing process sequence.', 'SCHEMA', N'Production', 'TABLE', N'WorkOrder-
Routing', 'COLUMN', N'OperationSequence'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Estimated manufacturing cost.',
'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting', 'COLUMN', N'PlannedCost'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Foreign key to

```

```

Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting',
'COLUMN', N'ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Planned manufacturing end date.',
'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting', 'COLUMN', N'ScheduledEndDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Planned manufacturing start date.',
'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting', 'COLUMN', N'ScheduledStart-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Foreign key to Work-
Order.WorkOrderID.', 'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting',
'COLUMN', N'WorkOrderID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [ActualCost] >
(0.00)', 'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting', 'CONSTRAINT', N'CK_-
WorkOrderRouting_ActualCost'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [ActualEndDate] >=
[ActualStartDate] OR [ActualEndDate] IS NULL OR [ActualStartDate] IS NULL',
'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting', 'CONSTRAINT', N'CK_WorkOrder-
Routing_ActualEndDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [ActualResource-
Hrs] >= (0.0000)', 'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting',
'CONSTRAINT', N'CK_WorkOrderRouting_ActualResourceHrs'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [PlannedCost] >
(0.00)', 'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting', 'CONSTRAINT', N'CK_-
WorkOrderRouting_PlannedCost'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [ScheduledEndDate]
>= [ScheduledStartDate]', 'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting',
'CONSTRAINT', N'CK_WorkOrderRouting_ScheduledEndDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting', 'CONSTRAINT',
N'DF_WorkOrderRouting_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Location.LocationID.', 'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting',
'CONSTRAINT', N'FK_WorkOrderRouting_Location_LocationID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
WorkOrder.WorkOrderID.', 'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting',
'CONSTRAINT', N'FK_WorkOrderRouting_WorkOrder_WorkOrderID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting', 'CONSTRAINT',
N'PK_WorkOrderRouting_WorkOrderID_ProductID_OperationSequence'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'WorkOrderRouting', 'INDEX', N'IX_WorkOrderRouting_Product-
ID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting',
'INDEX', N'PK_WorkOrderRouting_WorkOrderID_ProductID_OperationSequence'
GO

```

Uses

[Production].[Location]

[Production].[WorkOrder]

Production

 **[Purchasing].[ProductVendor]**
MS_Description

Cross-reference table mapping vendors with the products they supply.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	460
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
 	ProductID <i>Primary key. Foreign key to Product.ProductID.</i>	int	4	False	
 	BusinessEntityID <i>Primary key. Foreign key to Vendor.BusinessEntityID.</i>	int	4	False	
	AverageLeadTime <i>The average span of time (in days) between placing an order with the vendor and receiving the purchased product.</i>	int	4	False	
	StandardPrice <i>The vendor's usual selling price.</i>	money	8	False	
	LastReceiptCost <i>The selling price when last purchased.</i>	money	8	True	
	LastReceiptDate <i>Date the product was last received by the vendor.</i>	datetime	8	True	
	MinOrderQty <i>The maximum quantity that should be ordered.</i>	int	4	False	
	MaxOrderQty <i>The minimum quantity that should be ordered.</i>	int	4	False	
	OnOrderQty <i>The quantity currently on order.</i>	int	4	True	
	UnitMeasureCode <i>The product's unit of measure.</i>	nchar(3)	6	False	
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_ProductVendor_ProductID_BusinessEntityID <i>Primary key (clustered) constraint</i>	ProductID, Business-EntityID	True
	IX_ProductVendor_BusinessEntityID <i>Nonclustered index.</i>	BusinessEntityID	
	IX_ProductVendor_UnitMeasureCode <i>Nonclustered index.</i>	UnitMeasureCode	

Check Constraints

Name	On Column	Constraint
CK_ProductVendor_AverageLeadTime <i>Check constraint [AverageLeadTime] >= (1)</i>	AverageLeadTime	([AverageLeadTime]>=(1))
CK_ProductVendor_LastReceiptCost <i>Check constraint [LastReceiptCost] > (0.00)</i>	LastReceiptCost	([LastReceiptCost]>(0.00))
CK_ProductVendor_MaxOrderQty <i>Check constraint [MaxOrderQty] >= (1)</i>	MaxOrderQty	([MaxOrderQty]>=(1))
CK_ProductVendor_MinOrderQty <i>Check constraint [MinOrderQty] >= (1)</i>	MinOrderQty	([MinOrderQty]>=(1))
CK_ProductVendor_OnOrderQty <i>Check constraint [OnOrderQty] >= (0)</i>	OnOrderQty	([OnOrderQty]>=(0))
CK_ProductVendor_StandardPrice <i>Check constraint [StandardPrice] > (0.00)</i>	StandardPrice	([StandardPrice]>(0.00))

Foreign Keys

Name	Columns
FK_ProductVendor_Product_ProductID <i>Foreign key constraint referencing Product.ProductID.</i>	ProductID->[Production].[Product].[ProductID]
FK_ProductVendor_UnitMeasure_UnitMeasureCode <i>Foreign key constraint referencing UnitMeasure.UnitMeasureCode.</i>	UnitMeasureCode->[Production].[UnitMeasure].[UnitMeasureCode]
FK_ProductVendor_Vendor_BusinessEntityID <i>Foreign key constraint referencing Vendor.BusinessEntityID.</i>	BusinessEntityID->[Purchasing].[Vendor].[BusinessEntityID]

SQL Script

```
CREATE TABLE [Purchasing].[ProductVendor]
(
    [ProductID] [int] NOT NULL,
    [BusinessEntityID] [int] NOT NULL,
    [AverageLeadTime] [int] NOT NULL,
    [StandardPrice] [money] NOT NULL,
    [LastReceiptCost] [money] NULL,
    [LastReceiptDate] [datetime] NULL,
    [MinOrderQty] [int] NOT NULL,
    [MaxOrderQty] [int] NOT NULL,
    [OnOrderQty] [int] NULL,
    [UnitMeasureCode] [nvarchar] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_ProductVendor_ModifiedDate]
```

```

DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Purchasing].[ProductVendor] ADD CONSTRAINT [CK_ProductVendor_Average-
LeadTime] CHECK (([AverageLeadTime]>=(1)))
GO
ALTER TABLE [Purchasing].[ProductVendor] ADD CONSTRAINT [CK_ProductVendor_Last-
ReceiptCost] CHECK (([LastReceiptCost]>(0.00)))
GO
ALTER TABLE [Purchasing].[ProductVendor] ADD CONSTRAINT [CK_ProductVendor_MaxOrder-
Qty] CHECK (([MaxOrderQty]>=(1)))
GO
ALTER TABLE [Purchasing].[ProductVendor] ADD CONSTRAINT [CK_ProductVendor_MinOrder-
Qty] CHECK (([MinOrderQty]>=(1)))
GO
ALTER TABLE [Purchasing].[ProductVendor] ADD CONSTRAINT [CK_ProductVendor_OnOrder-
Qty] CHECK (([OnOrderQty]>=(0)))
GO
ALTER TABLE [Purchasing].[ProductVendor] ADD CONSTRAINT [CK_ProductVendor_Standard-
Price] CHECK (([StandardPrice]>(0.00)))
GO
ALTER TABLE [Purchasing].[ProductVendor] ADD CONSTRAINT [PK_ProductVendor_ProductID_-
BusinessEntityID] PRIMARY KEY CLUSTERED ([ProductID], [BusinessEntityID]) ON
[PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_ProductVendor_BusinessEntityID] ON
[Purchasing].[ProductVendor] ([BusinessEntityID]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_ProductVendor_UnitMeasureCode] ON
[Purchasing].[ProductVendor] ([UnitMeasureCode]) ON [PRIMARY]
GO
ALTER TABLE [Purchasing].[ProductVendor] ADD CONSTRAINT [FK_ProductVendor_Product_-
ProductID] FOREIGN KEY ([ProductID]) REFERENCES [Production].[Product] ([ProductID])
GO
ALTER TABLE [Purchasing].[ProductVendor] ADD CONSTRAINT [FK_ProductVendor_Unit-
Measure_UnitMeasureCode] FOREIGN KEY ([UnitMeasureCode]) REFERENCES
[Production].[UnitMeasure] ([UnitMeasureCode])
GO
ALTER TABLE [Purchasing].[ProductVendor] ADD CONSTRAINT [FK_ProductVendor_Vendor_-
BusinessEntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [Purchasing].[Vendor]
([BusinessEntityID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Cross-reference table mapping
vendors with the products they supply.', 'SCHEMA', N'Purchasing', 'TABLE', N'Product-
Vendor', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'The average span of time (in days)
between placing an order with the vendor and receiving the purchased product.',
'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'COLUMN', N'AverageLeadTime'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Foreign key to
Vendor.BusinessEntityID.', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor',
'COLUMN', N'BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'The selling price when last
purchased.', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'COLUMN', N'Last-
ReceiptCost'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date the product was last received
by the vendor.', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'COLUMN',
N'LastReceiptDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'The minimum quantity that should be
ordered.', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'COLUMN', N'MaxOrder-

```

```

Qty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'The maximum quantity that should be
ordered.', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'COLUMN', N'MinOrder-
Qty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'COLUMN', N'Modified-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'The quantity currently on order.',
'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'COLUMN', N'OnOrderQty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Foreign key to
Product.ProductID.', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'COLUMN',
N'ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'The vendor's usual selling
price.', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'COLUMN', N'Standard-
Price'
GO
EXEC sp_addextendedproperty N'MS_Description', N'The product's unit of measure.',
'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'COLUMN', N'UnitMeasureCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [AverageLeadTime]
>= (1)', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'CONSTRAINT', N'CK_-
ProductVendor_AverageLeadTime'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [LastReceiptCost]
> (0.00)', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'CONSTRAINT', N'CK_-
ProductVendor_LastReceiptCost'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [MaxOrderQty] >=
(1)', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'CONSTRAINT', N'CK_Product-
Vendor_MaxOrderQty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [MinOrderQty] >=
(1)', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'CONSTRAINT', N'CK_Product-
Vendor_MinOrderQty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [OnOrderQty] >=
(0)', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'CONSTRAINT', N'CK_Product-
Vendor_OnOrderQty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [StandardPrice] >
(0.00)', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'CONSTRAINT', N'CK_-
ProductVendor_StandardPrice'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'CONSTRAINT', N'DF_-
ProductVendor_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Product.ProductID.', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor',
'CONSTRAINT', N'FK_ProductVendor_Product_ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
UnitMeasure.UnitMeasureCode.', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor',
'CONSTRAINT', N'FK_ProductVendor_UnitMeasure_UnitMeasureCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Vendor.BusinessEntityID.', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor',
'CONSTRAINT', N'FK_ProductVendor_Vendor_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)

```

```
constraint', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'CONSTRAINT', N'PK_
ProductVendor_ProductID_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Purchasing', 'TABLE', N'ProductVendor', 'INDEX', N'IX_ProductVendor_BusinessEntity-
ID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Purchasing', 'TABLE', N'ProductVendor', 'INDEX', N'IX_ProductVendor_UnitMeasure-
Code'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor',
'INDEX', N'PK_ProductVendor_ProductID_BusinessEntityID'
GO
```

Uses

[Production].[Product]

[Production].[UnitMeasure]

[Purchasing].[Vendor]

Purchasing

 **[Purchasing].[PurchaseOrderDetail]**
MS_Description

Individual products associated with a specific purchase order. See PurchaseOrderHeader.

Properties

Property	Value
Row Count (~)	8845
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Computed	Max Length (Bytes)	Allow Nulls	Identity	Default
	PurchaseOrderID <i>Primary key. Foreign key to PurchaseOrderHeader.PurchaseOrderID.</i>	int		4	False		
	PurchaseOrderDetailID <i>Primary key. One line number per purchased product.</i>	int		4	False	1 - 1	
	DueDate <i>Date the product is expected to be received.</i>	datetime		8	False		
	OrderQty <i>Quantity ordered.</i>	smallint		2	False		
	ProductID <i>Product identification number. Foreign key to Product.ProductID.</i>	int		4	False		
	UnitPrice <i>Vendor's selling price of a single product.</i>	money		8	False		
	LineTotal <i>Per product subtotal. Computed as OrderQty * UnitPrice.</i>	money	True	8	False		
	ReceivedQty <i>Quantity actually received from the vendor.</i>	decimal(8,2)		5	False		
	RejectedQty <i>Quantity rejected during inspection.</i>	decimal(8,2)		5	False		
	StockedQty <i>Quantity accepted into inventory. Computed as ReceivedQty - RejectedQty.</i>	decimal(9,2)	True	5	False		
	ModifiedDate <i>Date and time the record</i>	datetime		8	False		(getdate())

	was last updated.						
--	-------------------	--	--	--	--	--	--

Computed columns

Name	Column definition
LineTotal	(isnull([OrderQty]*[UnitPrice],0.00))
StockedQty	(isnull([ReceivedQty]-[RejectedQty],0.00))

Indexes

Key	Name	Key Columns	Unique
	PK_PurchaseOrderDetail_PurchaseOrderID_PurchaseOrderDetailID <i>Primary key (clustered) constraint</i>	PurchaseOrderID, PurchaseOrderDetailID	True
	IX_PurchaseOrderDetail_ProductID <i>Nonclustered index.</i>	ProductID	

Triggers

Name	ANSI Nulls On	Quoted Identifier On	On
iPurchaseOrderDetail <i>AFTER INSERT trigger that inserts a row in the TransactionHistory table and updates the PurchaseOrderHeader.SubTotal column.</i>	True	True	After Insert
uPurchaseOrderDetail <i>AFTER UPDATE trigger that inserts a row in the TransactionHistory table, updates ModifiedDate in PurchaseOrderDetail and updates the PurchaseOrderHeader.SubTotal column.</i>	True	True	After Update

Check Constraints

Name	On Column	Constraint
CK_PurchaseOrderDetail_OrderQty <i>Check constraint [OrderQty] > (0)</i>	OrderQty	([OrderQty]>(0))
CK_PurchaseOrderDetail_ReceivedQty <i>Check constraint [ReceivedQty] >= (0.00)</i>	ReceivedQty	([ReceivedQty]>=(0.00))
CK_PurchaseOrderDetail_RejectedQty <i>Check constraint [RejectedQty] >= (0.00)</i>	RejectedQty	([RejectedQty]>=(0.00))
CK_PurchaseOrderDetail_UnitPrice <i>Check constraint [UnitPrice] >= (0.00)</i>	UnitPrice	([UnitPrice]>=(0.00))

Foreign Keys

Name	Columns
FK_PurchaseOrderDetail_Product_ProductID <i>Foreign key constraint referencing Product.ProductID.</i>	ProductID->[Production].[Product].[ProductID]
FK_PurchaseOrderDetail_PurchaseOrderHeader_PurchaseOrderID <i>Foreign key constraint referencing PurchaseOrder-</i>	PurchaseOrderID->[Purchasing].[PurchaseOrderHeader].[PurchaseOrderID]

Header.PurchaseOrderID.

SQL Script

```

CREATE TABLE [Purchasing].[PurchaseOrderDetail]
(
[PurchaseOrderID] [int] NOT NULL,
[PurchaseOrderDetailID] [int] NOT NULL IDENTITY(1, 1),
[DueDate] [datetime] NOT NULL,
[OrderQty] [smallint] NOT NULL,
[ProductID] [int] NOT NULL,
[UnitPrice] [money] NOT NULL,
[LineTotal] AS (isnull([OrderQty]*[UnitPrice],(0.00))),
[ReceivedQty] [decimal] (8, 2) NOT NULL,
[RejectedQty] [decimal] (8, 2) NOT NULL,
[StockedQty] AS (isnull([ReceivedQty]-[RejectedQty],(0.00))),
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_PurchaseOrderDetail_ModifiedDate]
DEFAULT (getdate())
) ON [PRIMARY]
GO

CREATE TRIGGER [Purchasing].[iPurchaseOrderDetail] ON [Purchasing].[PurchaseOrder-
Detail]
AFTER INSERT AS
BEGIN
    DECLARE @Count int;

    SET @Count = @@ROWCOUNT;
    IF @Count = 0
        RETURN;

    SET NOCOUNT ON;

    BEGIN TRY
        INSERT INTO [Production].[TransactionHistory]
            ([ProductID]
            , [ReferenceOrderID]
            , [ReferenceOrderLineID]
            , [TransactionType]
            , [TransactionDate]
            , [Quantity]
            , [ActualCost])
        SELECT
            inserted.[ProductID]
            , inserted.[PurchaseOrderID]
            , inserted.[PurchaseOrderDetailID]
            , 'P'
            , GETDATE()
            , inserted.[OrderQty]
            , inserted.[UnitPrice]
        FROM inserted
        INNER JOIN [Purchasing].[PurchaseOrderHeader]
            ON inserted.[PurchaseOrderID] = [Purchasing].[PurchaseOrder-
Header].[PurchaseOrderID];

        -- Update SubTotal in PurchaseOrderHeader record. Note that this causes the

```

```

-- PurchaseOrderHeader trigger to fire which will update the RevisionNumber.
UPDATE [Purchasing].[PurchaseOrderHeader]
SET [Purchasing].[PurchaseOrderHeader].[SubTotal] =
    (SELECT SUM([Purchasing].[PurchaseOrderDetail].[LineTotal])
     FROM [Purchasing].[PurchaseOrderDetail]
     WHERE [Purchasing].[PurchaseOrderHeader].[PurchaseOrderID] =
[Purchasing].[PurchaseOrderDetail].[PurchaseOrderID])
WHERE [Purchasing].[PurchaseOrderHeader].[PurchaseOrderID] IN (SELECT
inserted.[PurchaseOrderID] FROM inserted);
END TRY
BEGIN CATCH
    EXECUTE [dbo].[uspPrintError];

-- Rollback any active or uncommittable transactions before
-- inserting information in the ErrorLog
IF @@TRANCOUNT > 0
BEGIN
    ROLLBACK TRANSACTION;
END

EXECUTE [dbo].[uspLogError];
END CATCH;
END;
GO

CREATE TRIGGER [Purchasing].[uPurchaseOrderDetail] ON [Purchasing].[PurchaseOrder-
Detail]
AFTER UPDATE AS
BEGIN
    DECLARE @Count int;

    SET @Count = @@ROWCOUNT;
    IF @Count = 0
        RETURN;

    SET NOCOUNT ON;

    BEGIN TRY
        IF UPDATE([ProductID]) OR UPDATE([OrderQty]) OR UPDATE([UnitPrice])
        -- Insert record into TransactionHistory
        BEGIN
            INSERT INTO [Production].[TransactionHistory]
                ([ProductID]
                , [ReferenceOrderID]
                , [ReferenceOrderLineID]
                , [TransactionType]
                , [TransactionDate]
                , [Quantity]
                , [ActualCost])
            SELECT
                inserted.[ProductID]
                , inserted.[PurchaseOrderID]
                , inserted.[PurchaseOrderDetailID]
                , 'P'
                , GETDATE()
                , inserted.[OrderQty]
                , inserted.[UnitPrice]
            FROM inserted

```

```

        INNER JOIN [Purchasing].[PurchaseOrderDetail]
            ON inserted.[PurchaseOrderID] = [Purchasing].[PurchaseOrder-
Detail].[PurchaseOrderID];

-- Update SubTotal in PurchaseOrderHeader record. Note that this causes
the
-- PurchaseOrderHeader trigger to fire which will update the Revision-
Number.
UPDATE [Purchasing].[PurchaseOrderHeader]
SET [Purchasing].[PurchaseOrderHeader].[SubTotal] =
    (SELECT SUM([Purchasing].[PurchaseOrderDetail].[LineTotal])
    FROM [Purchasing].[PurchaseOrderDetail]
    WHERE [Purchasing].[PurchaseOrderHeader].[PurchaseOrderID]
        = [Purchasing].[PurchaseOrderDetail].[PurchaseOrderID])
WHERE [Purchasing].[PurchaseOrderHeader].[PurchaseOrderID]
    IN (SELECT inserted.[PurchaseOrderID] FROM inserted);

UPDATE [Purchasing].[PurchaseOrderDetail]
SET [Purchasing].[PurchaseOrderDetail].[ModifiedDate] = GETDATE()
FROM inserted
WHERE inserted.[PurchaseOrderID] = [Purchasing].[PurchaseOrder-
Detail].[PurchaseOrderID]
    AND inserted.[PurchaseOrderDetailID] = [Purchasing].[PurchaseOrder-
Detail].[PurchaseOrderDetailID];
END;
END TRY
BEGIN CATCH
    EXECUTE [dbo].[uspPrintError];

-- Rollback any active or uncommittable transactions before
-- inserting information in the ErrorLog
IF @@TRANCOUNT > 0
BEGIN
    ROLLBACK TRANSACTION;
END

EXECUTE [dbo].[uspLogError];
END CATCH;
END;
GO
ALTER TABLE [Purchasing].[PurchaseOrderDetail] ADD CONSTRAINT [CK_PurchaseOrder-
Detail_OrderQty] CHECK (([OrderQty]>(0)))
GO
ALTER TABLE [Purchasing].[PurchaseOrderDetail] ADD CONSTRAINT [CK_PurchaseOrder-
Detail_ReceivedQty] CHECK (([ReceivedQty]>=(0.00)))
GO
ALTER TABLE [Purchasing].[PurchaseOrderDetail] ADD CONSTRAINT [CK_PurchaseOrder-
Detail_RejectedQty] CHECK (([RejectedQty]>=(0.00)))
GO
ALTER TABLE [Purchasing].[PurchaseOrderDetail] ADD CONSTRAINT [CK_PurchaseOrder-
Detail_UnitPrice] CHECK (([UnitPrice]>=(0.00)))
GO
ALTER TABLE [Purchasing].[PurchaseOrderDetail] ADD CONSTRAINT [PK_PurchaseOrder-
Detail_PurchaseOrderID_PurchaseOrderDetailID] PRIMARY KEY CLUSTERED ([PurchaseOrder-
ID], [PurchaseOrderDetailID]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_PurchaseOrderDetail_ProductID] ON
[Purchasing].[PurchaseOrderDetail] ([ProductID]) ON [PRIMARY]
GO
ALTER TABLE [Purchasing].[PurchaseOrderDetail] ADD CONSTRAINT [FK_PurchaseOrder-

```

```

Detail_Product_ProductID] FOREIGN KEY ([ProductID]) REFERENCES
[Production].[Product] ([ProductID])
GO
ALTER TABLE [Purchasing].[PurchaseOrderDetail] ADD CONSTRAINT [FK_PurchaseOrder-
Detail_PurchaseOrderHeader_PurchaseOrderID] FOREIGN KEY ([PurchaseOrderID])
REFERENCES [Purchasing].[PurchaseOrderHeader] ([PurchaseOrderID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Individual products associated with
a specific purchase order. See PurchaseOrderHeader.', 'SCHEMA', N'Purchasing',
'TABLE', N'PurchaseOrderDetail', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date the product is expected to be
received.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderDetail', 'COLUMN',
N'DueDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Per product subtotal. Computed as
OrderQty * UnitPrice.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderDetail',
'COLUMN', N'LineTotal'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderDetail', 'COLUMN',
N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Quantity ordered.', 'SCHEMA',
N'Purchasing', 'TABLE', N'PurchaseOrderDetail', 'COLUMN', N'OrderQty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product identification number.
Foreign key to Product.ProductID.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase-
OrderDetail', 'COLUMN', N'ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. One line number per
purchased product.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderDetail',
'COLUMN', N'PurchaseOrderDetailID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Foreign key to
PurchaseOrderHeader.PurchaseOrderID.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase-
OrderDetail', 'COLUMN', N'PurchaseOrderID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Quantity actually received from the
vendor.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderDetail', 'COLUMN',
N'ReceivedQty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Quantity rejected during
inspection.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderDetail', 'COLUMN',
N'RejectedQty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Quantity accepted into inventory.
Computed as ReceivedQty - RejectedQty.', 'SCHEMA', N'Purchasing', 'TABLE',
N'PurchaseOrderDetail', 'COLUMN', N'StockedQty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Vendor''s selling price of a single
product.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderDetail', 'COLUMN',
N'UnitPrice'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [OrderQty] > (0)',
'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderDetail', 'CONSTRAINT', N'CK_-
PurchaseOrderDetail_OrderQty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [ReceivedQty] >=
(0.00)', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderDetail', 'CONSTRAINT',
N'CK_PurchaseOrderDetail_ReceivedQty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [RejectedQty] >=
(0.00)', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderDetail', 'CONSTRAINT',
N'CK_PurchaseOrderDetail_RejectedQty'

```

```

GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [UnitPrice] >=
(0.00)', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderDetail', 'CONSTRAINT',
N'CK_PurchaseOrderDetail_UnitPrice'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderDetail', 'CONSTRAINT',
N'DF_PurchaseOrderDetail_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Product.ProductID.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderDetail',
'CONSTRAINT', N'FK_PurchaseOrderDetail_Product_ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
PurchaseOrderHeader.PurchaseOrderID.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase-
OrderDetail', 'CONSTRAINT', N'FK_PurchaseOrderDetail_PurchaseOrderHeader_Purchase-
OrderID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderDetail', 'CONSTRAINT',
N'PK_PurchaseOrderDetail_PurchaseOrderID_PurchaseOrderDetailID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Purchasing', 'TABLE', N'PurchaseOrderDetail', 'INDEX', N'IX_PurchaseOrderDetail_-
ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderDetail',
'INDEX', N'PK_PurchaseOrderDetail_PurchaseOrderID_PurchaseOrderDetailID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'AFTER INSERT trigger that inserts a
row in the TransactionHistory table and updates the PurchaseOrderHeader.SubTotal
column.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderDetail', 'TRIGGER', N'i-
PurchaseOrderDetail'
GO
EXEC sp_addextendedproperty N'MS_Description', N'AFTER UPDATE trigger that inserts a
row in the TransactionHistory table, updates ModifiedDate in PurchaseOrderDetail and
updates the PurchaseOrderHeader.SubTotal column.', 'SCHEMA', N'Purchasing', 'TABLE',
N'PurchaseOrderDetail', 'TRIGGER', N'uPurchaseOrderDetail'
GO

```

Uses

[Production].[Product]
[Purchasing].[PurchaseOrderHeader]
Purchasing

 **[Purchasing].[PurchaseOrderHeader]**
MS_Description

General purchase order information. See PurchaseOrderDetail.

Properties

Property	Value
Row Count (~)	4012
Created	13:14:19 14 marca 2012
Last Modified	15:26:58 22 marca 2017

Columns

Key	Name	Data Type	Persisted	Computed	Max Length (Bytes)	Allow Nulls	Identity	Default
	PurchaseOrderID <i>Primary key.</i>	int			4	False	1 - 1	
	RevisionNumber <i>Incremental number to track changes to the purchase order over time.</i>	tinyint			1	False		((0))
	Status <i>Order current status. 1 = Pending; 2 = Approved; 3 = Rejected; 4 = Complete</i>	tinyint			1	False		((1))
	EmployeeID <i>Employee who created the purchase order. Foreign key to Employee.BusinessEntityID.</i>	int			4	False		
	VendorID <i>Vendor with whom the purchase order is placed. Foreign key to Vendor.BusinessEntityID.</i>	int			4	False		
	ShipMethodID <i>Shipping method. Foreign key to ShipMethod.ShipMethodID.</i>	int			4	False		
	OrderDate <i>Purchase order creation date.</i>	datetime			8	False		(getdate())
	ShipDate <i>Estimated shipment date from the vendor.</i>	datetime			8	True		
	SubTotal <i>Purchase order subtotal. Computed as SUM(PurchaseOrder-</i>	money			8	False		((0.00))

	<i>Detail.LineTotal</i>)for the appropriate PurchaseOrderID.							
	TaxAmt <i>Tax amount.</i>	money			8	False		((0.00))
	Freight <i>Shipping cost.</i>	money			8	False		((0.00))
	TotalDue <i>Total due to vendor. Computed as Subtotal + TaxAmt + Freight.</i>	money	True	True	8	False		
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime			8	False		(getdate())

Computed columns

Name	Column definition
TotalDue	(isnull(([SubTotal]+[TaxAmt])+[Freight],(0)))

Indexes

Key	Name	Key Columns	Unique
	PK_PurchaseOrderHeader_PurchaseOrderID <i>Primary key (clustered) constraint</i>	PurchaseOrderID	True
	IX_PurchaseOrderHeader_EmployeeID <i>Nonclustered index.</i>	EmployeeID	
	IX_PurchaseOrderHeader_VendorID <i>Nonclustered index.</i>	VendorID	

Triggers

Name	ANSI Nulls On	Quoted Identifier On	On
uPurchaseOrderHeader <i>AFTER UPDATE trigger that updates the Revision-Number and ModifiedDate columns in the PurchaseOrder-Header table.</i>	True	True	After Update

Check Constraints

Name	On Column	Constraint
CK_PurchaseOrderHeader_Freight <i>Check constraint [Freight] >= (0.00)</i>	Freight	([Freight]>=(0.00))
CK_PurchaseOrderHeader_ShipDate <i>Check constraint [ShipDate] >= [OrderDate] OR [ShipDate] IS NULL</i>		([ShipDate]>=[OrderDate] OR [ShipDate] IS NULL)
CK_PurchaseOrderHeader_Status <i>Check constraint [Status] BETWEEN (1) AND (4)</i>	Status	([Status]>=(1) AND [Status]<=(4))
CK_PurchaseOrderHeader_SubTotal <i>Check constraint [SubTotal] >= (0.00)</i>	SubTotal	([SubTotal]>=(0.00))

CK_PurchaseOrderHeader_TaxAmt <i>Check constraint [TaxAmt] >= (0.00)</i>	TaxAmt	([TaxAmt]>=(0.00))
--	--------	--------------------

Foreign Keys

Name	Columns
FK_PurchaseOrderHeader_Employee_EmployeeID <i>Foreign key constraint referencing Employee.EmployeeID.</i>	EmployeeID->[Human-Resources].[Employee].[BusinessEntityID]
FK_PurchaseOrderHeader_ShipMethod_ShipMethodID <i>Foreign key constraint referencing ShipMethod.ShipMethodID.</i>	ShipMethodID->[Purchasing].[ShipMethod].[ShipMethodID]
FK_PurchaseOrderHeader_Vendor_VendorID <i>Foreign key constraint referencing Vendor.VendorID.</i>	VendorID->[Purchasing].[Vendor].[BusinessEntityID]

SQL Script

```

CREATE TABLE [Purchasing].[PurchaseOrderHeader]
(
    [PurchaseOrderID] [int] NOT NULL IDENTITY(1, 1),
    [RevisionNumber] [tinyint] NOT NULL CONSTRAINT [DF_PurchaseOrderHeader_RevisionNumber] DEFAULT ((0)),
    [Status] [tinyint] NOT NULL CONSTRAINT [DF_PurchaseOrderHeader_Status] DEFAULT ((1)),
    [EmployeeID] [int] NOT NULL,
    [VendorID] [int] NOT NULL,
    [ShipMethodID] [int] NOT NULL,
    [OrderDate] [datetime] NOT NULL CONSTRAINT [DF_PurchaseOrderHeader_OrderDate] DEFAULT (getdate()),
    [ShipDate] [datetime] NULL,
    [SubTotal] [money] NOT NULL CONSTRAINT [DF_PurchaseOrderHeader_SubTotal] DEFAULT ((0.00)),
    [TaxAmt] [money] NOT NULL CONSTRAINT [DF_PurchaseOrderHeader_TaxAmt] DEFAULT ((0.00)),
    [Freight] [money] NOT NULL CONSTRAINT [DF_PurchaseOrderHeader_Freight] DEFAULT ((0.00)),
    [TotalDue] AS (isnull(([SubTotal]+[TaxAmt])+[Freight], (0))) PERSISTED NOT NULL,
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_PurchaseOrderHeader_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
GO

CREATE TRIGGER [Purchasing].[uPurchaseOrderHeader] ON [Purchasing].[PurchaseOrderHeader]
AFTER UPDATE AS
BEGIN
    DECLARE @Count int;

    SET @Count = @@ROWCOUNT;
    IF @Count = 0
        RETURN;

    SET NOCOUNT ON;

    BEGIN TRY
        -- Update RevisionNumber for modification of any field EXCEPT the Status.

```

```

IF NOT UPDATE([Status])
BEGIN
    UPDATE [Purchasing].[PurchaseOrderHeader]
    SET [Purchasing].[PurchaseOrderHeader].[RevisionNumber] =
        [Purchasing].[PurchaseOrderHeader].[RevisionNumber] + 1
    WHERE [Purchasing].[PurchaseOrderHeader].[PurchaseOrderID] IN
        (SELECT inserted.[PurchaseOrderID] FROM inserted);

    END;
END TRY
BEGIN CATCH
    EXECUTE [dbo].[uspPrintError];

    -- Rollback any active or uncommittable transactions before
    -- inserting information in the ErrorLog
    IF @@TRANCOUNT > 0
    BEGIN
        ROLLBACK TRANSACTION;
    END

    EXECUTE [dbo].[uspLogError];
END CATCH;
END;
GO
ALTER TABLE [Purchasing].[PurchaseOrderHeader] ADD CONSTRAINT [CK_PurchaseOrder-
Header_Freight] CHECK (([Freight]>=(0.00)))
GO
ALTER TABLE [Purchasing].[PurchaseOrderHeader] ADD CONSTRAINT [CK_PurchaseOrder-
Header_ShipDate] CHECK (([ShipDate]>=[OrderDate] OR [ShipDate] IS NULL))
GO
ALTER TABLE [Purchasing].[PurchaseOrderHeader] ADD CONSTRAINT [CK_PurchaseOrder-
Header_Status] CHECK (([Status]>=(1) AND [Status]<=(4)))
GO
ALTER TABLE [Purchasing].[PurchaseOrderHeader] ADD CONSTRAINT [CK_PurchaseOrder-
Header_SubTotal] CHECK (([SubTotal]>=(0.00)))
GO
ALTER TABLE [Purchasing].[PurchaseOrderHeader] ADD CONSTRAINT [CK_PurchaseOrder-
Header_TaxAmt] CHECK (([TaxAmt]>=(0.00)))
GO
ALTER TABLE [Purchasing].[PurchaseOrderHeader] ADD CONSTRAINT [PK_PurchaseOrder-
Header_PurchaseOrderID] PRIMARY KEY CLUSTERED ([PurchaseOrderID]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_PurchaseOrderHeader_EmployeeID] ON
[Purchasing].[PurchaseOrderHeader] ([EmployeeID]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_PurchaseOrderHeader_VendorID] ON
[Purchasing].[PurchaseOrderHeader] ([VendorID]) ON [PRIMARY]
GO
ALTER TABLE [Purchasing].[PurchaseOrderHeader] ADD CONSTRAINT [FK_PurchaseOrder-
Header_Employee_EmployeeID] FOREIGN KEY ([EmployeeID]) REFERENCES [Human-
Resources].[Employee] ([BusinessEntityID])
GO
ALTER TABLE [Purchasing].[PurchaseOrderHeader] ADD CONSTRAINT [FK_PurchaseOrder-
Header_ShipMethod_ShipMethodID] FOREIGN KEY ([ShipMethodID]) REFERENCES
[Purchasing].[ShipMethod] ([ShipMethodID])
GO
ALTER TABLE [Purchasing].[PurchaseOrderHeader] ADD CONSTRAINT [FK_PurchaseOrder-
Header_Vendor_VendorID] FOREIGN KEY ([VendorID]) REFERENCES [Purchasing].[Vendor]
([BusinessEntityID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'General purchase order information.
See PurchaseOrderDetail.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader',

```

```

NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Employee who created the purchase
order. Foreign key to Employee.BusinessEntityID.', 'SCHEMA', N'Purchasing', 'TABLE',
N'PurchaseOrderHeader', 'COLUMN', N'EmployeeID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Shipping cost.', 'SCHEMA',
N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'COLUMN', N'Freight'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'COLUMN',
N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Purchase order creation date.',
'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'COLUMN', N'OrderDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key.', 'SCHEMA',
N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'COLUMN', N'PurchaseOrderID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Incremental number to track changes
to the purchase order over time.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrder-
Header', 'COLUMN', N'RevisionNumber'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Estimated shipment date from the
vendor.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'COLUMN', N'Ship-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Shipping method. Foreign key to
ShipMethod.ShipMethodID.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader',
'COLUMN', N'ShipMethodID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Order current status. 1 = Pending;
2 = Approved; 3 = Rejected; 4 = Complete', 'SCHEMA', N'Purchasing', 'TABLE',
N'PurchaseOrderHeader', 'COLUMN', N'Status'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Purchase order subtotal. Computed
as SUM(PurchaseOrderDetail.LineTotal)for the appropriate PurchaseOrderID.',
'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'COLUMN', N'SubTotal'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Tax amount.', 'SCHEMA',
N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'COLUMN', N'TaxAmt'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Total due to vendor. Computed as
Subtotal + TaxAmt + Freight.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrder-
Header', 'COLUMN', N'TotalDue'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Vendor with whom the purchase order
is placed. Foreign key to Vendor.BusinessEntityID.', 'SCHEMA', N'Purchasing',
'TABLE', N'PurchaseOrderHeader', 'COLUMN', N'VendorID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Freight] >=
(0.00)', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'CONSTRAINT',
N'CK_PurchaseOrderHeader_Freight'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [ShipDate] >=
[OrderDate] OR [ShipDate] IS NULL', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase-
OrderHeader', 'CONSTRAINT', N'CK_PurchaseOrderHeader_ShipDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Status] BETWEEN
(1) AND (4)', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader',
'CONSTRAINT', N'CK_PurchaseOrderHeader_Status'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [SubTotal] >=
(0.00)', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'CONSTRAINT',

```

```

N'CK_PurchaseOrderHeader_SubTotal'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [TaxAmt] >=
(0.00)', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'CONSTRAINT',
N'CK_PurchaseOrderHeader_TaxAmt'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'CONSTRAINT', N'DF_-
PurchaseOrderHeader_Freight'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'CONSTRAINT',
N'DF_PurchaseOrderHeader_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'CONSTRAINT',
N'DF_PurchaseOrderHeader_OrderDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0',
'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'CONSTRAINT', N'DF_-
PurchaseOrderHeader_RevisionNumber'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 1',
'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'CONSTRAINT', N'DF_-
PurchaseOrderHeader_Status'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'CONSTRAINT', N'DF_-
PurchaseOrderHeader_SubTotal'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'CONSTRAINT', N'DF_-
PurchaseOrderHeader_TaxAmt'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Employee.EmployeeID.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader',
'CONSTRAINT', N'FK_PurchaseOrderHeader_Employee_EmployeeID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
ShipMethod.ShipMethodID.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader',
'CONSTRAINT', N'FK_PurchaseOrderHeader_ShipMethod_ShipMethodID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Vendor.VendorID.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader',
'CONSTRAINT', N'FK_PurchaseOrderHeader_Vendor_VendorID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'CONSTRAINT',
N'PK_PurchaseOrderHeader_PurchaseOrderID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'INDEX', N'IX_PurchaseOrderHeader_-
EmployeeID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'INDEX', N'IX_PurchaseOrderHeader_-
VendorID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader',
'INDEX', N'PK_PurchaseOrderHeader_PurchaseOrderID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'AFTER UPDATE trigger that updates
the RevisionNumber and ModifiedDate columns in the PurchaseOrderHeader table.',
'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'TRIGGER', N'uPurchase-

```

OrderHeader'

GO

Uses

[HumanResources].[Employee]

[Purchasing].[ShipMethod]

[Purchasing].[Vendor]

Purchasing

Used By

[Purchasing].[PurchaseOrderDetail]

 **[Purchasing].[ShipMethod]****MS_Description**

Shipping company lookup table.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	5
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	ShipMethodID <i>Primary key for ShipMethod records.</i>	int	4	False	1 - 1	
	Name <i>Shipping company name.</i>	[dbo].[Name]	100	False		
	ShipBase <i>Minimum shipping charge.</i>	money	8	False		((0.00))
	ShipRate <i>Shipping charge per pound.</i>	money	8	False		((0.00))
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False		(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_ShipMethod_ShipMethodID <i>Primary key (clustered) constraint</i>	ShipMethodID	True
	AK_ShipMethod_Name <i>Unique nonclustered index.</i>	Name	True
	AK_ShipMethod_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True

Check Constraints

Name	On Column	Constraint
CK_ShipMethod_ShipBase <i>Check constraint [ShipBase] > (0.00)</i>	ShipBase	([ShipBase]>(0.00))
CK_ShipMethod_ShipRate <i>Check constraint [ShipRate] > (0.00)</i>	ShipRate	([ShipRate]>(0.00))

SQL Script

```

CREATE TABLE [Purchasing].[ShipMethod]
(
    [ShipMethodID] [int] NOT NULL IDENTITY(1, 1),
    [Name] [dbo].[Name] NOT NULL,
    [ShipBase] [money] NOT NULL CONSTRAINT [DF_ShipMethod_ShipBase] DEFAULT ((0.00)),
    [ShipRate] [money] NOT NULL CONSTRAINT [DF_ShipMethod_ShipRate] DEFAULT ((0.00)),
    [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_ShipMethod_rowguid]
    DEFAULT (newid()),
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_ShipMethod_ModifiedDate] DEFAULT
    (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Purchasing].[ShipMethod] ADD CONSTRAINT [CK_ShipMethod_ShipBase] CHECK
(([ShipBase]>(0.00)))
GO
ALTER TABLE [Purchasing].[ShipMethod] ADD CONSTRAINT [CK_ShipMethod_ShipRate] CHECK
(([ShipRate]>(0.00)))
GO
ALTER TABLE [Purchasing].[ShipMethod] ADD CONSTRAINT [PK_ShipMethod_ShipMethodID]
PRIMARY KEY CLUSTERED ([ShipMethodID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_ShipMethod_Name] ON [Purchasing].[ShipMethod]
([Name]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_ShipMethod_rowguid] ON [Purchasing].[Ship-
Method] ([rowguid]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Shipping company lookup table.',
'SHEMA', N'Purchasing', 'TABLE', N'ShipMethod', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Purchasing', 'TABLE', N'ShipMethod', 'COLUMN', N'Modified-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Shipping company name.', 'SCHEMA',
N'Purchasing', 'TABLE', N'ShipMethod', 'COLUMN', N'Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Purchasing', 'TABLE', N'ShipMethod', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Minimum shipping charge.',
'SHEMA', N'Purchasing', 'TABLE', N'ShipMethod', 'COLUMN', N'ShipBase'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for ShipMethod
records.', 'SCHEMA', N'Purchasing', 'TABLE', N'ShipMethod', 'COLUMN', N'ShipMethod-
ID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Shipping charge per pound.',
'SHEMA', N'Purchasing', 'TABLE', N'ShipMethod', 'COLUMN', N'ShipRate'

```

```

GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [ShipBase] >
(0.00)', 'SCHEMA', N'Purchasing', 'TABLE', N'ShipMethod', 'CONSTRAINT', N'CK_Ship-
Method_ShipBase'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [ShipRate] >
(0.00)', 'SCHEMA', N'Purchasing', 'TABLE', N'ShipMethod', 'CONSTRAINT', N'CK_Ship-
Method_ShipRate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Purchasing', 'TABLE', N'ShipMethod', 'CONSTRAINT', N'DF_Ship-
Method_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Purchasing', 'TABLE', N'ShipMethod', 'CONSTRAINT', N'DF_Ship-
Method_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Purchasing', 'TABLE', N'ShipMethod', 'CONSTRAINT', N'DF_ShipMethod_Ship-
Base'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Purchasing', 'TABLE', N'ShipMethod', 'CONSTRAINT', N'DF_ShipMethod_Ship-
Rate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Purchasing', 'TABLE', N'ShipMethod', 'CONSTRAINT', N'PK_-
ShipMethod_ShipMethodID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SCHEMA', N'Purchasing', 'TABLE', N'ShipMethod', 'INDEX', N'AK_ShipMethod_Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Purchasing', 'TABLE', N'ShipMethod',
'INDEX', N'AK_ShipMethod_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Purchasing', 'TABLE', N'ShipMethod', 'INDEX',
N'PK_ShipMethod_ShipMethodID'
GO

```

Uses

[dbo].[Name]
Purchasing

Used By

[Purchasing].[PurchaseOrderHeader]
[Sales].[SalesOrderHeader]

 **[Purchasing].[Vendor]****MS_Description**

Companies from whom Adventure Works Cycles purchases parts or other goods.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	104
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	BusinessEntityID <i>Primary key for Vendor records. Foreign key to BusinessEntity.BusinessEntityID</i>	int	4	False	
	AccountNumber <i>Vendor account (identification) number.</i>	[dbo].[AccountNumber]	30	False	
	Name <i>Company name.</i>	[dbo].[Name]	100	False	
	CreditRating <i>1 = Superior, 2 = Excellent, 3 = Above average, 4 = Average, 5 = Below average</i>	tinyint	1	False	
	PreferredVendorStatus <i>0 = Do not use if another vendor is available. 1 = Preferred over other vendors supplying the same product.</i>	[dbo].[Flag]	1	False	((1))
	ActiveFlag <i>0 = Vendor no longer used. 1 = Vendor is actively used.</i>	[dbo].[Flag]	1	False	((1))
	PurchasingWebServiceURL <i>Vendor URL.</i>	nvarchar(1024)	2048	True	
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate ())

Indexes

Key	Name	Key Columns	Unique
	PK_Vendor_BusinessEntityID <i>Primary key (clustered) constraint</i>	BusinessEntityID	True
	AK_Vendor_AccountNumber <i>Unique nonclustered index.</i>	AccountNumber	True

Triggers

Name	ANSI Nulls On	Quoted Identifier On	On	Not For Replication
dVendor <i>INSTEAD OF DELETE trigger which keeps Vendors from being deleted.</i>	True	True	Instead Of Delete	True

Check Constraints

Name	On Column	Constraint
CK_Vendor_CreditRating <i>Check constraint [CreditRating] BETWEEN (1) AND (5)</i>	CreditRating	([CreditRating]>=(1) AND [CreditRating]<=(5))

Foreign Keys

Name	Columns
FK_Vendor_BusinessEntity_BusinessEntityID <i>Foreign key constraint referencing Business-Entity.BusinessEntityID</i>	BusinessEntityID->[Person].[BusinessEntity].[Business-EntityID]

SQL Script

```
CREATE TABLE [Purchasing].[Vendor]
(
  [BusinessEntityID] [int] NOT NULL,
  [AccountNumber] [dbo].[AccountNumber] NOT NULL,
  [Name] [dbo].[Name] NOT NULL,
  [CreditRating] [tinyint] NOT NULL,
  [PreferredVendorStatus] [dbo].[Flag] NOT NULL CONSTRAINT [DF_Vendor_PREFERREDVendor-Status] DEFAULT ((1)),
  [ActiveFlag] [dbo].[Flag] NOT NULL CONSTRAINT [DF_Vendor_ActiveFlag] DEFAULT ((1)),
  [PurchasingWebServiceURL] [nvarchar] (1024) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_Vendor_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
GO

CREATE TRIGGER [Purchasing].[dVendor] ON [Purchasing].[Vendor]
INSTEAD OF DELETE NOT FOR REPLICATION AS
BEGIN
  DECLARE @Count int;

  SET @Count = @@ROWCOUNT;
  IF @Count = 0
    RETURN;

  SET NOCOUNT ON;

  BEGIN TRY
    DECLARE @DeleteCount int;
```

```

SELECT @DeleteCount = COUNT(*) FROM deleted;
IF @DeleteCount > 0
BEGIN
    RAISERROR
        (N'Vendors cannot be deleted. They can only be marked as not
active.', -- Message
        10, -- Severity.
        1); -- State.

    -- Rollback any active or uncommittable transactions
    IF @@TRANCOUNT > 0
    BEGIN
        ROLLBACK TRANSACTION;
    END
END;
END TRY
BEGIN CATCH
    EXECUTE [dbo].[uspPrintError];

    -- Rollback any active or uncommittable transactions before
    -- inserting information in the ErrorLog
    IF @@TRANCOUNT > 0
    BEGIN
        ROLLBACK TRANSACTION;
    END

    EXECUTE [dbo].[uspLogError];
END CATCH;
END;
GO
ALTER TABLE [Purchasing].[Vendor] ADD CONSTRAINT [CK_Vendor_CreditRating] CHECK
(( [CreditRating] >= (1) AND [CreditRating] <= (5) ))
GO
ALTER TABLE [Purchasing].[Vendor] ADD CONSTRAINT [PK_Vendor_BusinessEntityID]
PRIMARY KEY CLUSTERED ([BusinessEntityID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_Vendor_AccountNumber] ON [Purchasing].[Vendor]
([AccountNumber]) ON [PRIMARY]
GO
ALTER TABLE [Purchasing].[Vendor] ADD CONSTRAINT [FK_Vendor_BusinessEntity_Business-
EntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [Person].[BusinessEntity]
([BusinessEntityID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Companies from whom Adventure Works
Cycles purchases parts or other goods.', 'SCHEMA', N'Purchasing', 'TABLE',
N'Vendor', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Vendor account (identification)
number.', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'COLUMN', N'AccountNumber'
GO
EXEC sp_addextendedproperty N'MS_Description', N'0 = Vendor no longer used. 1 =
Vendor is actively used.', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'COLUMN',
N'ActiveFlag'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Vendor records.
Foreign key to BusinessEntity.BusinessEntityID', 'SCHEMA', N'Purchasing', 'TABLE',
N'Vendor', 'COLUMN', N'BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'1 = Superior, 2 = Excellent, 3 =
Above average, 4 = Average, 5 = Below average', 'SCHEMA', N'Purchasing', 'TABLE',

```

```

N'Vendor', 'COLUMN', N'CreditRating'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Company name.', 'SCHEMA',
N'Purchasing', 'TABLE', N'Vendor', 'COLUMN', N'Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'0 = Do not use if another vendor is
available. 1 = Preferred over other vendors supplying the same product.', 'SCHEMA',
N'Purchasing', 'TABLE', N'Vendor', 'COLUMN', N'PreferredVendorStatus'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Vendor URL.', 'SCHEMA',
N'Purchasing', 'TABLE', N'Vendor', 'COLUMN', N'PurchasingWebServiceURL'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [CreditRating]
BETWEEN (1) AND (5)', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'CONSTRAINT',
N'CK_Vendor_CreditRating'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 1
(TRUE)', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'CONSTRAINT', N'DF_Vendor_
ActiveFlag'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'CONSTRAINT', N'DF_Vendor_
ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 1
(TRUE)', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'CONSTRAINT', N'DF_Vendor_
PreferredVendorStatus'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
BusinessEntity.BusinessEntityID', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor',
'CONSTRAINT', N'FK_Vendor_BusinessEntity_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'CONSTRAINT', N'PK_Vendor_
BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'INDEX', N'AK_Vendor_AccountNumber'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'INDEX',
N'PK_Vendor_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'INSTEAD OF DELETE trigger which
keeps Vendors from being deleted.', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor',
'TRIGGER', N'dVendor'
GO

```

Uses

[Person].[BusinessEntity]
 [dbo].[AccountNumber]
 [dbo].[Flag]
 [dbo].[Name]
 Purchasing

Used By

[Purchasing].[ProductVendor]
[Purchasing].[PurchaseOrderHeader]
[Purchasing].[vVendorWithAddresses]
[Purchasing].[vVendorWithContacts]
[dbo].[ufnGetContactInformation]

[Sales].[CountryRegionCurrency]

MS_Description

Cross-reference table mapping ISO currency codes to a country or region.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	109
Created	13:14:19 14 marca 2012
Last Modified	13:14:53 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
  	CountryRegionCode <i>ISO code for countries and regions. Foreign key to CountryRegion.CountryRegionCode.</i>	nvarchar(3)	6	False	
  	CurrencyCode <i>ISO standard currency code. Foreign key to Currency.CurrencyCode.</i>	nchar(3)	6	False	
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
 	PK_CountryRegionCurrency_CountryRegionCode_CurrencyCode <i>Primary key (clustered) constraint</i>	CountryRegionCode, CurrencyCode	True
	IX_CountryRegionCurrency_CurrencyCode <i>Nonclustered index.</i>	CurrencyCode	

Foreign Keys

Name	Columns
FK_CountryRegionCurrency_CountryRegion_CountryRegionCode <i>Foreign key constraint referencing CountryRegion.CountryRegionCode.</i>	CountryRegionCode->[Person].[CountryRegion].[CountryRegionCode]
FK_CountryRegionCurrency_Currency_CurrencyCode <i>Foreign key constraint referencing Currency.CurrencyCode.</i>	CurrencyCode->[Sales].[Currency].[CurrencyCode]

SQL Script

```

CREATE TABLE [Sales].[CountryRegionCurrency]
(
    [CountryRegionCode] [nvarchar] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [CurrencyCode] [nchar] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_CountryRegionCurrency_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Sales].[CountryRegionCurrency] ADD CONSTRAINT [PK_CountryRegion-Currency_CountryRegionCode_CurrencyCode] PRIMARY KEY CLUSTERED ([CountryRegionCode], [CurrencyCode]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_CountryRegionCurrency_CurrencyCode] ON [Sales].[CountryRegionCurrency] ([CurrencyCode]) ON [PRIMARY]
GO
ALTER TABLE [Sales].[CountryRegionCurrency] ADD CONSTRAINT [FK_CountryRegion-Currency_CountryRegion_CountryRegionCode] FOREIGN KEY ([CountryRegionCode]) REFERENCES [Person].[CountryRegion] ([CountryRegionCode])
GO
ALTER TABLE [Sales].[CountryRegionCurrency] ADD CONSTRAINT [FK_CountryRegion-Currency_Currency_CurrencyCode] FOREIGN KEY ([CurrencyCode]) REFERENCES [Sales].[Currency] ([CurrencyCode])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Cross-reference table mapping ISO currency codes to a country or region.', 'SCHEMA', N'Sales', 'TABLE', N'Country-RegionCurrency', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'ISO code for countries and regions. Foreign key to CountryRegion.CountryRegionCode.', 'SCHEMA', N'Sales', 'TABLE', N'CountryRegionCurrency', 'COLUMN', N'CountryRegionCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ISO standard currency code. Foreign key to Currency.CurrencyCode.', 'SCHEMA', N'Sales', 'TABLE', N'CountryRegion-Currency', 'COLUMN', N'CurrencyCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last updated.', 'SCHEMA', N'Sales', 'TABLE', N'CountryRegionCurrency', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'CountryRegionCurrency', 'CONSTRAINT', N'DF_CountryRegionCurrency_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing CountryRegion.CountryRegionCode.', 'SCHEMA', N'Sales', 'TABLE', N'CountryRegion-Currency', 'CONSTRAINT', N'FK_CountryRegionCurrency_CountryRegion_CountryRegionCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing Currency.CurrencyCode.', 'SCHEMA', N'Sales', 'TABLE', N'CountryRegionCurrency', 'CONSTRAINT', N'FK_CountryRegionCurrency_Currency_CurrencyCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Sales', 'TABLE', N'CountryRegionCurrency', 'CONSTRAINT', N'PK_CountryRegionCurrency_CountryRegionCode_CurrencyCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA', N'Sales', 'TABLE', N'CountryRegionCurrency', 'INDEX', N'IX_CountryRegionCurrency-CurrencyCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'CountryRegionCurrency', 'INDEX', N'PK_CountryRegionCurrency_CountryRegionCode_CurrencyCode'

```

GO

Uses

[Person].[CountryRegion]

[Sales].[Currency]

Sales

 **[Sales].[CreditCard]****MS_Description**

Customer credit card information.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	19118
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	CreditCardID <i>Primary key for CreditCard records.</i>	int	4	False	1 - 1	
	CardType <i>Credit card name.</i>	nvarchar(50)	100	False		
	CardNumber <i>Credit card number.</i>	nvarchar(25)	50	False		
	ExpMonth <i>Credit card expiration month.</i>	tinyint	1	False		
	ExpYear <i>Credit card expiration year.</i>	smallint	2	False		
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_CreditCard_CreditCardID <i>Primary key (clustered) constraint</i>	CreditCardID	True
	AK_CreditCard_CardNumber <i>Unique nonclustered index.</i>	CardNumber	True

SQL Script

```
CREATE TABLE [Sales].[CreditCard]
(
  [CreditCardID] [int] NOT NULL IDENTITY(1, 1),
  [CardType] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
  [CardNumber] [nvarchar] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
```

```

[ExpMonth] [tinyint] NOT NULL,
[ExpYear] [smallint] NOT NULL,
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_CreditCard_ModifiedDate] DEFAULT
(getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Sales].[CreditCard] ADD CONSTRAINT [PK_CreditCard_CreditCardID] PRIMARY
KEY CLUSTERED ([CreditCardID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_CreditCard_CardNumber] ON [Sales].[CreditCard]
([CardNumber]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Customer credit card information.',
'SHEMA', N'Sales', 'TABLE', N'CreditCard', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Credit card number.', 'SCHEMA',
N'Sales', 'TABLE', N'CreditCard', 'COLUMN', N'CardNumber'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Credit card name.', 'SCHEMA',
N'Sales', 'TABLE', N'CreditCard', 'COLUMN', N'CardType'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for CreditCard
records.', 'SCHEMA', N'Sales', 'TABLE', N'CreditCard', 'COLUMN', N'CreditCardID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Credit card expiration month.',
'SHEMA', N'Sales', 'TABLE', N'CreditCard', 'COLUMN', N'ExpMonth'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Credit card expiration year.',
'SHEMA', N'Sales', 'TABLE', N'CreditCard', 'COLUMN', N'ExpYear'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'CreditCard', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'CreditCard', 'CONSTRAINT', N'DF_Credit-
Card_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Sales', 'TABLE', N'CreditCard', 'CONSTRAINT', N'PK_Credit-
Card_CreditCardID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SHEMA', N'Sales', 'TABLE', N'CreditCard', 'INDEX', N'AK_CreditCard_CardNumber'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'CreditCard', 'INDEX', N'PK_
CreditCard_CreditCardID'
GO

```

Uses

Sales

Used By

[Sales].[PersonCreditCard]
[Sales].[SalesOrderHeader]

 **[Sales].[Currency]****MS_Description**

Lookup table containing standard ISO currencies.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	105
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	CurrencyCode <i>The ISO code for the Currency.</i>	nchar(3)	6	False	
	Name <i>Currency name.</i>	[dbo].[Name]	100	False	
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_Currency_CurrencyCode <i>Primary key (clustered) constraint</i>	CurrencyCode	True
	AK_Currency_Name <i>Unique nonclustered index.</i>	Name	True

SQL Script

```
CREATE TABLE [Sales].[Currency]
(
  [CurrencyCode] [nchar] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
  [Name] [dbo].[Name] NOT NULL,
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_Currency_ModifiedDate] DEFAULT
  (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Sales].[Currency] ADD CONSTRAINT [PK_Currency_CurrencyCode] PRIMARY KEY
CLUSTERED ([CurrencyCode]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_Currency_Name] ON [Sales].[Currency] ([Name])
ON [PRIMARY]
```

```
GO
EXEC sp_addextendedproperty N'MS_Description', N'Lookup table containing standard
ISO currencies.', 'SCHEMA', N'Sales', 'TABLE', N'Currency', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'The ISO code for the Currency.',
'SCHEMA', N'Sales', 'TABLE', N'Currency', 'COLUMN', N'CurrencyCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'Currency', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Currency name.', 'SCHEMA',
N'Sales', 'TABLE', N'Currency', 'COLUMN', N'Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Currency', 'CONSTRAINT', N'DF_Currency_
ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Sales', 'TABLE', N'Currency', 'CONSTRAINT', N'PK_Currency_
CurrencyCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SCHEMA', N'Sales', 'TABLE', N'Currency', 'INDEX', N'AK_Currency_Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'Currency', 'INDEX', N'PK_
Currency_CurrencyCode'
GO
```

Uses

[dbo].[Name]
Sales

Used By

[Sales].[CountryRegionCurrency]
[Sales].[CurrencyRate]

 **[Sales].[CurrencyRate]****MS_Description**

Currency exchange rates.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	13532
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	CurrencyRateID <i>Primary key for CurrencyRate records.</i>	int	4	False	1 - 1	
	CurrencyRateDate <i>Date and time the exchange rate was obtained.</i>	datetime	8	False		
	FromCurrencyCode <i>Exchange rate was converted from this currency code.</i>	nchar(3)	6	False		
	ToCurrencyCode <i>Exchange rate was converted to this currency code.</i>	nchar(3)	6	False		
	AverageRate <i>Average exchange rate for the day.</i>	money	8	False		
	EndOfDayRate <i>Final exchange rate for the day.</i>	money	8	False		
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_CurrencyRate_CurrencyRateID <i>Primary key (clustered) constraint</i>	CurrencyRate-ID	True
	AK_CurrencyRate_CurrencyRateDate_FromCurrencyCode_ToCurrencyCode <i>Unique nonclustered index.</i>	CurrencyRate-Date, From-Currency-Code, To-CurrencyCode	True

Foreign Keys

Name	Columns
FK_CurrencyRate_Currency_FromCurrencyCode <i>Foreign key constraint referencing Currency.From-CurrencyCode.</i>	FromCurrencyCode->[Sales].[Currency].[Currency-Code]
FK_CurrencyRate_Currency_ToCurrencyCode <i>Foreign key constraint referencing Currency.To-CurrencyCode.</i>	ToCurrencyCode->[Sales].[Currency].[CurrencyCode]

SQL Script

```

CREATE TABLE [Sales].[CurrencyRate]
(
  [CurrencyRateID] [int] NOT NULL IDENTITY(1, 1),
  [CurrencyRateDate] [datetime] NOT NULL,
  [FromCurrencyCode] [nchar] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
  [ToCurrencyCode] [nchar] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
  [AverageRate] [money] NOT NULL,
  [EndOfDayRate] [money] NOT NULL,
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_CurrencyRate_ModifiedDate] DEFAULT
(getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Sales].[CurrencyRate] ADD CONSTRAINT [PK_CurrencyRate_CurrencyRateID]
PRIMARY KEY CLUSTERED ([CurrencyRateID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_CurrencyRate_CurrencyRateDate_FromCurrencyCode_ToCurrencyCode] ON [Sales].[CurrencyRate] ([CurrencyRateDate], [FromCurrencyCode],
[ToCurrencyCode]) ON [PRIMARY]
GO
ALTER TABLE [Sales].[CurrencyRate] ADD CONSTRAINT [FK_CurrencyRate_Currency_FromCurrencyCode] FOREIGN KEY ([FromCurrencyCode]) REFERENCES [Sales].[Currency]
([CurrencyCode])
GO
ALTER TABLE [Sales].[CurrencyRate] ADD CONSTRAINT [FK_CurrencyRate_Currency_ToCurrencyCode] FOREIGN KEY ([ToCurrencyCode]) REFERENCES [Sales].[Currency]
([CurrencyCode])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Currency exchange rates.',
'SHEMA', N'Sales', 'TABLE', N'CurrencyRate', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Average exchange rate for the
day.', 'SCHEMA', N'Sales', 'TABLE', N'CurrencyRate', 'COLUMN', N'AverageRate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the exchange rate was
obtained.', 'SCHEMA', N'Sales', 'TABLE', N'CurrencyRate', 'COLUMN', N'CurrencyRate-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for CurrencyRate
records.', 'SCHEMA', N'Sales', 'TABLE', N'CurrencyRate', 'COLUMN', N'CurrencyRateID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Final exchange rate for the day.',
'SHEMA', N'Sales', 'TABLE', N'CurrencyRate', 'COLUMN', N'EndOfDayRate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Exchange rate was converted from
this currency code.', 'SCHEMA', N'Sales', 'TABLE', N'CurrencyRate', 'COLUMN', N'From-
CurrencyCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last

```

```
updated.', 'SCHEMA', N'Sales', 'TABLE', N'CurrencyRate', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Exchange rate was converted to this
currency code.', 'SCHEMA', N'Sales', 'TABLE', N'CurrencyRate', 'COLUMN', N'To-
CurrencyCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'CurrencyRate', 'CONSTRAINT', N'DF_-
CurrencyRate_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Currency.FromCurrencyCode.', 'SCHEMA', N'Sales', 'TABLE', N'CurrencyRate',
'CONSTRAINT', N'FK_CurrencyRate_Currency_FromCurrencyCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Currency.ToCurrencyCode.', 'SCHEMA', N'Sales', 'TABLE', N'CurrencyRate',
'CONSTRAINT', N'FK_CurrencyRate_Currency_ToCurrencyCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Sales', 'TABLE', N'CurrencyRate', 'CONSTRAINT', N'PK_-
CurrencyRate_CurrencyRateID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SCHEMA', N'Sales', 'TABLE', N'CurrencyRate', 'INDEX', N'AK_CurrencyRate_Currency-
RateDate_FromCurrencyCode_ToCurrencyCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'CurrencyRate', 'INDEX',
N'PK_CurrencyRate_CurrencyRateID'
GO
```

Uses

[Sales].[Currency]
Sales

Used By

[Sales].[SalesOrderHeader]

 **[Sales].[Customer]**
MS_Description

Current customer information. Also see the Person and Store tables.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	19820
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Computed	Max Length (Bytes)	Allow Nulls	Identity	Identity Replication	Default
	CustomerID <i>Primary key.</i>	int		4	False	1 - 1	False	
	PersonID <i>Foreign key to Person.BusinessEntityID</i>	int		4	True			
	StoreID <i>Foreign key to Store.BusinessEntityID</i>	int		4	True			
	TerritoryID <i>ID of the territory in which the customer is located. Foreign key to SalesTerritory.SalesTerritoryID.</i>	int		4	True			
	AccountNumber <i>Unique number identifying the customer assigned by the accounting system.</i>	varchar(10)	True	10	False			
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier		16	False			(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime		8	False			(getdate())

Computed columns

Name	Column definition
AccountNumber	(isnull('AW'+[dbo].[ufnLeadingZeros]([CustomerID]),''))

Indexes

Key	Name	Key Columns	Unique
	PK_Customer_CustomerID <i>Primary key (clustered) constraint</i>	CustomerID	True
	AK_Customer_AccountNumber <i>Unique nonclustered index.</i>	Account- Number	True
	AK_Customer_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True
	IX_Customer_TerritoryID <i>Nonclustered index.</i>	TerritoryID	

Foreign Keys

Name	Columns
FK_Customer_Person_PersonID <i>Foreign key constraint referencing Person.Business-EntityID.</i>	PersonID->[Person].[Person].[BusinessEntityID]
FK_Customer_SalesTerritory_TerritoryID <i>Foreign key constraint referencing Sales-Territory.TerritoryID.</i>	TerritoryID->[Sales].[SalesTerritory].[TerritoryID]
FK_Customer_Store_StoreID <i>Foreign key constraint referencing Store.BusinessEntity-ID.</i>	StoreID->[Sales].[Store].[BusinessEntityID]

SQL Script

```
CREATE TABLE [Sales].[Customer]
(
  [CustomerID] [int] NOT NULL IDENTITY(1, 1) NOT FOR REPLICATION,
  [PersonID] [int] NULL,
  [StoreID] [int] NULL,
  [TerritoryID] [int] NULL,
  [AccountNumber] AS (isnull('AW'+[dbo].[ufnLeadingZeros]([CustomerID]),'')),
  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_Customer_rowguid]
  DEFAULT (newid()),
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_Customer_ModifiedDate] DEFAULT
  (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Sales].[Customer] ADD CONSTRAINT [PK_Customer_CustomerID] PRIMARY KEY
  CLUSTERED ([CustomerID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_Customer_AccountNumber] ON [Sales].[Customer]
  ([AccountNumber]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_Customer_rowguid] ON [Sales].[Customer]
  ([rowguid]) ON [PRIMARY]
GO
```

```

CREATE NONCLUSTERED INDEX [IX_Customer_TerritoryID] ON [Sales].[Customer]
([TerritoryID]) ON [PRIMARY]
GO
ALTER TABLE [Sales].[Customer] ADD CONSTRAINT [FK_Customer_Person_PersonID] FOREIGN
KEY ([PersonID]) REFERENCES [Person].[Person] ([BusinessEntityID])
GO
ALTER TABLE [Sales].[Customer] ADD CONSTRAINT [FK_Customer_SalesTerritory_Territory-
ID] FOREIGN KEY ([TerritoryID]) REFERENCES [Sales].[SalesTerritory] ([TerritoryID])
GO
ALTER TABLE [Sales].[Customer] ADD CONSTRAINT [FK_Customer_Store_StoreID] FOREIGN
KEY ([StoreID]) REFERENCES [Sales].[Store] ([BusinessEntityID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Current customer information. Also
see the Person and Store tables.', 'SCHEMA', N'Sales', 'TABLE', N'Customer', NULL,
NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique number identifying the
customer assigned by the accounting system.', 'SCHEMA', N'Sales', 'TABLE',
N'Customer', 'COLUMN', N'AccountNumber'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key.', 'SCHEMA', N'Sales',
'TABLE', N'Customer', 'COLUMN', N'CustomerID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key to Person.Business-
EntityID', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'COLUMN', N'PersonID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Sales', 'TABLE', N'Customer', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key to Store.BusinessEntity-
ID', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'COLUMN', N'StoreID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ID of the territory in which the
customer is located. Foreign key to SalesTerritory.SalesTerritoryID.', 'SCHEMA',
N'Sales', 'TABLE', N'Customer', 'COLUMN', N'TerritoryID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'CONSTRAINT', N'DF_Customer_-
ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'CONSTRAINT', N'DF_-
Customer_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Person.BusinessEntityID.', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'CONSTRAINT',
N'FK_Customer_Person_PersonID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
SalesTerritory.TerritoryID.', 'SCHEMA', N'Sales', 'TABLE', N'Customer',
'CONSTRAINT', N'FK_Customer_SalesTerritory_TerritoryID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Store.BusinessEntityID.', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'CONSTRAINT',
N'FK_Customer_Store_StoreID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'CONSTRAINT', N'PK_Customer_-
CustomerID'
GO

```

```
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',  
'SCHEMA', N'Sales', 'TABLE', N'Customer', 'INDEX', N'AK_Customer_AccountNumber'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to  
support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'INDEX',  
N'AK_Customer_rowguid'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',  
N'Sales', 'TABLE', N'Customer', 'INDEX', N'IX_Customer_TerritoryID'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a  
primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'INDEX', N'PK_  
Customer_CustomerID'  
GO
```

Uses

[Person].[Person]
[Sales].[SalesTerritory]
[Sales].[Store]
[dbo].[ufnLeadingZeros]
Sales

Used By

[Sales].[SalesOrderHeader]
[Sales].[vIndividualCustomer]
[dbo].[ufnGetContactInformation]

[Sales].[PersonCreditCard]**MS_Description**

Cross-reference table mapping people to their credit card information in the CreditCard table.

Properties

Property	Value
Row Count (~)	19118
Created	13:14:19 14 marca 2012
Last Modified	13:14:54 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	BusinessEntityID <i>Business entity identification number. Foreign key to Person.BusinessEntityID.</i>	int	4	False	
	CreditCardID <i>Credit card identification number. Foreign key to CreditCard.CreditCardID.</i>	int	4	False	
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_PersonCreditCard_BusinessEntityID_CreditCardID <i>Primary key (clustered) constraint</i>	BusinessEntityID, CreditCardID	True

Foreign Keys

Name	Columns
FK_PersonCreditCard_CreditCard_CreditCardID <i>Foreign key constraint referencing CreditCard.CreditCardID.</i>	CreditCardID->[Sales].[CreditCard].[CreditCardID]
FK_PersonCreditCard_Person_BusinessEntityID <i>Foreign key constraint referencing Person.BusinessEntityID.</i>	BusinessEntityID->[Person].[Person].[BusinessEntityID]

SQL Script

```
CREATE TABLE [Sales].[PersonCreditCard]
(
  [BusinessEntityID] [int] NOT NULL,
```

```

[CreditCardID] [int] NOT NULL,
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_PersonCreditCard_ModifiedDate]
DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Sales].[PersonCreditCard] ADD CONSTRAINT [PK_PersonCreditCard_Business-
EntityID_CreditCardID] PRIMARY KEY CLUSTERED ([BusinessEntityID], [CreditCardID])
ON [PRIMARY]
GO
ALTER TABLE [Sales].[PersonCreditCard] ADD CONSTRAINT [FK_PersonCreditCard_Credit-
Card_CreditCardID] FOREIGN KEY ([CreditCardID]) REFERENCES [Sales].[CreditCard]
([CreditCardID])
GO
ALTER TABLE [Sales].[PersonCreditCard] ADD CONSTRAINT [FK_PersonCreditCard_Person_
BusinessEntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [Person].[Person]
([BusinessEntityID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Cross-reference table mapping
people to their credit card information in the CreditCard table. ', 'SCHEMA',
N'Sales', 'TABLE', N'PersonCreditCard', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Business entity identification
number. Foreign key to Person.BusinessEntityID.', 'SCHEMA', N'Sales', 'TABLE',
N'PersonCreditCard', 'COLUMN', N'BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Credit card identification number.
Foreign key to CreditCard.CreditCardID.', 'SCHEMA', N'Sales', 'TABLE', N'Person-
CreditCard', 'COLUMN', N'CreditCardID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'PersonCreditCard', 'COLUMN', N'Modified-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'PersonCreditCard', 'CONSTRAINT', N'DF_
PersonCreditCard_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
CreditCard.CreditCardID.', 'SCHEMA', N'Sales', 'TABLE', N'PersonCreditCard',
'CONSTRAINT', N'FK_PersonCreditCard_CreditCard_CreditCardID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Person.BusinessEntityID.', 'SCHEMA', N'Sales', 'TABLE', N'PersonCreditCard',
'CONSTRAINT', N'FK_PersonCreditCard_Person_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Sales', 'TABLE', N'PersonCreditCard', 'CONSTRAINT', N'PK_
PersonCreditCard_BusinessEntityID_CreditCardID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'PersonCreditCard', 'INDEX',
N'PK_PersonCreditCard_BusinessEntityID_CreditCardID'
GO

```

Uses

[Person].[Person]
[Sales].[CreditCard]
Sales

 **[Sales].[SalesOrderDetail]**
MS_Description

Individual products associated with a specific sales order. See SalesOrderHeader.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	121317
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Computed	Max Length (Bytes)	Allow Nulls	Identity	Default
	SalesOrderID <i>Primary key. Foreign key to SalesOrderHeader.SalesOrderID.</i>	int		4	False		
	SalesOrderDetailID <i>Primary key. One incremental unique number per product sold.</i>	int		4	False	1 - 1	
	CarrierTrackingNumber <i>Shipment tracking number supplied by the shipper.</i>	nvarchar(25)		50	True		
	OrderQty <i>Quantity ordered per product.</i>	smallint		2	False		
	ProductID <i>Product sold to customer. Foreign key to Product.ProductID.</i>	int		4	False		
	SpecialOfferID <i>Promotional code. Foreign key to SpecialOffer.SpecialOfferID.</i>	int		4	False		
	UnitPrice <i>Selling price of a single product.</i>	money		8	False		
	UnitPriceDiscount <i>Discount amount.</i>	money		8	False		((0.0))
	LineTotal <i>Per product subtotal. Computed as UnitPrice * (1 - UnitPriceDiscount) * OrderQty.</i>	numeric(38,6)	True	17	False		
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a</i>	uniqueidentifier		16	False		(newid())

	<i>merge replication sample.</i>					
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime		8	False	(getdate())

Computed columns

Name	Column definition
LineTotal	(isnull(([UnitPrice]*((1.0)-[UnitPriceDiscount]))*[OrderQty],(0.0)))

Indexes

Key	Name	Key Columns	Unique
	PK_SalesOrderDetail_SalesOrderID_SalesOrderDetailID <i>Primary key (clustered) constraint</i>	SalesOrderID, SalesOrderDetailID	True
	AK_SalesOrderDetail_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True
	IX_SalesOrderDetail_ProductID <i>Nonclustered index.</i>	ProductID	

Triggers

Name	ANSI Nulls On	Quoted Identifier On	On
iduSalesOrderDetail <i>AFTER INSERT, DELETE, UPDATE trigger that inserts a row in the TransactionHistory table, updates ModifiedDate in SalesOrderDetail and updates the SalesOrderHeader.SubTotal column.</i>	True	True	After Delete Insert Update

Check Constraints

Name	On Column	Constraint
CK_SalesOrderDetail_OrderQty <i>Check constraint [OrderQty] > (0)</i>	OrderQty	(([OrderQty]>(0))
CK_SalesOrderDetail_UnitPrice <i>Check constraint [UnitPrice] >= (0.00)</i>	UnitPrice	(([UnitPrice]>=(0.00))
CK_SalesOrderDetail_UnitPriceDiscount <i>Check constraint [UnitPriceDiscount] >= (0.00)</i>	UnitPriceDiscount	(([UnitPriceDiscount]>=(0.00))

Foreign Keys

Name	Delete	Columns
FK_SalesOrderDetail_SalesOrderHeader_SalesOrderID <i>Foreign key constraint referencing SalesOrderHeader.PurchaseOrderID.</i>	Cascade	SalesOrderID->[Sales].[SalesOrderHeader].[SalesOrderID]
FK_SalesOrderDetail_SpecialOfferProduct_SpecialOfferIDProductID <i>Foreign key constraint referencing SpecialOfferProduct.SpecialOfferIDProductID.</i>		SpecialOfferID->[Sales].[SpecialOfferProduct].[SpecialOfferID]

ProductID->[Sales].[SpecialOfferProduct].[ProductID]
--

SQL Script

```

CREATE TABLE [Sales].[SalesOrderDetail]
(
    [SalesOrderID] [int] NOT NULL,
    [SalesOrderDetailID] [int] NOT NULL IDENTITY(1, 1),
    [CarrierTrackingNumber] [nvarchar] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [OrderQty] [smallint] NOT NULL,
    [ProductID] [int] NOT NULL,
    [SpecialOfferID] [int] NOT NULL,
    [UnitPrice] [money] NOT NULL,
    [UnitPriceDiscount] [money] NOT NULL CONSTRAINT [DF_SalesOrderDetail_UnitPriceDiscount] DEFAULT ((0.0)),
    [LineTotal] AS (isnull(([UnitPrice]*((1.0)-[UnitPriceDiscount]))*[OrderQty],(0.0))),
    [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_SalesOrderDetail_rowguid] DEFAULT (newid()),
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_SalesOrderDetail_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
GO

CREATE TRIGGER [Sales].[iduSalesOrderDetail] ON [Sales].[SalesOrderDetail]
AFTER INSERT, DELETE, UPDATE AS
BEGIN
    DECLARE @Count int;

    SET @Count = @@ROWCOUNT;
    IF @Count = 0
        RETURN;

    SET NOCOUNT ON;

    BEGIN TRY
        -- If inserting or updating these columns
        IF UPDATE([ProductID]) OR UPDATE([OrderQty]) OR UPDATE([UnitPrice]) OR
        UPDATE([UnitPriceDiscount])
        -- Insert record into TransactionHistory
        BEGIN
            INSERT INTO [Production].[TransactionHistory]
                ([ProductID]
                , [ReferenceOrderID]
                , [ReferenceOrderLineID]
                , [TransactionType]
                , [TransactionDate]
                , [Quantity]
                , [ActualCost])
            SELECT
                inserted.[ProductID]
                , inserted.[SalesOrderID]
                , inserted.[SalesOrderDetailID]
                , 'S'
                , GETDATE()
                , inserted.[OrderQty]
        
```

```

        ,inserted.[UnitPrice]
    FROM inserted
    INNER JOIN [Sales].[SalesOrderHeader]
    ON inserted.[SalesOrderID] = [Sales].[SalesOrderHeader].[SalesOrder-
ID];

    UPDATE [Person].[Person]
    SET [Demographics].modify('declare default element namespace
        "http://schemas.microsoft.com/sqlserver/2004/07/adventure-
works/IndividualSurvey";
        replace value of (/IndividualSurvey/TotalPurchaseYTD) [1]
        with data(/IndividualSurvey/TotalPurchaseYTD) [1] + sql:column
("inserted.LineTotal")')
    FROM inserted
    INNER JOIN [Sales].[SalesOrderHeader] AS SOH
    ON inserted.[SalesOrderID] = SOH.[SalesOrderID]
    INNER JOIN [Sales].[Customer] AS C
    ON SOH.[CustomerID] = C.[CustomerID]
    WHERE C.[PersonID] = [Person].[Person].[BusinessEntityID];
END;

-- Update SubTotal in SalesOrderHeader record. Note that this causes the
-- SalesOrderHeader trigger to fire which will update the RevisionNumber.
UPDATE [Sales].[SalesOrderHeader]
SET [Sales].[SalesOrderHeader].[SubTotal] =
    (SELECT SUM([Sales].[SalesOrderDetail].[LineTotal])
    FROM [Sales].[SalesOrderDetail]
    WHERE [Sales].[SalesOrderHeader].[SalesOrderID] = [Sales].[Sales-
OrderDetail].[SalesOrderID])
    WHERE [Sales].[SalesOrderHeader].[SalesOrderID] IN (SELECT inserted.[Sales-
OrderID] FROM inserted);

    UPDATE [Person].[Person]
    SET [Demographics].modify('declare default element namespace
        "http://schemas.microsoft.com/sqlserver/2004/07/adventure-
works/IndividualSurvey";
        replace value of (/IndividualSurvey/TotalPurchaseYTD) [1]
        with data(/IndividualSurvey/TotalPurchaseYTD) [1] -
sql:column("deleted.LineTotal")')
    FROM deleted
    INNER JOIN [Sales].[SalesOrderHeader]
    ON deleted.[SalesOrderID] = [Sales].[SalesOrderHeader].[SalesOrderID]
    INNER JOIN [Sales].[Customer]
    ON [Sales].[Customer].[CustomerID] = [Sales].[SalesOrder-
Header].[CustomerID]
    WHERE [Sales].[Customer].[PersonID] = [Person].[Person].[BusinessEntityID];
END TRY
BEGIN CATCH
    EXECUTE [dbo].[uspPrintError];

-- Rollback any active or uncommittable transactions before
-- inserting information in the ErrorLog
IF @@TRANCOUNT > 0
BEGIN
    ROLLBACK TRANSACTION;
END

EXECUTE [dbo].[uspLogError];
END CATCH;

```

```

END;
GO
ALTER TABLE [Sales].[SalesOrderDetail] ADD CONSTRAINT [CK_SalesOrderDetail_OrderQty]
CHECK (([OrderQty]>(0)))
GO
ALTER TABLE [Sales].[SalesOrderDetail] ADD CONSTRAINT [CK_SalesOrderDetail_Unit-
Price] CHECK (([UnitPrice]>=(0.00)))
GO
ALTER TABLE [Sales].[SalesOrderDetail] ADD CONSTRAINT [CK_SalesOrderDetail_UnitPrice-
Discount] CHECK (([UnitPriceDiscount]>=(0.00)))
GO
ALTER TABLE [Sales].[SalesOrderDetail] ADD CONSTRAINT [PK_SalesOrderDetail_Sales-
OrderID_SalesOrderDetailID] PRIMARY KEY CLUSTERED ((SalesOrderID), [SalesOrder-
DetailID]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_SalesOrderDetail_ProductID] ON [Sales].[SalesOrder-
Detail] ([ProductID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_SalesOrderDetail_rowguid] ON [Sales].[Sales-
OrderDetail] ([rowguid]) ON [PRIMARY]
GO
ALTER TABLE [Sales].[SalesOrderDetail] ADD CONSTRAINT [FK_SalesOrderDetail_Sales-
OrderHeader_SalesOrderID] FOREIGN KEY ([SalesOrderID]) REFERENCES [Sales].[Sales-
OrderHeader] ([SalesOrderID]) ON DELETE CASCADE
GO
ALTER TABLE [Sales].[SalesOrderDetail] ADD CONSTRAINT [FK_SalesOrderDetail_Special-
OfferProduct_SpecialOfferIDProductID] FOREIGN KEY ([SpecialOfferID], [ProductID])
REFERENCES [Sales].[SpecialOfferProduct] ([SpecialOfferID], [ProductID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Individual products associated with
a specific sales order. See SalesOrderHeader.', 'SCHEMA', N'Sales', 'TABLE', N'Sales-
OrderDetail', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Shipment tracking number supplied
by the shipper.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail', 'COLUMN',
N'CarrierTrackingNumber'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Per product subtotal. Computed as
UnitPrice * (1 - UnitPriceDiscount) * OrderQty.', 'SCHEMA', N'Sales', 'TABLE',
N'SalesOrderDetail', 'COLUMN', N'LineTotal'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail', 'COLUMN', N'Modified-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Quantity ordered per product.',
'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail', 'COLUMN', N'OrderQty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product sold to customer. Foreign
key to Product.ProductID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail',
'COLUMN', N'ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesOrderDetail', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. One incremental unique
number per product sold.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail',
'COLUMN', N'SalesOrderDetailID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Foreign key to Sales-
OrderHeader.SalesOrderID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail',
'COLUMN', N'SalesOrderID'
GO

```

```

EXEC sp_addextendedproperty N'MS_Description', N'Promotional code. Foreign key to
SpecialOffer.SpecialOfferID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail',
'COLUMN', N'SpecialOfferID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Selling price of a single
product.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail', 'COLUMN', N'UnitPrice'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Discount amount.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesOrderDetail', 'COLUMN', N'UnitPriceDiscount'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [OrderQty] > (0)',
'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail', 'CONSTRAINT', N'CK_SalesOrder-
Detail_OrderQty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [UnitPrice] >=
(0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail', 'CONSTRAINT', N'CK_Sales-
OrderDetail_UnitPrice'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [UnitPrice-
Discount] >= (0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail',
'CONSTRAINT', N'CK_SalesOrderDetail_UnitPriceDiscount'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail', 'CONSTRAINT', N'DF_-
SalesOrderDetail_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail', 'CONSTRAINT', N'DF_Sales-
OrderDetail_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail', 'CONSTRAINT', N'DF_SalesOrder-
Detail_UnitPriceDiscount'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
SalesOrderHeader.PurchaseOrderID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrder-
Detail', 'CONSTRAINT', N'FK_SalesOrderDetail_SalesOrderHeader_SalesOrderID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
SpecialOfferProduct.SpecialOfferIDProductID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales-
OrderDetail', 'CONSTRAINT', N'FK_SalesOrderDetail_SpecialOfferProduct_SpecialOffer-
IDProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail', 'CONSTRAINT', N'PK_-
SalesOrderDetail_SalesOrderID_SalesOrderDetailID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail',
'INDEX', N'AK_SalesOrderDetail_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesOrderDetail', 'INDEX', N'IX_SalesOrderDetail_ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail', 'INDEX',
N'PK_SalesOrderDetail_SalesOrderID_SalesOrderDetailID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'AFTER INSERT, DELETE, UPDATE
trigger that inserts a row in the TransactionHistory table, updates ModifiedDate in
SalesOrderDetail and updates the SalesOrderHeader.SubTotal column.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesOrderDetail', 'TRIGGER', N'iduSalesOrderDetail'
GO

```

Uses

[Sales].[SalesOrderHeader]
[Sales].[SpecialOfferProduct]
Sales

 **[Sales].[SalesOrderHeader]**

MS_Description

General sales order information.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	31465
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Computed	Max Length (Bytes)	Allow Nulls	Identity	Identity Replication	Default
	SalesOrderID <i>Primary key.</i>	int		4	False	1 - 1	False	
	RevisionNumber <i>Incremental number to track changes to the sales order over time.</i>	tinyint		1	False			((0))
	OrderDate <i>Dates the sales order was created.</i>	datetime		8	False			(getdate())
	DueDate <i>Date the order is due to the customer.</i>	datetime		8	False			
	ShipDate <i>Date the order was shipped to the customer.</i>	datetime		8	True			
	Status <i>Order current status. 1 = In process; 2 = Approved; 3 = Backordered; 4 = Rejected; 5 = Shipped; 6 = Cancelled</i>	tinyint		1	False			((1))
	OnlineOrderFlag <i>0 = Order placed by sales person. 1 = Order placed online by customer.</i>	[dbo].[Flag]		1	False			((1))
	SalesOrderNumber <i>Unique sales order identification number.</i>	nvarchar(25)	True	50	False			
	PurchaseOrderNumber <i>Customer purchase order number</i>	[dbo].[Order-Number]		50	True			

	<i>reference.</i>						
	AccountNumber <i>Financial accounting number reference.</i>	[dbo].[Account-Number]		30	True		
	CustomerID <i>Customer identification number. Foreign key to Customer.BusinessEntityID.</i>	int		4	False		
	SalesPersonID <i>Sales person who created the sales order. Foreign key to SalesPerson.BusinessEntityID.</i>	int		4	True		
	TerritoryID <i>Territory in which the sale was made. Foreign key to SalesTerritory.SalesTerritoryID.</i>	int		4	True		
	BillToAddressID <i>Customer billing address. Foreign key to Address.AddressID.</i>	int		4	False		
	ShipToAddressID <i>Customer shipping address. Foreign key to Address.AddressID.</i>	int		4	False		
	ShipMethodID <i>Shipping method. Foreign key to ShipMethod.ShipMethodID.</i>	int		4	False		
	CreditCardID <i>Credit card identification number. Foreign key to CreditCard.CreditCardID.</i>	int		4	True		
	CreditCardApprovalCode <i>Approval code provided by the credit card company.</i>	varchar(15)		15	True		
	CurrencyRateID <i>Currency exchange rate used. Foreign key to CurrencyRate.CurrencyRateID.</i>	int		4	True		
	SubTotal <i>Sales subtotal. Computed as SUM(SalesOrderDetail.LineTotal)for the appropriate SalesOrderID.</i>	money		8	False		((0.00))
	TaxAmt <i>Tax amount.</i>	money		8	False		((0.00))
	Freight <i>Shipping cost.</i>	money		8	False		((0.00))
	TotalDue <i>Total due from customer. Computed as Subtotal + TaxAmt + Freight.</i>	money	True	8	False		

	Comment <i>Sales representative comments.</i>	nvarchar(128)		256	True		
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier		16	False		(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime		8	False		(getdate())

Computed columns

Name	Column definition
SalesOrderNumber	(isnull(N'SO'+CONVERT([nvarchar](23),[SalesOrderID]),(0)),N'*** ERROR ***)
TotalDue	(isnull(([SubTotal]+[TaxAmt])+[Freight]),(0))

Indexes

Key	Name	Key Columns	Unique
	PK_SalesOrderHeader_SalesOrderID <i>Primary key (clustered) constraint</i>	SalesOrderID	True
	AK_SalesOrderHeader_SalesOrderNumber <i>Unique nonclustered index.</i>	SalesOrderNumber	True
	AK_SalesOrderHeader_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True
	IX_SalesOrderHeader_CustomerID <i>Nonclustered index.</i>	CustomerID	
	IX_SalesOrderHeader_SalesPersonID <i>Nonclustered index.</i>	SalesPersonID	

Triggers

Name	ANSI Nulls On	Quoted Identifier On	On	Not For Replication
uSalesOrderHeader <i>AFTER UPDATE trigger that updates the RevisionNumber and ModifiedDate columns in the SalesOrderHeader table. Updates the SalesYTD column in the SalesPerson and SalesTerritory tables.</i>	True	True	After Update	True

Check Constraints

Name	On Column	Constraint
CK_SalesOrderHeader_DueDate <i>Check constraint [DueDate] >= [OrderDate]</i>		([DueDate]>=[OrderDate])
CK_SalesOrderHeader_Freight <i>Check constraint [Freight] >= (0.00)</i>	Freight	([Freight]>=(0.00))

CK_SalesOrderHeader_ShipDate <i>Check constraint [ShipDate] >= [OrderDate] OR [ShipDate] IS NULL</i>		(([ShipDate]>=[OrderDate]) OR [ShipDate] IS NULL)
CK_SalesOrderHeader_Status <i>Check constraint [Status] BETWEEN (0) AND (8)</i>	Status	(([Status]>=(0) AND [Status]<=(8))
CK_SalesOrderHeader_SubTotal <i>Check constraint [SubTotal] >= (0.00)</i>	SubTotal	(([SubTotal]>=(0.00))
CK_SalesOrderHeader_TaxAmt <i>Check constraint [TaxAmt] >= (0.00)</i>	TaxAmt	(([TaxAmt]>=(0.00))

Foreign Keys

Name	Columns
FK_SalesOrderHeader_Address_BillToAddressID <i>Foreign key constraint referencing Address.AddressID.</i>	BillToAddressID->[Person].[Address].[AddressID]
FK_SalesOrderHeader_Address_ShipToAddressID <i>Foreign key constraint referencing Address.AddressID.</i>	ShipToAddressID->[Person].[Address].[AddressID]
FK_SalesOrderHeader_CreditCard_CreditCardID <i>Foreign key constraint referencing CreditCard.CreditCardID.</i>	CreditCardID->[Sales].[CreditCard].[CreditCardID]
FK_SalesOrderHeader_CurrencyRate_CurrencyRateID <i>Foreign key constraint referencing CurrencyRate.CurrencyRateID.</i>	CurrencyRateID->[Sales].[CurrencyRate].[CurrencyRateID]
FK_SalesOrderHeader_Customer_CustomerID <i>Foreign key constraint referencing Customer.CustomerID.</i>	CustomerID->[Sales].[Customer].[CustomerID]
FK_SalesOrderHeader_SalesPerson_SalesPersonID <i>Foreign key constraint referencing SalesPerson.SalesPersonID.</i>	SalesPersonID->[Sales].[SalesPerson].[BusinessEntityID]
FK_SalesOrderHeader_SalesTerritory_TerritoryID <i>Foreign key constraint referencing SalesTerritory.TerritoryID.</i>	TerritoryID->[Sales].[SalesTerritory].[TerritoryID]
FK_SalesOrderHeader_ShipMethod_ShipMethodID <i>Foreign key constraint referencing ShipMethod.ShipMethodID.</i>	ShipMethodID->[Purchasing].[ShipMethod].[ShipMethodID]

SQL Script

```
CREATE TABLE [Sales].[SalesOrderHeader]
(
  [SalesOrderID] [int] NOT NULL IDENTITY(1, 1) NOT FOR REPLICATION,
  [RevisionNumber] [tinyint] NOT NULL CONSTRAINT [DF_SalesOrderHeader_RevisionNumber]
  DEFAULT ((0)),
  [OrderDate] [datetime] NOT NULL CONSTRAINT [DF_SalesOrderHeader_OrderDate] DEFAULT
  (getdate()),
  [DueDate] [datetime] NOT NULL,
  [ShipDate] [datetime] NULL,
  [Status] [tinyint] NOT NULL CONSTRAINT [DF_SalesOrderHeader_Status] DEFAULT ((1)),
  [OnlineOrderFlag] [dbo].[Flag] NOT NULL CONSTRAINT [DF_SalesOrderHeader_OnlineOrder-
  Flag] DEFAULT ((1)),
  [SalesOrderNumber] AS (isnull(N'SO'+CONVERT([nvarchar](23),[SalesOrderID],(0)),N'***
  ERROR ***')),
  [PurchaseOrderNumber] [dbo].[OrderNumber] NULL,
  [AccountNumber] [dbo].[AccountNumber] NULL,
  [CustomerID] [int] NOT NULL,
```

```

[SalesPersonID] [int] NULL,
[TerritoryID] [int] NULL,
[BillToAddressID] [int] NOT NULL,
[ShipToAddressID] [int] NOT NULL,
[ShipMethodID] [int] NOT NULL,
[CreditCardID] [int] NULL,
[CreditCardApprovalCode] [varchar] (15) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[CurrencyRateID] [int] NULL,
[SubTotal] [money] NOT NULL CONSTRAINT [DF_SalesOrderHeader_SubTotal] DEFAULT
((0.00)),
[TaxAmt] [money] NOT NULL CONSTRAINT [DF_SalesOrderHeader_TaxAmt] DEFAULT ((0.00)),
[Freight] [money] NOT NULL CONSTRAINT [DF_SalesOrderHeader_Freight] DEFAULT
((0.00)),
[TotalDue] AS (isnull(([SubTotal]+[TaxAmt])+[Freight],(0))),
[Comment] [nvarchar] (128) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_SalesOrder-
Header_rowguid] DEFAULT (newid()),
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_SalesOrderHeader_ModifiedDate]
DEFAULT (getdate())
) ON [PRIMARY]
GO

CREATE TRIGGER [Sales].[uSalesOrderHeader] ON [Sales].[SalesOrderHeader]
AFTER UPDATE NOT FOR REPLICATION AS
BEGIN
    DECLARE @Count int;

    SET @Count = @@ROWCOUNT;
    IF @Count = 0
        RETURN;

    SET NOCOUNT ON;

    BEGIN TRY
        -- Update RevisionNumber for modification of any field EXCEPT the Status.
        IF NOT UPDATE([Status])
            BEGIN
                UPDATE [Sales].[SalesOrderHeader]
                SET [Sales].[SalesOrderHeader].[RevisionNumber] =
                    [Sales].[SalesOrderHeader].[RevisionNumber] + 1
                WHERE [Sales].[SalesOrderHeader].[SalesOrderID] IN
                    (SELECT inserted.[SalesOrderID] FROM inserted);
            END;

        -- Update the SalesPerson SalesYTD when SubTotal is updated
        IF UPDATE([SubTotal])
            BEGIN
                DECLARE @StartDate datetime,
                        @EndDate datetime

                SET @StartDate = [dbo].[ufnGetAccountingStartDate]();
                SET @EndDate = [dbo].[ufnGetAccountingEndDate]();

                UPDATE [Sales].[SalesPerson]
                SET [Sales].[SalesPerson].[SalesYTD] =
                    (SELECT SUM([Sales].[SalesOrderHeader].[SubTotal])
                     FROM [Sales].[SalesOrderHeader]
                     WHERE [Sales].[SalesPerson].[BusinessEntityID] = [Sales].[SalesOrder-

```

```

Header].[SalesPersonID]
        AND ([Sales].[SalesOrderHeader].[Status] = 5) -- Shipped
        AND [Sales].[SalesOrderHeader].[OrderDate] BETWEEN @StartDate
AND @EndDate)
    WHERE [Sales].[SalesPerson].[BusinessEntityID]
        IN (SELECT DISTINCT inserted.[SalesPersonID] FROM inserted
            WHERE inserted.[OrderDate] BETWEEN @StartDate AND @EndDate);

-- Update the SalesTerritory SalesYTD when SubTotal is updated
UPDATE [Sales].[SalesTerritory]
SET [Sales].[SalesTerritory].[SalesYTD] =
    (SELECT SUM([Sales].[SalesOrderHeader].[SubTotal])
    FROM [Sales].[SalesOrderHeader]
    WHERE [Sales].[SalesTerritory].[TerritoryID] = [Sales].[SalesOrder-
Header].[TerritoryID]
        AND ([Sales].[SalesOrderHeader].[Status] = 5) -- Shipped
        AND [Sales].[SalesOrderHeader].[OrderDate] BETWEEN @StartDate
AND @EndDate)
    WHERE [Sales].[SalesTerritory].[TerritoryID]
        IN (SELECT DISTINCT inserted.[TerritoryID] FROM inserted
            WHERE inserted.[OrderDate] BETWEEN @StartDate AND @EndDate);

END;
END TRY
BEGIN CATCH
    EXECUTE [dbo].[uspPrintError];

-- Rollback any active or uncommittable transactions before
-- inserting information in the ErrorLog
IF @@TRANCOUNT > 0
BEGIN
    ROLLBACK TRANSACTION;
END

EXECUTE [dbo].[uspLogError];
END CATCH;
END;
GO
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [CK_SalesOrderHeader_DueDate]
CHECK (([DueDate]>=[OrderDate]))
GO
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [CK_SalesOrderHeader_Freight]
CHECK (([Freight]>=(0.00)))
GO
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [CK_SalesOrderHeader_ShipDate]
CHECK (([ShipDate]>=[OrderDate] OR [ShipDate] IS NULL))
GO
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [CK_SalesOrderHeader_Status]
CHECK (([Status]>=(0) AND [Status]<=(8)))
GO
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [CK_SalesOrderHeader_SubTotal]
CHECK (([SubTotal]>=(0.00)))
GO
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [CK_SalesOrderHeader_TaxAmt]
CHECK (([TaxAmt]>=(0.00)))
GO
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [PK_SalesOrderHeader_Sales-
OrderID] PRIMARY KEY CLUSTERED ([SalesOrderID]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_SalesOrderHeader_CustomerID] ON [Sales].[SalesOrder-
Header] ([CustomerID]) ON [PRIMARY]

```

```

GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_SalesOrderHeader_rowguid] ON [Sales].[Sales-
OrderHeader] ([rowguid]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_SalesOrderHeader_SalesOrderNumber] ON
[Sales].[SalesOrderHeader] ([SalesOrderNumber]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_SalesOrderHeader_SalesPersonID] ON [Sales].[SalesOrder-
Header] ([SalesPersonID]) ON [PRIMARY]
GO
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [FK_SalesOrderHeader_Address_
BillToAddressID] FOREIGN KEY ([BillToAddressID]) REFERENCES [Person].[Address]
([AddressID])
GO
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [FK_SalesOrderHeader_Address_
ShipToAddressID] FOREIGN KEY ([ShipToAddressID]) REFERENCES [Person].[Address]
([AddressID])
GO
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [FK_SalesOrderHeader_Credit-
Card_CreditCardID] FOREIGN KEY ([CreditCardID]) REFERENCES [Sales].[CreditCard]
([CreditCardID])
GO
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [FK_SalesOrderHeader_Currency-
Rate_CurrencyRateID] FOREIGN KEY ([CurrencyRateID]) REFERENCES [Sales].[Currency-
Rate] ([CurrencyRateID])
GO
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [FK_SalesOrderHeader_Customer_
CustomerID] FOREIGN KEY ([CustomerID]) REFERENCES [Sales].[Customer] ([CustomerID])
GO
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [FK_SalesOrderHeader_Sales-
Person_SalesPersonID] FOREIGN KEY ([SalesPersonID]) REFERENCES [Sales].[SalesPerson]
([BusinessEntityID])
GO
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [FK_SalesOrderHeader_Sales-
Territory_TerritoryID] FOREIGN KEY ([TerritoryID]) REFERENCES [Sales].[Sales-
Territory] ([TerritoryID])
GO
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [FK_SalesOrderHeader_Ship-
Method_ShipMethodID] FOREIGN KEY ([ShipMethodID]) REFERENCES [Purchasing].[Ship-
Method] ([ShipMethodID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'General sales order information.',
'SHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Financial accounting number
reference.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'COLUMN', N'Account-
Number'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Customer billing address. Foreign
key to Address.AddressID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader',
'COLUMN', N'BillToAddressID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Sales representative comments.',
'SHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'COLUMN', N'Comment'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Approval code provided by the
credit card company.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'COLUMN',
N'CreditCardApprovalCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Credit card identification number.
Foreign key to CreditCard.CreditCardID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrder-
Header', 'COLUMN', N'CreditCardID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Currency exchange rate used.

```

```

Foreign key to CurrencyRate.CurrencyRateID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales-
OrderHeader', 'COLUMN', N'CurrencyRateID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Customer identification number.
Foreign key to Customer.BusinessEntityID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales-
OrderHeader', 'COLUMN', N'CustomerID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date the order is due to the
customer.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'COLUMN', N'DueDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Shipping cost.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesOrderHeader', 'COLUMN', N'Freight'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'COLUMN', N'Modified-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'0 = Order placed by sales person. 1
= Order placed online by customer.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrder-
Header', 'COLUMN', N'OnlineOrderFlag'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Dates the sales order was
created.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'COLUMN', N'OrderDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Customer purchase order number
reference.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'COLUMN', N'Purchase-
OrderNumber'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Incremental number to track changes
to the sales order over time.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader',
'COLUMN', N'RevisionNumber'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesOrderHeader', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key.', 'SCHEMA', N'Sales',
'TABLE', N'SalesOrderHeader', 'COLUMN', N'SalesOrderID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique sales order identification
number.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'COLUMN', N'SalesOrder-
Number'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Sales person who created the sales
order. Foreign key to SalesPerson.BusinessEntityID.', 'SCHEMA', N'Sales', 'TABLE',
N'SalesOrderHeader', 'COLUMN', N'SalesPersonID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date the order was shipped to the
customer.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'COLUMN', N'ShipDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Shipping method. Foreign key to
ShipMethod.ShipMethodID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader',
'COLUMN', N'ShipMethodID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Customer shipping address. Foreign
key to Address.AddressID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader',
'COLUMN', N'ShipToAddressID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Order current status. 1 = In
process; 2 = Approved; 3 = Backordered; 4 = Rejected; 5 = Shipped; 6 = Cancelled',
'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'COLUMN', N'Status'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Sales subtotal. Computed as
SUM(SalesOrderDetail.LineTotal)for the appropriate SalesOrderID.', 'SCHEMA',

```

```

N'Sales', 'TABLE', N'SalesOrderHeader', 'COLUMN', N'SubTotal'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Tax amount.', 'SCHEMA', N'Sales',
'TABLE', N'SalesOrderHeader', 'COLUMN', N'TaxAmt'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Territory in which the sale was
made. Foreign key to SalesTerritory.SalesTerritoryID.', 'SCHEMA', N'Sales', 'TABLE',
N'SalesOrderHeader', 'COLUMN', N'TerritoryID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Total due from customer. Computed
as Subtotal + TaxAmt + Freight.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader',
'COLUMN', N'TotalDue'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [DueDate] >=
[OrderDate]', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'CK_
SalesOrderHeader_DueDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Freight] >=
(0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'CK_
SalesOrderHeader_Freight'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [ShipDate] >=
[OrderDate] OR [ShipDate] IS NULL', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrder
Header', 'CONSTRAINT', N'CK_SalesOrderHeader_ShipDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Status] BETWEEN
(0) AND (8)', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'CK_
SalesOrderHeader_Status'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [SubTotal] >=
(0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'CK_
SalesOrderHeader_SubTotal'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [TaxAmt] >=
(0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'CK_
SalesOrderHeader_TaxAmt'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'DF_
SalesOrderHeader_Freight'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'DF_
SalesOrderHeader_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 1
(TRUE)', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'DF_
SalesOrderHeader_OnlineOrderFlag'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'DF_
SalesOrderHeader_OrderDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0',
'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'DF_
SalesOrderHeader_RevisionNumber'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'DF_
SalesOrderHeader_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 1',
'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'DF_
SalesOrderHeader_Status'
GO

```

```

EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'DF_SalesOrder-
Header_SubTotal'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'DF_SalesOrder-
Header_TaxAmt'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Address.AddressID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT',
N'FK_SalesOrderHeader_Address_BillToAddressID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Address.AddressID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT',
N'FK_SalesOrderHeader_Address_ShipToAddressID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
CreditCard.CreditCardID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader',
'CONSTRAINT', N'FK_SalesOrderHeader_CreditCard_CreditCardID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
CurrencyRate.CurrencyRateID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader',
'CONSTRAINT', N'FK_SalesOrderHeader_CurrencyRate_CurrencyRateID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Customer.CustomerID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader',
'CONSTRAINT', N'FK_SalesOrderHeader_Customer_CustomerID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
SalesPerson.SalesPersonID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader',
'CONSTRAINT', N'FK_SalesOrderHeader_SalesPerson_SalesPersonID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
SalesTerritory.TerritoryID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader',
'CONSTRAINT', N'FK_SalesOrderHeader_SalesTerritory_TerritoryID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
ShipMethod.ShipMethodID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader',
'CONSTRAINT', N'FK_SalesOrderHeader_ShipMethod_ShipMethodID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'PK_-
SalesOrderHeader_SalesOrderID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader',
'INDEX', N'AK_SalesOrderHeader_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'INDEX', N'AK_SalesOrderHeader_-
SalesOrderNumber'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesOrderHeader', 'INDEX', N'IX_SalesOrderHeader_CustomerID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesOrderHeader', 'INDEX', N'IX_SalesOrderHeader_SalesPerson-
ID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'INDEX',
N'PK_SalesOrderHeader_SalesOrderID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'AFTER UPDATE trigger that updates
the RevisionNumber and ModifiedDate columns in the SalesOrderHeader table.Updates

```

```
the SalesYTD column in the SalesPerson and SalesTerritory tables.', 'SCHEMA',  
N'Sales', 'TABLE', N'SalesOrderHeader', 'TRIGGER', N'uSalesOrderHeader'  
GO
```

Uses

[Person].[Address]
[Purchasing].[ShipMethod]
[Sales].[CreditCard]
[Sales].[CurrencyRate]
[Sales].[Customer]
[Sales].[SalesPerson]
[Sales].[SalesTerritory]
[dbo].[AccountNumber]
[dbo].[Flag]
[dbo].[OrderNumber]
Sales

Used By

[Sales].[SalesOrderDetail]
[Sales].[SalesOrderHeaderSalesReason]
[Sales].[vSalesPersonSalesByFiscalYears]

 **[Sales].[SalesOrderHeaderSalesReason]****MS_Description**

Cross-reference table mapping sales orders to sales reason codes.

Properties

Property	Value
Row Count (~)	27647
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	SalesOrderID <i>Primary key. Foreign key to SalesOrder-Header.SalesOrderID.</i>	int	4	False	
	SalesReasonID <i>Primary key. Foreign key to Sales-Reason.SalesReasonID.</i>	int	4	False	
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_SalesOrderHeaderSalesReason_SalesOrderID_SalesReasonID <i>Primary key (clustered) constraint</i>	SalesOrderID, SalesReasonID	True

Foreign Keys

Name	Delete	Columns
FK_SalesOrderHeaderSalesReason_SalesOrderHeader_SalesOrderID <i>Foreign key constraint referencing SalesOrderHeader.SalesOrderID.</i>	Cascade	SalesOrderID->[Sales].[SalesOrderHeader].[SalesOrderID]
FK_SalesOrderHeaderSalesReason_SalesReason_SalesReasonID <i>Foreign key constraint referencing SalesReason.SalesReasonID.</i>		SalesReasonID->[Sales].[SalesReason].[SalesReasonID]

SQL Script

```
CREATE TABLE [Sales].[SalesOrderHeaderSalesReason]
```

```

(
[SalesOrderID] [int] NOT NULL,
[SalesReasonID] [int] NOT NULL,
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_SalesOrderHeaderSalesReason_
ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Sales].[SalesOrderHeaderSalesReason] ADD CONSTRAINT [PK_SalesOrder-
HeaderSalesReason_SalesOrderID_SalesReasonID] PRIMARY KEY CLUSTERED ([SalesOrder-
ID], [SalesReasonID]) ON [PRIMARY]
GO
ALTER TABLE [Sales].[SalesOrderHeaderSalesReason] ADD CONSTRAINT [FK_SalesOrder-
HeaderSalesReason_SalesOrderHeader_SalesOrderID] FOREIGN KEY ([SalesOrderID])
REFERENCES [Sales].[SalesOrderHeader] ([SalesOrderID]) ON DELETE CASCADE
GO
ALTER TABLE [Sales].[SalesOrderHeaderSalesReason] ADD CONSTRAINT [FK_SalesOrder-
HeaderSalesReason_SalesReason_SalesReasonID] FOREIGN KEY ([SalesReasonID])
REFERENCES [Sales].[SalesReason] ([SalesReasonID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Cross-reference table mapping sales
orders to sales reason codes.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeaderSales-
Reason', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeaderSalesReason', 'COLUMN',
N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Foreign key to Sales-
OrderHeader.SalesOrderID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeaderSales-
Reason', 'COLUMN', N'SalesOrderID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Foreign key to Sales-
Reason.SalesReasonID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeaderSalesReason',
'COLUMN', N'SalesReasonID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeaderSalesReason',
'CONSTRAINT', N'DF_SalesOrderHeaderSalesReason_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
SalesOrderHeader.SalesOrderID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader-
SalesReason', 'CONSTRAINT', N'FK_SalesOrderHeaderSalesReason_SalesOrderHeader_Sales-
OrderID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
SalesReason.SalesReasonID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeaderSales-
Reason', 'CONSTRAINT', N'FK_SalesOrderHeaderSalesReason_SalesReason_SalesReasonID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeaderSalesReason',
'CONSTRAINT', N'PK_SalesOrderHeaderSalesReason_SalesOrderID_SalesReasonID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeaderSales-
Reason', 'INDEX', N'PK_SalesOrderHeaderSalesReason_SalesOrderID_SalesReasonID'
GO

```

Uses

[Sales].[SalesOrderHeader]

[Sales].[SalesReason]

Sales

Author: author

Copyright 2017 - All Rights Reserved

Page 232 of 400

 **[Sales].[SalesPerson]**
MS_Description

Sales representative current information.

Properties

Property	Value
Row Count (~)	17
Created	13:14:19 14 marca 2012
Last Modified	15:26:58 22 marca 2017

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
  	BusinessEntityID <i>Primary key for SalesPerson records. Foreign key to Employee.BusinessEntityID</i>	int	4	False	
	TerritoryID <i>Territory currently assigned to. Foreign key to SalesTerritory.SalesTerritoryID.</i>	int	4	True	
	SalesQuota <i>Projected yearly sales.</i>	money	8	True	
	Bonus <i>Bonus due if quota is met.</i>	money	8	False	((0.00))
	CommissionPct <i>Commision percent received per sale.</i>	smallmoney	4	False	((0.00))
	SalesYTD <i>Sales total year to date.</i>	money	8	False	((0.00))
	SalesLastYear <i>Sales total of previous year.</i>	money	8	False	((0.00))
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False	(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
 	PK_SalesPerson_BusinessEntityID <i>Primary key (clustered) constraint</i>	BusinessEntityID	True
	AK_SalesPerson_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True

Check Constraints

Name	On Column	Constraint
CK_SalesPerson_Bonus <i>Check constraint [Bonus] >= (0.00)</i>	Bonus	((Bonus)>=(0.00))
CK_SalesPerson_CommissionPct <i>Check constraint [CommissionPct] >= (0.00)</i>	CommissionPct	((CommissionPct)>=(0.00))
CK_SalesPerson_SalesLastYear <i>Check constraint [SalesLastYear] >= (0.00)</i>	SalesLastYear	((SalesLastYear)>=(0.00))
CK_SalesPerson_SalesQuota <i>Check constraint [SalesQuota] > (0.00)</i>	SalesQuota	((SalesQuota)>(0.00))
CK_SalesPerson_SalesYTD <i>Check constraint [SalesYTD] >= (0.00)</i>	SalesYTD	((SalesYTD)>=(0.00))

Foreign Keys

Name	Columns
FK_SalesPerson_Employee_BusinessEntityID <i>Foreign key constraint referencing Employee.EmployeeID.</i>	BusinessEntityID->[Human-Resources].[Employee].[BusinessEntityID]
FK_SalesPerson_SalesTerritory_TerritoryID <i>Foreign key constraint referencing SalesTerritory.TerritoryID.</i>	TerritoryID->[Sales].[SalesTerritory].[TerritoryID]

SQL Script

```
CREATE TABLE [Sales].[SalesPerson]
(
    [BusinessEntityID] [int] NOT NULL,
    [TerritoryID] [int] NULL,
    [SalesQuota] [money] NULL,
    [Bonus] [money] NOT NULL CONSTRAINT [DF_SalesPerson_Bonus] DEFAULT ((0.00)),
    [CommissionPct] [smallmoney] NOT NULL CONSTRAINT [DF_SalesPerson_CommissionPct]
    DEFAULT ((0.00)),
    [SalesYTD] [money] NOT NULL CONSTRAINT [DF_SalesPerson_SalesYTD] DEFAULT ((0.00)),
    [SalesLastYear] [money] NOT NULL CONSTRAINT [DF_SalesPerson_SalesLastYear] DEFAULT
    ((0.00)),
    [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_SalesPerson_rowguid]
    DEFAULT (newid()),
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_SalesPerson_ModifiedDate] DEFAULT
    (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Sales].[SalesPerson] ADD CONSTRAINT [CK_SalesPerson_Bonus] CHECK
(( [Bonus] >= (0.00) ))
GO
ALTER TABLE [Sales].[SalesPerson] ADD CONSTRAINT [CK_SalesPerson_CommissionPct]
CHECK (( [CommissionPct] >= (0.00) ))
GO
ALTER TABLE [Sales].[SalesPerson] ADD CONSTRAINT [CK_SalesPerson_SalesLastYear]
CHECK (( [SalesLastYear] >= (0.00) ))
GO
ALTER TABLE [Sales].[SalesPerson] ADD CONSTRAINT [CK_SalesPerson_SalesQuota] CHECK
(( [SalesQuota] > (0.00) ))
GO
ALTER TABLE [Sales].[SalesPerson] ADD CONSTRAINT [CK_SalesPerson_SalesYTD] CHECK
```

```

((SalesYTD)>=(0.00))
GO
ALTER TABLE [Sales].[SalesPerson] ADD CONSTRAINT [PK_SalesPerson_BusinessEntityID]
PRIMARY KEY CLUSTERED ([BusinessEntityID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_SalesPerson_rowguid] ON [Sales].[SalesPerson]
(rowguid) ON [PRIMARY]
GO
ALTER TABLE [Sales].[SalesPerson] ADD CONSTRAINT [FK_SalesPerson_Employee_Business-
EntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [HumanResources].[Employee]
([BusinessEntityID])
GO
ALTER TABLE [Sales].[SalesPerson] ADD CONSTRAINT [FK_SalesPerson_SalesTerritory_
TerritoryID] FOREIGN KEY ([TerritoryID]) REFERENCES [Sales].[SalesTerritory]
([TerritoryID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Sales representative current
information.', 'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Bonus due if quota is met.',
'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'COLUMN', N'Bonus'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for SalesPerson
records. Foreign key to Employee.BusinessEntityID', 'SCHEMA', N'Sales', 'TABLE',
N'SalesPerson', 'COLUMN', N'BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Commision percent received per
sale.', 'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'COLUMN', N'CommissionPct'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesPerson', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Sales total of previous year.',
'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'COLUMN', N'SalesLastYear'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Projected yearly sales.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesPerson', 'COLUMN', N'SalesQuota'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Sales total year to date.',
'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'COLUMN', N'SalesYTD'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Territory currently assigned to.
Foreign key to SalesTerritory.SalesTerritoryID.', 'SCHEMA', N'Sales', 'TABLE',
N'SalesPerson', 'COLUMN', N'TerritoryID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Bonus] >=
(0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'CONSTRAINT', N'CK_Sales-
Person_Bonus'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [CommissionPct] >=
(0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'CONSTRAINT', N'CK_Sales-
Person_CommissionPct'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [SalesLastYear] >=
(0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'CONSTRAINT', N'CK_Sales-
Person_SalesLastYear'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [SalesQuota] >
(0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'CONSTRAINT', N'CK_Sales-
Person_SalesQuota'

```

```

GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [SalesYTD] >=
(0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'CONSTRAINT', N'CK_Sales-
Person_SalesYTD'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'CONSTRAINT', N'DF_SalesPerson_Bonus'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'CONSTRAINT', N'DF_SalesPerson_-
CommissionPct'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'CONSTRAINT', N'DF_Sales-
Person_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'CONSTRAINT', N'DF_Sales-
Person_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'CONSTRAINT', N'DF_SalesPerson_Sales-
LastYear'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'CONSTRAINT', N'DF_SalesPerson_Sales-
YTD'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Employee.EmployeeID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'CONSTRAINT',
N'FK_SalesPerson_Employee_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
SalesTerritory.TerritoryID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesPerson',
'CONSTRAINT', N'FK_SalesPerson_SalesTerritory_TerritoryID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'CONSTRAINT', N'PK_Sales-
Person_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'INDEX',
N'AK_SalesPerson_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'INDEX',
N'PK_SalesPerson_BusinessEntityID'
GO

```

Uses

[HumanResources].[Employee]

[Sales].[SalesTerritory]

Sales

Used By

[Sales].[SalesOrderHeader]

[Sales].[SalesPersonQuotaHistory]

[Sales].[SalesTerritoryHistory]

[Sales].[Store]

Project > Isrep17 > User databases > AdventureWorks > Tables > Sales.SalesPerson

[Sales].[vSalesPerson]

[Sales].[vSalesPersonSalesByFiscalYears]

 **[Sales].[SalesPersonQuotaHistory]****MS_Description**

Sales performance tracking.

Properties

Property	Value
Row Count (~)	163
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	BusinessEntityID <i>Sales person identification number. Foreign key to SalesPerson.BusinessEntityID.</i>	int	4	False	
	QuotaDate <i>Sales quota date.</i>	datetime	8	False	
	SalesQuota <i>Sales quota amount.</i>	money	8	False	
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False	(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_SalesPersonQuotaHistory_BusinessEntityID_QuotaDate <i>Primary key (clustered) constraint</i>	BusinessEntityID, QuotaDate	True
	AK_SalesPersonQuotaHistory_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True

Check Constraints

Name	On Column	Constraint
CK_SalesPersonQuotaHistory_SalesQuota <i>Check constraint [SalesQuota] > (0.00)</i>	SalesQuota	([SalesQuota]>(0.00))

Foreign Keys

Name	Columns
FK_SalesPersonQuotaHistory_SalesPerson_BusinessEntityID <i>Foreign key constraint referencing SalesPerson.SalesPersonID.</i>	BusinessEntityID->[Sales].[SalesPerson].[BusinessEntityID]

SQL Script

```

CREATE TABLE [Sales].[SalesPersonQuotaHistory]
(
  [BusinessEntityID] [int] NOT NULL,
  [QuotaDate] [datetime] NOT NULL,
  [SalesQuota] [money] NOT NULL,
  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_SalesPersonQuotaHistory_rowguid] DEFAULT (newid()),
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_SalesPersonQuotaHistory_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Sales].[SalesPersonQuotaHistory] ADD CONSTRAINT [CK_SalesPersonQuotaHistory_SalesQuota] CHECK (([SalesQuota]>(0.00)))
GO
ALTER TABLE [Sales].[SalesPersonQuotaHistory] ADD CONSTRAINT [PK_SalesPersonQuotaHistory_BusinessEntityID_QuotaDate] PRIMARY KEY CLUSTERED ([BusinessEntityID], [QuotaDate]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_SalesPersonQuotaHistory_rowguid] ON [Sales].[SalesPersonQuotaHistory] ([rowguid]) ON [PRIMARY]
GO
ALTER TABLE [Sales].[SalesPersonQuotaHistory] ADD CONSTRAINT [FK_SalesPersonQuotaHistory_SalesPerson_BusinessEntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [Sales].[SalesPerson] ([BusinessEntityID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Sales performance tracking.', 'SCHEMA', N'Sales', 'TABLE', N'SalesPersonQuotaHistory', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Sales person identification number. Foreign key to SalesPerson.BusinessEntityID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesPersonQuotaHistory', 'COLUMN', N'BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last updated.', 'SCHEMA', N'Sales', 'TABLE', N'SalesPersonQuotaHistory', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Sales quota date.', 'SCHEMA', N'Sales', 'TABLE', N'SalesPersonQuotaHistory', 'COLUMN', N'QuotaDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Sales', 'TABLE', N'SalesPersonQuotaHistory', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Sales quota amount.', 'SCHEMA', N'Sales', 'TABLE', N'SalesPersonQuotaHistory', 'COLUMN', N'SalesQuota'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [SalesQuota] > (0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesPersonQuotaHistory', 'CONSTRAINT', N'CK_SalesPersonQuotaHistory_SalesQuota'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of

```

```
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'SalesPersonQuotaHistory', 'CONSTRAINT',
N'DF_SalesPersonQuotaHistory_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Sales', 'TABLE', N'SalesPersonQuotaHistory', 'CONSTRAINT',
N'DF_SalesPersonQuotaHistory_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
SalesPerson.SalesPersonID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesPersonQuota-
History', 'CONSTRAINT', N'FK_SalesPersonQuotaHistory_SalesPerson_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Sales', 'TABLE', N'SalesPersonQuotaHistory', 'CONSTRAINT',
N'PK_SalesPersonQuotaHistory_BusinessEntityID_QuotaDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'SalesPersonQuota-
History', 'INDEX', N'AK_SalesPersonQuotaHistory_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'SalesPersonQuotaHistory',
'INDEX', N'PK_SalesPersonQuotaHistory_BusinessEntityID_QuotaDate'
GO
```

Uses

[Sales].[SalesPerson]
Sales

 **[Sales].[SalesReason]****MS_Description**

Lookup table of customer purchase reasons.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	10
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	SalesReasonID <i>Primary key for SalesReason records.</i>	int	4	False	1 - 1	
	Name <i>Sales reason description.</i>	[dbo].[Name]	100	False		
	ReasonType <i>Category the sales reason belongs to.</i>	[dbo].[Name]	100	False		
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_SalesReason_SalesReasonID <i>Primary key (clustered) constraint</i>	SalesReasonID	True

SQL Script

```
CREATE TABLE [Sales].[SalesReason]
(
  [SalesReasonID] [int] NOT NULL IDENTITY(1, 1),
  [Name] [dbo].[Name] NOT NULL,
  [ReasonType] [dbo].[Name] NOT NULL,
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_SalesReason_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Sales].[SalesReason] ADD CONSTRAINT [PK_SalesReason_SalesReasonID]
PRIMARY KEY CLUSTERED ([SalesReasonID]) ON [PRIMARY]
```

```
GO
EXEC sp_addextendedproperty N'MS_Description', N'Lookup table of customer purchase reasons.', 'SCHEMA', N'Sales', 'TABLE', N'SalesReason', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last updated.', 'SCHEMA', N'Sales', 'TABLE', N'SalesReason', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Sales reason description.', 'SCHEMA', N'Sales', 'TABLE', N'SalesReason', 'COLUMN', N'Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Category the sales reason belongs to.', 'SCHEMA', N'Sales', 'TABLE', N'SalesReason', 'COLUMN', N'ReasonType'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for SalesReason records.', 'SCHEMA', N'Sales', 'TABLE', N'SalesReason', 'COLUMN', N'SalesReasonID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'SalesReason', 'CONSTRAINT', N'DF_SalesReason_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Sales', 'TABLE', N'SalesReason', 'CONSTRAINT', N'PK_SalesReason_SalesReasonID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'SalesReason', 'INDEX', N'PK_SalesReason_SalesReasonID'
GO
```

Uses

[dbo].[Name]
Sales

Used By

[Sales].[SalesOrderHeaderSalesReason]

 **[Sales].[SalesTaxRate]**

MS_Description

Tax rate lookup table.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	29
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	SalesTaxRateID <i>Primary key for SalesTaxRate records.</i>	int	4	False	1 - 1	
	StateProvinceID <i>State, province, or country/region the sales tax applies to.</i>	int	4	False		
	TaxType <i>1 = Tax applied to retail transactions, 2 = Tax applied to wholesale transactions, 3 = Tax applied to all sales (retail and wholesale) transactions.</i>	tinyint	1	False		
	TaxRate <i>Tax rate amount.</i>	smallmoney	4	False		((0.00))
	Name <i>Tax rate description.</i>	[dbo].[Name]	100	False		
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False		(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_SalesTaxRate_SalesTaxRateID <i>Primary key (clustered) constraint</i>	SalesTaxRateID	True
	AK_SalesTaxRate_StateProvinceID_TaxType <i>Unique nonclustered index.</i>	StateProvinceID, TaxType	True

AK_SalesTaxRate_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True
---	---------	------

Check Constraints

Name	On Column	Constraint
CK_SalesTaxRate_TaxType <i>Check constraint [TaxType] BETWEEN (1) AND (3)</i>	TaxType	([TaxType]>=(1) AND [TaxType]<=(3))

Foreign Keys

Name	Columns
FK_SalesTaxRate_StateProvince_StateProvinceID <i>Foreign key constraint referencing StateProvince.StateProvinceID.</i>	StateProvinceID->[Person].[StateProvince].[StateProvinceID]

SQL Script

```
CREATE TABLE [Sales].[SalesTaxRate]
(
    [SalesTaxRateID] [int] NOT NULL IDENTITY(1, 1),
    [StateProvinceID] [int] NOT NULL,
    [TaxType] [tinyint] NOT NULL,
    [TaxRate] [smallmoney] NOT NULL CONSTRAINT [DF_SalesTaxRate_TaxRate] DEFAULT
    ((0.00)),
    [Name] [dbo].[Name] NOT NULL,
    [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_SalesTax-
    Rate_rowguid] DEFAULT (newid()),
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_SalesTaxRate_ModifiedDate] DEFAULT
    (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Sales].[SalesTaxRate] ADD CONSTRAINT [CK_SalesTaxRate_TaxType] CHECK
((([TaxType]>=(1) AND [TaxType]<=(3)))
GO
ALTER TABLE [Sales].[SalesTaxRate] ADD CONSTRAINT [PK_SalesTaxRate_SalesTaxRateID]
PRIMARY KEY CLUSTERED ([SalesTaxRateID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_SalesTaxRate_rowguid] ON [Sales].[SalesTaxRate]
([rowguid]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_SalesTaxRate_StateProvinceID_TaxType] ON
[Sales].[SalesTaxRate] ([StateProvinceID], [TaxType]) ON [PRIMARY]
GO
ALTER TABLE [Sales].[SalesTaxRate] ADD CONSTRAINT [FK_SalesTaxRate_StateProvince_
StateProvinceID] FOREIGN KEY ([StateProvinceID]) REFERENCES [Person].[StateProvince]
([StateProvinceID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Tax rate lookup table.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesTaxRate', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTaxRate', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Tax rate description.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesTaxRate', 'COLUMN', N'Name'
```

```

GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesTaxRate', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for SalesTaxRate
records.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTaxRate', 'COLUMN', N'SalesTaxRateID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'State, province, or country/region
the sales tax applies to.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTaxRate', 'COLUMN',
N'StateProvinceID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Tax rate amount.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesTaxRate', 'COLUMN', N'TaxRate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'1 = Tax applied to retail
transactions, 2 = Tax applied to wholesale transactions, 3 = Tax applied to all
sales (retail and wholesale) transactions.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTax-
Rate', 'COLUMN', N'TaxType'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [TaxType] BETWEEN
(1) AND (3)', 'SCHEMA', N'Sales', 'TABLE', N'SalesTaxRate', 'CONSTRAINT', N'CK_Sales-
TaxRate_TaxType'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'SalesTaxRate', 'CONSTRAINT', N'DF_Sales-
TaxRate_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Sales', 'TABLE', N'SalesTaxRate', 'CONSTRAINT', N'DF_SalesTax-
Rate_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Sales', 'TABLE', N'SalesTaxRate', 'CONSTRAINT', N'DF_SalesTaxRate_Tax-
Rate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
StateProvince.StateProvinceID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTaxRate',
'CONSTRAINT', N'FK_SalesTaxRate_StateProvince_StateProvinceID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Sales', 'TABLE', N'SalesTaxRate', 'CONSTRAINT', N'PK_Sales-
TaxRate_SalesTaxRateID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTaxRate',
'INDEX', N'AK_SalesTaxRate_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SCHEMA', N'Sales', 'TABLE', N'SalesTaxRate', 'INDEX', N'AK_SalesTaxRate_State-
ProvinceID_TaxType'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTaxRate', 'INDEX',
N'PK_SalesTaxRate_SalesTaxRateID'
GO

```

Uses

[Person].[StateProvince]

[dbo].[Name]

Sales

 **[Sales].[SalesTerritory]**
MS_Description

Sales territory lookup table.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	10
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	TerritoryID <i>Primary key for SalesTerritory records.</i>	int	4	False	1 - 1	
	Name <i>Sales territory description</i>	[dbo].[Name]	100	False		
	CountryRegionCode <i>ISO standard country or region code. Foreign key to Country-Region.CountryRegionCode.</i>	nvarchar(3)	6	False		
	Group <i>Geographic area to which the sales territory belong.</i>	nvarchar(50)	100	False		
	SalesYTD <i>Sales in the territory year to date.</i>	money	8	False		((0.00))
	SalesLastYear <i>Sales in the territory the previous year.</i>	money	8	False		((0.00))
	CostYTD <i>Business costs in the territory year to date.</i>	money	8	False		((0.00))
	CostLastYear <i>Business costs in the territory the previous year.</i>	money	8	False		((0.00))
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False		(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_SalesTerritory_TerritoryID <i>Primary key (clustered) constraint</i>	TerritoryID	True
	AK_SalesTerritory_Name <i>Unique nonclustered index.</i>	Name	True
	AK_SalesTerritory_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True

Check Constraints

Name	On Column	Constraint
CK_SalesTerritory_CostLastYear <i>Check constraint [CostLastYear] >= (0.00)</i>	CostLastYear	([CostLastYear]>=(0.00))
CK_SalesTerritory_CostYTD <i>Check constraint [CostYTD] >= (0.00)</i>	CostYTD	([CostYTD]>=(0.00))
CK_SalesTerritory_SalesLastYear <i>Check constraint [SalesLastYear] >= (0.00)</i>	SalesLastYear	([SalesLastYear]>=(0.00))
CK_SalesTerritory_SalesYTD <i>Check constraint [SalesYTD] >= (0.00)</i>	SalesYTD	([SalesYTD]>=(0.00))

Foreign Keys

Name	Columns
FK_SalesTerritory_CountryRegion_CountryRegion-Code <i>Foreign key constraint referencing Country-Region.CountryRegionCode.</i>	CountryRegionCode->[Person].[Country-Region].[CountryRegionCode]

SQL Script

```
CREATE TABLE [Sales].[SalesTerritory]
(
  [TerritoryID] [int] NOT NULL IDENTITY(1, 1),
  [Name] [dbo].[Name] NOT NULL,
  [CountryRegionCode] [nvarchar] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
  [Group] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
  [SalesYTD] [money] NOT NULL CONSTRAINT [DF_SalesTerritory_SalesYTD] DEFAULT
  ((0.00)),
  [SalesLastYear] [money] NOT NULL CONSTRAINT [DF_SalesTerritory_SalesLastYear]
  DEFAULT ((0.00)),
  [CostYTD] [money] NOT NULL CONSTRAINT [DF_SalesTerritory_CostYTD] DEFAULT ((0.00)),
  [CostLastYear] [money] NOT NULL CONSTRAINT [DF_SalesTerritory_CostLastYear] DEFAULT
  ((0.00)),
  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_Sales-
  Territory_rowguid] DEFAULT (newid()),
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_SalesTerritory_ModifiedDate]
  DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Sales].[SalesTerritory] ADD CONSTRAINT [CK_SalesTerritory_CostLastYear]
CHECK (([CostLastYear]>=(0.00)))
GO
ALTER TABLE [Sales].[SalesTerritory] ADD CONSTRAINT [CK_SalesTerritory_CostYTD]
```

```

CHECK (([CostYTD]>=(0.00)))
GO
ALTER TABLE [Sales].[SalesTerritory] ADD CONSTRAINT [CK_SalesTerritory_SalesLastYear] CHECK (([SalesLastYear]>=(0.00)))
GO
ALTER TABLE [Sales].[SalesTerritory] ADD CONSTRAINT [CK_SalesTerritory_SalesYTD] CHECK (([SalesYTD]>=(0.00)))
GO
ALTER TABLE [Sales].[SalesTerritory] ADD CONSTRAINT [PK_SalesTerritory_TerritoryID] PRIMARY KEY CLUSTERED ([TerritoryID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_SalesTerritory_Name] ON [Sales].[SalesTerritory] ([Name]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_SalesTerritory_rowguid] ON [Sales].[SalesTerritory] ([rowguid]) ON [PRIMARY]
GO
ALTER TABLE [Sales].[SalesTerritory] ADD CONSTRAINT [FK_SalesTerritory_CountryRegion_CountryRegionCode] FOREIGN KEY ([CountryRegionCode]) REFERENCES [Person].[CountryRegion] ([CountryRegionCode])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Sales territory lookup table.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Business costs in the territory the previous year.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'COLUMN', N'CostLastYear'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Business costs in the territory year to date.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'COLUMN', N'CostYTD'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ISO standard country or region code. Foreign key to CountryRegion.CountryRegionCode. ', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'COLUMN', N'CountryRegionCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Geographic area to which the sales territory belong.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'COLUMN', N'Group'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last updated.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Sales territory description.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'COLUMN', N'Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Sales in the territory the previous year.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'COLUMN', N'SalesLastYear'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Sales in the territory year to date.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'COLUMN', N'SalesYTD'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for SalesTerritory records.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'COLUMN', N'TerritoryID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [CostLastYear] >= (0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'CK_SalesTerritory_CostLastYear'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [CostYTD] >=

```

```

(0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'CK_Sales-
Territory_CostYTD'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [SalesLastYear] >=
(0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'CK_Sales-
Territory_SalesLastYear'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [SalesYTD] >=
(0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'CK_Sales-
Territory_SalesYTD'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'DF_SalesTerritory_-
CostLastYear'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'DF_SalesTerritory_-
CostYTD'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'DF_Sales-
Territory_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'DF_Sales-
Territory_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'DF_SalesTerritory_-
SalesLastYear'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'DF_SalesTerritory_-
SalesYTD'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
CountryRegion.CountryRegionCode.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory',
'CONSTRAINT', N'FK_SalesTerritory_CountryRegion_CountryRegionCode'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'PK_-
SalesTerritory_TerritoryID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.',
'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'INDEX', N'AK_SalesTerritory_Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory',
'INDEX', N'AK_SalesTerritory_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'INDEX',
N'PK_SalesTerritory_TerritoryID'
GO

```

Uses

[Person].[CountryRegion]
[dbo].[Name]
Sales

Used By

[Person].[StateProvince]
[Sales].[Customer]
[Sales].[SalesOrderHeader]
[Sales].[SalesPerson]
[Sales].[SalesTerritoryHistory]
[Sales].[vSalesPerson]
[Sales].[vSalesPersonSalesByFiscalYears]

 **[Sales].[SalesTerritoryHistory]**
MS_Description

Sales representative transfers to other sales territories.

Properties

Property	Value
Row Count (~)	17
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	BusinessEntityID <i>Primary key. The sales rep. Foreign key to SalesPerson.BusinessEntityID.</i>	int	4	False	
	TerritoryID <i>Primary key. Territory identification number. Foreign key to SalesTerritory.SalesTerritoryID.</i>	int	4	False	
	StartDate <i>Primary key. Date the sales representative started work in the territory.</i>	datetime	8	False	
	EndDate <i>Date the sales representative left work in the territory.</i>	datetime	8	True	
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False	(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_SalesTerritoryHistory_BusinessEntityID_StartDate_TerritoryID <i>Primary key (clustered) constraint</i>	BusinessEntityID, StartDate, TerritoryID	True
	AK_SalesTerritoryHistory_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True

Check Constraints

Name	Constraint
CK_SalesTerritoryHistory_EndDate <i>Check constraint [EndDate] >= [StartDate] OR [EndDate] IS NULL</i>	(([EndDate]>=[StartDate] OR [EndDate] IS NULL)

Foreign Keys

Name	Columns
FK_SalesTerritoryHistory_SalesPerson_BusinessEntityID <i>Foreign key constraint referencing SalesPerson.SalesPersonID.</i>	BusinessEntityID->[Sales].[SalesPerson].[BusinessEntityID]
FK_SalesTerritoryHistory_SalesTerritory_TerritoryID <i>Foreign key constraint referencing SalesTerritory.TerritoryID.</i>	TerritoryID->[Sales].[SalesTerritory].[TerritoryID]

SQL Script

```

CREATE TABLE [Sales].[SalesTerritoryHistory]
(
    [BusinessEntityID] [int] NOT NULL,
    [TerritoryID] [int] NOT NULL,
    [StartDate] [datetime] NOT NULL,
    [EndDate] [datetime] NULL,
    [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_SalesTerritoryHistory_rowguid] DEFAULT (newid()),
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_SalesTerritoryHistory_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Sales].[SalesTerritoryHistory] ADD CONSTRAINT [CK_SalesTerritoryHistory_EndDate] CHECK (([EndDate]>=[StartDate] OR [EndDate] IS NULL))
GO
ALTER TABLE [Sales].[SalesTerritoryHistory] ADD CONSTRAINT [PK_SalesTerritoryHistory_BusinessEntityID_StartDate_TerritoryID] PRIMARY KEY CLUSTERED ([BusinessEntityID], [StartDate], [TerritoryID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_SalesTerritoryHistory_rowguid] ON [Sales].[SalesTerritoryHistory] ([rowguid]) ON [PRIMARY]
GO
ALTER TABLE [Sales].[SalesTerritoryHistory] ADD CONSTRAINT [FK_SalesTerritoryHistory_SalesPerson_BusinessEntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [Sales].[SalesPerson] ([BusinessEntityID])
GO
ALTER TABLE [Sales].[SalesTerritoryHistory] ADD CONSTRAINT [FK_SalesTerritoryHistory_SalesTerritory_TerritoryID] FOREIGN KEY ([TerritoryID]) REFERENCES [Sales].[SalesTerritory] ([TerritoryID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Sales representative transfers to other sales territories.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritoryHistory', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. The sales rep. Foreign key to SalesPerson.BusinessEntityID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritoryHistory', 'COLUMN', N'BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date the sales representative left

```

```

work in the territory.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritoryHistory',
'COLUMN', N'EndDate'

GO

EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritoryHistory', 'COLUMN',
N'ModifiedDate'

GO

EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesTerritoryHistory', 'COLUMN', N'rowguid'

GO

EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Date the sales
representative started work in the territory.', 'SCHEMA', N'Sales', 'TABLE', N'Sales-
TerritoryHistory', 'COLUMN', N'StartDate'

GO

EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Territory
identification number. Foreign key to SalesTerritory.SalesTerritoryID.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesTerritoryHistory', 'COLUMN', N'TerritoryID'

GO

EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [EndDate] >=
[StartDate] OR [EndDate] IS NULL', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory-
History', 'CONSTRAINT', N'CK_SalesTerritoryHistory_EndDate'

GO

EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritoryHistory', 'CONSTRAINT',
N'DF_SalesTerritoryHistory_ModifiedDate'

GO

EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritoryHistory', 'CONSTRAINT', N'DF_
SalesTerritoryHistory_rowguid'

GO

EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
SalesPerson.SalesPersonID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritoryHistory',
'CONSTRAINT', N'FK_SalesTerritoryHistory_SalesPerson_BusinessEntityID'

GO

EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
SalesTerritory.TerritoryID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritoryHistory',
'CONSTRAINT', N'FK_SalesTerritoryHistory_SalesTerritory_TerritoryID'

GO

EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritoryHistory', 'CONSTRAINT',
N'PK_SalesTerritoryHistory_BusinessEntityID_StartDate_TerritoryID'

GO

EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory-
History', 'INDEX', N'AK_SalesTerritoryHistory_rowguid'

GO

EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritoryHistory',
'INDEX', N'PK_SalesTerritoryHistory_BusinessEntityID_StartDate_TerritoryID'

GO

```

Uses

[Sales].[SalesPerson]
 [Sales].[SalesTerritory]
 Sales

[Sales].[ShoppingCartItem]

MS_Description

Contains online customer orders until the order is submitted or cancelled.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	3
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	ShoppingCartItemID <i>Primary key for ShoppingCartItem records.</i>	int	4	False	1 - 1	
	ShoppingCartID <i>Shopping cart identification number.</i>	nvarchar(50)	100	False		
	Quantity <i>Product quantity ordered.</i>	int	4	False		((1))
	ProductID <i>Product ordered. Foreign key to Product.ProductID.</i>	int	4	False		
	DateCreated <i>Date the time the record was created.</i>	datetime	8	False		(getdate())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_ShoppingCartItem_ShoppingCartItemID <i>Primary key (clustered) constraint</i>	ShoppingCartItemID	True
	IX_ShoppingCartItem_ShoppingCartID_ProductID <i>Nonclustered index.</i>	ShoppingCartID, ProductID	

Check Constraints

Name	On Column	Constraint
CK_ShoppingCartItem_Quantity	Quantity	([Quantity]>=(1))

Check constraint [Quantity] >= (1)		
------------------------------------	--	--

Foreign Keys

Name	Columns
FK_ShoppingCartItem_Product_ProductID <i>Foreign key constraint referencing Product.ProductID.</i>	ProductID->[Production].[Product].[ProductID]

SQL Script

```

CREATE TABLE [Sales].[ShoppingCartItem]
(
    [ShoppingCartItemID] [int] NOT NULL IDENTITY(1, 1),
    [ShoppingCartID] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [Quantity] [int] NOT NULL CONSTRAINT [DF_ShoppingCartItem_Quantity] DEFAULT ((1)),
    [ProductID] [int] NOT NULL,
    [DateCreated] [datetime] NOT NULL CONSTRAINT [DF_ShoppingCartItem_DateCreated]
    DEFAULT (getdate()),
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_ShoppingCartItem_ModifiedDate]
    DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Sales].[ShoppingCartItem] ADD CONSTRAINT [CK_ShoppingCartItem_Quantity]
CHECK (([Quantity]>=(1)))
GO
ALTER TABLE [Sales].[ShoppingCartItem] ADD CONSTRAINT [PK_ShoppingCartItem_Shopping-
CartItemID] PRIMARY KEY CLUSTERED ([ShoppingCartItemID]) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_ShoppingCartItem_ShoppingCartID_ProductID] ON
[Sales].[ShoppingCartItem] ([ShoppingCartID], [ProductID]) ON [PRIMARY]
GO
ALTER TABLE [Sales].[ShoppingCartItem] ADD CONSTRAINT [FK_ShoppingCartItem_Product_
ProductID] FOREIGN KEY ([ProductID]) REFERENCES [Production].[Product] ([ProductID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Contains online customer orders
until the order is submitted or cancelled.', 'SCHEMA', N'Sales', 'TABLE', N'Shopping-
CartItem', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date the time the record was
created.', 'SCHEMA', N'Sales', 'TABLE', N'ShoppingCartItem', 'COLUMN', N'Date-
Created'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'ShoppingCartItem', 'COLUMN', N'Modified-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product ordered. Foreign key to
Product.ProductID.', 'SCHEMA', N'Sales', 'TABLE', N'ShoppingCartItem', 'COLUMN',
N'ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product quantity ordered.',
'SCHEMA', N'Sales', 'TABLE', N'ShoppingCartItem', 'COLUMN', N'Quantity'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Shopping cart identification
number.', 'SCHEMA', N'Sales', 'TABLE', N'ShoppingCartItem', 'COLUMN', N'ShoppingCart-
ID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for ShoppingCartItem
records.', 'SCHEMA', N'Sales', 'TABLE', N'ShoppingCartItem', 'COLUMN', N'Shopping-

```

```
CartItemID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Quantity] >=
(1)', 'SCHEMA', N'Sales', 'TABLE', N'ShoppingCartItem', 'CONSTRAINT', N'CK_Shopping-
CartItem_Quantity'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'ShoppingCartItem', 'CONSTRAINT', N'DF_-
ShoppingCartItem_DateCreated'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'ShoppingCartItem', 'CONSTRAINT', N'DF_-
ShoppingCartItem_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 1',
'SCHEMA', N'Sales', 'TABLE', N'ShoppingCartItem', 'CONSTRAINT', N'DF_ShoppingCart-
Item_Quantity'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Product.ProductID.', 'SCHEMA', N'Sales', 'TABLE', N'ShoppingCartItem', 'CONSTRAINT',
N'FK_ShoppingCartItem_Product_ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Sales', 'TABLE', N'ShoppingCartItem', 'CONSTRAINT', N'PK_-
ShoppingCartItem_ShoppingCartItemID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Sales', 'TABLE', N'ShoppingCartItem', 'INDEX', N'IX_ShoppingCartItem_ShoppingCart-
ID_ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'ShoppingCartItem', 'INDEX',
N'PK_ShoppingCartItem_ShoppingCartItemID'
GO
```

Uses

[Production].[Product]
Sales

 **[Sales].[SpecialOffer]**

MS_Description

Sale discounts lookup table.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	16
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	SpecialOfferID <i>Primary key for SpecialOffer records.</i>	int	4	False	1 - 1	
	Description <i>Discount description.</i>	nvarchar(255)	510	False		
	DiscountPct <i>Discount percentage.</i>	smallmoney	4	False		((0.00))
	Type <i>Discount type category.</i>	nvarchar(50)	100	False		
	Category <i>Group the discount applies to such as Reseller or Customer.</i>	nvarchar(50)	100	False		
	StartDate <i>Discount start date.</i>	datetime	8	False		
	EndDate <i>Discount end date.</i>	datetime	8	False		
	MinQty <i>Minimum discount percent allowed.</i>	int	4	False		((0))
	MaxQty <i>Maximum discount percent allowed.</i>	int	4	True		
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False		(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False		(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_SpecialOffer_SpecialOfferID <i>Primary key (clustered) constraint</i>	SpecialOfferID	True
	AK_SpecialOffer_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True

Check Constraints

Name	On Column	Constraint
CK_SpecialOffer_DiscountPct <i>Check constraint [DiscountPct] >= (0.00)</i>	DiscountPct	([DiscountPct]>=(0.00))
CK_SpecialOffer_EndDate <i>Check constraint [EndDate] >= [StartDate]</i>		([EndDate]>=[StartDate])
CK_SpecialOffer_MaxQty <i>Check constraint [MaxQty] >= (0)</i>	MaxQty	([MaxQty]>=(0))
CK_SpecialOffer_MinQty <i>Check constraint [MinQty] >= (0)</i>	MinQty	([MinQty]>=(0))

SQL Script

```

CREATE TABLE [Sales].[SpecialOffer]
(
    [SpecialOfferID] [int] NOT NULL IDENTITY(1, 1),
    [Description] [nvarchar] (255) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [DiscountPct] [smallmoney] NOT NULL CONSTRAINT [DF_SpecialOffer_DiscountPct] DEFAULT ((0.00)),
    [Type] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [Category] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [StartDate] [datetime] NOT NULL,
    [EndDate] [datetime] NOT NULL,
    [MinQty] [int] NOT NULL CONSTRAINT [DF_SpecialOffer_MinQty] DEFAULT ((0)),
    [MaxQty] [int] NULL,
    [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_SpecialOffer_rowguid] DEFAULT (newid()),
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_SpecialOffer_ModifiedDate] DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Sales].[SpecialOffer] ADD CONSTRAINT [CK_SpecialOffer_DiscountPct] CHECK (([DiscountPct]>=(0.00)))
GO
ALTER TABLE [Sales].[SpecialOffer] ADD CONSTRAINT [CK_SpecialOffer_EndDate] CHECK (([EndDate]>=[StartDate]))
GO
ALTER TABLE [Sales].[SpecialOffer] ADD CONSTRAINT [CK_SpecialOffer_MaxQty] CHECK (([MaxQty]>=(0)))
GO
ALTER TABLE [Sales].[SpecialOffer] ADD CONSTRAINT [CK_SpecialOffer_MinQty] CHECK (([MinQty]>=(0)))
GO
ALTER TABLE [Sales].[SpecialOffer] ADD CONSTRAINT [PK_SpecialOffer_SpecialOfferID] PRIMARY KEY CLUSTERED ([SpecialOfferID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_SpecialOffer_rowguid] ON [Sales].[SpecialOffer]

```

```

([rowguid]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Sale discounts lookup table.',
'SHEMA', N'Sales', 'TABLE', N'SpecialOffer', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Group the discount applies to such
as Reseller or Customer.', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'COLUMN',
N'Category'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Discount description.', 'SCHEMA',
N'Sales', 'TABLE', N'SpecialOffer', 'COLUMN', N'Description'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Discount presentage.', 'SCHEMA',
N'Sales', 'TABLE', N'SpecialOffer', 'COLUMN', N'DiscountPct'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Discount end date.', 'SCHEMA',
N'Sales', 'TABLE', N'SpecialOffer', 'COLUMN', N'EndDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Maximum discount percent allowed.',
'SHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'COLUMN', N'MaxQty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Minimum discount percent allowed.',
'SHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'COLUMN', N'MinQty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Sales', 'TABLE', N'SpecialOffer', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for SpecialOffer
records.', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'COLUMN', N'SpecialOfferID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Discount start date.', 'SCHEMA',
N'Sales', 'TABLE', N'SpecialOffer', 'COLUMN', N'StartDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Discount type category.', 'SCHEMA',
N'Sales', 'TABLE', N'SpecialOffer', 'COLUMN', N'Type'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [DiscountPct] >=
(0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'CONSTRAINT', N'CK_Special-
Offer_DiscountPct'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [EndDate] >=
[StartDate]', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'CONSTRAINT', N'CK_-
SpecialOffer_EndDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [MaxQty] >= (0)',
'SHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'CONSTRAINT', N'CK_SpecialOffer_Max-
Qty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [MinQty] >= (0)',
'SHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'CONSTRAINT', N'CK_SpecialOffer_Min-
Qty'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'CONSTRAINT', N'DF_SpecialOffer_-
DiscountPct'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'CONSTRAINT', N'DF_SpecialOffer_Min-
Qty'

```

```
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'CONSTRAINT', N'DF_Special-
Offer_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'CONSTRAINT', N'DF_Special-
Offer_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'CONSTRAINT', N'PK_-
SpecialOffer_SpecialOfferID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOffer',
'INDEX', N'AK_SpecialOffer_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'INDEX',
N'PK_SpecialOffer_SpecialOfferID'
GO
```

Uses

Sales

Used By

[Sales].[SpecialOfferProduct]

[Sales].[SpecialOfferProduct]

MS_Description

Cross-reference table mapping products to special offer discounts.

Properties

Property	Value
Row Count (~)	538
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
	SpecialOfferID <i>Primary key for SpecialOfferProduct records.</i>	int	4	False	
	ProductID <i>Product identification number. Foreign key to Product.ProductID.</i>	int	4	False	
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False	(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Unique
	PK_SpecialOfferProduct_SpecialOfferID_ProductID <i>Primary key (clustered) constraint</i>	SpecialOfferID, ProductID	True
	AK_SpecialOfferProduct_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid	True
	IX_SpecialOfferProduct_ProductID <i>Nonclustered index.</i>	ProductID	

Foreign Keys

Name	Columns
FK_SpecialOfferProduct_Product_ProductID <i>Foreign key constraint referencing Product.ProductID.</i>	ProductID->[Production].[Product].[ProductID]
FK_SpecialOfferProduct_SpecialOffer_SpecialOfferID <i>Foreign key constraint referencing SpecialOffer.SpecialOfferID.</i>	SpecialOfferID->[Sales].[SpecialOffer].[SpecialOfferID]

SQL Script

```

CREATE TABLE [Sales].[SpecialOfferProduct]
(
    [SpecialOfferID] [int] NOT NULL,
    [ProductID] [int] NOT NULL,
    [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_SpecialOffer-
Product_rowguid] DEFAULT (newid()),
    [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_SpecialOfferProduct_ModifiedDate]
DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Sales].[SpecialOfferProduct] ADD CONSTRAINT [PK_SpecialOfferProduct_
SpecialOfferID_ProductID] PRIMARY KEY CLUSTERED ([SpecialOfferID], [ProductID]) ON
[PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_SpecialOfferProduct_ProductID] ON [Sales].[Special-
OfferProduct] ([ProductID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_SpecialOfferProduct_rowguid] ON
[Sales].[SpecialOfferProduct] ([rowguid]) ON [PRIMARY]
GO
ALTER TABLE [Sales].[SpecialOfferProduct] ADD CONSTRAINT [FK_SpecialOfferProduct_
Product_ProductID] FOREIGN KEY ([ProductID]) REFERENCES [Production].[Product]
([ProductID])
GO
ALTER TABLE [Sales].[SpecialOfferProduct] ADD CONSTRAINT [FK_SpecialOfferProduct_
SpecialOffer_SpecialOfferID] FOREIGN KEY ([SpecialOfferID]) REFERENCES
[Sales].[SpecialOffer] ([SpecialOfferID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Cross-reference table mapping
products to special offer discounts.', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOffer-
Product', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOfferProduct', 'COLUMN', N'Modified-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product identification number.
Foreign key to Product.ProductID.', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOffer-
Product', 'COLUMN', N'ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Sales', 'TABLE', N'SpecialOfferProduct', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for SpecialOfferProduct
records.', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOfferProduct', 'COLUMN', N'Special-
OfferID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOfferProduct', 'CONSTRAINT', N'DF_
SpecialOfferProduct_ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOfferProduct', 'CONSTRAINT', N'DF_
SpecialOfferProduct_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Product.ProductID.', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOfferProduct',
'CONSTRAINT', N'FK_SpecialOfferProduct_Product_ProductID'

```

```
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
SpecialOffer.SpecialOfferID.', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOfferProduct',
'CONSTRAINT', N'FK_SpecialOfferProduct_SpecialOffer_SpecialOfferID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOfferProduct', 'CONSTRAINT',
N'PK_SpecialOfferProduct_SpecialOfferID_ProductID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOfferProduct',
'INDEX', N'AK_SpecialOfferProduct_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Sales', 'TABLE', N'SpecialOfferProduct', 'INDEX', N'IX_SpecialOfferProduct_Product-
ID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOfferProduct',
'INDEX', N'PK_SpecialOfferProduct_SpecialOfferID_ProductID'
GO
```

Uses

[Production].[Product]
[Sales].[SpecialOffer]
Sales

Used By

[Sales].[SalesOrderDetail]

 [Sales].[Store]**MS_Description**

Customers (resellers) of Adventure Works products.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	701
Created	13:14:19 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)	Allow Nulls	Default
 	BusinessEntityID <i>Primary key. Foreign key to Customer.BusinessEntityID.</i>	int	4	False	
	Name <i>Name of the store.</i>	[dbo].[Name]	100	False	
	SalesPersonID <i>ID of the sales person assigned to the customer. Foreign key to SalesPerson.BusinessEntityID.</i>	int	4	True	
	Demographics <i>Demographic information about the store such as the number of employees, annual sales and store type.</i>	xml([Sales].[StoreSurvey-SchemaCollection])	max	True	
	rowguid <i>ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.</i>	uniqueidentifier	16	False	(newid())
	ModifiedDate <i>Date and time the record was last updated.</i>	datetime	8	False	(getdate())

Indexes

Key	Name	Key Columns	Type	Unique	XML Type
 	PK_Store_BusinessEntityID <i>Primary key (clustered) constraint</i>	Business-EntityID		True	
	AK_Store_rowguid <i>Unique nonclustered index. Used to support replication samples.</i>	rowguid		True	
	IX_Store_SalesPersonID <i>Nonclustered index.</i>	SalesPerson-ID			

PXML_Store_Demographics <i>Primary XML index.</i>	Demographi cs	xml		Primary
--	------------------	-----	--	---------

Foreign Keys

Name	Columns
FK_Store_BusinessEntity_BusinessEntityID <i>Foreign key constraint referencing Business-Entity.BusinessEntityID</i>	BusinessEntityID->[Person].[BusinessEntity].[Business-EntityID]
FK_Store_SalesPerson_SalesPersonID <i>Foreign key constraint referencing SalesPerson.SalesPersonID</i>	SalesPersonID->[Sales].[SalesPerson].[BusinessEntity-ID]

SQL Script

```

CREATE TABLE [Sales].[Store]
(
  [BusinessEntityID] [int] NOT NULL,
  [Name] [dbo].[Name] NOT NULL,
  [SalesPersonID] [int] NULL,
  [Demographics] [xml] (CONTENT [Sales].[StoreSurveySchemaCollection]) NULL,
  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_Store_rowguid]
  DEFAULT (newid()),
  [ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_Store_ModifiedDate] DEFAULT
  (getdate())
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
ALTER TABLE [Sales].[Store] ADD CONSTRAINT [PK_Store_BusinessEntityID] PRIMARY KEY
CLUSTERED ([BusinessEntityID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK_Store_rowguid] ON [Sales].[Store] ((rowguid))
ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_Store_SalesPersonID] ON [Sales].[Store] ((SalesPerson-
ID)) ON [PRIMARY]
GO
CREATE PRIMARY XML INDEX [PXML_Store_Demographics]
ON [Sales].[Store] ([Demographics])
GO
ALTER TABLE [Sales].[Store] ADD CONSTRAINT [FK_Store_BusinessEntity_BusinessEntity-
ID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [Person].[BusinessEntity] ([Business-
EntityID])
GO
ALTER TABLE [Sales].[Store] ADD CONSTRAINT [FK_Store_SalesPerson_SalesPersonID]
FOREIGN KEY ([SalesPersonID]) REFERENCES [Sales].[SalesPerson] ([BusinessEntityID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Customers (resellers) of Adventure
Works products.', 'SCHEMA', N'Sales', 'TABLE', N'Store', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Foreign key to
Customer.BusinessEntityID.', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'COLUMN',
N'BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Demographic informationg about the
store such as the number of employees, annual sales and store type.', 'SCHEMA',
N'Sales', 'TABLE', N'Store', 'COLUMN', N'Demographics'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last

```

```

updated.', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'COLUMN', N'ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Name of the store.', 'SCHEMA',
N'Sales', 'TABLE', N'Store', 'COLUMN', N'Name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ROWGUIDCOL number uniquely
identifying the record. Used to support a merge replication sample.', 'SCHEMA',
N'Sales', 'TABLE', N'Store', 'COLUMN', N'rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'ID of the sales person assigned to
the customer. Foreign key to SalesPerson.BusinessEntityID.', 'SCHEMA', N'Sales',
'TABLE', N'Store', 'COLUMN', N'SalesPersonID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'CONSTRAINT', N'DF_Store_Modified-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID()', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'CONSTRAINT', N'DF_Store_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
BusinessEntity.BusinessEntityID', 'SCHEMA', N'Sales', 'TABLE', N'Store',
'CONSTRAINT', N'FK_Store_BusinessEntity_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
SalesPerson.SalesPersonID', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'CONSTRAINT',
N'FK_Store_SalesPerson_SalesPersonID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered)
constraint', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'CONSTRAINT', N'PK_Store_-
BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'INDEX', N'AK_-
Store_rowguid'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Sales', 'TABLE', N'Store', 'INDEX', N'IX_Store_SalesPersonID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a
primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'INDEX', N'PK_-
Store_BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary XML index.', 'SCHEMA',
N'Sales', 'TABLE', N'Store', 'INDEX', N'PXML_Store_Demographics'
GO

```

Uses

[Person].[BusinessEntity]
 [Sales].[SalesPerson]
 [dbo].[Name]
 Sales
 [Sales].[StoreSurveySchemaCollection]

Used By

[Sales].[Customer]
 [Sales].[vStoreWithAddresses]
 [Sales].[vStoreWithContacts]

Project > Isrep17 > User databases > AdventureWorks > Tables > Sales.Store

[Sales].[vStoreWithDemographics]

[dbo].[ufnGetContactInformation]

 **Views**
Objects

Name
HumanResources.vEmployee <i>Employee names and addresses.</i>
HumanResources.vEmployeeDepartment <i>Returns employee name, title, and current department.</i>
HumanResources.vEmployeeDepartmentHistory <i>Returns employee name and current and previous departments.</i>
HumanResources.vJobCandidate <i>Job candidate names and resumes.</i>
HumanResources.vJobCandidateEducation <i>Displays the content from each education related element in the xml column Resume in the HumanResources.JobCandidate table. The content has been localized into French, Simplified Chinese and Thai. Some data may not display correctly unless supplemental language support is installed.</i>
HumanResources.vJobCandidateEmployment <i>Displays the content from each employment history related element in the xml column Resume in the HumanResources.JobCandidate table. The content has been localized into French, Simplified Chinese and Thai. Some data may not display correctly unless supplemental language support is installed.</i>
Person.vAdditionalContactInfo <i>Displays the contact name and content from each element in the xml column AdditionalContactInfo for that person.</i>
Person.vStateProvinceCountryRegion <i>Joins StateProvince table with CountryRegion table.</i>
Production.vProductAndDescription <i>Product names and descriptions. Product descriptions are provided in multiple languages.</i>
Production.vProductModelCatalogDescription <i>Displays the content from each element in the xml column CatalogDescription for each product in the Production.ProductModel table that has catalog data.</i>
Production.vProductModelInstructions <i>Displays the content from each element in the xml column Instructions for each product in the Production.ProductModel table that has manufacturing instructions.</i>
Purchasing.vVendorWithAddresses <i>Vendor (company) names and addresses .</i>
Purchasing.vVendorWithContacts <i>Vendor (company) names and the names of vendor employees to contact.</i>
Sales.vIndividualCustomer <i>Individual customers (names and addresses) that purchase Adventure Works Cycles products online.</i>
Sales.vPersonDemographics <i>Displays the content from each element in the xml column Demographics for each customer in the Person.Person table.</i>
Sales.vSalesPerson <i>Sales representatives (names and addresses) and their sales-related information.</i>
Sales.vSalesPersonSalesByFiscalYears <i>Uses PIVOT to return aggregated sales information for each sales representative.</i>
Sales.vStoreWithAddresses <i>Stores (including store addresses) that sell Adventure Works Cycles products to consumers.</i>
Sales.vStoreWithContacts <i>Stores (including store contacts) that sell Adventure Works Cycles products to consumers.</i>
Sales.vStoreWithDemographics <i>Stores (including demographics) that sell Adventure Works Cycles products to consumers.</i>

 **[HumanResources].[vEmployee]**
MS_Description

Employee names and addresses.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Created	13:14:55 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Name	Data Type	Max Length (Bytes)
BusinessEntityID	int	4
Title	nvarchar(8)	16
FirstName	[dbo].[Name]	100
MiddleName	[dbo].[Name]	100
LastName	[dbo].[Name]	100
Suffix	nvarchar(10)	20
JobTitle	nvarchar(50)	100
PhoneNumber	[dbo].[Phone]	50
PhoneNumberType	[dbo].[Name]	100
EmailAddress	nvarchar(50)	100
EmailPromotion	int	4
AddressLine1	nvarchar(60)	120
AddressLine2	nvarchar(60)	120
City	nvarchar(30)	60
StateProvinceName	[dbo].[Name]	100
PostalCode	nvarchar(15)	30
CountryRegionName	[dbo].[Name]	100
AdditionalContactInfo	xml	max

SQL Script

```
CREATE VIEW [HumanResources].[vEmployee]
AS
SELECT
    e.[BusinessEntityID]
```

```

,p.[Title]
,p.[FirstName]
,p.[MiddleName]
,p.[LastName]
,p.[Suffix]
,e.[JobTitle]
,pp.[PhoneNumber]
,pnt.[Name] AS [PhoneNumberType]
,ea.[EmailAddress]
,p.[EmailPromotion]
,a.[AddressLine1]
,a.[AddressLine2]
,a.[City]
,sp.[Name] AS [StateProvinceName]
,a.[PostalCode]
,cr.[Name] AS [CountryRegionName]
,p.[AdditionalContactInfo]
FROM [HumanResources].[Employee] e
INNER JOIN [Person].[Person] p
ON p.[BusinessEntityID] = e.[BusinessEntityID]
INNER JOIN [Person].[BusinessEntityAddress] bea
ON bea.[BusinessEntityID] = e.[BusinessEntityID]
INNER JOIN [Person].[Address] a
ON a.[AddressID] = bea.[AddressID]
INNER JOIN [Person].[StateProvince] sp
ON sp.[StateProvinceID] = a.[StateProvinceID]
INNER JOIN [Person].[CountryRegion] cr
ON cr.[CountryRegionCode] = sp.[CountryRegionCode]
LEFT OUTER JOIN [Person].[PersonPhone] pp
ON pp.BusinessEntityID = p.[BusinessEntityID]
LEFT OUTER JOIN [Person].[PhoneNumberType] pnt
ON pp.[PhoneNumberTypeID] = pnt.[PhoneNumberTypeID]
LEFT OUTER JOIN [Person].[EmailAddress] ea
ON p.[BusinessEntityID] = ea.[BusinessEntityID];
GO
EXEC sp_addextendedproperty N'MS_Description', N'Employee names and addresses.',
'SHEMA', N'HumanResources', 'VIEW', N'vEmployee', NULL, NULL
GO

```

Uses

[HumanResources].[Employee]
[Person].[Address]
[Person].[BusinessEntityAddress]
[Person].[CountryRegion]
[Person].[EmailAddress]
[Person].[Person]
[Person].[PersonPhone]
[Person].[PhoneNumberType]
[Person].[StateProvince]
[dbo].[Name]
[dbo].[Phone]
HumanResources

 **[HumanResources].[vEmployeeDepartment]**
MS_Description

Returns employee name, title, and current department.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Created	13:14:55 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Name	Data Type	Max Length (Bytes)
BusinessEntityID	int	4
Title	nvarchar(8)	16
FirstName	[dbo].[Name]	100
MiddleName	[dbo].[Name]	100
LastName	[dbo].[Name]	100
Suffix	nvarchar(10)	20
JobTitle	nvarchar(50)	100
Department	[dbo].[Name]	100
GroupName	[dbo].[Name]	100
StartDate	date	3

SQL Script

```
CREATE VIEW [HumanResources].[vEmployeeDepartment]
AS
SELECT
    e.[BusinessEntityID]
    ,p.[Title]
    ,p.[FirstName]
    ,p.[MiddleName]
    ,p.[LastName]
    ,p.[Suffix]
    ,e.[JobTitle]
    ,d.[Name] AS [Department]
    ,d.[GroupName]
    ,edh.[StartDate]
FROM [HumanResources].[Employee] e
```

```
INNER JOIN [Person].[Person] p
ON p.[BusinessEntityID] = e.[BusinessEntityID]
INNER JOIN [HumanResources].[EmployeeDepartmentHistory] edh
ON e.[BusinessEntityID] = edh.[BusinessEntityID]
INNER JOIN [HumanResources].[Department] d
ON edh.[DepartmentID] = d.[DepartmentID]
WHERE edh.EndDate IS NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Returns employee name, title, and
current department.', 'SCHEMA', N'HumanResources', 'VIEW', N'vEmployeeDepartment',
NULL, NULL
GO
```

Uses

[HumanResources].[Department]
[HumanResources].[Employee]
[HumanResources].[EmployeeDepartmentHistory]
[Person].[Person]
[dbo].[Name]
HumanResources

[HumanResources].[vEmployeeDepartmentHistory]

MS_Description

Returns employee name and current and previous departments.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Created	13:14:55 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Name	Data Type	Max Length (Bytes)
BusinessEntityID	int	4
Title	nvarchar(8)	16
FirstName	[dbo].[Name]	100
MiddleName	[dbo].[Name]	100
LastName	[dbo].[Name]	100
Suffix	nvarchar(10)	20
Shift	[dbo].[Name]	100
Department	[dbo].[Name]	100
GroupName	[dbo].[Name]	100
StartDate	date	3
EndDate	date	3

SQL Script

```
CREATE VIEW [HumanResources].[vEmployeeDepartmentHistory]
AS
SELECT
    e.[BusinessEntityID]
    ,p.[Title]
    ,p.[FirstName]
    ,p.[MiddleName]
    ,p.[LastName]
    ,p.[Suffix]
    ,s.[Name] AS [Shift]
    ,d.[Name] AS [Department]
    ,d.[GroupName]
    ,edh.[StartDate]
```

```
,edh.[EndDate]
FROM [HumanResources].[Employee] e
INNER JOIN [Person].[Person] p
ON p.[BusinessEntityID] = e.[BusinessEntityID]
INNER JOIN [HumanResources].[EmployeeDepartmentHistory] edh
ON e.[BusinessEntityID] = edh.[BusinessEntityID]
INNER JOIN [HumanResources].[Department] d
ON edh.[DepartmentID] = d.[DepartmentID]
INNER JOIN [HumanResources].[Shift] s
ON s.[ShiftID] = edh.[ShiftID];
GO
EXEC sp_addextendedproperty N'MS_Description', N'Returns employee name and current
and previous departments.', 'SCHEMA', N'HumanResources', 'VIEW', N'vEmployee-
DepartmentHistory', NULL, NULL
GO
```

Uses

[HumanResources].[Department]
[HumanResources].[Employee]
[HumanResources].[EmployeeDepartmentHistory]
[HumanResources].[Shift]
[Person].[Person]
[dbo].[Name]
HumanResources


[HumanResources].[vJobCandidate]
MS_Description

Job candidate names and resumes.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Created	13:14:55 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Name	Data Type	Max Length (Bytes)	Identity
JobCandidateID	int	4	0 - 0
BusinessEntityID	int	4	
Name.Prefix	nvarchar(30)	60	
Name.First	nvarchar(30)	60	
Name.Middle	nvarchar(30)	60	
Name.Last	nvarchar(30)	60	
Name.Suffix	nvarchar(30)	60	
Skills	nvarchar(max)	max	
Addr.Type	nvarchar(30)	60	
Addr.Loc.CountryRegion	nvarchar(100)	200	
Addr.Loc.State	nvarchar(100)	200	
Addr.Loc.City	nvarchar(100)	200	
Addr.PostalCode	nvarchar(20)	40	
EMail	nvarchar(max)	max	
WebSite	nvarchar(max)	max	
ModifiedDate	datetime	8	

SQL Script

```
CREATE VIEW [HumanResources].[vJobCandidate]
AS
SELECT
    jc.[JobCandidateID]
    ,jc.[BusinessEntityID]
    , [Resume].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
```

```

        (/Resume/Name/Name.Prefix)[1]', 'nvarchar(30)') AS [Name.Prefix]
    , [Resume].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
        (/Resume/Name/Name.First)[1]', 'nvarchar(30)') AS [Name.First]
    , [Resume].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
        (/Resume/Name/Name.Middle)[1]', 'nvarchar(30)') AS [Name.Middle]
    , [Resume].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
        (/Resume/Name/Name.Last)[1]', 'nvarchar(30)') AS [Name.Last]
    , [Resume].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
        (/Resume/Name/Name.Suffix)[1]', 'nvarchar(30)') AS [Name.Suffix]
    , [Resume].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
        (/Resume/Skills)[1]', 'nvarchar(max)') AS [Skills]
    , [Resume].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
        (Address/Addr.Type)[1]', 'nvarchar(30)') AS [Addr.Type]
    , [Resume].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
        (Address/Addr.Location/Location/Loc.CountryRegion)[1]', 'nvarchar(100)') AS
[Addr.Loc.CountryRegion]
    , [Resume].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
        (Address/Addr.Location/Location/Loc.State)[1]', 'nvarchar(100)') AS
[Addr.Loc.State]
    , [Resume].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
        (Address/Addr.Location/Location/Loc.City)[1]', 'nvarchar(100)') AS
[Addr.Loc.City]
    , [Resume].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
        (Address/Addr.PostalCode)[1]', 'nvarchar(20)') AS [Addr.PostalCode]
    , [Resume].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
        (/Resume/EMail)[1]', 'nvarchar(max)') AS [EMail]
    , [Resume].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
        (/Resume/WebSite)[1]', 'nvarchar(max)') AS [WebSite]
    , jc.[ModifiedDate]
FROM [HumanResources].[JobCandidate] jc
CROSS APPLY jc.[Resume].nodes(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
        /Resume') AS Resume(ref);
GO
EXEC sp_addextendedproperty N'MS_Description', N'Job candidate names and resumes.',
'SHEMA', N'HumanResources', 'VIEW', N'vJobCandidate', NULL, NULL
GO

```

Uses

[HumanResources].[JobCandidate]
HumanResources


[HumanResources].[vJobCandidateEducation]

MS_Description

Displays the content from each education related element in the xml column Resume in the HumanResources.JobCandidate table. The content has been localized into French, Simplified Chinese and Thai. Some data may not display correctly unless supplemental language support is installed.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Created	13:14:55 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Name	Data Type	Max Length (Bytes)	Identity
JobCandidateID	int	4	0 - 0
Edu.Level	nvarchar(max)	max	
Edu.StartDate	datetime	8	
Edu.EndDate	datetime	8	
Edu.Degree	nvarchar(50)	100	
Edu.Major	nvarchar(50)	100	
Edu.Minor	nvarchar(50)	100	
Edu.GPA	nvarchar(5)	10	
Edu.GPAScale	nvarchar(5)	10	
Edu.School	nvarchar(100)	200	
Edu.Loc.CountryRegion	nvarchar(100)	200	
Edu.Loc.State	nvarchar(100)	200	
Edu.Loc.City	nvarchar(100)	200	

SQL Script

```
CREATE VIEW [HumanResources].[vJobCandidateEducation]
AS
SELECT
    jc.[JobCandidateID]
    , [Education].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
    (Edu.Level) [1]', 'nvarchar(max)') AS [Edu.Level]
    , CONVERT(datetime, REPLACE([Education].ref.value(N'declare default element
namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
```

```

        (Edu.StartDate) [1]', 'nvarchar(20)' , 'Z', ''), 101) AS [Edu.StartDate]
    , CONVERT(datetime, REPLACE([Education].ref.value(N'declare default element
namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
    (Edu.EndDate) [1]', 'nvarchar(20)' , 'Z', ''), 101) AS [Edu.EndDate]
    , [Education].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
    (Edu.Degree) [1]', 'nvarchar(50)') AS [Edu.Degree]
    , [Education].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
    (Edu.Major) [1]', 'nvarchar(50)') AS [Edu.Major]
    , [Education].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
    (Edu.Minor) [1]', 'nvarchar(50)') AS [Edu.Minor]
    , [Education].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
    (Edu.GPA) [1]', 'nvarchar(5)') AS [Edu.GPA]
    , [Education].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
    (Edu.GPAScale) [1]', 'nvarchar(5)') AS [Edu.GPAScale]
    , [Education].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
    (Edu.School) [1]', 'nvarchar(100)') AS [Edu.School]
    , [Education].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
    (Edu.Location/Location/Loc.CountryRegion) [1]', 'nvarchar(100)') AS
[Edu.Loc.CountryRegion]
    , [Education].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
    (Edu.Location/Location/Loc.State) [1]', 'nvarchar(100)') AS [Edu.Loc.State]
    , [Education].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
    (Edu.Location/Location/Loc.City) [1]', 'nvarchar(100)') AS [Edu.Loc.City]
FROM [HumanResources].[JobCandidate] jc
CROSS APPLY jc.[Resume].nodes(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
/Resume/Education') AS [Education] (ref);
GO
EXEC sp_addextendedproperty N'MS_Description', N'Displays the content from each
education related element in the xml column Resume in the HumanResources.Job-
Candidate table. The content has been localized into French, Simplified Chinese and
Thai. Some data may not display correctly unless supplemental language support is
installed.', 'SCHEMA', N'HumanResources', 'VIEW', N'vJobCandidateEducation', NULL,
NULL
GO

```

Uses

[HumanResources].[JobCandidate]
HumanResources

[HumanResources].[vJobCandidateEmployment]

MS_Description

Displays the content from each employment history related element in the xml column Resume in the Human-Resources.JobCandidate table. The content has been localized into French, Simplified Chinese and Thai. Some data may not display correctly unless supplemental language support is installed.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Created	13:14:55 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Name	Data Type	Max Length (Bytes)	Identity
JobCandidateID	int	4	0 - 0
Emp.StartDate	datetime	8	
Emp.EndDate	datetime	8	
Emp.OrgName	nvarchar(100)	200	
Emp.JobTitle	nvarchar(100)	200	
Emp.Responsibility	nvarchar(max)	max	
Emp.FunctionCategory	nvarchar(max)	max	
Emp.IndustryCategory	nvarchar(max)	max	
Emp.Loc.CountryRegion	nvarchar(max)	max	
Emp.Loc.State	nvarchar(max)	max	
Emp.Loc.City	nvarchar(max)	max	

SQL Script

```
CREATE VIEW [HumanResources].[vJobCandidateEmployment]
AS
SELECT
    jc.[JobCandidateID]
    , CONVERT(datetime, REPLACE([Employment].ref.value(N'declare default element
namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
(Emp.StartDate)[1]', 'nvarchar(20)') , 'Z', ''), 101) AS [Emp.StartDate]
    , CONVERT(datetime, REPLACE([Employment].ref.value(N'declare default element
namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
(Emp.EndDate)[1]', 'nvarchar(20)') , 'Z', ''), 101) AS [Emp.EndDate]
    , [Employment].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
```

```
(Emp.OrgName) [1]', 'nvarchar(100)') AS [Emp.OrgName]
, [Employment].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
(Emp.JobTitle) [1]', 'nvarchar(100)') AS [Emp.JobTitle]
, [Employment].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
(Emp.Responsibility) [1]', 'nvarchar(max)') AS [Emp.Responsibility]
, [Employment].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
(Emp.FunctionCategory) [1]', 'nvarchar(max)') AS [Emp.FunctionCategory]
, [Employment].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
(Emp.IndustryCategory) [1]', 'nvarchar(max)') AS [Emp.IndustryCategory]
, [Employment].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
(Emp.Location/Location/Loc.CountryRegion) [1]', 'nvarchar(max)') AS
[Emp.Loc.CountryRegion]
, [Employment].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
(Emp.Location/Location/Loc.State) [1]', 'nvarchar(max)') AS [Emp.Loc.State]
, [Employment].ref.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
(Emp.Location/Location/Loc.City) [1]', 'nvarchar(max)') AS [Emp.Loc.City]
FROM [HumanResources].[JobCandidate] jc
CROSS APPLY jc.[Resume].nodes(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";
/Resume/Employment') AS Employment(ref);
GO
EXEC sp_addextendedproperty N'MS_Description', N'Displays the content from each
employment history related element in the xml column Resume in the Human-
Resources.JobCandidate table. The content has been localized into French, Simplified
Chinese and Thai. Some data may not display correctly unless supplemental language
support is installed.', 'SCHEMA', N'HumanResources', 'VIEW', N'vJobCandidate-
Employment', NULL, NULL
GO
```

Uses

[HumanResources].[JobCandidate]
HumanResources

 **[Person].[vAdditionalContactInfo]**
MS_Description

Displays the contact name and content from each element in the xml column AdditionalContactInfo for that person.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Created	13:14:55 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Name	Data Type	Max Length (Bytes)
BusinessEntityID	int	4
FirstName	[dbo].[Name]	100
MiddleName	[dbo].[Name]	100
LastName	[dbo].[Name]	100
TelephoneNumber	nvarchar(50)	100
TelephoneSpecialInstructions	nvarchar(max)	max
Street	nvarchar(50)	100
City	nvarchar(50)	100
StateProvince	nvarchar(50)	100
PostalCode	nvarchar(50)	100
CountryRegion	nvarchar(50)	100
HomeAddressSpecialInstructions	nvarchar(max)	max
EMailAddress	nvarchar(128)	256
EMailSpecialInstructions	nvarchar(max)	max
EMailTelephoneNumber	nvarchar(50)	100
rowguid	uniqueidentifier	16
ModifiedDate	datetime	8

SQL Script

```
CREATE VIEW [Person].[vAdditionalContactInfo]
AS
SELECT
    [BusinessEntityID]
    , [FirstName]
```

```

, [MiddleName]
, [LastName]
, [ContactInfo].ref.value(N'declare namespace
ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";
declare namespace
act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";
(act:telephoneNumber)[1]/act:number', 'nvarchar(50)') AS [TelephoneNumber]
, LTRIM(RTRIM([ContactInfo].ref.value(N'declare namespace
ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";
declare namespace
act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";
(act:telephoneNumber/act:SpecialInstructions/text())[1]', 'nvarchar(max)'))
AS [TelephoneSpecialInstructions]
, [ContactInfo].ref.value(N'declare namespace
ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";
declare namespace
act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";
(act:homePostalAddress/act:Street)[1]', 'nvarchar(50)') AS [Street]
, [ContactInfo].ref.value(N'declare namespace
ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";
declare namespace
act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";
(act:homePostalAddress/act:City)[1]', 'nvarchar(50)') AS [City]
, [ContactInfo].ref.value(N'declare namespace
ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";
declare namespace
act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";
(act:homePostalAddress/act:StateProvince)[1]', 'nvarchar(50)') AS [State-
Province]
, [ContactInfo].ref.value(N'declare namespace
ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";
declare namespace
act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";
(act:homePostalAddress/act:PostalCode)[1]', 'nvarchar(50)') AS [PostalCode]
, [ContactInfo].ref.value(N'declare namespace
ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";
declare namespace
act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";
(act:homePostalAddress/act:CountryRegion)[1]', 'nvarchar(50)') AS [Country-
Region]
, [ContactInfo].ref.value(N'declare namespace
ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";
declare namespace
act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";
(act:homePostalAddress/act:SpecialInstructions/text())[1]', 'nvarchar(max)')
AS [HomeAddressSpecialInstructions]
, [ContactInfo].ref.value(N'declare namespace
ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";
declare namespace
act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";
(act:eMail/act:EmailAddress)[1]', 'nvarchar(128)') AS [EmailAddress]
, LTRIM(RTRIM([ContactInfo].ref.value(N'declare namespace
ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";
declare namespace
act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";
(act:eMail/act:SpecialInstructions/text())[1]', 'nvarchar(max)'))
AS [Email-
SpecialInstructions]
, [ContactInfo].ref.value(N'declare namespace
ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";
declare namespace
act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";
(act:eMail/act:SpecialInstructions/act:telephoneNumber/act:number)[1]',
'nvarchar(50)') AS [EmailTelephoneNumber]
, [rowguid]

```

```
    , [ModifiedDate]
FROM [Person].[Person]
OUTER APPLY [AdditionalContactInfo].nodes(
    'declare namespace ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";
    /ci:AdditionalContactInfo') AS ContactInfo(ref)
WHERE [AdditionalContactInfo] IS NOT NULL;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Displays the contact name and content from each element in the xml column AdditionalContactInfo for that person.',
'SHEMA', N'Person', 'VIEW', N'vAdditionalContactInfo', NULL, NULL
GO
```

Uses

[Person].[Person]

[dbo].[Name]

Person

 **[Person].[vStateProvinceCountryRegion]****MS_Description**

Joins StateProvince table with CountryRegion table.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Schema Bound	True
Created	13:14:55 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)
	StateProvinceID	int	4
	StateProvinceCode	nchar(3)	6
	IsOnlyStateProvinceFlag	[dbo].[Flag]	1
	StateProvinceName	[dbo].[Name]	100
	TerritoryID	int	4
	CountryRegionCode	nvarchar(3)	6
	CountryRegionName	[dbo].[Name]	100

Indexes

Key	Name	Key Columns	Unique
	IX_vStateProvinceCountryRegion <i>Clustered index on the view vStateProvinceCountryRegion.</i>	StateProvince-ID, Country-RegionCode	True

SQL Script

```
CREATE VIEW [Person].[vStateProvinceCountryRegion]
WITH SCHEMABINDING
AS
SELECT
    sp.[StateProvinceID]
    , sp.[StateProvinceCode]
    , sp.[IsOnlyStateProvinceFlag]
    , sp.[Name] AS [StateProvinceName]
```

```
,sp.[TerritoryID]
,cr.[CountryRegionCode]
,cr.[Name] AS [CountryRegionName]
FROM [Person].[StateProvince] sp
INNER JOIN [Person].[CountryRegion] cr
ON sp.[CountryRegionCode] = cr.[CountryRegionCode];
GO
CREATE UNIQUE CLUSTERED INDEX [IX_vStateProvinceCountryRegion] ON [Person].[vState-
ProvinceCountryRegion] ([StateProvinceID], [CountryRegionCode]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Joins StateProvince table with
CountryRegion table.', 'SCHEMA', N'Person', 'VIEW', N'vStateProvinceCountryRegion',
NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index on the view vState-
ProvinceCountryRegion.', 'SCHEMA', N'Person', 'VIEW', N'vStateProvinceCountry-
Region', 'INDEX', N'IX_vStateProvinceCountryRegion'
GO
```

Uses

[Person].[CountryRegion]
[Person].[StateProvince]
[dbo].[Flag]
[dbo].[Name]
Person

 **[Production].[vProductAndDescription]**
MS_Description

Product names and descriptions. Product descriptions are provided in multiple languages.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Schema Bound	True
Created	13:14:55 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Key	Name	Data Type	Max Length (Bytes)
	ProductID	int	4
	Name	[dbo].[Name]	100
	ProductModel	[dbo].[Name]	100
	CultureID	nchar(6)	12
	Description	nvarchar(400)	800

Indexes

Key	Name	Key Columns	Unique
	IX_vProductAndDescription <i>Clustered index on the view vProductAndDescription.</i>	CultureID, ProductID	True

SQL Script

```
CREATE VIEW [Production].[vProductAndDescription]
WITH SCHEMABINDING
AS
-- View (indexed or standard) to display products and product descriptions by
language.
SELECT
    p.[ProductID]
    ,p.[Name]
    ,pm.[Name] AS [ProductModel]
    ,pmx.[CultureID]
    ,pd.[Description]
FROM [Production].[Product] p
```

```
INNER JOIN [Production].[ProductModel] pm
ON p.[ProductModelID] = pm.[ProductModelID]
INNER JOIN [Production].[ProductModelProductDescriptionCulture] pmx
ON pm.[ProductModelID] = pmx.[ProductModelID]
INNER JOIN [Production].[ProductDescription] pd
ON pmx.[ProductDescriptionID] = pd.[ProductDescriptionID];
GO
CREATE UNIQUE CLUSTERED INDEX [IX_vProductAndDescription] ON [Production].[vProduct-
AndDescription] ([CultureID], [ProductID]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product names and descriptions.
Product descriptions are provided in multiple languages.', 'SCHEMA', N'Production',
'VIEW', N'vProductAndDescription', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index on the view v-
ProductAndDescription.', 'SCHEMA', N'Production', 'VIEW', N'vProductAndDescription',
'INDEX', N'IX_vProductAndDescription'
GO
```

Uses

[Production].[Product]
[Production].[ProductDescription]
[Production].[ProductModel]
[Production].[ProductModelProductDescriptionCulture]
[dbo].[Name]
Production

[Production].[vProductModelCatalogDescription]

MS_Description

Displays the content from each element in the xml column CatalogDescription for each product in the Production.ProductModel table that has catalog data.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Created	13:14:55 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Name	Data Type	Max Length (Bytes)	Identity
ProductModelID	int	4	0 - 0
Name	[dbo].[Name]	100	
Summary	nvarchar(max)	max	
Manufacturer	nvarchar(max)	max	
Copyright	nvarchar(30)	60	
ProductURL	nvarchar(256)	512	
WarrantyPeriod	nvarchar(256)	512	
WarrantyDescription	nvarchar(256)	512	
NoOfYears	nvarchar(256)	512	
MaintenanceDescription	nvarchar(256)	512	
Wheel	nvarchar(256)	512	
Saddle	nvarchar(256)	512	
Pedal	nvarchar(256)	512	
BikeFrame	nvarchar(max)	max	
Crankset	nvarchar(256)	512	
PictureAngle	nvarchar(256)	512	
PictureSize	nvarchar(256)	512	
ProductPhotoID	nvarchar(256)	512	
Material	nvarchar(256)	512	
Color	nvarchar(256)	512	
ProductLine	nvarchar(256)	512	
Style	nvarchar(256)	512	
RiderExperience	nvarchar(1024)	2048	
rowguid	uniqueidentifier	16	

ModifiedDate	datetime	8	
--------------	----------	---	--

SQL Script

```

CREATE VIEW [Production].[vProductModelCatalogDescription]
AS
SELECT
    [ProductModelID]
    , [Name]
    , [CatalogDescription].value(N'declare namespace
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
        declare namespace html="http://www.w3.org/1999/xhtml";
        (/p1:ProductDescription/p1:Summary/html:p) [1]', 'nvarchar(max)') AS
[Summary]
    , [CatalogDescription].value(N'declare namespace
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
        (/p1:ProductDescription/p1:Manufacturer/p1:Name) [1]', 'nvarchar(max)') AS
[Manufacturer]
    , [CatalogDescription].value(N'declare namespace
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
        (/p1:ProductDescription/p1:Manufacturer/p1:Copyright) [1]', 'nvarchar(30)')
AS [Copyright]
    , [CatalogDescription].value(N'declare namespace
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
        (/p1:ProductDescription/p1:Manufacturer/p1:ProductURL) [1]', 'nvarchar(256)')
AS [ProductURL]
    , [CatalogDescription].value(N'declare namespace
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
        declare namespace
wm="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelWarr-
AndMain";
        (/p1:ProductDescription/p1:Features/wm:Warranty/wm:WarrantyPeriod) [1]',
'nvarchar(256)') AS [WarrantyPeriod]
    , [CatalogDescription].value(N'declare namespace
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
        declare namespace
wm="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelWarr-
AndMain";
        (/p1:ProductDescription/p1:Features/wm:Warranty/wm:Description) [1]',
'nvarchar(256)') AS [WarrantyDescription]
    , [CatalogDescription].value(N'declare namespace
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
        declare namespace
wm="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelWarr-
AndMain";
        (/p1:ProductDescription/p1:Features/wm:Maintenance/wm:NoOfYears) [1]',
'nvarchar(256)') AS [NoOfYears]
    , [CatalogDescription].value(N'declare namespace
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
        declare namespace
wm="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelWarr-
AndMain";
        (/p1:ProductDescription/p1:Features/wm:Maintenance/wm:Description) [1]',
'nvarchar(256)') AS [MaintenanceDescription]

```

```
, [CatalogDescription].value(N'declare namespace
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
    declare namespace wf="http://www.adventure-works.com/schemas/OtherFeatures";
    (/p1:ProductDescription/p1:Features/wf:wheel) [1]', 'nvarchar(256)') AS
[Wheel]
, [CatalogDescription].value(N'declare namespace
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
    declare namespace wf="http://www.adventure-works.com/schemas/OtherFeatures";
    (/p1:ProductDescription/p1:Features/wf:saddle) [1]', 'nvarchar(256)') AS
[Saddle]
, [CatalogDescription].value(N'declare namespace
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
    declare namespace wf="http://www.adventure-works.com/schemas/OtherFeatures";
    (/p1:ProductDescription/p1:Features/wf:pedal) [1]', 'nvarchar(256)') AS
[Pedal]
, [CatalogDescription].value(N'declare namespace
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
    declare namespace wf="http://www.adventure-works.com/schemas/OtherFeatures";
    (/p1:ProductDescription/p1:Features/wf:BikeFrame) [1]', 'nvarchar(max)') AS
[BikeFrame]
, [CatalogDescription].value(N'declare namespace
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
    declare namespace wf="http://www.adventure-works.com/schemas/OtherFeatures";
    (/p1:ProductDescription/p1:Features/wf:crankset) [1]', 'nvarchar(256)') AS
[Crankset]
, [CatalogDescription].value(N'declare namespace
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
    (/p1:ProductDescription/p1:Picture/p1:Angle) [1]', 'nvarchar(256)') AS
[PictureAngle]
, [CatalogDescription].value(N'declare namespace
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
    (/p1:ProductDescription/p1:Picture/p1:Size) [1]', 'nvarchar(256)') AS
[PictureSize]
, [CatalogDescription].value(N'declare namespace
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
    (/p1:ProductDescription/p1:Picture/p1:ProductPhotoID) [1]', 'nvarchar(256)')
AS [ProductPhotoID]
, [CatalogDescription].value(N'declare namespace
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
    (/p1:ProductDescription/p1:Specifications/Material) [1]', 'nvarchar(256)') AS
[Material]
, [CatalogDescription].value(N'declare namespace
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
    (/p1:ProductDescription/p1:Specifications/Color) [1]', 'nvarchar(256)') AS
[Color]
, [CatalogDescription].value(N'declare namespace
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
    (/p1:ProductDescription/p1:Specifications/ProductLine) [1]', 'nvarchar(256)')
AS [ProductLine]
, [CatalogDescription].value(N'declare namespace
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
    (/p1:ProductDescription/p1:Specifications/Style) [1]', 'nvarchar(256)') AS
[Style]
, [CatalogDescription].value(N'declare namespace
```

```
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-  
Description";  
    (/p1:ProductDescription/p1:Specifications/RiderExperience) [1]',  
'nvarchar(1024)') AS [RiderExperience]  
    , [rowguid]  
    , [ModifiedDate]  
FROM [Production].[ProductModel]  
WHERE [CatalogDescription] IS NOT NULL;  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Displays the content from each  
element in the xml column CatalogDescription for each product in the  
Production.ProductModel table that has catalog data.', 'SCHEMA', N'Production',  
'VIEW', N'vProductModelCatalogDescription', NULL, NULL  
GO
```

Uses

[Production].[ProductModel]
[dbo].[Name]
Production

 **[Production].[vProductModelInstructions]**
MS_Description

Displays the content from each element in the xml column Instructions for each product in the Production.Product-Model table that has manufacturing instructions.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Created	13:14:55 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Name	Data Type	Max Length (Bytes)	Identity
ProductModelID	int	4	0 - 0
Name	[dbo].[Name]	100	
Instructions	nvarchar(max)	max	
LocationID	int	4	
SetupHours	decimal(9,4)	5	
MachineHours	decimal(9,4)	5	
LaborHours	decimal(9,4)	5	
LotSize	int	4	
Step	nvarchar(1024)	2048	
rowguid	uniqueidentifier	16	
ModifiedDate	datetime	8	

SQL Script

```
CREATE VIEW [Production].[vProductModelInstructions]
AS
SELECT
    [ProductModelID]
    , [Name]
    , [Instructions].value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelManu-
Instructions";
    (/root/text())[1]', 'nvarchar(max)') AS [Instructions]
    , [MfgInstructions].ref.value('@LocationID[1]', 'int') AS [LocationID]
    , [MfgInstructions].ref.value('@SetupHours[1]', 'decimal(9, 4)') AS [SetupHours]
    , [MfgInstructions].ref.value('@MachineHours[1]', 'decimal(9, 4)') AS [Machine-
Hours]
```

```
, [MfgInstructions].ref.value('@LaborHours[1]', 'decimal(9, 4)') AS [LaborHours]
, [MfgInstructions].ref.value('@LotSize[1]', 'int') AS [LotSize]
, [Steps].ref.value('string(.)[1]', 'nvarchar(1024)') AS [Step]
, [rowguid]
, [ModifiedDate]
FROM [Production].[ProductModel]
CROSS APPLY [Instructions].nodes(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelManu-
Instructions";
    /root/Location') MfgInstructions(ref)
CROSS APPLY [MfgInstructions].ref.nodes('declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelManu-
Instructions";
    step') Steps(ref);
GO
EXEC sp_addextendedproperty N'MS_Description', N'Displays the content from each
element in the xml column Instructions for each product in the Production.Product-
Model table that has manufacturing instructions.', 'SCHEMA', N'Production', 'VIEW',
N'vProductModelInstructions', NULL, NULL
GO
```

Uses

[Production].[ProductModel]
[dbo].[Name]
Production

 **[Purchasing].[vVendorWithAddresses]**
MS_Description

Vendor (company) names and addresses .

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Created	13:14:55 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Name	Data Type	Max Length (Bytes)
BusinessEntityID	int	4
Name	[dbo].[Name]	100
AddressType	[dbo].[Name]	100
AddressLine1	nvarchar(60)	120
AddressLine2	nvarchar(60)	120
City	nvarchar(30)	60
StateProvinceName	[dbo].[Name]	100
PostalCode	nvarchar(15)	30
CountryRegionName	[dbo].[Name]	100

SQL Script

```

CREATE VIEW [Purchasing].[vVendorWithAddresses] AS
SELECT
    v.[BusinessEntityID]
    ,v.[Name]
    ,at.[Name] AS [AddressType]
    ,a.[AddressLine1]
    ,a.[AddressLine2]
    ,a.[City]
    ,sp.[Name] AS [StateProvinceName]
    ,a.[PostalCode]
    ,cr.[Name] AS [CountryRegionName]
FROM [Purchasing].[Vendor] v
INNER JOIN [Person].[BusinessEntityAddress] bea
ON bea.[BusinessEntityID] = v.[BusinessEntityID]
INNER JOIN [Person].[Address] a
ON a.[AddressID] = bea.[AddressID]

```

```
INNER JOIN [Person].[StateProvince] sp
ON sp.[StateProvinceID] = a.[StateProvinceID]
INNER JOIN [Person].[CountryRegion] cr
ON cr.[CountryRegionCode] = sp.[CountryRegionCode]
INNER JOIN [Person].[AddressType] at
ON at.[AddressTypeID] = bea.[AddressTypeID];
GO
EXEC sp_addextendedproperty N'MS_Description', N'Vendor (company) names and
addresses .', 'SCHEMA', N'Purchasing', 'VIEW', N'vVendorWithAddresses', NULL, NULL
GO
```

Uses

[Person].[Address]
[Person].[AddressType]
[Person].[BusinessEntityAddress]
[Person].[CountryRegion]
[Person].[StateProvince]
[Purchasing].[Vendor]
[dbo].[Name]
Purchasing

 **[Purchasing].[vVendorWithContacts]**
MS_Description

Vendor (company) names and the names of vendor employees to contact.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Created	13:14:55 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Name	Data Type	Max Length (Bytes)
BusinessEntityID	int	4
Name	[dbo].[Name]	100
ContactType	[dbo].[Name]	100
Title	nvarchar(8)	16
FirstName	[dbo].[Name]	100
MiddleName	[dbo].[Name]	100
LastName	[dbo].[Name]	100
Suffix	nvarchar(10)	20
PhoneNumber	[dbo].[Phone]	50
PhoneNumberType	[dbo].[Name]	100
EmailAddress	nvarchar(50)	100
EmailPromotion	int	4

SQL Script

```
CREATE VIEW [Purchasing].[vVendorWithContacts] AS
SELECT
    v.[BusinessEntityID]
    ,v.[Name]
    ,ct.[Name] AS [ContactType]
    ,p.[Title]
    ,p.[FirstName]
    ,p.[MiddleName]
    ,p.[LastName]
    ,p.[Suffix]
    ,pp.[PhoneNumber]
    ,pnt.[Name] AS [PhoneNumberType]
```

```

,ea.[EmailAddress]
,p.[EmailPromotion]
FROM [Purchasing].[Vendor] v
INNER JOIN [Person].[BusinessEntityContact] bec
ON bec.[BusinessEntityID] = v.[BusinessEntityID]
INNER JOIN [Person].[ContactType] ct
ON ct.[ContactTypeID] = bec.[ContactTypeID]
INNER JOIN [Person].[Person] p
ON p.[BusinessEntityID] = bec.[PersonID]
LEFT OUTER JOIN [Person].[EmailAddress] ea
ON ea.[BusinessEntityID] = p.[BusinessEntityID]
LEFT OUTER JOIN [Person].[PersonPhone] pp
ON pp.[BusinessEntityID] = p.[BusinessEntityID]
LEFT OUTER JOIN [Person].[PhoneNumberType] pnt
ON pnt.[PhoneNumberTypeID] = pp.[PhoneNumberTypeID];
GO
EXEC sp_addextendedproperty N'MS_Description', N'Vendor (company) names and the
names of vendor employees to contact.', 'SCHEMA', N'Purchasing', 'VIEW', N'vVendor-
WithContacts', NULL, NULL
GO

```

Uses

[Person].[BusinessEntityContact]
[Person].[ContactType]
[Person].[EmailAddress]
[Person].[Person]
[Person].[PersonPhone]
[Person].[PhoneNumberType]
[Purchasing].[Vendor]
[dbo].[Name]
[dbo].[Phone]
Purchasing

 **[Sales].[vIndividualCustomer]**
MS_Description

Individual customers (names and addresses) that purchase Adventure Works Cycles products online.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Created	13:14:55 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Name	Data Type	Max Length (Bytes)
BusinessEntityID	int	4
Title	nvarchar(8)	16
FirstName	[dbo].[Name]	100
MiddleName	[dbo].[Name]	100
LastName	[dbo].[Name]	100
Suffix	nvarchar(10)	20
PhoneNumber	[dbo].[Phone]	50
PhoneNumberType	[dbo].[Name]	100
EmailAddress	nvarchar(50)	100
EmailPromotion	int	4
AddressType	[dbo].[Name]	100
AddressLine1	nvarchar(60)	120
AddressLine2	nvarchar(60)	120
City	nvarchar(30)	60
StateProvinceName	[dbo].[Name]	100
PostalCode	nvarchar(15)	30
CountryRegionName	[dbo].[Name]	100
Demographics	xml	max

SQL Script

```
CREATE VIEW [Sales].[vIndividualCustomer]
AS
SELECT
    p.[BusinessEntityID]
```

```

,p.[Title]
,p.[FirstName]
,p.[MiddleName]
,p.[LastName]
,p.[Suffix]
,pp.[PhoneNumber]
,pnt.[Name] AS [PhoneNumberType]
,ea.[EmailAddress]
,p.[EmailPromotion]
,at.[Name] AS [AddressType]
,a.[AddressLine1]
,a.[AddressLine2]
,a.[City]
,[StateProvinceName] = sp.[Name]
,a.[PostalCode]
,[CountryRegionName] = cr.[Name]
,p.[Demographics]
FROM [Person].[Person] p
INNER JOIN [Person].[BusinessEntityAddress] bea
ON bea.[BusinessEntityID] = p.[BusinessEntityID]
INNER JOIN [Person].[Address] a
ON a.[AddressID] = bea.[AddressID]
INNER JOIN [Person].[StateProvince] sp
ON sp.[StateProvinceID] = a.[StateProvinceID]
INNER JOIN [Person].[CountryRegion] cr
ON cr.[CountryRegionCode] = sp.[CountryRegionCode]
INNER JOIN [Person].[AddressType] at
ON at.[AddressTypeID] = bea.[AddressTypeID]
INNER JOIN [Sales].[Customer] c
ON c.[PersonID] = p.[BusinessEntityID]
LEFT OUTER JOIN [Person].[EmailAddress] ea
ON ea.[BusinessEntityID] = p.[BusinessEntityID]
LEFT OUTER JOIN [Person].[PersonPhone] pp
ON pp.[BusinessEntityID] = p.[BusinessEntityID]
LEFT OUTER JOIN [Person].[PhoneNumberType] pnt
ON pnt.[PhoneNumberTypeID] = pp.[PhoneNumberTypeID]
WHERE c.StoreID IS NULL;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Individual customers (names and
addresses) that purchase Adventure Works Cycles products online.', 'SCHEMA',
N'Sales', 'VIEW', N'vIndividualCustomer', NULL, NULL
GO

```

Uses

[Person].[Address]
[Person].[AddressType]
[Person].[BusinessEntityAddress]
[Person].[CountryRegion]
[Person].[EmailAddress]
[Person].[Person]
[Person].[PersonPhone]
[Person].[PhoneNumberType]
[Person].[StateProvince]
[Sales].[Customer]
[dbo].[Name]

Project > Isrep17 > User databases > AdventureWorks > Views > Sales.vIndividualCustomer

[dbo].[Phone]
Sales

 **[Sales].[vPersonDemographics]**
MS_Description

Displays the content from each element in the xml column Demographics for each customer in the Person.Person table.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Created	13:14:55 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Name	Data Type	Max Length (Bytes)
BusinessEntityID	int	4
TotalPurchaseYTD	money	8
DateFirstPurchase	datetime	8
BirthDate	datetime	8
MaritalStatus	nvarchar(1)	2
YearlyIncome	nvarchar(30)	60
Gender	nvarchar(1)	2
TotalChildren	int	4
NumberChildrenAtHome	int	4
Education	nvarchar(30)	60
Occupation	nvarchar(30)	60
HomeOwnerFlag	bit	1
NumberCarsOwned	int	4

SQL Script

```
CREATE VIEW [Sales].[vPersonDemographics]
AS
SELECT
    p.[BusinessEntityID]
    , [IndividualSurvey].[ref].[value] (N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey";
    TotalPurchaseYTD[1]', 'money') AS [TotalPurchaseYTD]
    , CONVERT(datetime, REPLACE([IndividualSurvey].[ref].[value] (N'declare default
element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-
works/IndividualSurvey";
    DateFirstPurchase[1]', 'nvarchar(20)') , 'Z', ''), 101) AS [DateFirst-
```

```

Purchase]
    , CONVERT(datetime, REPLACE([IndividualSurvey].[ref].[value] (N'declare default
element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-
works/IndividualSurvey";
        BirthDate[1]', 'nvarchar(20)'), 'Z', ''), 101) AS [BirthDate]
    , [IndividualSurvey].[ref].[value] (N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey";
        MaritalStatus[1]', 'nvarchar(1)') AS [MaritalStatus]
    , [IndividualSurvey].[ref].[value] (N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey";
        YearlyIncome[1]', 'nvarchar(30)') AS [YearlyIncome]
    , [IndividualSurvey].[ref].[value] (N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey";
        Gender[1]', 'nvarchar(1)') AS [Gender]
    , [IndividualSurvey].[ref].[value] (N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey";
        TotalChildren[1]', 'integer') AS [TotalChildren]
    , [IndividualSurvey].[ref].[value] (N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey";
        NumberChildrenAtHome[1]', 'integer') AS [NumberChildrenAtHome]
    , [IndividualSurvey].[ref].[value] (N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey";
        Education[1]', 'nvarchar(30)') AS [Education]
    , [IndividualSurvey].[ref].[value] (N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey";
        Occupation[1]', 'nvarchar(30)') AS [Occupation]
    , [IndividualSurvey].[ref].[value] (N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey";
        HomeOwnerFlag[1]', 'bit') AS [HomeOwnerFlag]
    , [IndividualSurvey].[ref].[value] (N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey";
        NumberCarsOwned[1]', 'integer') AS [NumberCarsOwned]
FROM [Person].[Person] p
CROSS APPLY p.[Demographics].nodes(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey";
    /IndividualSurvey') AS [IndividualSurvey] (ref)
WHERE [Demographics] IS NOT NULL;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Displays the content from each
element in the xml column Demographics for each customer in the Person.Person
table.', 'SCHEMA', N'Sales', 'VIEW', N'vPersonDemographics', NULL, NULL
GO

```

Uses

[Person].[Person]
Sales



MS_Description

Sales representatives (names and addresses) and their sales-related information.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Created	13:14:55 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Name	Data Type	Max Length (Bytes)
BusinessEntityID	int	4
Title	nvarchar(8)	16
FirstName	[dbo].[Name]	100
MiddleName	[dbo].[Name]	100
LastName	[dbo].[Name]	100
Suffix	nvarchar(10)	20
JobTitle	nvarchar(50)	100
PhoneNumber	[dbo].[Phone]	50
PhoneNumberType	[dbo].[Name]	100
EmailAddress	nvarchar(50)	100
EmailPromotion	int	4
AddressLine1	nvarchar(60)	120
AddressLine2	nvarchar(60)	120
City	nvarchar(30)	60
StateProvinceName	[dbo].[Name]	100
PostalCode	nvarchar(15)	30
CountryRegionName	[dbo].[Name]	100
TerritoryName	[dbo].[Name]	100
TerritoryGroup	nvarchar(50)	100
SalesQuota	money	8
SalesYTD	money	8
SalesLastYear	money	8

SQL Script

```

CREATE VIEW [Sales].[vSalesPerson]
AS
SELECT
    s.[BusinessEntityID]
    ,p.[Title]
    ,p.[FirstName]
    ,p.[MiddleName]
    ,p.[LastName]
    ,p.[Suffix]
    ,e.[JobTitle]
    ,pp.[PhoneNumber]
    ,pnt.[Name] AS [PhoneNumberType]
    ,ea.[EmailAddress]
    ,p.[EmailPromotion]
    ,a.[AddressLine1]
    ,a.[AddressLine2]
    ,a.[City]
    ,[StateProvinceName] = sp.[Name]
    ,a.[PostalCode]
    ,[CountryRegionName] = cr.[Name]
    ,[TerritoryName] = st.[Name]
    ,[TerritoryGroup] = st.[Group]
    ,s.[SalesQuota]
    ,s.[SalesYTD]
    ,s.[SalesLastYear]
FROM [Sales].[SalesPerson] s
    INNER JOIN [HumanResources].[Employee] e
ON e.[BusinessEntityID] = s.[BusinessEntityID]
    INNER JOIN [Person].[Person] p
ON p.[BusinessEntityID] = s.[BusinessEntityID]
    INNER JOIN [Person].[BusinessEntityAddress] bea
ON bea.[BusinessEntityID] = s.[BusinessEntityID]
    INNER JOIN [Person].[Address] a
ON a.[AddressID] = bea.[AddressID]
    INNER JOIN [Person].[StateProvince] sp
ON sp.[StateProvinceID] = a.[StateProvinceID]
    INNER JOIN [Person].[CountryRegion] cr
ON cr.[CountryRegionCode] = sp.[CountryRegionCode]
    LEFT OUTER JOIN [Sales].[SalesTerritory] st
ON st.[TerritoryID] = s.[TerritoryID]
    LEFT OUTER JOIN [Person].[EmailAddress] ea
ON ea.[BusinessEntityID] = p.[BusinessEntityID]
    LEFT OUTER JOIN [Person].[PersonPhone] pp
ON pp.[BusinessEntityID] = p.[BusinessEntityID]
    LEFT OUTER JOIN [Person].[PhoneNumberType] pnt
ON pnt.[PhoneNumberTypeID] = pp.[PhoneNumberTypeID];
GO
EXEC sp_addextendedproperty N'MS_Description', N'Sales representatives (names and
addresses) and their sales-related information.', 'SCHEMA', N'Sales', 'VIEW', N'v-
SalesPerson', NULL, NULL
GO

```

Uses

[HumanResources].[Employee]
[Person].[Address]
[Person].[BusinessEntityAddress]
[Person].[CountryRegion]
[Person].[EmailAddress]
[Person].[Person]
[Person].[PersonPhone]
[Person].[PhoneNumberType]
[Person].[StateProvince]
[Sales].[SalesPerson]
[Sales].[SalesTerritory]
[dbo].[Name]
[dbo].[Phone]
Sales

 **[Sales].[vSalesPersonSalesByFiscalYears]**
MS_Description

Uses PIVOT to return aggregated sales information for each sales representative.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Created	13:14:55 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Name	Data Type	Max Length (Bytes)
SalesPersonID	int	4
FullName	nvarchar(152)	304
JobTitle	nvarchar(50)	100
SalesTerritory	[dbo].[Name]	100
2002	money	8
2003	money	8
2004	money	8

SQL Script

```
CREATE VIEW [Sales].[vSalesPersonSalesByFiscalYears]
AS
SELECT
    pvt.[SalesPersonID]
    ,pvt.[FullName]
    ,pvt.[JobTitle]
    ,pvt.[SalesTerritory]
    ,pvt.[2002]
    ,pvt.[2003]
    ,pvt.[2004]
FROM (SELECT
    soh.[SalesPersonID]
    ,p.[FirstName] + ' ' + COALESCE(p.[MiddleName], '') + ' ' + p.[LastName] AS
[FullName]
    ,e.[JobTitle]
    ,st.[Name] AS [SalesTerritory]
    ,soh.[SubTotal]
    ,YEAR(DATEADD(m, 6, soh.[OrderDate])) AS [FiscalYear]
```

```
FROM [Sales].[SalesPerson] sp
INNER JOIN [Sales].[SalesOrderHeader] soh
ON sp.[BusinessEntityID] = soh.[SalesPersonID]
INNER JOIN [Sales].[SalesTerritory] st
ON sp.[TerritoryID] = st.[TerritoryID]
INNER JOIN [HumanResources].[Employee] e
ON soh.[SalesPersonID] = e.[BusinessEntityID]
INNER JOIN [Person].[Person] p
ON p.[BusinessEntityID] = sp.[BusinessEntityID]
) AS soh
PIVOT
(
    SUM([SubTotal])
    FOR [FiscalYear]
    IN ([2002], [2003], [2004])
) AS pvt;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Uses PIVOT to return aggregated
sales information for each sales representative.', 'SCHEMA', N'Sales', 'VIEW', N'v-
SalesPersonSalesByFiscalYears', NULL, NULL
GO
```

Uses

[HumanResources].[Employee]

[Person].[Person]

[Sales].[SalesOrderHeader]

[Sales].[SalesPerson]

[Sales].[SalesTerritory]

[dbo].[Name]

Sales

 **[Sales].[vStoreWithAddresses]**
MS_Description

Stores (including store addresses) that sell Adventure Works Cycles products to consumers.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Created	13:14:55 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Name	Data Type	Max Length (Bytes)
BusinessEntityID	int	4
Name	[dbo].[Name]	100
AddressType	[dbo].[Name]	100
AddressLine1	nvarchar(60)	120
AddressLine2	nvarchar(60)	120
City	nvarchar(30)	60
StateProvinceName	[dbo].[Name]	100
PostalCode	nvarchar(15)	30
CountryRegionName	[dbo].[Name]	100

SQL Script

```
CREATE VIEW [Sales].[vStoreWithAddresses] AS
SELECT
    s.[BusinessEntityID]
    ,s.[Name]
    ,at.[Name] AS [AddressType]
    ,a.[AddressLine1]
    ,a.[AddressLine2]
    ,a.[City]
    ,sp.[Name] AS [StateProvinceName]
    ,a.[PostalCode]
    ,cr.[Name] AS [CountryRegionName]
FROM [Sales].[Store] s
INNER JOIN [Person].[BusinessEntityAddress] bea
ON bea.[BusinessEntityID] = s.[BusinessEntityID]
INNER JOIN [Person].[Address] a
ON a.[AddressID] = bea.[AddressID]
```

```
INNER JOIN [Person].[StateProvince] sp
ON sp.[StateProvinceID] = a.[StateProvinceID]
INNER JOIN [Person].[CountryRegion] cr
ON cr.[CountryRegionCode] = sp.[CountryRegionCode]
INNER JOIN [Person].[AddressType] at
ON at.[AddressTypeID] = bea.[AddressTypeID];
GO
EXEC sp_addextendedproperty N'MS_Description', N'Stores (including store addresses)
that sell Adventure Works Cycles products to consumers.', 'SCHEMA', N'Sales',
'VIEW', N'vStoreWithAddresses', NULL, NULL
GO
```

Uses

[Person].[Address]
[Person].[AddressType]
[Person].[BusinessEntityAddress]
[Person].[CountryRegion]
[Person].[StateProvince]
[Sales].[Store]
[dbo].[Name]
Sales

 **[Sales].[vStoreWithContacts]**
MS_Description

Stores (including store contacts) that sell Adventure Works Cycles products to consumers.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Created	13:14:55 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Name	Data Type	Max Length (Bytes)
BusinessEntityID	int	4
Name	[dbo].[Name]	100
ContactType	[dbo].[Name]	100
Title	nvarchar(8)	16
FirstName	[dbo].[Name]	100
MiddleName	[dbo].[Name]	100
LastName	[dbo].[Name]	100
Suffix	nvarchar(10)	20
PhoneNumber	[dbo].[Phone]	50
PhoneNumberType	[dbo].[Name]	100
EmailAddress	nvarchar(50)	100
EmailPromotion	int	4

SQL Script

```
CREATE VIEW [Sales].[vStoreWithContacts] AS
SELECT
    s.[BusinessEntityID]
    ,s.[Name]
    ,ct.[Name] AS [ContactType]
    ,p.[Title]
    ,p.[FirstName]
    ,p.[MiddleName]
    ,p.[LastName]
    ,p.[Suffix]
    ,pp.[PhoneNumber]
    ,pnt.[Name] AS [PhoneNumberType]
```

```
,ea.[EmailAddress]
,p.[EmailPromotion]
FROM [Sales].[Store] s
INNER JOIN [Person].[BusinessEntityContact] bec
ON bec.[BusinessEntityID] = s.[BusinessEntityID]
INNER JOIN [Person].[ContactType] ct
ON ct.[ContactTypeID] = bec.[ContactTypeID]
INNER JOIN [Person].[Person] p
ON p.[BusinessEntityID] = bec.[PersonID]
LEFT OUTER JOIN [Person].[EmailAddress] ea
ON ea.[BusinessEntityID] = p.[BusinessEntityID]
LEFT OUTER JOIN [Person].[PersonPhone] pp
ON pp.[BusinessEntityID] = p.[BusinessEntityID]
LEFT OUTER JOIN [Person].[PhoneNumberType] pnt
ON pnt.[PhoneNumberTypeID] = pp.[PhoneNumberTypeID];
GO
EXEC sp_addextendedproperty N'MS_Description', N'Stores (including store contacts)
that sell Adventure Works Cycles products to consumers.', 'SCHEMA', N'Sales',
'VIEW', N'vStoreWithContacts', NULL, NULL
GO
```

Uses

[Person].[BusinessEntityContact]
[Person].[ContactType]
[Person].[EmailAddress]
[Person].[Person]
[Person].[PersonPhone]
[Person].[PhoneNumberType]
[Sales].[Store]
[dbo].[Name]
[dbo].[Phone]
Sales

 **[Sales].[vStoreWithDemographics]**
MS_Description

Stores (including demographics) that sell Adventure Works Cycles products to consumers.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Created	13:14:55 14 marca 2012
Last Modified	13:14:55 14 marca 2012

Columns

Name	Data Type	Max Length (Bytes)
BusinessEntityID	int	4
Name	[dbo].[Name]	100
AnnualSales	money	8
AnnualRevenue	money	8
BankName	nvarchar(50)	100
BusinessType	nvarchar(5)	10
YearOpened	int	4
Specialty	nvarchar(50)	100
SquareFeet	int	4
Brands	nvarchar(30)	60
Internet	nvarchar(30)	60
NumberEmployees	int	4

SQL Script

```
CREATE VIEW [Sales].[vStoreWithDemographics] AS
SELECT
    s.[BusinessEntityID]
    ,s.[Name]
    ,s.[Demographics].value('declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey";
(/StoreSurvey/AnnualSales) [1]', 'money') AS [AnnualSales]
    ,s.[Demographics].value('declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey";
(/StoreSurvey/AnnualRevenue) [1]', 'money') AS [AnnualRevenue]
    ,s.[Demographics].value('declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey";
```

```

        (/StoreSurvey/BankName) [1]', 'nvarchar(50)') AS [BankName]
    ,s.[Demographics].value('declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey";
        (/StoreSurvey/BusinessType) [1]', 'nvarchar(5)') AS [BusinessType]
    ,s.[Demographics].value('declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey";
        (/StoreSurvey/YearOpened) [1]', 'integer') AS [YearOpened]
    ,s.[Demographics].value('declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey";
        (/StoreSurvey/Specialty) [1]', 'nvarchar(50)') AS [Specialty]
    ,s.[Demographics].value('declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey";
        (/StoreSurvey/SquareFeet) [1]', 'integer') AS [SquareFeet]
    ,s.[Demographics].value('declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey";
        (/StoreSurvey/Brands) [1]', 'nvarchar(30)') AS [Brands]
    ,s.[Demographics].value('declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey";
        (/StoreSurvey/Internet) [1]', 'nvarchar(30)') AS [Internet]
    ,s.[Demographics].value('declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey";
        (/StoreSurvey/NumberEmployees) [1]', 'integer') AS [NumberEmployees]
FROM [Sales].[Store] s;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Stores (including demographics)
that sell Adventure Works Cycles products to consumers.', 'SCHEMA', N'Sales',
'VIEW', N'vStoreWithDemographics', NULL, NULL
GO

```

Uses

[Sales].[Store]
[dbo].[Name]
Sales

 **Stored Procedures**
Objects

Name
dbo.uspGetBillOfMaterials <i>Stored procedure using a recursive query to return a multi-level bill of material for the specified ProductID.</i>
dbo.uspGetEmployeeManagers <i>Stored procedure using a recursive query to return the direct and indirect managers of the specified employee.</i>
dbo.uspGetManagerEmployees <i>Stored procedure using a recursive query to return the direct and indirect employees of the specified manager.</i>
dbo.uspGetWhereUsedProductID <i>Stored procedure using a recursive query to return all components or assemblies that directly or indirectly use the specified ProductID.</i>
dbo.uspLogError <i>Logs error information in the ErrorLog table about the error that caused execution to jump to the CATCH block of a TRY...CATCH construct. Should be executed from within the scope of a CATCH block otherwise it will return without inserting error information.</i>
dbo.uspPrintError <i>Prints error information about the error that caused execution to jump to the CATCH block of a TRY...CATCH construct. Should be executed from within the scope of a CATCH block otherwise it will return without printing any error information.</i>
dbo.uspSearchCandidateResumes
HumanResources.uspUpdateEmployeeHireInfo <i>Updates the Employee table and inserts a new row in the EmployeePayHistory table with the values specified in the input parameters.</i>
HumanResources.uspUpdateEmployeeLogin <i>Updates the Employee table with the values specified in the input parameters for the given BusinessEntityID.</i>
HumanResources.uspUpdateEmployeePersonalInfo <i>Updates the Employee table with the values specified in the input parameters for the given EmployeeID.</i>

[dbo].[uspGetBillOfMaterials]

MS_Description

Stored procedure using a recursive query to return a multi-level bill of material for the specified ProductID.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@StartProductID <i>Input parameter for the stored procedure uspGetBillOfMaterials. Enter a valid ProductID from the Production.Product table.</i>	int	4
@CheckDate <i>Input parameter for the stored procedure uspGetBillOfMaterials used to eliminate components not used after that date. Enter a valid date.</i>	datetime	8

SQL Script

```
CREATE PROCEDURE [dbo].[uspGetBillOfMaterials]
    @StartProductID [int],
    @CheckDate [datetime]
AS
BEGIN
    SET NOCOUNT ON;

    -- Use recursive query to generate a multi-level Bill of Material (i.e. all
    level 1
    -- components of a level 0 assembly, all level 2 components of a level 1
    assembly)
    -- The CheckDate eliminates any components that are no longer used in the
    product on this date.
    WITH [BOM_cte] ([ProductAssemblyID], [ComponentID], [ComponentDesc], [PerAssembly-
    Qty], [StandardCost], [ListPrice], [BOMLevel], [RecursionLevel]) -- CTE name and
    columns
    AS (
        SELECT b.[ProductAssemblyID], b.[ComponentID], p.[Name], b.[PerAssemblyQty],
        p.[StandardCost], p.[ListPrice], b.[BOMLevel], 0 -- Get the initial list of
        components for the bike assembly
        FROM [Production].[BillOfMaterials] b
            INNER JOIN [Production].[Product] p
            ON b.[ComponentID] = p.[ProductID]
        WHERE b.[ProductAssemblyID] = @StartProductID
            AND @CheckDate >= b.[StartDate]
            AND @CheckDate <= ISNULL(b.[EndDate], @CheckDate)
        UNION ALL
```

```
SELECT b.[ProductAssemblyID], b.[ComponentID], p.[Name], b.[PerAssemblyQty],
p.[StandardCost], p.[ListPrice], b.[BOMLevel], [RecursionLevel] + 1 -- Join
recursive member to anchor
FROM [BOM_cte] cte
INNER JOIN [Production].[BillOfMaterials] b
ON b.[ProductAssemblyID] = cte.[ComponentID]
INNER JOIN [Production].[Product] p
ON b.[ComponentID] = p.[ProductID]
WHERE @CheckDate >= b.[StartDate]
AND @CheckDate <= ISNULL(b.[EndDate], @CheckDate)
)
-- Outer select from the CTE
SELECT b.[ProductAssemblyID], b.[ComponentID], b.[ComponentDesc], SUM(b.[Per-
AssemblyQty]) AS [TotalQuantity] , b.[StandardCost], b.[ListPrice], b.[BOMLevel],
b.[RecursionLevel]
FROM [BOM_cte] b
GROUP BY b.[ComponentID], b.[ComponentDesc], b.[ProductAssemblyID],
b.[BOMLevel], b.[RecursionLevel], b.[StandardCost], b.[ListPrice]
ORDER BY b.[BOMLevel], b.[ProductAssemblyID], b.[ComponentID]
OPTION (MAXRECURSION 25)
END;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Stored procedure using a recursive
query to return a multi-level bill of material for the specified ProductID.',
'SHEMA', N'dbo', 'PROCEDURE', N'uspGetBillOfMaterials', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored
procedure uspGetBillOfMaterials used to eliminate components not used after that
date. Enter a valid date.', 'SCHEMA', N'dbo', 'PROCEDURE', N'uspGetBillOfMaterials',
'PARAMETER', N'@CheckDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored
procedure uspGetBillOfMaterials. Enter a valid ProductID from the Production.Product
table.', 'SCHEMA', N'dbo', 'PROCEDURE', N'uspGetBillOfMaterials', 'PARAMETER',
N'@StartProductID'
GO
```

Uses

[Production].[BillOfMaterials]

[Production].[Product]

[dbo].[uspGetEmployeeManagers]

MS_Description

Stored procedure using a recursive query to return the direct and indirect managers of the specified employee.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@BusinessEntityID <i>Input parameter for the stored procedure uspGetEmployeeManagers. Enter a valid BusinessEntityID from the HumanResources.Employee table.</i>	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[uspGetEmployeeManagers]
    @BusinessEntityID [int]
AS
BEGIN
    SET NOCOUNT ON;

    -- Use recursive query to list out all Employees required for a particular
    Manager
    WITH [EMP_cte] ([BusinessEntityID], [OrganizationNode], [FirstName], [LastName],
    [JobTitle], [RecursionLevel]) -- CTE name and columns
    AS (
        SELECT e.[BusinessEntityID], e.[OrganizationNode], p.[FirstName], p.[Last-
        Name], e.[JobTitle], 0 -- Get the initial Employee
        FROM [HumanResources].[Employee] e
            INNER JOIN [Person].[Person] as p
            ON p.[BusinessEntityID] = e.[BusinessEntityID]
        WHERE e.[BusinessEntityID] = @BusinessEntityID
        UNION ALL
        SELECT e.[BusinessEntityID], e.[OrganizationNode], p.[FirstName], p.[Last-
        Name], e.[JobTitle], [RecursionLevel] + 1 -- Join recursive member to anchor
        FROM [HumanResources].[Employee] e
            INNER JOIN [EMP_cte]
            ON e.[OrganizationNode] = [EMP_cte].[OrganizationNode].GetAncestor(1)
            INNER JOIN [Person].[Person] p
            ON p.[BusinessEntityID] = e.[BusinessEntityID]
    )
    -- Join back to Employee to return the manager name
    SELECT [EMP_cte].[RecursionLevel], [EMP_cte].[BusinessEntityID],
    [EMP_cte].[FirstName], [EMP_cte].[LastName],
    [EMP_cte].[OrganizationNode].ToString() AS [OrganizationNode], p.[FirstName]
```

```
AS 'ManagerFirstName', p.[LastName] AS 'ManagerLastName' -- Outer select from the
CTE
FROM [EMP_cte]
INNER JOIN [HumanResources].[Employee] e
ON [EMP_cte].[OrganizationNode].GetAncestor(1) = e.[OrganizationNode]
INNER JOIN [Person].[Person] p
ON p.[BusinessEntityID] = e.[BusinessEntityID]
ORDER BY [RecursionLevel], [EMP_cte].[OrganizationNode].ToString()
OPTION (MAXRECURSION 25)
END;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Stored procedure using a recursive
query to return the direct and indirect managers of the specified employee.',
'SHEMA', N'dbo', 'PROCEDURE', N'uspGetEmployeeManagers', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored
procedure uspGetEmployeeManagers. Enter a valid BusinessEntityID from the Human-
Resources.Employee table.', 'SCHEMA', N'dbo', 'PROCEDURE', N'uspGetEmployee-
Managers', 'PARAMETER', N'@BusinessEntityID'
GO
```

Uses

[HumanResources].[Employee]
[Person].[Person]

[dbo].[uspGetManagerEmployees]

MS_Description

Stored procedure using a recursive query to return the direct and indirect employees of the specified manager.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@BusinessEntityID <i>Input parameter for the stored procedure uspGetManagerEmployees. Enter a valid BusinessEntityID of the manager from the HumanResources.Employee table.</i>	int	4

SQL Script

```
CREATE PROCEDURE [dbo].[uspGetManagerEmployees]
    @BusinessEntityID [int]
AS
BEGIN
    SET NOCOUNT ON;

    -- Use recursive query to list out all Employees required for a particular
    Manager
    WITH [EMP_cte]([BusinessEntityID], [OrganizationNode], [FirstName], [LastName],
    [RecursionLevel]) -- CTE name and columns
    AS (
        SELECT e.[BusinessEntityID], e.[OrganizationNode], p.[FirstName], p.[Last-
        Name], 0 -- Get the initial list of Employees for Manager n
        FROM [HumanResources].[Employee] e
            INNER JOIN [Person].[Person] p
            ON p.[BusinessEntityID] = e.[BusinessEntityID]
        WHERE e.[BusinessEntityID] = @BusinessEntityID
        UNION ALL
        SELECT e.[BusinessEntityID], e.[OrganizationNode], p.[FirstName], p.[Last-
        Name], [RecursionLevel] + 1 -- Join recursive member to anchor
        FROM [HumanResources].[Employee] e
            INNER JOIN [EMP_cte]
            ON e.[OrganizationNode].GetAncestor(1) = [EMP_cte].[OrganizationNode]
            INNER JOIN [Person].[Person] p
            ON p.[BusinessEntityID] = e.[BusinessEntityID]
    )
    -- Join back to Employee to return the manager name
    SELECT [EMP_cte].[RecursionLevel], [EMP_cte].[OrganizationNode].ToString() as
    [OrganizationNode], p.[FirstName] AS 'ManagerFirstName', p.[LastName] AS 'Manager-
```

```
LastName',
    [EMP_cte].[BusinessEntityID], [EMP_cte].[FirstName], [EMP_cte].[LastName] --
Outer select from the CTE
FROM [EMP_cte]
    INNER JOIN [HumanResources].[Employee] e
    ON [EMP_cte].[OrganizationNode].GetAncestor(1) = e.[OrganizationNode]
    INNER JOIN [Person].[Person] p
    ON p.[BusinessEntityID] = e.[BusinessEntityID]
ORDER BY [RecursionLevel], [EMP_cte].[OrganizationNode].ToString()
OPTION (MAXRECURSION 25)
END;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Stored procedure using a recursive
query to return the direct and indirect employees of the specified manager.',
'SHEMA', N'dbo', 'PROCEDURE', N'uspGetManagerEmployees', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored
procedure uspGetManagerEmployees. Enter a valid BusinessEntityID of the manager from
the HumanResources.Employee table.', 'SCHEMA', N'dbo', 'PROCEDURE', N'uspGetManager-
Employees', 'PARAMETER', N'@BusinessEntityID'
GO
```

Uses

[HumanResources].[Employee]
[Person].[Person]

[dbo].[uspGetWhereUsedProductID]

MS_Description

Stored procedure using a recursive query to return all components or assemblies that directly or indirectly use the specified ProductID.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@StartProductID <i>Input parameter for the stored procedure uspGetWhereUsedProductID. Enter a valid ProductID from the Production.Product table.</i>	int	4
@CheckDate <i>Input parameter for the stored procedure uspGetWhereUsedProductID used to eliminate components not used after that date. Enter a valid date.</i>	datetime	8

SQL Script

```
CREATE PROCEDURE [dbo].[uspGetWhereUsedProductID]
    @StartProductID [int],
    @CheckDate [datetime]
AS
BEGIN
    SET NOCOUNT ON;

    --Use recursive query to generate a multi-level Bill of Material (i.e. all level
    1 components of a level 0 assembly, all level 2 components of a level 1 assembly)
    WITH [BOM_cte] ([ProductAssemblyID], [ComponentID], [ComponentDesc], [PerAssemblyQty], [StandardCost], [ListPrice], [BOMLevel], [RecursionLevel]) -- CTE name and
    columns
    AS (
        SELECT b.[ProductAssemblyID], b.[ComponentID], p.[Name], b.[PerAssemblyQty],
        p.[StandardCost], p.[ListPrice], b.[BOMLevel], 0 -- Get the initial list of
        components for the bike assembly
        FROM [Production].[BillOfMaterials] b
        INNER JOIN [Production].[Product] p
        ON b.[ProductAssemblyID] = p.[ProductID]
        WHERE b.[ComponentID] = @StartProductID
        AND @CheckDate >= b.[StartDate]
        AND @CheckDate <= ISNULL(b.[EndDate], @CheckDate)
        UNION ALL
        SELECT b.[ProductAssemblyID], b.[ComponentID], p.[Name], b.[PerAssemblyQty],
        p.[StandardCost], p.[ListPrice], b.[BOMLevel], [RecursionLevel] + 1 -- Join
        recursive member to anchor
```

```
FROM [BOM_cte] cte
    INNER JOIN [Production].[BillOfMaterials] b
    ON cte.[ProductAssemblyID] = b.[ComponentID]
    INNER JOIN [Production].[Product] p
    ON b.[ProductAssemblyID] = p.[ProductID]
WHERE @CheckDate >= b.[StartDate]
    AND @CheckDate <= ISNULL(b.[EndDate], @CheckDate)
)
-- Outer select from the CTE
SELECT b.[ProductAssemblyID], b.[ComponentID], b.[ComponentDesc], SUM(b.[Per-
AssemblyQty]) AS [TotalQuantity], b.[StandardCost], b.[ListPrice], b.[BOMLevel],
b.[RecursionLevel]
FROM [BOM_cte] b
GROUP BY b.[ComponentID], b.[ComponentDesc], b.[ProductAssemblyID],
b.[BOMLevel], b.[RecursionLevel], b.[StandardCost], b.[ListPrice]
ORDER BY b.[BOMLevel], b.[ProductAssemblyID], b.[ComponentID]
OPTION (MAXRECURSION 25)
END;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Stored procedure using a recursive
query to return all components or assemblies that directly or indirectly use the
specified ProductID.', 'SCHEMA', N'dbo', 'PROCEDURE', N'uspGetWhereUsedProductID',
NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored
procedure uspGetWhereUsedProductID used to eliminate components not used after that
date. Enter a valid date.', 'SCHEMA', N'dbo', 'PROCEDURE', N'uspGetWhereUsedProduct-
ID', 'PARAMETER', N'@CheckDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored
procedure uspGetWhereUsedProductID. Enter a valid ProductID from the
Production.Product table.', 'SCHEMA', N'dbo', 'PROCEDURE', N'uspGetWhereUsedProduct-
ID', 'PARAMETER', N'@StartProductID'
GO
```

Uses

[Production].[BillOfMaterials]

[Production].[Product]

[dbo].[uspLogError]

MS_Description

Logs error information in the ErrorLog table about the error that caused execution to jump to the CATCH block of a TRY...CATCH construct. Should be executed from within the scope of a CATCH block otherwise it will return without inserting error information.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)	Direction
@ErrorLogID <i>Output parameter for the stored procedure uspLogError. Contains the ErrorLogID value corresponding to the row inserted by uspLogError in the ErrorLog table.</i>	int	4	Out

SQL Script

```
-- uspLogError logs error information in the ErrorLog table about the
-- error that caused execution to jump to the CATCH block of a
-- TRY...CATCH construct. This should be executed from within the scope
-- of a CATCH block otherwise it will return without inserting error
-- information.
CREATE PROCEDURE [dbo].[uspLogError]
    @ErrorLogID [int] = 0 OUTPUT -- contains the ErrorLogID of the row inserted
AS                                     -- by uspLogError in the ErrorLog table
BEGIN
    SET NOCOUNT ON;

    -- Output parameter value of 0 indicates that error
    -- information was not logged
    SET @ErrorLogID = 0;

    BEGIN TRY
        -- Return if there is no error information to log
        IF ERROR_NUMBER() IS NULL
            RETURN;

        -- Return if inside an uncommittable transaction.
        -- Data insertion/modification is not allowed when
        -- a transaction is in an uncommittable state.
        IF XACT_STATE() = -1
            BEGIN
```

```
        PRINT 'Cannot log error since the current transaction is in an
uncommittable state. '
        + 'Rollback the transaction before executing uspLogError in order to
successfully log error information.';

        RETURN;
    END

    INSERT [dbo].[ErrorLog]
    (
        [UserName],
        [ErrorNumber],
        [ErrorSeverity],
        [ErrorState],
        [ErrorProcedure],
        [ErrorLine],
        [ErrorMessage]
    )
    VALUES
    (
        CONVERT(sysname, CURRENT_USER),
        ERROR_NUMBER(),
        ERROR_SEVERITY(),
        ERROR_STATE(),
        ERROR_PROCEDURE(),
        ERROR_LINE(),
        ERROR_MESSAGE()
    );

    -- Pass back the ErrorLogID of the row inserted
    SET @ErrorLogID = @@IDENTITY;
END TRY
BEGIN CATCH
    PRINT 'An error occurred in stored procedure uspLogError: ';
    EXECUTE [dbo].[uspPrintError];
    RETURN -1;
END CATCH
END;
GO

EXEC sp_addextendedproperty N'MS_Description', N'Logs error information in the Error-
Log table about the error that caused execution to jump to the CATCH block of a
TRY...CATCH construct. Should be executed from within the scope of a CATCH block
otherwise it will return without inserting error information.', 'SCHEMA', N'dbo',
'PROCEDURE', N'uspLogError', NULL, NULL
GO

EXEC sp_addextendedproperty N'MS_Description', N'Output parameter for the stored
procedure uspLogError. Contains the ErrorLogID value corresponding to the row
inserted by uspLogError in the ErrorLog table.', 'SCHEMA', N'dbo', 'PROCEDURE',
N'uspLogError', 'PARAMETER', N'@ErrorLogID'
GO
```

Uses

[dbo].[ErrorLog]
[dbo].[uspPrintError]

Used By

[HumanResources].[uspUpdateEmployeeHireInfo]

[HumanResources].[uspUpdateEmployeeLogin]

[HumanResources].[uspUpdateEmployeePersonalInfo]

[dbo].[uspPrintError]

MS_Description

Prints error information about the error that caused execution to jump to the CATCH block of a TRY...CATCH construct. Should be executed from within the scope of a CATCH block otherwise it will return without printing any error information.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
-- uspPrintError prints error information about the error that caused
-- execution to jump to the CATCH block of a TRY...CATCH construct.
-- Should be executed from within the scope of a CATCH block otherwise
-- it will return without printing any error information.
CREATE PROCEDURE [dbo].[uspPrintError]
AS
BEGIN
    SET NOCOUNT ON;

    -- Print error information.
    PRINT 'Error ' + CONVERT(varchar(50), ERROR_NUMBER()) +
        ', Severity ' + CONVERT(varchar(5), ERROR_SEVERITY()) +
        ', State ' + CONVERT(varchar(5), ERROR_STATE()) +
        ', Procedure ' + ISNULL(ERROR_PROCEDURE(), '-') +
        ', Line ' + CONVERT(varchar(5), ERROR_LINE());
    PRINT ERROR_MESSAGE();
END;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Prints error information about the
error that caused execution to jump to the CATCH block of a TRY...CATCH construct.
Should be executed from within the scope of a CATCH block otherwise it will return
without printing any error information.', 'SCHEMA', N'dbo', 'PROCEDURE', N'uspPrint-
Error', NULL, NULL
GO
```

Used By

[dbo].[uspLogError]

[dbo].[uspSearchCandidateResumes]

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@searchString	nvarchar(1000)	2000
@useInflectional	bit	1
@useThesaurus	bit	1
@language	int	4

SQL Script

```
--A stored procedure which demonstrates integrated full text search

CREATE PROCEDURE [dbo].[uspSearchCandidateResumes]
    @searchString [nvarchar](1000),
    @useInflectional [bit]=0,
    @useThesaurus [bit]=0,
    @language[int]=0

WITH EXECUTE AS CALLER
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @string nvarchar(1050)
    --setting the lcid to the default instance LCID if needed
    IF @language = NULL OR @language = 0
    BEGIN
        SELECT @language =CONVERT(int, serverproperty('lcid'))
    END

    --FREETEXTTABLE case as inflectional and Thesaurus were required
    IF @useThesaurus = 1 AND @useInflectional = 1
    BEGIN
        SELECT FT_TBL.[JobCandidateID], KEY_TBL.[RANK] FROM [Human-Resources].[JobCandidate] AS FT_TBL
            INNER JOIN FREETEXTTABLE([HumanResources].[JobCandidate],*,
            @searchString,LANGUAGE @language) AS KEY_TBL
            ON FT_TBL.[JobCandidateID] =KEY_TBL.[KEY]
```

```
        END

        ELSE IF @useThesaurus = 1
        BEGIN
            SELECT @string = 'FORMSOF(THESAURUS, '''+@searchString + '''+)''
            SELECT FT_TBL.[JobCandidateID], KEY_TBL.[RANK] FROM [Human-
Resources].[JobCandidate] AS FT_TBL
                INNER JOIN CONTAINSTABLE([HumanResources].[JobCandidate], *,
@string, LANGUAGE @language) AS KEY_TBL
                ON FT_TBL.[JobCandidateID] =KEY_TBL.[KEY]
        END

        ELSE IF @useInflectional = 1
        BEGIN
            SELECT @string = 'FORMSOF(INFLECTIONAL, '''+@searchString + '''+)''
            SELECT FT_TBL.[JobCandidateID], KEY_TBL.[RANK] FROM [Human-
Resources].[JobCandidate] AS FT_TBL
                INNER JOIN CONTAINSTABLE([HumanResources].[JobCandidate], *,
@string, LANGUAGE @language) AS KEY_TBL
                ON FT_TBL.[JobCandidateID] =KEY_TBL.[KEY]
        END

        ELSE --base case, plain CONTAINSTABLE
        BEGIN
            SELECT @string= '''+@searchString + '''+
            SELECT FT_TBL.[JobCandidateID],KEY_TBL.[RANK] FROM [Human-
Resources].[JobCandidate] AS FT_TBL
                INNER JOIN CONTAINSTABLE([HumanResources].[Job-
Candidate], *,@string, LANGUAGE @language) AS KEY_TBL
                ON FT_TBL.[JobCandidateID] =KEY_TBL.[KEY]
        END

    END;
GO
```

Uses

[HumanResources].[JobCandidate]

[HumanResources].[uspUpdateEmployeeHireInfo]

MS_Description

Updates the Employee table and inserts a new row in the EmployeePayHistory table with the values specified in the input parameters.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@BusinessEntityID <i>Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a valid BusinessEntityID from the Employee table.</i>	int	4
@JobTitle <i>Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a title for the employee.</i>	nvarchar(50)	100
@HireDate <i>Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a hire date for the employee.</i>	datetime	8
@RateChangeDate <i>Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the date the rate changed for the employee.</i>	datetime	8
@Rate <i>Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the new rate for the employee.</i>	money	8
@PayFrequency <i>Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the pay frequency for the employee.</i>	tinyint	1
@CurrentFlag <i>Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the current flag for the employee.</i>	Flag	1

SQL Script

```
CREATE PROCEDURE [HumanResources].[uspUpdateEmployeeHireInfo]
    @BusinessEntityID [int],
    @JobTitle [nvarchar] (50),
    @HireDate [datetime],
    @RateChangeDate [datetime],
    @Rate [money],
    @PayFrequency [tinyint],
    @CurrentFlag [dbo].[Flag]
WITH EXECUTE AS CALLER
AS
```

```
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        BEGIN TRANSACTION;

        UPDATE [HumanResources].[Employee]
        SET [JobTitle] = @JobTitle
            ,[HireDate] = @HireDate
            ,[CurrentFlag] = @CurrentFlag
        WHERE [BusinessEntityID] = @BusinessEntityID;

        INSERT INTO [HumanResources].[EmployeePayHistory]
            ([BusinessEntityID]
            ,[RateChangeDate]
            ,[Rate]
            ,[PayFrequency])
        VALUES (@BusinessEntityID, @RateChangeDate, @Rate, @PayFrequency);

        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        -- Rollback any active or uncommittable transactions before
        -- inserting information in the ErrorLog
        IF @@TRANCOUNT > 0
            BEGIN
                ROLLBACK TRANSACTION;
            END

        EXECUTE [dbo].[uspLogError];
    END CATCH;
END;
GO

EXEC sp_addextendedproperty N'MS_Description', N'Updates the Employee table and
inserts a new row in the EmployeePayHistory table with the values specified in the
input parameters.', 'SCHEMA', N'HumanResources', 'PROCEDURE', N'uspUpdateEmployee-
HireInfo', NULL, NULL
GO

EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored
procedure uspUpdateEmployeeHireInfo. Enter a valid BusinessEntityID from the
Employee table.', 'SCHEMA', N'HumanResources', 'PROCEDURE', N'uspUpdateEmployeeHire-
Info', 'PARAMETER', N'@BusinessEntityID'
GO

EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored
procedure uspUpdateEmployeeHireInfo. Enter the current flag for the employee.',
'SCHEMA', N'HumanResources', 'PROCEDURE', N'uspUpdateEmployeeHireInfo', 'PARAMETER',
N'@CurrentFlag'
GO

EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored
procedure uspUpdateEmployeeHireInfo. Enter a hire date for the employee.', 'SCHEMA',
N'HumanResources', 'PROCEDURE', N'uspUpdateEmployeeHireInfo', 'PARAMETER', N'@Hire-
Date'
GO

EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored
procedure uspUpdateEmployeeHireInfo. Enter a title for the employee.', 'SCHEMA',
N'HumanResources', 'PROCEDURE', N'uspUpdateEmployeeHireInfo', 'PARAMETER', N'@Job-
Title'
GO

EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored
procedure uspUpdateEmployeeHireInfo. Enter the pay frequency for the employee.',
'SCHEMA', N'HumanResources', 'PROCEDURE', N'uspUpdateEmployeeHireInfo', 'PARAMETER',
```

```
N'@PayFrequency'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored  
procedure uspUpdateEmployeeHireInfo. Enter the new rate for the employee.',  
'SCHEMA', N'HumanResources', 'PROCEDURE', N'uspUpdateEmployeeHireInfo', 'PARAMETER',  
N'@Rate'  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored  
procedure uspUpdateEmployeeHireInfo. Enter the date the rate changed for the  
employee.', 'SCHEMA', N'HumanResources', 'PROCEDURE', N'uspUpdateEmployeeHireInfo',  
'PARAMETER', N'@RateChangeDate'  
GO
```

Uses

[HumanResources].[Employee]
[HumanResources].[EmployeePayHistory]
[dbo].[uspLogError]
[dbo].[Flag]
HumanResources

[HumanResources].[uspUpdateEmployeeLogin]

MS_Description

Updates the Employee table with the values specified in the input parameters for the given BusinessEntityID.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@BusinessEntityID <i>Input parameter for the stored procedure uspUpdateEmployeeLogin. Enter a valid EmployeeID from the Employee table.</i>	int	4
@OrganizationNode <i>Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a valid ManagerID for the employee.</i>	hierarchyid	892
@LoginID <i>Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a valid login for the employee.</i>	nvarchar(256)	512
@JobTitle <i>Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a title for the employee.</i>	nvarchar(50)	100
@HireDate <i>Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a hire date for the employee.</i>	datetime	8
@CurrentFlag <i>Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the current flag for the employee.</i>	Flag	1

SQL Script

```
CREATE PROCEDURE [HumanResources].[uspUpdateEmployeeLogin]
    @BusinessEntityID [int],
    @OrganizationNode [hierarchyid],
    @LoginID [nvarchar] (256),
    @JobTitle [nvarchar] (50),
    @HireDate [datetime],
    @CurrentFlag [dbo].[Flag]
WITH EXECUTE AS CALLER
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
```

```
UPDATE [HumanResources].[Employee]
SET [OrganizationNode] = @OrganizationNode
    ,[LoginID] = @LoginID
    ,[JobTitle] = @JobTitle
    ,[HireDate] = @HireDate
    ,[CurrentFlag] = @CurrentFlag
WHERE [BusinessEntityID] = @BusinessEntityID;
END TRY
BEGIN CATCH
    EXECUTE [dbo].[uspLogError];
END CATCH;
END;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Updates the Employee table with the values specified in the input parameters for the given BusinessEntityID.', 'SCHEMA', N'HumanResources', 'PROCEDURE', N'uspUpdateEmployeeLogin', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored procedure uspUpdateEmployeeLogin. Enter a valid EmployeeID from the Employee table.', 'SCHEMA', N'HumanResources', 'PROCEDURE', N'uspUpdateEmployeeLogin', 'PARAMETER', N'@BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the current flag for the employee.', 'SCHEMA', N'HumanResources', 'PROCEDURE', N'uspUpdateEmployeeLogin', 'PARAMETER', N'@CurrentFlag'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a hire date for the employee.', 'SCHEMA', N'HumanResources', 'PROCEDURE', N'uspUpdateEmployeeLogin', 'PARAMETER', N'@HireDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a title for the employee.', 'SCHEMA', N'HumanResources', 'PROCEDURE', N'uspUpdateEmployeeLogin', 'PARAMETER', N'@JobTitle'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a valid login for the employee.', 'SCHEMA', N'HumanResources', 'PROCEDURE', N'uspUpdateEmployeeLogin', 'PARAMETER', N'@LoginID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a valid ManagerID for the employee.', 'SCHEMA', N'HumanResources', 'PROCEDURE', N'uspUpdateEmployeeLogin', 'PARAMETER', N'@OrganizationNode'
GO
```

Uses

[HumanResources].[Employee]
[dbo].[uspLogError]
[dbo].[Flag]
HumanResources

[HumanResources].[uspUpdateEmployeePersonalInfo]

MS_Description

Updates the Employee table with the values specified in the input parameters for the given EmployeeID.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@BusinessEntityID <i>Input parameter for the stored procedure uspUpdateEmployeePersonalInfo. Enter a valid BusinessEntityID from the HumanResources.Employee table.</i>	int	4
@NationalIDNumber <i>Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a national ID for the employee.</i>	nvarchar(15)	30
@BirthDate <i>Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a birth date for the employee.</i>	datetime	8
@MaritalStatus <i>Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a marital status for the employee.</i>	nchar	1
@Gender <i>Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a gender for the employee.</i>	nchar	1

SQL Script

```
CREATE PROCEDURE [HumanResources].[uspUpdateEmployeePersonalInfo]
    @BusinessEntityID [int],
    @NationalIDNumber [nvarchar](15),
    @BirthDate [datetime],
    @MaritalStatus [nchar](1),
    @Gender [nchar](1)
WITH EXECUTE AS CALLER
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        UPDATE [HumanResources].[Employee]
        SET [NationalIDNumber] = @NationalIDNumber
            , [BirthDate] = @BirthDate
```

```
        ,[MaritalStatus] = @MaritalStatus
        ,[Gender] = @Gender
    WHERE [BusinessEntityID] = @BusinessEntityID;
END TRY
BEGIN CATCH
    EXECUTE [dbo].[uspLogError];
END CATCH;
END;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Updates the Employee table with the
values specified in the input parameters for the given EmployeeID.', 'SCHEMA',
N'HumanResources', 'PROCEDURE', N'uspUpdateEmployeePersonalInfo', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored
procedure uspUpdateEmployeeHireInfo. Enter a birth date for the employee.',
'SCHEMA', N'HumanResources', 'PROCEDURE', N'uspUpdateEmployeePersonalInfo',
'PARAMETER', N'@BirthDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored
procedure uspUpdateEmployeePersonalInfo. Enter a valid BusinessEntityID from the
HumanResources.Employee table.', 'SCHEMA', N'HumanResources', 'PROCEDURE', N'usp-
UpdateEmployeePersonalInfo', 'PARAMETER', N'@BusinessEntityID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored
procedure uspUpdateEmployeeHireInfo. Enter a gender for the employee.', 'SCHEMA',
N'HumanResources', 'PROCEDURE', N'uspUpdateEmployeePersonalInfo', 'PARAMETER',
N'@Gender'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored
procedure uspUpdateEmployeeHireInfo. Enter a marital status for the employee.',
'SCHEMA', N'HumanResources', 'PROCEDURE', N'uspUpdateEmployeePersonalInfo',
'PARAMETER', N'@MaritalStatus'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored
procedure uspUpdateEmployeeHireInfo. Enter a national ID for the employee.',
'SCHEMA', N'HumanResources', 'PROCEDURE', N'uspUpdateEmployeePersonalInfo',
'PARAMETER', N'@NationalIDNumber'
GO
```

Uses

[HumanResources].[Employee]
[dbo].[uspLogError]
HumanResources

Table-valued Functions

Objects

Name
dbo.ufnGetContactInformation <i>Table value function returning the first name, last name, job title and contact type for a given contact.</i>

[dbo].[ufnGetContactInformation]

MS_Description

Table value function returning the first name, last name, job title and contact type for a given contact.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@PersonID <i>Input parameter for the table value function ufnGetContactInformation. Enter a valid PersonID from the Person.Contact table.</i>	int	4

SQL Script

```
CREATE FUNCTION [dbo].[ufnGetContactInformation] (@PersonID int)
RETURNS @retContactInformation TABLE
(
    -- Columns returned by the function
    [PersonID] int NOT NULL,
    [FirstName] [nvarchar](50) NULL,
    [LastName] [nvarchar](50) NULL,
    [JobTitle] [nvarchar](50) NULL,
    [BusinessEntityType] [nvarchar](50) NULL
)
AS
-- Returns the first name, last name, job title and business entity type for the
specified contact.
-- Since a contact can serve multiple roles, more than one row may be returned.
BEGIN
    IF @PersonID IS NOT NULL
        BEGIN
            IF EXISTS(SELECT * FROM [HumanResources].[Employee] e
                WHERE e.[BusinessEntityID] = @PersonID)
                INSERT INTO @retContactInformation
                SELECT @PersonID, p.FirstName, p.LastName, e.[JobTitle], 'Employee'
                FROM [HumanResources].[Employee] AS e
                INNER JOIN [Person].[Person] p
                ON p.[BusinessEntityID] = e.[BusinessEntityID]
                WHERE e.[BusinessEntityID] = @PersonID;

            IF EXISTS(SELECT * FROM [Purchasing].[Vendor] AS v
                INNER JOIN [Person].[BusinessEntityContact] bec
```

```
        ON bec.[BusinessEntityID] = v.[BusinessEntityID]
        WHERE bec.[PersonID] = @PersonID)
INSERT INTO @retContactInformation
    SELECT @PersonID, p.FirstName, p.LastName, ct.[Name], 'Vendor
Contact'
    FROM [Purchasing].[Vendor] AS v
        INNER JOIN [Person].[BusinessEntityContact] bec
        ON bec.[BusinessEntityID] = v.[BusinessEntityID]
        INNER JOIN [Person].ContactType ct
        ON ct.[ContactTypeID] = bec.[ContactTypeID]
        INNER JOIN [Person].[Person] p
        ON p.[BusinessEntityID] = bec.[PersonID]
    WHERE bec.[PersonID] = @PersonID;

IF EXISTS(SELECT * FROM [Sales].[Store] AS s
        INNER JOIN [Person].[BusinessEntityContact] bec
        ON bec.[BusinessEntityID] = s.[BusinessEntityID]
        WHERE bec.[PersonID] = @PersonID)
INSERT INTO @retContactInformation
    SELECT @PersonID, p.FirstName, p.LastName, ct.[Name], 'Store
Contact'
    FROM [Sales].[Store] AS s
        INNER JOIN [Person].[BusinessEntityContact] bec
        ON bec.[BusinessEntityID] = s.[BusinessEntityID]
        INNER JOIN [Person].ContactType ct
        ON ct.[ContactTypeID] = bec.[ContactTypeID]
        INNER JOIN [Person].[Person] p
        ON p.[BusinessEntityID] = bec.[PersonID]
    WHERE bec.[PersonID] = @PersonID;

IF EXISTS(SELECT * FROM [Person].[Person] AS p
        INNER JOIN [Sales].[Customer] AS c
        ON c.[PersonID] = p.[BusinessEntityID]
        WHERE p.[BusinessEntityID] = @PersonID AND c.[StoreID] IS NULL)
INSERT INTO @retContactInformation
    SELECT @PersonID, p.FirstName, p.LastName, NULL, 'Consumer'
    FROM [Person].[Person] AS p
        INNER JOIN [Sales].[Customer] AS c
        ON c.[PersonID] = p.[BusinessEntityID]
        WHERE p.[BusinessEntityID] = @PersonID AND c.[StoreID] IS NULL;

END

RETURN;
END;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Table value function returning the
first name, last name, job title and contact type for a given contact.', 'SCHEMA',
N'dbo', 'FUNCTION', N'ufnGetContactInformation', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the table value
function ufnGetContactInformation. Enter a valid PersonID from the Person.Contact
table.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufnGetContactInformation', 'PARAMETER',
N'@PersonID'
GO
```

Uses

[HumanResources].[Employee]
[Person].[BusinessEntityContact]
[Person].[ContactType]
[Person].[Person]
[Purchasing].[Vendor]
[Sales].[Customer]
[Sales].[Store]

Scalar-valued Functions

Objects

Name
dbo.ufnGetAccountingEndDate <i>Scalar function used in the uSalesOrderHeader trigger to set the starting account date.</i>
dbo.ufnGetAccountingStartDate <i>Scalar function used in the uSalesOrderHeader trigger to set the ending account date.</i>
dbo.ufnGetDocumentStatusText <i>Scalar function returning the text representation of the Status column in the Document table.</i>
dbo.ufnGetProductDealerPrice <i>Scalar function returning the dealer price for a given product on a particular order date.</i>
dbo.ufnGetProductListPrice <i>Scalar function returning the list price for a given product on a particular order date.</i>
dbo.ufnGetProductStandardCost <i>Scalar function returning the standard cost for a given product on a particular order date.</i>
dbo.ufnGetPurchaseOrderStatusText <i>Scalar function returning the text representation of the Status column in the PurchaseOrderHeader table.</i>
dbo.ufnGetSalesOrderStatusText <i>Scalar function returning the text representation of the Status column in the SalesOrderHeader table.</i>
dbo.ufnGetStock <i>Scalar function returning the quantity of inventory in LocationID 6 (Miscellaneous Storage) for a specified Product-ID.</i>
dbo.ufnLeadingZeros <i>Scalar function used by the Sales.Customer table to help set the account number.</i>

[dbo].[ufnGetAccountingEndDate]

MS_Description

Scalar function used in the uSalesOrderHeader trigger to set the starting account date.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE FUNCTION [dbo].[ufnGetAccountingEndDate]()
RETURNS [datetime]
AS
BEGIN
    RETURN DATEADD(millisecond, -2, CONVERT(datetime, '20040701', 112));
END;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Scalar function used in the uSales-
OrderHeader trigger to set the starting account date.', 'SCHEMA', N'dbo',
'FUNCTION', N'ufnGetAccountingEndDate', NULL, NULL
GO
```

[dbo].[ufnGetAccountingStartDate]

MS_Description

Scalar function used in the uSalesOrderHeader trigger to set the ending account date.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

SQL Script

```
CREATE FUNCTION [dbo].[ufnGetAccountingStartDate] ()
RETURNS [datetime]
AS
BEGIN
    RETURN CONVERT(datetime, '20030701', 112);
END;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Scalar function used in the uSales-
OrderHeader trigger to set the ending account date.', 'SCHEMA', N'dbo', 'FUNCTION',
N'ufnGetAccountingStartDate', NULL, NULL
GO
```

[dbo].[ufnGetDocumentStatusText]

MS_Description

Scalar function returning the text representation of the Status column in the Document table.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Status <i>Input parameter for the scalar function ufnGetDocumentStatusText. Enter a valid integer.</i>	tinyint	1

SQL Script

```
CREATE FUNCTION [dbo].[ufnGetDocumentStatusText] (@Status [tinyint])
RETURNS [nvarchar] (16)
AS
-- Returns the sales order status text representation for the status value.
BEGIN
    DECLARE @ret [nvarchar] (16);

    SET @ret =
        CASE @Status
            WHEN 1 THEN N'Pending approval'
            WHEN 2 THEN N'Approved'
            WHEN 3 THEN N'Obsolete'
            ELSE N'** Invalid **'
        END;

    RETURN @ret
END;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Scalar function returning the text
representation of the Status column in the Document table.', 'SCHEMA', N'dbo',
'FUNCTION', N'ufnGetDocumentStatusText', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the scalar
function ufnGetDocumentStatusText. Enter a valid integer.', 'SCHEMA', N'dbo',
'FUNCTION', N'ufnGetDocumentStatusText', 'PARAMETER', N'@Status'
GO
```

[dbo].[ufnGetProductDealerPrice]

MS_Description

Scalar function returning the dealer price for a given product on a particular order date.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ProductID <i>Input parameter for the scalar function ufnGetProductDealerPrice. Enter a valid ProductID from the Production.Product table.</i>	int	4
@OrderDate <i>Input parameter for the scalar function ufnGetProductDealerPrice. Enter a valid order date.</i>	datetime	8

SQL Script

```
CREATE FUNCTION [dbo].[ufnGetProductDealerPrice] (@ProductID [int], @OrderDate
[datetime])
RETURNS [money]
AS
-- Returns the dealer price for the product on a specific date.
BEGIN
    DECLARE @DealerPrice money;
    DECLARE @DealerDiscount money;

    SET @DealerDiscount = 0.60 -- 60% of list price

    SELECT @DealerPrice = plph.[ListPrice] * @DealerDiscount
    FROM [Production].[Product] p
        INNER JOIN [Production].[ProductListPriceHistory] plph
            ON p.[ProductID] = plph.[ProductID]
            AND p.[ProductID] = @ProductID
            AND @OrderDate BETWEEN plph.[StartDate] AND COALESCE(plph.[EndDate],
CONVERT(datetime, '99991231', 112)); -- Make sure we get all the prices!

    RETURN @DealerPrice;
END;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Scalar function returning the
dealer price for a given product on a particular order date.', 'SCHEMA', N'dbo',
```

```
'FUNCTION', N'ufnGetProductDealerPrice', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the scalar
function ufnGetProductDealerPrice. Enter a valid order date.', 'SCHEMA', N'dbo',
'FUNCTION', N'ufnGetProductDealerPrice', 'PARAMETER', N'@OrderDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the scalar
function ufnGetProductDealerPrice. Enter a valid ProductID from the
Production.Product table.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufnGetProductDealer-
Price', 'PARAMETER', N'@ProductID'
GO
```

Uses

[Production].[Product]

[Production].[ProductListPriceHistory]

[dbo].[ufnGetProductListPrice]

MS_Description

Scalar function returning the list price for a given product on a particular order date.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ProductID <i>Input parameter for the scalar function ufnGetProductListPrice. Enter a valid ProductID from the Production.Product table.</i>	int	4
@OrderDate <i>Input parameter for the scalar function ufnGetProductListPrice. Enter a valid order date.</i>	datetime	8

SQL Script

```
CREATE FUNCTION [dbo].[ufnGetProductListPrice](@ProductID [int], @OrderDate
[datetime])
RETURNS [money]
AS
BEGIN
    DECLARE @ListPrice money;

    SELECT @ListPrice = plph.[ListPrice]
    FROM [Production].[Product] p
    INNER JOIN [Production].[ProductListPriceHistory] plph
    ON p.[ProductID] = plph.[ProductID]
    AND p.[ProductID] = @ProductID
    AND @OrderDate BETWEEN plph.[StartDate] AND COALESCE(plph.[EndDate],
CONVERT(datetime, '99991231', 112)); -- Make sure we get all the prices!

    RETURN @ListPrice;
END;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Scalar function returning the list
price for a given product on a particular order date.', 'SCHEMA', N'dbo',
'FUNCTION', N'ufnGetProductListPrice', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the scalar
function ufnGetProductListPrice. Enter a valid order date.', 'SCHEMA', N'dbo',
'FUNCTION', N'ufnGetProductListPrice', 'PARAMETER', N'@OrderDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the scalar
```

```
function ufnGetProductListPrice. Enter a valid ProductID from the Production.Product  
table.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufnGetProductListPrice', 'PARAMETER',  
N'@ProductID'  
GO
```

Uses

[Production].[Product]

[Production].[ProductListPriceHistory]

[dbo].[ufnGetProductStandardCost]

MS_Description

Scalar function returning the standard cost for a given product on a particular order date.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ProductID <i>Input parameter for the scalar function ufnGetProductStandardCost. Enter a valid ProductID from the Production.Product table.</i>	int	4
@OrderDate <i>Input parameter for the scalar function ufnGetProductStandardCost. Enter a valid order date.</i>	datetime	8

SQL Script

```
CREATE FUNCTION [dbo].[ufnGetProductStandardCost] (@ProductID [int], @OrderDate [datetime])
RETURNS [money]
AS
-- Returns the standard cost for the product on a specific date.
BEGIN
    DECLARE @StandardCost money;

    SELECT @StandardCost = pch.[StandardCost]
    FROM [Production].[Product] p
        INNER JOIN [Production].[ProductCostHistory] pch
            ON p.[ProductID] = pch.[ProductID]
            AND p.[ProductID] = @ProductID
            AND @OrderDate BETWEEN pch.[StartDate] AND COALESCE(pch.[EndDate],
CONVERT(datetime, '99991231', 112)); -- Make sure we get all the prices!

    RETURN @StandardCost;
END;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Scalar function returning the
standard cost for a given product on a particular order date.', 'SCHEMA', N'dbo',
'FUNCTION', N'ufnGetProductStandardCost', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the scalar
function ufnGetProductStandardCost. Enter a valid order date.', 'SCHEMA', N'dbo',
'FUNCTION', N'ufnGetProductStandardCost', 'PARAMETER', N'@OrderDate'
GO
```

```
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the scalar  
function ufnGetProductStandardCost. Enter a valid ProductID from the  
Production.Product table.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufnGetProductStandard-  
Cost', 'PARAMETER', N'@ProductID'  
GO
```

Uses

[Production].[Product]

[Production].[ProductCostHistory]

[dbo].[ufnGetPurchaseOrderStatusText]

MS_Description

Scalar function returning the text representation of the Status column in the PurchaseOrderHeader table.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Status <i>Input parameter for the scalar function ufnGetPurchaseOrderStatusText. Enter a valid integer.</i>	tinyint	1

SQL Script

```
CREATE FUNCTION [dbo].[ufnGetPurchaseOrderStatusText] (@Status [tinyint])
RETURNS [nvarchar] (15)
AS
-- Returns the sales order status text representation for the status value.
BEGIN
    DECLARE @ret [nvarchar] (15);

    SET @ret =
        CASE @Status
            WHEN 1 THEN 'Pending'
            WHEN 2 THEN 'Approved'
            WHEN 3 THEN 'Rejected'
            WHEN 4 THEN 'Complete'
            ELSE '** Invalid **'
        END;

    RETURN @ret
END;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Scalar function returning the text
representation of the Status column in the PurchaseOrderHeader table.', 'SCHEMA',
N'dbo', 'FUNCTION', N'ufnGetPurchaseOrderStatusText', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the scalar
function ufnGetPurchaseOrderStatusText. Enter a valid integer.', 'SCHEMA', N'dbo',
'FUNCTION', N'ufnGetPurchaseOrderStatusText', 'PARAMETER', N'@Status'
GO
```

[dbo].[ufnGetSalesOrderStatusText]

MS_Description

Scalar function returning the text representation of the Status column in the SalesOrderHeader table.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@Status <i>Input parameter for the scalar function ufnGetSalesOrderStatusText. Enter a valid integer.</i>	tinyint	1

SQL Script

```
CREATE FUNCTION [dbo].[ufnGetSalesOrderStatusText](@Status [tinyint])
RETURNS [nvarchar] (15)
AS
-- Returns the sales order status text representation for the status value.
BEGIN
    DECLARE @ret [nvarchar] (15);

    SET @ret =
        CASE @Status
            WHEN 1 THEN 'In process'
            WHEN 2 THEN 'Approved'
            WHEN 3 THEN 'Backordered'
            WHEN 4 THEN 'Rejected'
            WHEN 5 THEN 'Shipped'
            WHEN 6 THEN 'Cancelled'
            ELSE '** Invalid **'
        END;

    RETURN @ret
END;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Scalar function returning the text
representation of the Status column in the SalesOrderHeader table.', 'SCHEMA',
N'dbo', 'FUNCTION', N'ufnGetSalesOrderStatusText', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the scalar
function ufnGetSalesOrderStatusText. Enter a valid integer.', 'SCHEMA', N'dbo',
'FUNCTION', N'ufnGetSalesOrderStatusText', 'PARAMETER', N'@Status'
GO
```



[dbo].[ufnGetStock]

MS_Description

Scalar function returning the quantity of inventory in LocationID 6 (Miscellaneous Storage)for a specified ProductID.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True

Parameters

Name	Data Type	Max Length (Bytes)
@ProductID <i>Input parameter for the scalar function ufnGetStock. Enter a valid Product-ID from the Production.ProductInventory table.</i>	int	4

SQL Script

```
CREATE FUNCTION [dbo].[ufnGetStock] (@ProductID [int])
RETURNS [int]
AS
-- Returns the stock level for the product. This function is used internally only
BEGIN
    DECLARE @ret int;

    SELECT @ret = SUM(p.[Quantity])
    FROM [Production].[ProductInventory] p
    WHERE p.[ProductID] = @ProductID
        AND p.[LocationID] = '6'; -- Only look at inventory in the misc storage

    IF (@ret IS NULL)
        SET @ret = 0

    RETURN @ret
END;
GO
EXEC sp_addextendedproperty N'MS_Description', N'Scalar function returning the
quantity of inventory in LocationID 6 (Miscellaneous Storage)for a specified Product-
ID.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufnGetStock', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the scalar
function ufnGetStock. Enter a valid ProductID from the Production.ProductInventory
table.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufnGetStock', 'PARAMETER', N'@ProductID'
GO
```

Uses

[Production].[ProductInventory]

[dbo].[ufnLeadingZeros]

MS_Description

Scalar function used by the Sales.Customer table to help set the account number.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True
Schema Bound	True

Parameters

Name	Data Type	Max Length (Bytes)
@Value <i>Input parameter for the scalar function ufnLeadingZeros. Enter a valid integer.</i>	int	4

SQL Script

```
CREATE FUNCTION [dbo].[ufnLeadingZeros] (  
    @Value int  
)  
RETURNS varchar(8)  
WITH SCHEMABINDING  
AS  
BEGIN  
    DECLARE @ReturnValue varchar(8);  
  
    SET @ReturnValue = CONVERT(varchar(8), @Value);  
    SET @ReturnValue = REPLICATE('0', 8 - DATALENGTH(@ReturnValue)) + @ReturnValue;  
  
    RETURN (@ReturnValue);  
END;  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Scalar function used by the  
Sales.Customer table to help set the account number.', 'SCHEMA', N'dbo', 'FUNCTION',  
N'ufnLeadingZeros', NULL, NULL  
GO  
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the scalar  
function ufnLeadingZeros. Enter a valid integer.', 'SCHEMA', N'dbo', 'FUNCTION',  
N'ufnLeadingZeros', 'PARAMETER', N'@Value'  
GO
```

Used By

[Sales].[Customer]

Database Triggers

Objects

Name
ddlDatabaseTriggerLog <i>Database trigger to audit all of the DDL changes made to the AdventureWorks 2012 database.</i>

ddlDatabaseTriggerLog

MS_Description

Database trigger to audit all of the DDL changes made to the AdventureWorks 2012 database.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True
Disabled	True

SQL Script

```
CREATE TRIGGER [ddlDatabaseTriggerLog] ON DATABASE
FOR DDL_DATABASE_LEVEL_EVENTS AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @data XML;
    DECLARE @schema sysname;
    DECLARE @object sysname;
    DECLARE @eventType sysname;

    SET @data = EVENTDATA();
    SET @eventType = @data.value('/EVENT_INSTANCE/EventType [1]', 'sysname');
    SET @schema = @data.value('/EVENT_INSTANCE/SchemaName [1]', 'sysname');
    SET @object = @data.value('/EVENT_INSTANCE/ObjectName [1]', 'sysname')

    IF @object IS NOT NULL
        PRINT ' ' + @eventType + ' - ' + @schema + '.' + @object;
    ELSE
        PRINT ' ' + @eventType + ' - ' + @schema;

    IF @eventType IS NULL
        PRINT CONVERT(nvarchar(max), @data);

    INSERT [dbo].[DatabaseLog]
    (
        [PostTime],
        [DatabaseUser],
        [Event],
        [Schema],
        [Object],
        [TSQL],
        [XmlEvent]
    )
    VALUES
    (
```

```
        GETDATE(),
        CONVERT(sysname, CURRENT_USER),
        @eventType,
        CONVERT(sysname, @schema),
        CONVERT(sysname, @object),
        @data.value('(/EVENT_INSTANCE/TSQLCommand)[1]', 'nvarchar(max)'),
        @data
    );
END;
GO
DISABLE TRIGGER ddlDatabaseTriggerLog ON DATABASE
GO
EXEC sp_addextendedproperty N'MS_Description', N'Database trigger to audit all of
the DDL changes made to the AdventureWorks 2012 database.', 'TRIGGER', N'ddlDatabase-
TriggerLog', NULL, NULL, NULL, NULL
GO
```

 **User-Defined Data Types**

Objects

Name
dbo.AccountNumber
dbo.Flag
dbo.Name
dbo.NameStyle
dbo.OrderNumber
dbo.Phone

 **[dbo].[AccountNumber]**

Properties

Property	Value
Allow Nulls	True
Base Type Name	nvarchar
Length	15

SQL Script

```
CREATE TYPE [dbo].[AccountNumber] FROM nvarchar (15) NULL  
GO
```

Used By

[Purchasing].[Vendor]
[Sales].[SalesOrderHeader]



Properties

Property	Value
Allow Nulls	False
Base Type Name	bit
Length	1

SQL Script

```
CREATE TYPE [dbo].[Flag] FROM bit NOT NULL  
GO
```

Used By

[HumanResources].[Employee]
[Person].[StateProvince]
[Production].[Product]
[Production].[ProductProductPhoto]
[Purchasing].[Vendor]
[Sales].[SalesOrderHeader]
[Person].[vStateProvinceCountryRegion]
[HumanResources].[uspUpdateEmployeeHireInfo]
[HumanResources].[uspUpdateEmployeeLogin]


[dbo].[Name]

Properties

Property	Value
Allow Nulls	True
Base Type Name	nvarchar
Length	50

SQL Script

```
CREATE TYPE [dbo].[Name] FROM nvarchar (50) NULL
GO
```

Used By

- [HumanResources].[Department]
- [HumanResources].[Shift]
- [Person].[AddressType]
- [Person].[ContactType]
- [Person].[CountryRegion]
- [Person].[Person]
- [Person].[PhoneNumberType]
- [Person].[StateProvince]
- [Production].[Culture]
- [Production].[Location]
- [Production].[Product]
- [Production].[ProductCategory]
- [Production].[ProductModel]
- [Production].[ProductReview]
- [Production].[ProductSubcategory]
- [Production].[ScrapReason]
- [Production].[UnitMeasure]
- [Purchasing].[ShipMethod]
- [Purchasing].[Vendor]
- [Sales].[Currency]
- [Sales].[SalesReason]
- [Sales].[SalesTaxRate]
- [Sales].[SalesTerritory]
- [Sales].[Store]
- [HumanResources].[vEmployee]
- [HumanResources].[vEmployeeDepartment]
- [HumanResources].[vEmployeeDepartmentHistory]
- [Person].[vAdditionalContactInfo]
- [Person].[vStateProvinceCountryRegion]
- [Production].[vProductAndDescription]
- [Production].[vProductModelCatalogDescription]
- [Production].[vProductModelInstructions]
- [Purchasing].[vVendorWithAddresses]

Project > Isrep17 > User databases > AdventureWorks > Programmability > Types > User-Defined Data Types
> dbo.Name

[Purchasing].[vVendorWithContacts]
[Sales].[vIndividualCustomer]
[Sales].[vSalesPerson]
[Sales].[vSalesPersonSalesByFiscalYears]
[Sales].[vStoreWithAddresses]
[Sales].[vStoreWithContacts]
[Sales].[vStoreWithDemographics]

 **[dbo].[NameStyle]**

Properties

Property	Value
Allow Nulls	False
Base Type Name	bit
Length	1

SQL Script

```
CREATE TYPE [dbo].[NameStyle] FROM bit NOT NULL  
GO
```

Used By

[Person].[Person]

 **[dbo].[OrderNumber]**

Properties

Property	Value
Allow Nulls	True
Base Type Name	nvarchar
Length	25

SQL Script

```
CREATE TYPE [dbo].[OrderNumber] FROM nvarchar (25) NULL  
GO
```

Used By

[Sales].[SalesOrderHeader]



Properties

Property	Value
Allow Nulls	True
Base Type Name	nvarchar
Length	25

SQL Script

```
CREATE TYPE [dbo].[Phone] FROM nvarchar (25) NULL  
GO
```

Used By

[Person].[PersonPhone]
[HumanResources].[vEmployee]
[Purchasing].[vVendorWithContacts]
[Sales].[vIndividualCustomer]
[Sales].[vSalesPerson]
[Sales].[vStoreWithContacts]

XML Schema Collections

Objects

Name
HumanResources.HRResumeSchemaCollection <i>Collection of XML schemas for the Resume column in the HumanResources.JobCandidate table.</i>
Person.AdditionalContactInfoSchemaCollection <i>Collection of XML schemas for the AdditionalContactInfo column in the Person.Contact table.</i>
Person.IndividualSurveySchemaCollection <i>Collection of XML schemas for the Demographics column in the Person.Person table.</i>
Production.ManuInstructionsSchemaCollection <i>Collection of XML schemas for the Instructions column in the Production.ProductModel table.</i>
Production.ProductDescriptionSchemaCollection <i>Collection of XML schemas for the CatalogDescription column in the Production.ProductModel table.</i>
Sales.StoreSurveySchemaCollection <i>Collection of XML schemas for the Demographics column in the Sales.Store table.</i>

[HumanResources].[HRRResumeSchemaCollection]

MS_Description

Collection of XML schemas for the Resume column in the HumanResources.JobCandidate table.

Dependent Columns

- [HumanResources].[JobCandidate].[Resume]

SQL Script

```
CREATE XML SCHEMA COLLECTION [HumanResources].[HRRResumeSchemaCollection]
AS N'<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:t="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume"
targetNamespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-
works/Resume" elementFormDefault="qualified">
  <xsd:element name="Address" type="t:AddressType" />
  <xsd:element name="Education" type="t:EducationType" />
  <xsd:element name="Employment" type="t:EmploymentType" />
  <xsd:element name="Location" type="t:LocationType" />
  <xsd:element name="Name" type="t:NameType" />
  <xsd:element name="Resume" type="t:ResumeType" />
  <xsd:element name="Telephone" type="t:TelephoneType" />
  <xsd:complexType name="AddressType">
    <xsd:complexContent>
      <xsd:restriction base="xsd:anyType">
        <xsd:sequence>
          <xsd:element name="Addr.Type" type="xsd:string" />
          <xsd:element name="Addr.OrgName" type="xsd:string" minOccurs="0" />
          <xsd:element name="Addr.Street" type="xsd:string" maxOccurs="unbounded" />
          <xsd:element name="Addr.Location">
            <xsd:complexType>
              <xsd:complexContent>
                <xsd:restriction base="xsd:anyType">
                  <xsd:sequence>
                    <xsd:element ref="t:Location" />
                  </xsd:sequence>
                </xsd:restriction>
              </xsd:complexContent>
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="Addr.PostalCode" type="xsd:string" />
          <xsd:element name="Addr.Telephone" minOccurs="0">
            <xsd:complexType>
              <xsd:complexContent>
                <xsd:restriction base="xsd:anyType">
                  <xsd:sequence>
                    <xsd:element ref="t:Telephone" maxOccurs="unbounded" />
                  </xsd:sequence>
                </xsd:restriction>
              </xsd:complexContent>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```

```
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="EducationType">
    <xsd:complexContent>
      <xsd:restriction base="xsd:anyType">
        <xsd:sequence>
          <xsd:element name="Edu.Level" type="xsd:string" />
          <xsd:element name="Edu.StartDate" type="xsd:date" />
          <xsd:element name="Edu.EndDate" type="xsd:date" />
          <xsd:element name="Edu.Degree" type="xsd:string" minOccurs="0" />
          <xsd:element name="Edu.Major" type="xsd:string" minOccurs="0" />
          <xsd:element name="Edu.Minor" type="xsd:string" minOccurs="0" />
          <xsd:element name="Edu.GPA" type="xsd:string" minOccurs="0" />
          <xsd:element name="Edu.GPAAternate" type="xsd:decimal" minOccurs="0" />
          <xsd:element name="Edu.GPAScale" type="xsd:decimal" minOccurs="0" />
          <xsd:element name="Edu.School" type="xsd:string" minOccurs="0" />
          <xsd:element name="Edu.Location" minOccurs="0">
            <xsd:complexType>
              <xsd:complexContent>
                <xsd:restriction base="xsd:anyType">
                  <xsd:sequence>
                    <xsd:element ref="t:Location" />
                  </xsd:sequence>
                </xsd:restriction>
              </xsd:complexContent>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="EmploymentType">
    <xsd:complexContent>
      <xsd:restriction base="xsd:anyType">
        <xsd:sequence>
          <xsd:element name="Emp.StartDate" type="xsd:date" minOccurs="0" />
          <xsd:element name="Emp.EndDate" type="xsd:date" minOccurs="0" />
          <xsd:element name="Emp.OrgName" type="xsd:string" />
          <xsd:element name="Emp.JobTitle" type="xsd:string" />
          <xsd:element name="Emp.Responsibility" type="xsd:string" />
          <xsd:element name="Emp.FunctionCategory" type="xsd:string" minOccurs="0" />
        </xsd:sequence>
        <xsd:element name="Emp.IndustryCategory" type="xsd:string" minOccurs="0" />
        <xsd:element name="Emp.Location" minOccurs="0">
          <xsd:complexType>
            <xsd:complexContent>
              <xsd:restriction base="xsd:anyType">
                <xsd:sequence>
                  <xsd:element ref="t:Location" />
                </xsd:sequence>
              </xsd:restriction>
            </xsd:complexContent>
          </xsd:complexType>
        </xsd:element>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>
```

```
        </xsd:sequence>
    </xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="LocationType">
    <xsd:complexContent>
        <xsd:restriction base="xsd:anyType">
            <xsd:sequence>
                <xsd:element name="Loc.CountryRegion" type="xsd:string" />
                <xsd:element name="Loc.State" type="xsd:string" minOccurs="0" />
                <xsd:element name="Loc.City" type="xsd:string" minOccurs="0" />
            </xsd:sequence>
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="NameType">
    <xsd:complexContent>
        <xsd:restriction base="xsd:anyType">
            <xsd:sequence>
                <xsd:element name="Name.Prefix" type="xsd:string" minOccurs="0" />
                <xsd:element name="Name.First" type="xsd:string" />
                <xsd:element name="Name.Middle" type="xsd:string" minOccurs="0" />
                <xsd:element name="Name.Last" type="xsd:string"/>
                <xsd:element name="Name.Suffix" type="xsd:string" minOccurs="0" />
            </xsd:sequence>
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ResumeType">
    <xsd:complexContent>
        <xsd:restriction base="xsd:anyType">
            <xsd:sequence>
                <xsd:element ref="t:Name" />
                <xsd:element name="Skills" type="xsd:string" minOccurs="0" />
                <xsd:element ref="t:Employment" maxOccurs="unbounded" />
                <xsd:element ref="t:Education" maxOccurs="unbounded" />
                <xsd:element ref="t:Address" maxOccurs="unbounded" />
                <xsd:element ref="t:Telephone" minOccurs="0" />
                <xsd:element name="EMail" type="xsd:string" minOccurs="0" />
                <xsd:element name="WebSite" type="xsd:string" minOccurs="0" />
            </xsd:sequence>
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="TelephoneType">
    <xsd:complexContent>
        <xsd:restriction base="xsd:anyType">
            <xsd:sequence>
                <xsd:element name="Tel.Type" type="xsd:anyType" minOccurs="0" />
                <xsd:element name="Tel.IntlCode" type="xsd:int" minOccurs="0" />
                <xsd:element name="Tel.AreaCode" type="xsd:int" minOccurs="0" />
                <xsd:element name="Tel.Number" type="xsd:string" />
                <xsd:element name="Tel.Extension" type="xsd:int" minOccurs="0" />
            </xsd:sequence>
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>
```

Project > Isrep17 > User databases > AdventureWorks > Programmability > Types > XML Schema Collections
> HumanResources.HRRResumeSchemaCollection

```
</xsd:schema>'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Collection of XML schemas for the
Resume column in the HumanResources.JobCandidate table.', 'SCHEMA', N'Human-
Resources', 'XML_SCHEMA_COLLECTION', N'HRRResumeSchemaCollection', NULL, NULL
GO
```

Uses

HumanResources

Used By

[HumanResources].[JobCandidate]

[Person].[AdditionalContactInfoSchemaCollection]

MS_Description

Collection of XML schemas for the AdditionalContactInfo column in the Person.Contact table.

Dependent Columns

- [Person].[Person].[AdditionalContactInfo]

SQL Script

```
CREATE XML SCHEMA COLLECTION [Person].[AdditionalContactInfoSchemaCollection]
AS N'<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:t="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo"
targetNamespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-
works/ContactInfo">
  <xsd:element name="AdditionalContactInfo">
    <xsd:complexType mixed="true">
      <xsd:complexContent mixed="true">
        <xsd:restriction base="xsd:anyType">
          <xsd:sequence>
            <xsd:any
namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact-
Record" http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes"
minOccurs="0" maxOccurs="unbounded" />
          </xsd:sequence>
        </xsd:restriction>
      </xsd:complexContent>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:t="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact-
Record" targetNamespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-
works/ContactRecord">
  <xsd:element name="ContactRecord">
    <xsd:complexType mixed="true">
      <xsd:complexContent mixed="true">
        <xsd:restriction base="xsd:anyType">
          <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:any
namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact-
Types" />
            </xsd:choice>
            <xsd:attribute name="date" type="xsd:date" />
          </xsd:restriction>
        </xsd:complexContent>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:t="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact-
Types" targetNamespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-
works/ContactTypes" elementFormDefault="qualified">
  <xsd:element name="eMail" type="t:eMailType" />
  <xsd:element name="facsimileTelephoneNumber" type="t:phoneNumberType" />
</xsd:schema>
```

```

<xsd:element name="homePostalAddress" type="t:addressType" />
<xsd:element name="internationalISDNNNumber" type="t:phoneNumberType" />
<xsd:element name="mobile" type="t:phoneNumberType" />
<xsd:element name="pager" type="t:phoneNumberType" />
<xsd:element name="physicalDeliveryOfficeName" type="t:addressType" />
<xsd:element name="registeredAddress" type="t:addressType" />
<xsd:element name="telephoneNumber" type="t:phoneNumberType" />
<xsd:element name="telexNumber" type="t:phoneNumberType" />
<xsd:complexType name="addressType">
  <xsd:complexContent>
    <xsd:restriction base="xsd:anyType">
      <xsd:sequence>
        <xsd:element name="Street" type="xsd:string" maxOccurs="2" />
        <xsd:element name="City" type="xsd:string" />
        <xsd:element name="StateProvince" type="xsd:string" />
        <xsd:element name="PostalCode" type="xsd:string" minOccurs="0" />
        <xsd:element name="CountryRegion" type="xsd:string" />
        <xsd:element name="SpecialInstructions" type="t:specialInstructionsType"
minOccurs="0" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="eMailType">
  <xsd:complexContent>
    <xsd:restriction base="xsd:anyType">
      <xsd:sequence>
        <xsd:element name="eMailAddress" type="xsd:string" />
        <xsd:element name="SpecialInstructions" type="t:specialInstructionsType"
minOccurs="0" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="phoneNumberType">
  <xsd:complexContent>
    <xsd:restriction base="xsd:anyType">
      <xsd:sequence>
        <xsd:element name="number">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:pattern value="[0-9\(\)\-]*" />
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="SpecialInstructions" type="t:specialInstructionsType"
minOccurs="0" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="specialInstructionsType" mixed="true">
  <xsd:complexContent mixed="true">
    <xsd:restriction base="xsd:anyType">
      <xsd:sequence>
        <xsd:any
namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact-
Types" minOccurs="0" maxOccurs="unbounded" />

```

```
</xsd:sequence>
  </xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
</xsd:schema>'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Collection of XML schemas for the
AdditionalContactInfo column in the Person.Contact table.', 'SCHEMA', N'Person',
'XML_SCHEMA_COLLECTION', N'AdditionalContactInfoSchemaCollection', NULL, NULL
GO
```

Uses

Person

Used By

[Person].[Person]

[Person].[IndividualSurveySchemaCollection]

MS_Description

Collection of XML schemas for the Demographics column in the Person.Person table.

Dependent Columns

- [Person].[Person].[Demographics]

SQL Script

```
CREATE XML SCHEMA COLLECTION [Person].[IndividualSurveySchemaCollection]
AS N'<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:t="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Individual-
Survey" targetNamespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-
works/IndividualSurvey" elementFormDefault="qualified">
  <xsd:element name="IndividualSurvey">
    <xsd:complexType>
      <xsd:complexContent>
        <xsd:restriction base="xsd:anyType">
          <xsd:sequence>
            <xsd:element name="TotalPurchaseYTD" type="xsd:decimal" minOccurs="0" />
            <xsd:element name="DateFirstPurchase" type="xsd:date" minOccurs="0" />
            <xsd:element name="BirthDate" type="xsd:date" minOccurs="0" />
            <xsd:element name="MaritalStatus" type="xsd:string" minOccurs="0" />
            <xsd:element name="YearlyIncome" type="t:SalaryType" minOccurs="0" />
            <xsd:element name="Gender" type="xsd:string" minOccurs="0" />
            <xsd:element name="TotalChildren" type="xsd:int" minOccurs="0" />
            <xsd:element name="NumberChildrenAtHome" type="xsd:int" minOccurs="0" />
            <xsd:element name="Education" type="xsd:string" minOccurs="0" />
            <xsd:element name="Occupation" type="xsd:string" minOccurs="0" />
            <xsd:element name="HomeOwnerFlag" type="xsd:string" minOccurs="0" />
            <xsd:element name="NumberCarsOwned" type="xsd:int" minOccurs="0" />
            <xsd:element name="Hobby" type="xsd:string" minOccurs="0" max-
Occurs="unbounded" />
            <xsd:element name="CommuteDistance" type="t:MileRangeType" minOccurs="0"
/>
            <xsd:element name="Comments" type="xsd:string" minOccurs="0" />
          </xsd:sequence>
        </xsd:restriction>
      </xsd:complexContent>
    </xsd:complexType>
  </xsd:element>
  <xsd:simpleType name="MileRangeType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="0-1 Miles" />
      <xsd:enumeration value="1-2 Miles" />
      <xsd:enumeration value="2-5 Miles" />
      <xsd:enumeration value="5-10 Miles" />
      <xsd:enumeration value="10+ Miles" />
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="SalaryType">
    <xsd:restriction base="xsd:string">
```

```
<xsd:enumeration value="0-25000" />
<xsd:enumeration value="25001-50000" />
<xsd:enumeration value="50001-75000" />
<xsd:enumeration value="75001-100000" />
<xsd:enumeration value="greater than 100000" />
</xsd:restriction>
</xsd:simpleType>
</xsd:schema>'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Collection of XML schemas for the
Demographics column in the Person.Person table.', 'SCHEMA', N'Person', 'XML SCHEMA
COLLECTION', N'IndividualSurveySchemaCollection', NULL, NULL
GO
```

Uses

Person

Used By

[Person].[Person]

[Production].[ManuInstructionsSchemaCollection]

MS_Description

Collection of XML schemas for the Instructions column in the Production.ProductModel table.

Dependent Columns

- [Production].[ProductModel].[Instructions]

SQL Script

```
CREATE XML SCHEMA COLLECTION [Production].[ManuInstructionsSchemaCollection]
AS N'<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:t="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
ManuInstructions" target-
Namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product-
ModelManuInstructions" elementFormDefault="qualified">
  <xsd:element name="root">
    <xsd:complexType mixed="true">
      <xsd:complexContent mixed="true">
        <xsd:restriction base="xsd:anyType">
          <xsd:sequence>
            <xsd:element name="Location" maxOccurs="unbounded">
              <xsd:complexType mixed="true">
                <xsd:complexContent mixed="true">
                  <xsd:restriction base="xsd:anyType">
                    <xsd:sequence>
                      <xsd:element name="step" type="t:StepType" max-
Occurs="unbounded" />
                    </xsd:sequence>
                    <xsd:attribute name="LocationID" type="xsd:integer"
use="required" />
                    <xsd:attribute name="SetupHours" type="xsd:decimal" />
                    <xsd:attribute name="MachineHours" type="xsd:decimal" />
                    <xsd:attribute name="LaborHours" type="xsd:decimal" />
                    <xsd:attribute name="LotSize" type="xsd:decimal" />
                  </xsd:restriction>
                </xsd:complexContent>
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
        </xsd:restriction>
      </xsd:complexContent>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="StepType" mixed="true">
    <xsd:complexContent mixed="true">
      <xsd:restriction base="xsd:anyType">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
          <xsd:element name="tool" type="xsd:string" />
          <xsd:element name="material" type="xsd:string" />
          <xsd:element name="blueprint" type="xsd:string" />
          <xsd:element name="specs" type="xsd:string" />
          <xsd:element name="diag" type="xsd:string" />
        </xsd:choice>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```

```
</xsd:choice>
  </xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
</xsd:schema>'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Collection of XML schemas for the
Instructions column in the Production.ProductModel table.', 'SCHEMA', N'Production',
'XML_SCHEMA_COLLECTION', N'ManuInstructionsSchemaCollection', NULL, NULL
GO
```

Uses

Production

Used By

[Production].[ProductModel]

[Production].[ProductDescriptionSchemaCollection]

MS_Description

Collection of XML schemas for the CatalogDescription column in the Production.ProductModel table.

Dependent Columns

- [Production].[ProductModel].[CatalogDescription]

SQL Script

```
CREATE XML SCHEMA COLLECTION [Production].[ProductDescriptionSchemaCollection]
AS N'<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:t="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
WarrAndMain" target-
Namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product-
ModelWarrAndMain" elementFormDefault="qualified">
  <xsd:element name="Maintenance">
    <xsd:complexType>
      <xsd:complexContent>
        <xsd:restriction base="xsd:anyType">
          <xsd:sequence>
            <xsd:element name="NoOfYears" type="xsd:string" />
            <xsd:element name="Description" type="xsd:string" />
          </xsd:sequence>
        </xsd:restriction>
      </xsd:complexContent>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Warranty">
    <xsd:complexType>
      <xsd:complexContent>
        <xsd:restriction base="xsd:anyType">
          <xsd:sequence>
            <xsd:element name="WarrantyPeriod" type="xsd:string" />
            <xsd:element name="Description" type="xsd:string" />
          </xsd:sequence>
        </xsd:restriction>
      </xsd:complexContent>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ns1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product-
ModelWarrAndMain" xmlns:t="http://schemas.microsoft.com/sqlserver/2004/07/adventure-
works/ProductModelDescription" target-
Namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product-
ModelDescription" elementFormDefault="qualified">
  <xsd:import namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-
works/ProductModelWarrAndMain" />
  <xsd:element name="Code" type="xsd:string" />
  <xsd:element name="Description" type="xsd:string" />
  <xsd:element name="ProductDescription" type="t:ProductDescription" />
  <xsd:element name="Taxonomy" type="xsd:string" />
  <xsd:complexType name="Category">
```

```
<xsd:complexContent>
  <xsd:restriction base="xsd:anyType">
    <xsd:sequence>
      <xsd:element ref="t:Taxonomy" />
      <xsd:element ref="t:Code" />
      <xsd:element ref="t:Description" minOccurs="0" />
    </xsd:sequence>
  </xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="Features" mixed="true">
  <xsd:complexContent mixed="true">
    <xsd:restriction base="xsd:anyType">
      <xsd:sequence>
        <xsd:element ref="ns1:Warranty" />
        <xsd:element ref="ns1:Maintenance" />
        <xsd:any namespace="##other" processContents="skip" minOccurs="0" max-
Occurs="unbounded" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="Manufacturer">
  <xsd:complexContent>
    <xsd:restriction base="xsd:anyType">
      <xsd:sequence>
        <xsd:element name="Name" type="xsd:string" minOccurs="0" />
        <xsd:element name="CopyrightURL" type="xsd:string" minOccurs="0" />
        <xsd:element name="Copyright" type="xsd:string" minOccurs="0" />
        <xsd:element name="ProductURL" type="xsd:string" minOccurs="0" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="Picture">
  <xsd:complexContent>
    <xsd:restriction base="xsd:anyType">
      <xsd:sequence>
        <xsd:element name="Name" type="xsd:string" minOccurs="0" />
        <xsd:element name="Angle" type="xsd:string" minOccurs="0" />
        <xsd:element name="Size" type="xsd:string" minOccurs="0" />
        <xsd:element name="ProductPhotoID" type="xsd:integer" minOccurs="0" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ProductDescription">
  <xsd:complexContent>
    <xsd:restriction base="xsd:anyType">
      <xsd:sequence>
        <xsd:element name="Summary" type="t:Summary" minOccurs="0" />
        <xsd:element name="Manufacturer" type="t:Manufacturer" minOccurs="0" />
        <xsd:element name="Features" type="t:Features" minOccurs="0" max-
Occurs="unbounded" />
        <xsd:element name="Picture" type="t:Picture" minOccurs="0" max-
Occurs="unbounded" />
        <xsd:element name="Category" type="t:Category" minOccurs="0" max-
Occurs="unbounded" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
```

```
<xsd:element name="Specifications" type="t:Specifications" minOccurs="0"
maxOccurs="unbounded" />
</xsd:sequence>
<xsd:attribute name="ProductModelID" type="xsd:string" />
<xsd:attribute name="ProductModelName" type="xsd:string" />
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="Specifications" mixed="true">
<xsd:complexContent mixed="true">
<xsd:restriction base="xsd:anyType">
<xsd:sequence>
<xsd:any processContents="skip" minOccurs="0" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="Summary" mixed="true">
<xsd:complexContent mixed="true">
<xsd:restriction base="xsd:anyType">
<xsd:sequence>
<xsd:any namespace="http://www.w3.org/1999/xhtml" processContents="skip"
minOccurs="0" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
</xsd:schema>'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Collection of XML schemas for the
CatalogDescription column in the Production.ProductModel table.', 'SCHEMA',
N'Production', 'XML_SCHEMA_COLLECTION', N'ProductDescriptionSchemaCollection', NULL,
NULL
GO
```

Uses

Production

Used By

[Production].[ProductModel]

[Sales].[StoreSurveySchemaCollection]

MS_Description

Collection of XML schemas for the Demographics column in the Sales.Store table.

Dependent Columns

- [Sales].[Store].[Demographics]

SQL Script

```
CREATE XML SCHEMA COLLECTION [Sales].[StoreSurveySchemaCollection]
AS N'<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:t="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey"
targetNamespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-
works/StoreSurvey" elementFormDefault="qualified">
  <xsd:element name="StoreSurvey">
    <xsd:complexType>
      <xsd:complexContent>
        <xsd:restriction base="xsd:anyType">
          <xsd:sequence>
            <xsd:element name="ContactName" type="xsd:string" minOccurs="0" />
            <xsd:element name="JobTitle" type="xsd:string" minOccurs="0" />
            <xsd:element name="AnnualSales" type="xsd:decimal" minOccurs="0" />
            <xsd:element name="AnnualRevenue" type="xsd:decimal" minOccurs="0" />
            <xsd:element name="BankName" type="xsd:string" minOccurs="0" />
            <xsd:element name="BusinessType" type="t:BusinessType" minOccurs="0" />
            <xsd:element name="YearOpened" type="xsd:gYear" minOccurs="0" />
            <xsd:element name="Specialty" type="t:SpecialtyType" minOccurs="0" />
            <xsd:element name="SquareFeet" type="xsd:float" minOccurs="0" />
            <xsd:element name="Brands" type="t:BrandType" minOccurs="0" />
            <xsd:element name="Internet" type="t:InternetType" minOccurs="0" />
            <xsd:element name="NumberEmployees" type="xsd:int" minOccurs="0" />
            <xsd:element name="Comments" type="xsd:string" minOccurs="0" />
          </xsd:sequence>
        </xsd:restriction>
      </xsd:complexContent>
    </xsd:complexType>
  </xsd:element>
  <xsd:simpleType name="BrandType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="AW" />
      <xsd:enumeration value="2" />
      <xsd:enumeration value="3" />
      <xsd:enumeration value="4+" />
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="BusinessType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="BM" />
      <xsd:enumeration value="BS" />
      <xsd:enumeration value="D" />
      <xsd:enumeration value="OS" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>
```

```
<xsd:enumeration value="SGS" />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="InternetType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="56kb" />
    <xsd:enumeration value="ISDN" />
    <xsd:enumeration value="DSL" />
    <xsd:enumeration value="T1" />
    <xsd:enumeration value="T2" />
    <xsd:enumeration value="T3" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="SpecialtyType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Family" />
    <xsd:enumeration value="Kids" />
    <xsd:enumeration value="BMX" />
    <xsd:enumeration value="Touring" />
    <xsd:enumeration value="Road" />
    <xsd:enumeration value="Mountain" />
    <xsd:enumeration value="All" />
  </xsd:restriction>
</xsd:simpleType>
</xsd:schema>'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Collection of XML schemas for the
Demographics column in the Sales.Store table.', 'SCHEMA', N'Sales', 'XML SCHEMA
COLLECTION', N'StoreSurveySchemaCollection', NULL, NULL
GO
```

Uses

Sales

Used By

[Sales].[Store]

 **Full Text Catalogs**

Objects

Name

dbo.AW2008FullTextCatalog

AW2008FullTextCatalog

Properties

Property	Value
Owner	dbo
Default	True
Accent Sensitive	True

Tables

- JobCandidate
- Document
- ProductReview

SQL Script

```
CREATE FULLTEXT CATALOG [AW2008FullTextCatalog]
WITH ACCENT_SENSITIVITY = ON
AS DEFAULT
AUTHORIZATION [dbo]
GO
CREATE FULLTEXT INDEX ON [HumanResources].[JobCandidate] KEY INDEX [PK_JobCandidate_
JobCandidateID] ON [AW2008FullTextCatalog]
GO
CREATE FULLTEXT INDEX ON [Production].[Document] KEY INDEX [PK_Document_Document-
Node] ON [AW2008FullTextCatalog]
GO
CREATE FULLTEXT INDEX ON [Production].[ProductReview] KEY INDEX [PK_ProductReview_
ProductReviewID] ON [AW2008FullTextCatalog]
GO
```

Uses

[HumanResources].[JobCandidate]
[Production].[Document]
[Production].[ProductReview]

 **Users**

Objects

Name

piotrkononow

 **piotrkononow**

Properties

Property	Value
Type	SqlUser
Login Name	piotrkononow
Default Schema	dbo

Database Level Permissions

Type	Action
CONNECT	Grant

SQL Script

```
IF NOT EXISTS (SELECT * FROM master.dbo.syslogins WHERE loginname = N'piotrkononow')
CREATE LOGIN [piotrkononow] WITH PASSWORD = 'p@ssw0rd'
GO
CREATE USER [piotrkononow] FOR LOGIN [piotrkononow]
GO
```

Database Roles

Objects

Name
db_accessadmin
db_backupoperator
db_datareader
db_datawriter
db_ddladmin
db_denydatareader
db_denydatawriter
db_owner
db_securityadmin
public

db_accessadmin

Properties

Property	Value
Owner	dbo

db_backupoperator

Properties

Property	Value
Owner	dbo

db_datareader

Properties

Property	Value
Owner	dbo

Members

- piotrkononow

SQL Script

```
EXEC sp_addrolemember N'db_datareader', N'piotrkononow'  
GO
```

Uses

piotrkononow

db_datawriter

Properties

Property	Value
Owner	dbo

db_ddladmin

Properties

Property	Value
Owner	dbo

Members

- piotrkononow

SQL Script

```
EXEC sp_addrolemember N'db_ddladmin', N'piotrkononow'  
GO
```

Uses

piotrkononow

 **db_denydatareader**

Properties

Property	Value
Owner	dbo

 **db_denydatawriter**

Properties

Property	Value
Owner	dbo

 **db_owner**

Properties

Property	Value
Owner	dbo

 **db_securityadmin**

Properties

Property	Value
Owner	dbo

 **public**

Properties

Property	Value
Owner	dbo

Schemas

Objects

Name
HumanResources <i>Contains objects related to employees and departments.</i>
Person <i>Contains objects related to names and addresses of customers, vendors, and employees</i>
Production <i>Contains objects related to products, inventory, and manufacturing.</i>
Purchasing <i>Contains objects related to vendors and purchase orders.</i>
Sales <i>Contains objects related to customers, sales orders, and sales territories.</i>

HumanResources

MS_Description

Contains objects related to employees and departments.

Properties

Property	Value
Owner	dbo

SQL Script

```
CREATE SCHEMA [HumanResources]
AUTHORIZATION [dbo]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Contains objects related to
employees and departments.', 'SCHEMA', N'HumanResources', NULL, NULL, NULL, NULL
GO
```

Used By

[HumanResources].[Department]
 [HumanResources].[Employee]
 [HumanResources].[EmployeeDepartmentHistory]
 [HumanResources].[EmployeePayHistory]
 [HumanResources].[JobCandidate]
 [HumanResources].[Shift]
 [HumanResources].[vEmployee]
 [HumanResources].[vEmployeeDepartment]
 [HumanResources].[vEmployeeDepartmentHistory]
 [HumanResources].[vJobCandidate]
 [HumanResources].[vJobCandidateEducation]
 [HumanResources].[vJobCandidateEmployment]
 [HumanResources].[uspUpdateEmployeeHireInfo]
 [HumanResources].[uspUpdateEmployeeLogin]
 [HumanResources].[uspUpdateEmployeePersonalInfo]
 [HumanResources].[HRResumeSchemaCollection]

Person

MS_Description

Contains objects related to names and addresses of customers, vendors, and employees

Properties

Property	Value
Owner	dbo

SQL Script

```
CREATE SCHEMA [Person]
AUTHORIZATION [dbo]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Contains objects related to names
and addresses of customers, vendors, and employees', 'SCHEMA', N'Person', NULL,
NULL, NULL, NULL
GO
```

Used By

[Person].[Address]
[Person].[AddressType]
[Person].[BusinessEntity]
[Person].[BusinessEntityAddress]
[Person].[BusinessEntityContact]
[Person].[ContactType]
[Person].[CountryRegion]
[Person].[EmailAddress]
[Person].[Password]
[Person].[Person]
[Person].[PersonPhone]
[Person].[PhoneNumberType]
[Person].[StateProvince]
[Person].[vAdditionalContactInfo]
[Person].[vStateProvinceCountryRegion]
[Person].[AdditionalContactInfoSchemaCollection]
[Person].[IndividualSurveySchemaCollection]

Production

MS_Description

Contains objects related to products, inventory, and manufacturing.

Properties

Property	Value
Owner	dbo

SQL Script

```
CREATE SCHEMA [Production]
AUTHORIZATION [dbo]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Contains objects related to
products, inventory, and manufacturing.', 'SCHEMA', N'Production', NULL, NULL, NULL,
NULL
GO
```

Used By

[Production].[BillOfMaterials]
 [Production].[Culture]
 [Production].[Document]
 [Production].[Illustration]
 [Production].[Location]
 [Production].[Product]
 [Production].[ProductCategory]
 [Production].[ProductCostHistory]
 [Production].[ProductDescription]
 [Production].[ProductDocument]
 [Production].[ProductInventory]
 [Production].[ProductListPriceHistory]
 [Production].[ProductModel]
 [Production].[ProductModelIllustration]
 [Production].[ProductModelProductDescriptionCulture]
 [Production].[ProductPhoto]
 [Production].[ProductProductPhoto]
 [Production].[ProductReview]
 [Production].[ProductSubcategory]
 [Production].[ScrapReason]
 [Production].[TransactionHistory]
 [Production].[TransactionHistoryArchive]
 [Production].[UnitMeasure]
 [Production].[WorkOrder]
 [Production].[WorkOrderRouting]
 [Production].[vProductAndDescription]
 [Production].[vProductModelCatalogDescription]

Project > Isrep17 > User databases > AdventureWorks > Security > Schemas > Production

[Production].[vProductModelInstructions]

[Production].[ManuInstructionsSchemaCollection]

[Production].[ProductDescriptionSchemaCollection]

Purchasing

MS_Description

Contains objects related to vendors and purchase orders.

Properties

Property	Value
Owner	dbo

SQL Script

```
CREATE SCHEMA [Purchasing]
AUTHORIZATION [dbo]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Contains objects related to vendors
and purchase orders.', 'SCHEMA', N'Purchasing', NULL, NULL, NULL, NULL
GO
```

Used By

[Purchasing].[ProductVendor]
[Purchasing].[PurchaseOrderDetail]
[Purchasing].[PurchaseOrderHeader]
[Purchasing].[ShipMethod]
[Purchasing].[Vendor]
[Purchasing].[vVendorWithAddresses]
[Purchasing].[vVendorWithContacts]

Sales

MS_Description

Contains objects related to customers, sales orders, and sales territories.

Properties

Property	Value
Owner	dbo

SQL Script

```
CREATE SCHEMA [Sales]
AUTHORIZATION [dbo]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Contains objects related to
customers, sales orders, and sales territories.', 'SCHEMA', N'Sales', NULL, NULL,
NULL, NULL
GO
```

Used By

[Sales].[CountryRegionCurrency]
 [Sales].[CreditCard]
 [Sales].[Currency]
 [Sales].[CurrencyRate]
 [Sales].[Customer]
 [Sales].[PersonCreditCard]
 [Sales].[SalesOrderDetail]
 [Sales].[SalesOrderHeader]
 [Sales].[SalesOrderHeaderSalesReason]
 [Sales].[SalesPerson]
 [Sales].[SalesPersonQuotaHistory]
 [Sales].[SalesReason]
 [Sales].[SalesTaxRate]
 [Sales].[SalesTerritory]
 [Sales].[SalesTerritoryHistory]
 [Sales].[ShoppingCartItem]
 [Sales].[SpecialOffer]
 [Sales].[SpecialOfferProduct]
 [Sales].[Store]
 [Sales].[vIndividualCustomer]
 [Sales].[vPersonDemographics]
 [Sales].[vSalesPerson]
 [Sales].[vSalesPersonSalesByFiscalYears]
 [Sales].[vStoreWithAddresses]
 [Sales].[vStoreWithContacts]
 [Sales].[vStoreWithDemographics]
 [Sales].[StoreSurveySchemaCollection]

