localhost Documentation

AdventureWorks2016

Server localhost

Author dnagykrisztina

Created 2021. március 21., vasárnap 18:47:33

File Path C:\Users\dnagykrisztina\Desktop\Szakdolgozat\Redgate-MSSQL\Dokumentáció\documenter_pdf-2021-

03-21T18-47-33.pdf

Table of Contents

able of Contents	2
localhost	7
User databases	
AdventureWorks2016 Database	10
Ⅲ Tables	13
[dbo].[AWBuildVersion]	17
[dbo].[DatabaseLog]	19
[dbo].[ErrorLog]	21
[HumanResources].[Department]	24
[HumanResources].[Employee]	26
[HumanResources].[EmployeeDepartmentHistory]	34
[HumanResources].[EmployeePayHistory]	38
[HumanResources].[JobCandidate]	41
[HumanResources].[Shift]	44
[Person].[Address]	47
[Person].[AddressType]	51
[Person].[BusinessEntity]	54
[Person].[BusinessEntityAddress]	56
[Person].[BusinessEntityContact]	60
[Person].[ContactType]	64
[Person].[CountryRegion]	66
[Person].[EmailAddress]	68
[Person].[Password]	71
[Person].[Person]	74
[Person].[PersonPhone]	82
[Person].[Pet]	85
[Person].[PhoneNumberType]	87
[Person].[StateProvince]	89
[Production].[BillOfMaterials]	93
[Production].[Culture]	98
[Production].[Document]	100
[Production].[Illustration]	105
[Production].[Location]	107
[Production].[Product]	110
[Production].[ProductCategory]	119
[Production].[ProductCostHistory]	122
[Production].[ProductDescription]	125

■	[Production].[ProductDocument]	128
■■	[Production].[ProductInventory]	131
■■	[Production].[ProductListPriceHistory]	135
■■	[Production].[ProductModel]	138
	[Production].[ProductModelIllustration]	141
≡≡	[Production].[ProductModelProductDescriptionCulture]	144
	[Production].[ProductPhoto]	147
≣≣	[Production].[ProductProductPhoto]	149
≣≣	[Production].[ProductReview]	152
■■	[Production].[ProductSubcategory]	156
■■	[Production].[ScrapReason]	159
■■	[Production].[TransactionHistory]	161
■■	[Production].[TransactionHistoryArchive]	165
	[Production].[UnitMeasure]	169
■■	[Production].[WorkOrder]	171
■■	[Production].[WorkOrderRouting]	177
■■	[Purchasing].[ProductVendor]	182
■■	[Purchasing].[PurchaseOrderDetail]	187
■■	[Purchasing].[PurchaseOrderHeader]	195
	[Purchasing].[ShipMethod]	202
	[Purchasing].[Vendor]	205
	[Sales].[CountryRegionCurrency]	210
	[Sales].[CreditCard]	213
≡≡	[Sales].[Currency]	216
≡≡	[Sales].[CurrencyRate]	218
	[Sales].[Customer]	221
≡≡	[Sales].[PersonCreditCard]	225
≡≡	[Sales].[SalesOrderDetail]	228
■■	[Sales].[SalesOrderHeader]	235
≡≡	[Sales].[SalesOrderHeaderSalesReason]	246
≡≡	[Sales].[SalesPerson]	249
	[Sales].[SalesPersonQuotaHistory]	254
■■	[Sales].[SalesReason]	257
	[Sales].[SalesTaxRate]	
	[Sales].[SalesTerritory]	263
	[Sales].[SalesTerritoryHistory]	268

		[Sales].[SpecialOfferProduct]	279
	≡≡	[Sales].[Store]	282
Œ	Vie	ws	286
		[HumanResources].[vEmployee]	288
	Œ	[HumanResources].[vEmployeeDepartment]	291
		[HumanResources].[vEmployeeDepartmentHistory]	293
		[HumanResources].[vJobCandidate]	295
		[HumanResources].[vJobCandidateEducation]	298
	Œ	[HumanResources].[vJobCandidateEmployment]	300
		[Person].[vAdditionalContactInfo]	302
	Œ	[Person].[vStateProvinceCountryRegion]	305
	Œ	[Production].[vProductAndDescription]	307
		[Production].[vProductModelCatalogDescription]	309
	Œ	[Production].[vProductModelInstructions]	313
		[Purchasing].[vVendorWithAddresses]	315
		[Purchasing].[vVendorWithContacts]	317
		[Sales].[vIndividualCustomer]	319
		[Sales].[vPersonDemographics]	322
	Œ	[Sales].[vSalesPerson]	324
		[Sales].[vSalesPersonSalesByFiscalYears]	327
	Œ	[Sales].[vStoreWithAddresses]	329
	Œ	[Sales].[vStoreWithContacts]	331
	Œ	[Sales].[vStoreWithDemographics]	333
	Sto	ored Procedures	335
		[dbo].[uspGetBillOfMaterials]	
		[dbo].[uspGetEmployeeManagers]	338
		[dbo].[uspGetManagerEmployees]	340
		[dbo].[uspGetWhereUsedProductID]	342
		[dbo].[uspLogError]	344
		[dbo].[uspPrintError]	347
		[dbo].[uspSearchCandidateResumes]	348
		[HumanResources].[uspUpdateEmployeeHireInfo]	350
		[HumanResources].[uspUpdateEmployeeLogin]	353
		[HumanResources].[uspUpdateEmployeePersonalInfo]	
f,		ole-valued Functions	
		[dbo].[ufnGetContactInformation]	
101 01 f >	Sca	alar-valued Functions	361
	101 01 fx	[dbo].[ufnGetAccountingEndDate]	362

	[dbo].[ufnGetAccountingStartDate]	363
	[dbo].[ufnGetDocumentStatusText]	364
	[dbo].[ufnGetProductDealerPrice]	
	[dbo].[ufnGetProductListPrice]	
	[dbo].[ufnGetProductStandardCost]	370
	[dbo].[ufnGetPurchaseOrderStatusText]	372
	[dbo].[ufnGetSalesOrderStatusText]	374
	[dbo].[ufnGetStock]	376
	[dbo].[ufnLeadingZeros]	378
4	Database Triggers	380
	ddlDatabaseTriggerLog	381
iii E	User-Defined Data Types	383
	[dbo].[AccountNumber]	384
	[dbo].[Flag]	
	[dbo].[Name]	
	[dbo].[NameStyle]	388
	[dbo].[OrderNumber]	
	[dbo].[Phone]	390
Å	XML Schema Collections	391
	HumanResources].[HRResumeSchemaCollection]	392
	[Person].[AdditionalContactInfoSchemaCollection]	396
	[Person].[IndividualSurveySchemaCollection]	399
	[Production].[ManuInstructionsSchemaCollection]	401
	[Production].[ProductDescriptionSchemaCollection]	403
_	[Sales].[StoreSurveySchemaCollection]	
	Full Text Catalogs	408
_	AW2016FullTextCatalog	
1	Users	
	≜ dbo	
	1 guest	
٠.	Database Roles	413
	db_accessadmin	
	db_backupoperator	
	db_datareader	
	db_datawriter	
	db_ddladmin	
	db_denydatareader	
	db_denydatawriter	415

db_owner	415
db_securityadmin	415
public	416
^ Schemas	417
	418
A Person	419
⚠ Production	420
Purchasing	422
⚠ Sales	423

■ localhost

Databases (1)

• AdventureWorks2016

Server Properties

Property	Value
Product	Microsoft SQL Server
Version	14.0.3381.3
Language	English
Platform	NT x64
Edition	Developer Edition (64-bit)
Engine Edition	3 (Enterprise)
Processors	6
OS Version	6.2 (9200)
Physical Memory	10187
Is Clustered	False
Root Directory	C:\
Collation	SQL_Latin1_General_CP1_CI_AS

Server Settings

Property	Value
Default data file path	/var/opt/mssql/data/
Default backup file path	/var/opt/mssql/data
Default log file path	/var/opt/mssql/data/
Recovery Interval (minutes)	0
Default index fill factor	0
Default backup media retention	0
Compress Backup	False

Advanced Server Settings

Property	Value
Full text upgrade option	2
Locks	0
Nested triggers enabled	True
Allow triggers to fire others	True

English
4096
1033
2049
10
-1
65536
5
0
16
2147483647
False
False
False
0
False
False
True

C	_	. .	_
LJ U	lser	data	bases

Databases (1)

• AdventureWorks2016

■ AdventureWorks2016 Database

MS_Description

AdventureWorks 2016 Sample OLTP Database

Database Properties

Property	Value
SQL Server Version	SQL Server 2017
Compatibility Level	SQL Server 2016
Last backup time	04/09/2018
Last log backup time	-
Creation date	Mar 21 2021
Users	4
Database Encryption Enabled	False
Database Encryption Algorithm	None
Database size	209.63 MB
Unallocated space	1.19 MB

Database Options

Property	Value
Compatibility Level	130
Database collation	SQL_Latin1_General_CP1_CI_AS
Restrict access	MULTI_USER
Is read-only	False
Auto close	False
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

ANIOLAULI J. J. f It	E.G.
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	b19fa6ba-05ee-4ea3-9ba5-b08f9ef06c91
Service broker enabled	False
Log reuse wait	NOTHING
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	60
Database owner	sa

Files

Name	Туре	Size	Maxsize	Autogrowth	File Name
AdventureWorks2016_Data	Data	207,63 MB	unlimited	16,00 MB	/var/opt/mssql/data/Advent

					ureWorks2016_Data.mdf
AdventureWorks2016_Log	Log	2,00 MB	2048,00 GB	16,00 MB	/var/opt/mssql/data/Advent ureWorks2016_Log.ldf

■ Tables

Objects

Name

dbo.AWBuildVersion

Current version number of the AdventureWorks 2016 sample database.

dbo.DatabaseLog

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

dbo.ErrorLog

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.

HumanResources.Department

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

HumanResources.Employee

Employee information such as salary, department, and title.

HumanResources.EmployeeDepartmentHistory

Employee department transfers

HumanResources.EmployeePayHistory

Employee pay history.

HumanResources.JobCandidate

Résumés submitted to Human Resources by job applicants.

HumanResources.Shift

Work shift lookup table.

Person.Address

Street address information for customers, employees, and vendors.

Person.AddressType

Types of addresses stored in the Address table.

Person.BusinessEntity

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

Person.BusinessEntityAddress

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

Person.BusinessEntityContact

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

Person.ContactType

Lookup table containing the types of business entity contacts.

Person.CountryRegion

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

Person.EmailAddress

Where to send a person email.

Person.Password

One way hashed authentication information

Person.Person

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

Person.PersonPhone

Telephone number and type of a person.

Person.Pet

Person.PhoneNumberType

Type of phone number of a person.

Person.StateProvince

State and province lookup table.

Production.BillOfMaterials

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

Production.Culture

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

Production.Document

Product maintenance documents.

Production.Illustration

Bicycle assembly diagrams.

Production.Location

Product inventory and manufacturing locations.

Production.Product

Products sold or used in the manfacturing of sold products.

Production.ProductCategory

High-level product categorization.

Production.ProductCostHistory

Changes in the cost of a product over time.

Production.ProductDescription

Product descriptions in several languages.

Production.ProductDocument

Cross-reference table mapping products to related product documents.

Production.ProductInventory

Product inventory information.

Production.ProductListPriceHistory

Changes in the list price of a product over time.

Production.ProductModel

Product model classification.

Production.ProductModelIllustration

Cross-reference table mapping product models and illustrations.

Production. Product Model Product Description Culture

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

Production.ProductPhoto

Product images.

Production.ProductProductPhoto

Cross-reference table mapping products and product photos.

Production.ProductReview

Customer reviews of products they have purchased.

Production.ProductSubcategory

Product subcategories. See ProductCategory table.

Production.ScrapReason

Manufacturing failure reasons lookup table.

Production.TransactionHistory

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

Production.TransactionHistoryArchive

Transactions for previous years.

Production.UnitMeasure

Unit of measure lookup table.

Production.WorkOrder

Manufacturing work orders.

Production.WorkOrderRouting

Work order details.

Purchasing.ProductVendor

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

Purchasing.PurchaseOrderDetail

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

Purchasing.PurchaseOrderHeader

General purchase order information. See PurchaseOrderDetail.

Purchasing.ShipMethod

Shipping company lookup table.

Purchasing.Vendor

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

Sales.CountryRegionCurrency

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

Sales.CreditCard

Customer credit card information.

Sales.Currency

Lookup table containing standard ISO currencies.

Sales.CurrencyRate

Currency exchange rates.

Sales.Customer

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

Sales.PersonCreditCard

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

Sales.SalesOrderDetail

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

Sales.SalesOrderHeader

General sales order information.

Sales.SalesOrderHeaderSalesReason

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

Sales.SalesPerson

Sales representative current information.

Sales.SalesPersonQuotaHistory

Sales performance tracking.

Sales.SalesReason

Lookup table of customer purchase reasons.

Sales.SalesTaxRate

Tax rate lookup table.

Sales.SalesTerritory

Sales territory lookup table.

Sales.SalesTerritoryHistory

Sales representative transfers to other sales territories.

Sales.ShoppingCartItem

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

Sales.SpecialOffer

Sale discounts lookup table.

Sales.SpecialOfferProduct

Cross-reference table mapping products to special offer discounts.

Sales.Store

Customers (resellers) of Adventure Works products.



Author: dnagykrisztina

[dbo].[AWBuildVersion]

MS_Description

Current version number of the AdventureWorks 2016 sample database.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	1
Created	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:45 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
₽ <mark>≯</mark> C	SystemInformationID Primary key for AWBuildVersion records.	tinyint	1	NOT NULL	1 - 1	
	Database Version Version number of the database in 9.yy.mm.dd.00 format.	nvarchar(25)	50	NOT NULL		
	VersionDate Date and time the record was last updated.	datetime	8	NOT NULL		
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

Key	Name	Key Columns	Unique
PK	PK_AWBuildVersion_SystemInformationID Primary key (clustered) constraint	SystemInformationID	True

```
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 SystemInformation-
ID] PRIMARY KEY CLUSTERED ([SystemInformationID]) ON [PRIMARY]
EXEC sp addextendedproperty N'MS Description', N'Current version number of the
AdventureWorks 2016 sample database. ', 'SCHEMA', N'dbo', 'TABLE', N'AWBuildVersion',
NULL, NULL
GO
EXEC sp addextendedproperty N'MS Description', N'Version 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'
EXEC sp addextendedproperty N'MS Description', N'Primary key for AWBuildVersion
records.', 'SCHEMA', N'dbo', 'TABLE', N'AWBuildVersion', 'COLUMN', N'SystemInformation-
ID'
GO
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'dbo', 'TABLE', N'AWBuildVersion', 'COLUMN', N'VersionDate'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'dbo', 'TABLE', N'AWBuildVersion', 'CONSTRAINT', N'DF AWBuild-
Version ModifiedDate
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint', 'SCHEMA', N'dbo', 'TABLE', N'AWBuildVersion', 'CONSTRAINT', N'PK_AWBuildVersion_System-
InformationID'
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a primary
key constraint.', 'SCHEMA', N'dbo', 'TABLE', N'AWBuildVersion', 'INDEX', N'PK AWBuild-
Version SystemInformationID'
```

[dbo].[DatabaseLog]

MS_Description

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

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Неар	True
Row Count (~)	1597
Created	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:45 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK	DatabaseLogID Primary key for DatabaseLog records.	int	4	NOT NULL	1 - 1
	PostTime The date and time the DDL change occurred.	datetime	8	NOT NULL	
	DatabaseUser The user who implemented the DDL change.	[sys].[sysname]	256	NOT NULL	
	Event The type of DDL statement that was executed.	[sys].[sysname]	256	NOT NULL	
	Schema The schema to which the changed object belongs.	[sys].[sysname]	256	NULL allowed	
	Object The object that was changed by the DDL statment.	[sys].[sysname]	256	NULL allowed	
	TSQL The exact Transact-SQL statement that was executed.	nvarchar(max)	max	NOT NULL	
	XmlEvent The raw XML data generated by database trigger.	xml	max	NOT NULL	

Indexes

Key	Name	Key Columns	Unique
PK	PK_DatabaseLog_DatabaseLogID Primary key (nonclustered) constraint	DatabaseLogID	True

```
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]
GO
ALTER TABLE [dbo].[DatabaseLog] ADD CONSTRAINT [PK DatabaseLog DatabaseLogID] PRIMARY
KEY NONCLUSTERED ([DatabaseLogID]) ON [PRIMARY]
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 ddl-
DatabaseTriggerLog.', 'SCHEMA', N'dbo', 'TABLE', N'DatabaseLog', NULL, NULL
EXEC sp addextendedproperty N'MS Description', N'Primary key for DatabaseLog records.',
'SCHEMA', N'dbo', 'TABLE', N'DatabaseLog', 'COLUMN', N'DatabaseLogID'
EXEC sp addextendedproperty N'MS Description', N'The user who implemented the DDL
change.', 'SCHEMA', N'dbo', 'TABLE', N'DatabaseLog', 'COLUMN', N'DatabaseUser'
EXEC sp addextendedproperty N'MS Description', N'The type of DDL statement that was
executed.', 'SCHEMA', N'dbo', 'TABLE', N'DatabaseLog', 'COLUMN', N'Event'
EXEC sp addextendedproperty N'MS Description', N'The object that was changed by the DDL
statment.', 'SCHEMA', N'dbo', 'TABLE', N'DatabaseLog', 'COLUMN', N'Object'
EXEC sp addextendedproperty N'MS Description', N'The date and time the DDL change
occurred.', 'SCHEMA', N'dbo', 'TABLE', N'DatabaseLog', 'COLUMN', N'PostTime'
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'
EXEC sp addextendedproperty N'MS Description', N'The raw XML data generated by database
trigger.', 'SCHEMA', N'dbo', 'TABLE', N'DatabaseLog', 'COLUMN', N'XmlEvent'
EXEC sp addextendedproperty N'MS Description', N'Primary key (nonclustered)
constraint', 'SCHEMA', N'dbo', 'TABLE', N'DatabaseLog', 'CONSTRAINT', N'PK DatabaseLog -
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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:40 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullabilit y	Identity	Default
PK <mark>P</mark> C	ErrorLogID Primary key for ErrorLog records.	int	4	NOT NULL	1 - 1	
	ErrorTime The date and time at which the error occurred.	datetime	8	NOT NULL		(getdate())
	UserName The user who executed the batch in which the error occurred.	[sys].[sysname]	256	NOT NULL		
	ErrorNumber The error number of the error that occurred.	int	4	NOT NULL		
	ErrorSeverity The severity of the error that occurred.	int	4	NULL allowed		
	ErrorState The state number of the error that occurred.	int	4	NULL allowed		
	ErrorProcedure The name of the stored procedure or trigger where the error occurred.	nvarchar(126)	252	NULL allowed		
	ErrorLine The line number at which the error occurred.	int	4	NULL allowed		
	ErrorMessage The message text of the error that occurred.	nvarchar(4000)	8000	NOT NULL		

Indexes

Key	Name	Key Columns	Unique
PK	PK_ErrorLog_ErrorLogID Primary key (clustered) constraint	ErrorLogID	True

```
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]
ALTER TABLE [dbo]. [ErrorLog] ADD CONSTRAINT [PK ErrorLog ErrorLogID] PRIMARY KEY
CLUSTERED ([ErrorLogID]) ON [PRIMARY]
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'
EXEC sp addextendedproperty N'MS Description', N'Primary key for ErrorLog records.',
'SCHEMA', N'dbo', 'TABLE', N'ErrorLog', 'COLUMN', N'ErrorLogID'
EXEC sp addextendedproperty N'MS Description', N'The message text of the error that
occurred.', 'SCHEMA', N'dbo', 'TABLE', N'ErrorLog', 'COLUMN', N'ErrorMessage'
EXEC sp addextendedproperty N'MS Description', N'The error number of the error that
occurred.', 'SCHEMA', N'dbo', 'TABLE', N'ErrorLog', 'COLUMN', N'ErrorNumber'
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'
EXEC sp addextendedproperty N'MS Description', N'The severity of the error that
occurred.', 'SCHEMA', N'dbo', 'TABLE', N'ErrorLog', 'COLUMN', N'ErrorSeverity'
EXEC sp addextendedproperty N'MS Description', N'The state number of the error that
occurred.', 'SCHEMA', N'dbo', 'TABLE', N'ErrorLog', 'COLUMN', N'ErrorState'
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'
```

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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
PK <mark>P</mark> C	DepartmentID Primary key for Department records.	smallint	2	NOT NULL	1 - 1	
<u>.</u>	Name Name of the department.	[dbo].[Name]	100	NOT NULL		
	GroupName Name of the group to which the department belongs.	[dbo].[Name]	100	NOT NULL		
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

Key	Name	Key Columns	Unique
PK	PK_Department_DepartmentID Primary key (clustered) constraint	DepartmentID	True
	AK_Department_Name Unique nonclustered index.	Name	True

```
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 DepartmentID]
PRIMARY KEY CLUSTERED ([DepartmentID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK Department Name] ON [HumanResources].[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'HumanResources',
'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'
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'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'HumanResources', 'TABLE', N'Department', 'COLUMN', N'Modified-
EXEC sp addextendedproperty N'MS Description', N'Name of the department.', 'SCHEMA',
N'HumanResources', 'TABLE', N'Department', 'COLUMN', N'Name'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'HumanResources', 'TABLE', N'Department', 'CONSTRAINT', N'DF -
Department ModifiedDate'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint',
'SCHEMA', N'HumanResources', 'TABLE', N'Department', 'CONSTRAINT', N'PK Department -
DepartmentID'
EXEC sp addextendedproperty N'MS Description', N'Unique nonclustered index.', 'SCHEMA',
N'HumanResources', 'TABLE', N'Department', 'INDEX', N'AK Department Name'
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]

Ⅲ [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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:52 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Compute d	Max Length (Bytes)	Nullabilit y	Default
PK OFKO	BusinessEntityID Primary key for Employee records. Foreign key to Business- Entity.BusinessEntityID.	int		4	NOT NULL	
. 	NationalIDNumber Unique national identification number such as a social security number.	nvarchar(15)		30	NOT NULL	
.t.	LoginID Network login.	nvarchar(256)		512	NOT NULL	
 (2)	OrganizationNode Where the employee is located in corporate hierarchy.	hierarchyid		892	NULL allowed	
. 	OrganizationLevel The depth of the employee in the corporate hierarchy.	smallint	True	2	NULL allowed	
	JobTitle Work title such as Buyer or Sales Representative.	nvarchar(50)		100	NOT NULL	
	BirthDate Date of birth.	date		3	NOT NULL	
	MaritalStatus M = Married, S = Single	nchar(1)		2	NOT NULL	
	Gender M = Male, F = Female	nchar(1)		2	NOT NULL	
	HireDate Employee hired on this date.	date		3	NOT NULL	
	SalariedFlag Job classification. 0 = Hourly, not exempt from collective	[dbo].[Flag]		1	NOT NULL	((1))

	bargaining. 1 = Salaried, exempt from collective bargaining.				
=	VacationHours Number of available vacation hours.	smallint	2	NOT NULL	((0))
=	SickLeaveHours Number of available sick leave hours.	smallint	2	NOT NULL	((0))
	CurrentFlag 0 = Inactive, 1 = Active	[dbo].[Flag]	1	NOT NULL	((1))
ů.	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL	(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate()

Computed columns

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

Indexes

Key	Name	Key Columns	Unique
PK <mark>P</mark> C	PK_Employee_BusinessEntityID Primary key (clustered) constraint	BusinessEntity- ID	True
	AK_Employee_LoginID Unique nonclustered index.	LoginID	True
	AK_Employee_NationalIDNumber Unique nonclustered index.	National- IDNumber	True
	AK_Employee_rowguid Unique nonclustered index. Used to support replication samples.	rowguid	True
	IX_Employee_OrganizationLevel_OrganizationNode Unique nonclustered index.	Organization- Level, Organization- Node	
	IX_Employee_OrganizationNode Unique nonclustered index.	Organization- Node	

Triggers

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

Check Constraints

Name	On Column	Constraint
CK_Employee_BirthDate Check constraint [BirthDate] >= '1930-01-01' AND [BirthDate] <= dateadd(year,(-18),GETDATE())	BirthDate	([Birth- Date]>='1930- 01-01' AND [Birth- Date]<=dateadd(year,(- 18),getdate()))
CK_Employee_Gender Check constraint [Gender]='f' OR [Gender]='m' OR [Gender]='F' OR [Gender]='M'	Gender	(upper([Gender]) ='F' OR upper([Gender]) ='M')
CK_Employee_HireDate Check constraint [HireDate] >= '1996-07-01' AND [HireDate] <= dateadd(day,(1),GETDATE())	HireDate	([Hire- Date]>='1996- 07-01' AND [Hire- Date]<=dateadd(day,(1),getdate()))
CK_Employee_MaritalStatus Check constraint [MaritalStatus]='s' OR [MaritalStatus]='m' OR [MaritalStatus]='S' OR [MaritalStatus]='M'	MaritalStatus	(upper([Marital- Status])='S' OR upper([Marital- Status])='M')
CK_Employee_SickLeaveHours Check constraint [SickLeaveHours] >= (0) AND [SickLeaveHours] <= (120)	SickLeaveHours	([SickLeave- Hours]>=(0) AND [SickLeave- Hours]<=(120))
CK_Employee_VacationHours Check constraint [VacationHours] >= (-40) AND [VacationHours] <= (240)	VacationHours	([Vacation- Hours]>=((-40)) AND [Vacation- Hours]<=(240))

Foreign Keys

Name	Columns
FK_Employee_Person_BusinessEntityID Foreign key constraint referencing Person.BusinessEntity-ID.	BusinessEntityID->[Person].[Person].[BusinessEntityID]

```
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
[SickLeaveHours] [smallint] NOT NULL CONSTRAINT [DF Employee SickLeaveHours] DEFAULT
[CurrentFlaq] [dbo].[Flaq] NOT NULL CONSTRAINT [DF Employee CurrentFlaq] 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]
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;
ALTER TABLE [HumanResources].[Employee] ADD CONSTRAINT [CK Employee BirthDate] CHECK
(([BirthDate]>='1930-01-01' AND [BirthDate]<=dateadd(year, (-18),getdate())))
ALTER TABLE [HumanResources].[Employee] ADD CONSTRAINT [CK Employee Gender] CHECK
((upper([Gender])='F' OR upper([Gender])='M'))
ALTER TABLE [HumanResources].[Employee] ADD CONSTRAINT [CK Employee HireDate] CHECK
(([HireDate]>='1996-07-01' AND [HireDate]<=dateadd(day,(1),getdate())))
ALTER TABLE [HumanResources].[Employee] ADD CONSTRAINT [CK Employee MaritalStatus]
CHECK ((upper([MaritalStatus])='S' OR upper([MaritalStatus])='M'))
ALTER TABLE [HumanResources].[Employee] ADD CONSTRAINT [CK Employee SickLeaveHours]
CHECK (([SickLeaveHours]>=(0) AND [SickLeaveHours]<=(120)))</pre>
ALTER TABLE [HumanResources].[Employee] ADD CONSTRAINT [CK Employee VacationHours]
```

```
CHECK (([VacationHours]>=((-40)) AND [VacationHours]<=(240)))
ALTER TABLE [HumanResources].[Employee] ADD CONSTRAINT [PK Employee BusinessEntityID]
PRIMARY KEY CLUSTERED ([BusinessEntityID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK Employee LoginID] ON [HumanResources].[Employee]
([LoginID]) ON [PRIMARY]
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK Employee NationalIDNumber] ON [Human-
Resources].[Employee] ([NationalIDNumber]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX Employee OrganizationLevel OrganizationNode] ON [Human-
Resources].[Employee] ([OrganizationLevel], [OrganizationNode]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX Employee OrganizationNode] ON [HumanResources]. [Employee]
([OrganizationNode]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK Employee rowquid] ON [HumanResources].[Employee]
([rowguid]) ON [PRIMARY]
ALTER TABLE [HumanResources].[Employee] ADD CONSTRAINT [FK Employee Person Business-
EntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [Person].[Person] ([Business-
EntityID])
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'
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'
EXEC sp addextendedproperty N'MS Description', N'0 = Inactive, 1 = Active', 'SCHEMA',
N'HumanResources', 'TABLE', N'Employee', 'COLUMN', N'CurrentFlag'
EXEC sp addextendedproperty N'MS Description', N'M = Male, F = Female', 'SCHEMA',
N'HumanResources', 'TABLE', N'Employee', 'COLUMN', N'Gender'
EXEC sp addextendedproperty N'MS Description', N'Employee hired on this date.',
'SCHEMA, N'HumanResources', 'TABLE', N'Employee', 'COLUMN', N'HireDate'
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'
EXEC sp addextendedproperty N'MS Description', N'Network login.', 'SCHEMA', N'Human-
Resources', 'TABLE', N'Employee', 'COLUMN', N'LoginID'
EXEC sp addextendedproperty N'MS Description', N'M = Married, S = Single', 'SCHEMA',
N'HumanResources', 'TABLE', N'Employee', 'COLUMN', N'MaritalStatus'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'COLUMN', N'ModifiedDate'
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'
EXEC sp addextendedproperty N'MS Description', N'Where the employee is located in
corporate hierarchy.', 'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'COLUMN',
N'OrganizationNode'
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'rowquid'
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'
EXEC sp addextendedproperty N'MS Description', N'Number of available sick leave
hours.', 'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'COLUMN', N'SickLeaveHours'
GO
EXEC sp addextendedproperty N'MS Description', N'Number of available vacation hours.',
'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'COLUMN', N'VacationHours'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [BirthDate] >=
''1930-\overline{0}1-01'' AND [BirthDate] <\overline{=} dateadd(year,(-18),GETDATE())', 'SCHEMA', N'Human-
Resources', 'TABLE', N'Employee', 'CONSTRAINT', N'CK_Employee_BirthDate'
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'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [HireDate] >= ''1996-
07-01'' AND [HireDate] <= dateadd(day,(1),GETDATE())', 'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'CONSTRAINT', N'CK_Employee_HireDate'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [MaritalStatus]=''s''
OR [MaritalStatus]=''m'' OR [MaritalStatus]=''S'' OR [MaritalStatus]=''M''', 'SCHEMA',
N'HumanResources', 'TABLE', N'Employee', 'CONSTRAINT', N'CK_Employee_MaritalStatus'
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'
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'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of 1',
'SCHEMA, N'HumanResources', 'TABLE', N'Employee', 'CONSTRAINT', N'DF Employee Current-
Flag'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'CONSTRAINT', N'DF_-
Employee ModifiedDate'
GO
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of NEWID()',
'SCHEMA, N'HumanResources', 'TABLE', N'Employee', 'CONSTRAINT', N'DF Employee rowguid'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of 1 (TRUE)',
'SCHEMA, N'HumanResources', 'TABLE', N'Employee', 'CONSTRAINT', N'DF Employee Salaried-
```

```
Flag'
GO
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of 0',
'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'CONSTRAINT', N'DF Employee Sick-
GO
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of 0',
'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'CONSTRAINT', N'DF Employee Vacation-
GO
{\tt EXEC} \ {\tt sp\_addextended} property \ {\tt 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'Primary key (clustered) constraint',
'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'CONSTRAINT', N'PK_Employee_Business-
EntityID'
GO
EXEC sp addextendedproperty N'MS Description', N'Unique nonclustered index.', 'SCHEMA',
N'HumanResources', 'TABLE', N'Employee', 'INDEX', N'AK Employee LoginID'
EXEC sp addextendedproperty N'MS Description', N'Unique nonclustered index.', 'SCHEMA',
N'HumanResources', 'TABLE', N'Employee', 'INDEX', N'AK Employee NationalIDNumber'
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'
EXEC sp addextendedproperty N'MS Description', N'Clustered index created by a primary
key constraint.', 'SCHEMA', N'HumanResources', 'TABLE', N'Employee', 'INDEX', N'PK -
Employee BusinessEntityID'
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]
[Production].[Document]

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PK P FKP	BusinessEntityID Employee identification number. Foreign key to Employee.BusinessEntityID.	int	4	NOT NULL	
PKO III.	DepartmentID Department in which the employee worked including currently. Foreign key to Department.DepartmentID.	smallint	2	NOT NULL	
PKO	ShiftID Identifies which 8-hour shift the employee works. Foreign key to Shift.Shift.ID.	tinyint	1	NOT NULL	
PK/C	StartDate Date the employee started work in the department.	date	3	NOT NULL	
	EndDate Date the employee left the department. NULL = Current department.	date	3	NULL allowed	
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Key	Name	Key Columns	Unique
PK <mark>P</mark> C	PK_EmployeeDepartmentHistory_BusinessEntityID_StartDate_DepartmentID	BusinessEntity- ID, StartDate, DepartmentID, ShiftID	True
	IX_EmployeeDepartmentHistory_DepartmentID Nonclustered index.	DepartmentID	
	IX_EmployeeDepartmentHistory_ShiftID Nonclustered index.	ShiftID	

Check Constraints

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

Foreign Keys

Name	Columns	
FK_EmployeeDepartmentHistory_Department DepartmentID Foreign key constraint referencing Department.DepartmentID.	DepartmentID->[Human-Resources].[Department].[DepartmentID]	
FK_EmployeeDepartmentHistory_Employee_Business- EntityID Foreign key constraint referencing Employee.Employee- ID.	BusinessEntityID->[Human- Resources].[Employee].[BusinessEntityID]	
FK_EmployeeDepartmentHistory_Shift_ShiftID Foreign key constraint referencing Shift.ShiftID	ShiftID->[HumanResources].[Shift].[ShiftID]	

```
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 Modified-
Date] DEFAULT (getdate())
) ON [PRIMARY]
ALTER TABLE [HumanResources].[EmployeeDepartmentHistory] ADD CONSTRAINT [CK_Employee-
DepartmentHistory EndDate] CHECK (([EndDate]>=[StartDate] OR [EndDate] IS NULL))
ALTER TABLE [HumanResources].[EmployeeDepartmentHistory] ADD CONSTRAINT [PK Employee-
DepartmentHistory BusinessEntityID StartDate DepartmentID] PRIMARY KEY CLUSTERED
([BusinessEntityID], [StartDate], [DepartmentID], [ShiftID]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX EmployeeDepartmentHistory_DepartmentID] ON [Human-
Resources].[EmployeeDepartmentHistory] ([DepartmentID]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX EmployeeDepartmentHistory ShiftID] ON [Human-
Resources].[EmployeeDepartmentHistory] ([ShiftID]) ON [PRIMARY]
ALTER TABLE [HumanResources].[EmployeeDepartmentHistory] ADD CONSTRAINT [FK Employee-
DepartmentHistory_Department_DepartmentID] FOREIGN KEY ([DepartmentID]) REFERENCES
[HumanResources].[Department] ([DepartmentID])
```

```
ALTER TABLE [HumanResources].[EmployeeDepartmentHistory] ADD CONSTRAINT [FK Employee-
DepartmentHistory Employee BusinessEntityID] FOREIGN KEY ([BusinessEntityID])
REFERENCES [HumanResources].[Employee] ([BusinessEntityID])
ALTER TABLE [HumanResources].[EmployeeDepartmentHistory] ADD CONSTRAINT [FK Employee-
DepartmentHistory Shift ShiftID] FOREIGN KEY ([ShiftID]) REFERENCES [Human-
Resources].[Shift] ([ShiftID])
EXEC sp_addextendedproperty N'MS_Description', N'Employee department transfers.',
'SCHEMA', N'HumanResources', 'TABLE', N'EmployeeDepartmentHistory', NULL, NULL
EXEC sp_addextendedproperty N'MS_Description', N'Employee identification number.
Foreign key to Employee.BusinessEntityID.', 'SCHEMA', N'HumanResources', 'TABLE',
N'EmployeeDepartmentHistory', 'COLUMN', N'BusinessEntityID'
EXEC sp addextendedproperty N'MS Description', N'Department in which the employee
worked including currently. Foreign key to Department.DepartmentID.', 'SCHEMA', N'Human-
Resources', 'TABLE', N'EmployeeDepartmentHistory', 'COLUMN', N'DepartmentID'
EXEC sp addextendedproperty N'MS Description', N'Date the employee left the department.
NULL = Current department.', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeeDepartment-
History', 'COLUMN', N'EndDate'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeeDepartmentHistory',
'COLUMN', N'ModifiedDate'
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'
EXEC sp addextendedproperty N'MS Description', N'Date the employee started work in the
department.', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeeDepartmentHistory',
'COLUMN', N'StartDate'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [EndDate] >= [Start-
Date] OR [EndDate] IS NUL', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeeDepartment-
History', 'CONSTRAINT', N'CK EmployeeDepartmentHistory EndDate'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeeDepartmentHistory',
'CONSTRAINT', N'DF EmployeeDepartmentHistory ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Foreign key constraint referencing
Department.DepartmentID.', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeeDepartment-History', 'CONSTRAINT', N'FK_EmployeeDepartmentHistory_Department_DepartmentID'
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'
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'
EXEC sp addextendedproperty N'MS Description', N'Primary key (clustered) constraint',
'SCHEMA', N'HumanResources', 'TABLE', N'EmployeeDepartmentHistory', 'CONSTRAINT', N'PK_-
EmployeeDepartmentHistory BusinessEntityID StartDate DepartmentID'
EXEC sp addextendedproperty N'MS Description', N'Nonclustered index.', 'SCHEMA',
N'HumanResources', 'TABLE', N'EmployeeDepartmentHistory', 'INDEX', N'IX Employee-
```

Project > localhost > User databases > AdventureWorks2016 > Tables > HumanResources.EmployeeDepartment-History

```
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'EmployeeDepartmentHistory',
'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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKC FKD	BusinessEntityID Employee identification number. Foreign key to Employee.BusinessEntityID.	int	4	NOT NULL	
PKP C	RateChangeDate Date the change in pay is effective	datetime	8	NOT NULL	
	Rate Salary hourly rate.	money	8	NOT NULL	
	PayFrequency 1 = Salary received monthly, 2 = Salary received biweekly	tinyint	1	NOT NULL	
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Key	Name	Key Columns	Unique
PKC	PK_EmployeePayHistory_BusinessEntityID_RateChangeDate Primary key (clustered) constraint	BusinessEntityID, RateChangeDate	True

Check Constraints

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

Foreign Keys

Name	Columns
FK_EmployeePayHistory_Employee_BusinessEntityID Foreign key constraint referencing Employee.Employee-ID.	BusinessEntityID->[Human- Resources].[Employee].[BusinessEntityID]

```
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)))
ALTER TABLE [HumanResources].[EmployeePayHistory] ADD CONSTRAINT [CK EmployeePay-
History Rate] CHECK (([Rate]>=(6.50) AND [Rate]<=(200.00)))
ALTER TABLE [HumanResources].[EmployeePayHistory] ADD CONSTRAINT [PK EmployeePay-
History BusinessEntityID RateChangeDate] PRIMARY KEY CLUSTERED ([BusinessEntityID],
[RateChangeDate]) ON [PRIMARY]
ALTER TABLE [HumanResources].[EmployeePayHistory] ADD CONSTRAINT [FK EmployeePay-
History Employee BusinessEntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [Human-
Resources].[Employee] ([BusinessEntityID])
EXEC sp addextendedproperty N'MS Description', N'Employee pay history.', 'SCHEMA',
N'HumanResources', 'TABLE', N'EmployeePayHistory', NULL, NULL
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'EmployeePayHistory',
'COLUMN', N'PayFrequency'
GO
EXEC sp addextendedproperty N'MS Description', N'Salary hourly rate.', 'SCHEMA',
N'HumanResources', 'TABLE', N'EmployeePayHistory', 'COLUMN', N'Rate'
EXEC sp addextendedproperty N'MS Description', N'Date the change in pay is effective',
'SCHEMA', N'HumanResources', 'TABLE', N'EmployeePayHistory', 'COLUMN', N'RateChange-
Date'
```

```
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'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [Rate] >= (6.50) AND
[Rate] <= (200.00)', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeePayHistory',
'CONSTRAINT', N'CK_EmployeePayHistory_Rate'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeePayHistory', 'CONSTRAINT',
N'DF_EmployeePayHistory ModifiedDate'
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'
EXEC sp addextendedproperty N'MS_Description', N'Primary key (clustered) constraint',
'SCHEMA', N'HumanResources', 'TABLE', N'EmployeePayHistory', 'CONSTRAINT', N'PK -
EmployeePayHistory_BusinessEntityID_RateChangeDate'
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a primary
key constraint.', 'SCHEMA', N'HumanResources', 'TABLE', N'EmployeePayHistory', 'INDEX',
N'PK EmployeePayHistory BusinessEntityID RateChangeDate'
```

[HumanResources].[Employee] HumanResources

Used By

[HumanResources]. [uspUpdateEmployeeHireInfo]

Ⅲ [HumanResources].[JobCandidate]

MS_Description

Résumés submitted to Human Resources by job applicants.

Properties

Property	Value
Full Text Catalog	AW2016FullTextCatalog
Full Text Key Index	PK_JobCandidate_JobCandidateID
Row Count (~)	13
Created	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Lengt h (Bytes)	Nullab ility	Full Text Index ed	Langu age	ldentit y	Defaul t
P <mark>/</mark> C	JobCandidateID Primary key for Job- Candidate records.	int	4	NOT NULL			1 - 1	
F P	BusinessEntityID Employee identification number if applicant was hired. Foreign key to Employee.Business- EntityID.	int	4	NULL allowe d				
	Resume Résumé in XML format.	xml([Human- Resources].[HRRes umeSchema- Collection])	max	NULL allowe d	True	1033		
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL				(getda te())

Indexes

Key	Name	Key Columns	Unique
PK	PK_JobCandidate_JobCandidateID Primary key (clustered) constraint	JobCandidateID	True
	IX_JobCandidate_BusinessEntityID Nonclustered index.	BusinessEntityID	

Foreign Keys

Name	Columns
FK_JobCandidate_Employee_BusinessEntityID Foreign key constraint referencing Employee.Employee- ID.	BusinessEntityID->[Human- Resources].[Employee].[BusinessEntityID]

```
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]
ALTER TABLE [HumanResources].[JobCandidate] ADD CONSTRAINT [PK JobCandidate Job-
CandidateID] PRIMARY KEY CLUSTERED ([JobCandidateID]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX JobCandidate BusinessEntityID] ON [HumanResources].[Job-
Candidate] ([BusinessEntityID]) ON [PRIMARY]
ALTER TABLE [HumanResources].[JobCandidate] ADD CONSTRAINT [FK JobCandidate Employee -
BusinessEntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [Human-
Resources].[Employee] ([BusinessEntityID])
EXEC sp addextendedproperty N'MS Description', N'Résumés submitted to Human Resources
by job applicants.', 'SCHEMA', N'HumanResources', 'TABLE', N'JobCandidate', NULL, NULL
EXEC sp_addextendedproperty N'MS_Description', N'Employee identification number if
applicant was hired. Foreign key to Employee.BusinessEntityID.', 'SCHEMA', N'Human-
Resources', 'TABLE', N'JobCandidate', 'COLUMN', N'BusinessEntityID'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for JobCandidate
records.', 'SCHEMA', N'HumanResources', 'TABLE', N'JobCandidate', 'COLUMN', N'Job-
CandidateID'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'HumanResources', 'TABLE', N'JobCandidate', 'COLUMN', N'Modified-
Date'
EXEC sp_addextendedproperty N'MS_Description', N'Résumé in XML format.', 'SCHEMA', N'HumanResources', 'TABLE', N'JobCandidate', 'COLUMN', N'Resume'
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'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint',
'SCHEMA', N'HumanResources', 'TABLE', N'JobCandidate', 'CONSTRAINT', N'PK_JobCandidate -
JobCandidateID'
```

```
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA', N'HumanResources', 'TABLE', N'JobCandidate', 'INDEX', N'IX_JobCandidate_BusinessEntity-ID'

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_JobCandidate_JobCandidate_JobCandidate_JobCandidateID'

GO

CREATE FULLTEXT INDEX ON [HumanResources].[JobCandidate] KEY INDEX [PK_JobCandidate_JobCandidateID] ON [AW2016FullTextCatalog]

GO

ALTER FULLTEXT INDEX ON [HumanResources].[JobCandidate] ADD ([Resume] LANGUAGE 1033)

GO
```

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

Used By

[HumanResources].[vJobCandidate] [HumanResources].[vJobCandidateEducation] [HumanResources].[vJobCandidateEmployment] [dbo].[uspSearchCandidateResumes] AW2016FullTextCatalog

[HumanResources].[Shift]

MS_Description

Work shift lookup table.

Properties

Property	Value	
Collation	SQL_Latin1_General_CP1_CI_AS	
Row Count (~)	3	
Created	17:24:40 2017. október 19., csütörtök	
Last Modified	17:24:51 2017. október 19., csütörtök	

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
PK <mark>P</mark> G	ShiftID Primary key for Shift records.	tinyint	1	NOT NULL	1 - 1	
.	Name Shift description.	[dbo].[Name]	100	NOT NULL		
ı.t.	StartTime Shift start time.	time	5	NOT NULL		
<u>.</u>	EndTime Shift end time.	time	5	NOT NULL		
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

Key	Name	Key Columns	Unique
PK G	PK_Shift_ShiftID Primary key (clustered) constraint	ShiftID	True
	AK_Shift_Name Unique nonclustered index.	Name	True
	AK_Shift_StartTime_EndTime Unique nonclustered index.	StartTime, EndTime	True

```
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]
ALTER TABLE [HumanResources].[Shift] ADD CONSTRAINT [PK Shift ShiftID] PRIMARY KEY
CLUSTERED ([ShiftID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK_Shift_Name] ON [HumanResources].[Shift] ([Name])
ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK Shift StartTime EndTime] ON [Human-
Resources].[Shift] ([StartTime], [EndTime]) ON [PRIMARY]
EXEC sp addextendedproperty N'MS Description', N'Work shift lookup table.', 'SCHEMA',
N'HumanResources', 'TABLE', N'Shift', NULL, NULL
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'
EXEC sp addextendedproperty N'MS Description', N'Shift description.', 'SCHEMA', N'Human-
Resources', 'TABLE', N'Shift', 'COLUMN', N'Name'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Shift records.',
'SCHEMA', N'HumanResources', 'TABLE', N'Shift', 'COLUMN', N'ShiftID'
EXEC sp addextendedproperty N'MS Description', N'Shift start time.', 'SCHEMA', N'Human-
Resources', 'TABLE', N'Shift', 'COLUMN', N'StartTime'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'HumanResources', 'TABLE', N'Shift', 'CONSTRAINT', N'DF Shift -
ModifiedDate'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint',
'SCHEMA', N'HumanResources', 'TABLE', N'Shift', 'CONSTRAINT', N'PK Shift ShiftID'
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.', 'SCHEMA',
N'HumanResources', 'TABLE', N'Shift', 'INDEX', N'AK Shift Name'
EXEC sp addextendedproperty N'MS Description', N'Unique nonclustered index.', 'SCHEMA',
N'HumanResources', 'TABLE', N'Shift', 'INDEX', N'AK Shift StartTime EndTime'
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
```

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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullabil ity	Identity	Identity Replica tion	Default
P <mark>//</mark> C	AddressID Primary key for Address records.	int	4	NOT NULL	1 - 1	False	
ı.	AddressLine1 First street address line.	nvarchar(60)	120	NOT NULL			
ı.	AddressLine2 Second street address line.	nvarchar(60)	120	NULL allowed			
ı.	City Name of the city.	nvarchar(30)	60	NOT NULL			
(2) ►	StateProvinceID Unique identification number for the state or province. Foreign key to StateProvince table.	int	4	NOT NULL			
ф	PostalCode Postal code for the street address.	nvarchar(15)	30	NOT NULL			
	SpatialLocation Latitude and longitude of this address.	geography	max	NULL allowed			
ů.	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL			(newid()
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL			(getdate ())

Indexes

Key	Name	Key Columns	Unique
PK	PK_Address_AddressID Primary key (clustered) constraint	AddressID	True
	AK_Address_rowguid Unique nonclustered index. Used to support replication samples.	rowguid	True
	IX_Address_AddressLine1_AddressLine2_City_StateProvinceID_PostalCode Nonclustered index.	AddressLine1, AddressLine2, City, State- ProvinceID, PostalCode	True
	IX_Address_StateProvinceID Nonclustered index.	StateProvince-ID	

Foreign Keys

Name	Columns
FK_Address_StateProvince_StateProvinceID Foreign key constraint referencing StateProvince.State- ProvinceID.	StateProvinceID->[Person].[StateProvince].[State-ProvinceID]

```
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]
CREATE UNIQUE NONCLUSTERED INDEX [IX_Address_AddressLine1_AddressLine2_City_State-
ProvinceID_PostalCode] ON [Person].[Address] ([AddressLine1], [AddressLine2], [City],
[StateProvinceID], [PostalCode]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK_Address_rowguid] ON [Person].[Address] ([rowguid])
ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX_Address_StateProvinceID] ON [Person].[Address] ([State-
ProvinceID]) ON [PRIMARY]
ALTER TABLE [Person].[Address] ADD CONSTRAINT [FK Address StateProvince StateProvince
```

```
ID] FOREIGN KEY ([StateProvinceID]) REFERENCES [Person].[StateProvince] ([StateProvince-
ID1)
EXEC sp_addextendedproperty N'MS_Description', N'Street address information for
customers, employees, and vendors.', 'SCHEMA', N'Person', 'TABLE', N'Address', NULL,
EXEC sp addextendedproperty N'MS Description', N'Primary key for Address records.',
'SCHEMA', N'Person', 'TABLE', N'Address', 'COLUMN', N'AddressID'
EXEC sp_addextendedproperty N'MS_Description', N'First street address line.', 'SCHEMA',
N'Person', 'TABLE', N'Address', 'COLUMN', N'AddressLine1'
EXEC sp addextendedproperty N'MS Description', N'Second street address line.',
'SCHEMA', N'Person', 'TABLE', N'Address', 'COLUMN', N'AddressLine2'
EXEC sp addextendedproperty N'MS Description', N'Name of the city.', 'SCHEMA',
N'Person', 'TABLE', N'Address', 'COLUMN', N'City'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Person', 'TABLE', N'Address', 'COLUMN', N'ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Postal code for the street address.',
'SCHEMA', N'Person', 'TABLE', N'Address', 'COLUMN', N'PostalCode'
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'
EXEC sp addextendedproperty N'MS Description', N'Latitude and longitude of this
address.', 'SCHEMA', N'Person', 'TABLE', N'Address', 'COLUMN', N'SpatialLocation'
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'
EXEC sp addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'Address', 'CONSTRAINT', N'DF Address -
ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of NEWID()',
'SCHEMA', N'Person', 'TABLE', N'Address', 'CONSTRAINT', N'DF Address rowquid'
CO
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'
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'
EXEC sp addextendedproperty N'MS Description', N'Nonclustered index.', 'SCHEMA',
N'Person', 'TABLE', N'Address', "INDEX', N'IX_Address_AddressLine1_AddressLine2_City_-
StateProvinceID PostalCode'
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

GO
```

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
₽ <mark>%</mark> G	AddressTypeID Primary key for AddressType records.	int	4	NOT NULL	1 - 1	
.t.	Name Address type description. For example, Billing, Home, or Shipping.	[dbo].[Name]	100	NOT NULL		
#	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL		(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

Key	Name	Key Columns	Unique
PK G	PK_AddressType_AddressTypeID Primary key (clustered) constraint	AddressTypeID	True
	AK_AddressType_Name Unique nonclustered index.	Name	True
	AK_AddressType_rowguid Unique nonclustered index. Used to support replication samples.	rowguid	True

```
CREATE TABLE [Person].[AddressType]
[AddressTypeID] [int] NOT NULL IDENTITY(1, 1),
[Name] [dbo].[Name] NOT NULL,
[rowquid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF AddressType rowquid]
DEFAULT (newid()),
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF AddressType ModifiedDate] DEFAULT
(getdate())
) ON [PRIMARY]
ALTER TABLE [Person].[AddressType] ADD CONSTRAINT [PK AddressType AddressTypeID]
PRIMARY KEY CLUSTERED ([AddressTypeID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK AddressType_Name] ON [Person].[AddressType]
([Name]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK AddressType rowguid] ON [Person].[AddressType]
([rowquid]) ON [PRIMARY]
EXEC sp_addextendedproperty N'MS_Description', N'Types of addresses stored in the
Address table. ', 'SCHEMA', N'Person', 'TABLE', N'AddressType', NULL, NULL
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for AddressType records.', 'SCHEMA', N'Person', 'TABLE', N'AddressType', 'COLUMN', N'AddressTypeID'
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',
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_AddressType_rowguid'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Person', 'TABLE', N'AddressType', 'CONSTRAINT', N'PK_AddressType_Address-
TypeID'
EXEC sp addextendedproperty N'MS Description', N'Unique nonclustered index.', 'SCHEMA',
N'Person', 'TABLE', N'AddressType', 'INDEX', N'AK AddressType Name'
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_Address-
```

Type_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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullabili ty	Identity	Identity Replicati on	Default
PK <mark>P</mark> C	BusinessEntityID Primary key for all customers, vendors, and employees.	int	4	NOT NULL	1 - 1	False	
ф	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL			(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL			(getdate())

Indexes

Key	Name	Key Columns	Unique
PK	PK_BusinessEntity_BusinessEntityID Primary key (clustered) constraint	BusinessEntity- ID	True
	AK_BusinessEntity_rowguid Unique nonclustered index. Used to support replication samples.	rowguid	True

```
CREATE TABLE [Person].[BusinessEntity]

(
[BusinessEntityID] [int] NOT NULL IDENTITY(1, 1) NOT FOR REPLICATION,

[rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF_BusinessEntity_rowguid]

DEFAULT (newid()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_BusinessEntity_ModifiedDate] DEFAULT
```

```
(getdate())
) ON [PRIMARY]
ALTER TABLE [Person]. [BusinessEntity] ADD CONSTRAINT [PK BusinessEntity BusinessEntity-
ID] PRIMARY KEY CLUSTERED ([BusinessEntityID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK BusinessEntity rowguid] ON [Person].[Business-
Entity] ([rowguid]) ON [PRIMARY]
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
EXEC sp addextendedproperty N'MS Description', N'Primary key for all customers,
vendors, and employees.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntity', 'COLUMN',
N'BusinessEntityID'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntity', 'COLUMN', N'ModifiedDate'
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'rowquid'
{\tt EXEC} \ {\tt sp\_addextended} property \ {\tt N'MS\_Description'}, \ {\tt N'Default} \ {\tt constraint} \ {\tt value} \ {\tt of}
GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntity', 'CONSTRAINT', N'DF -
BusinessEntity_ModifiedDate'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of NEWID()',
'SCHEMA', N'Person', 'TABLE', N'BusinessEntity', 'CONSTRAINT', N'DF Business-
Entity rowguid'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Person', 'TABLE', N'BusinessEntity', 'CONSTRAINT', N'PK BusinessEntity -
BusinessEntityID'
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'
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'
```

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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKEFKE	BusinessEntityID Primary key. Foreign key to Business- Entity.BusinessEntityID.	int	4	NOT NULL	
P <mark>/</mark> P	AddressID Primary key. Foreign key to Address.AddressID.	int	4	NOT NULL	
7 2	AddressTypeID Primary key. Foreign key to Address- Type.AddressTypeID.	int	4	NOT NULL	
. 1.	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL	(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Key	Name	Key Columns	Unique
PK P G	PK_BusinessEntityAddress_BusinessEntityID_AddressID_AddressTypeID Primary key (clustered) constraint	BusinessEntity- ID, AddressID, AddressTypeID	True
	AK_BusinessEntityAddress_rowguid Unique nonclustered index. Used to support replication samples.	rowguid	True
	IX_BusinessEntityAddress_AddressID Nonclustered index.	AddressID	
	IX_BusinessEntityAddress_AddressTypeID Nonclustered index.	AddressTypeID	

Foreign Keys

Name	Columns
FK_BusinessEntityAddress_Address_AddressID Foreign key constraint referencing Address.AddressID.	AddressID->[Person].[Address].[AddressID]
FK_BusinessEntityAddress_AddressType_AddressType-ID Foreign key constraint referencing AddressType.Address-TypeID.	AddressTypeID->[Person].[AddressType].[AddressType-ID]
FK_BusinessEntityAddress_BusinessEntity_Business- EntityID Foreign key constraint referencing Business- Entity.BusinessEntityID.	BusinessEntityID->[Person].[BusinessEntity].[Business-EntityID]

```
CREATE TABLE [Person].[BusinessEntityAddress]
[BusinessEntityID] [int] NOT NULL,
[AddressID] [int] NOT NULL,
[AddressTypeID] [int] NOT NULL,
[rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF BusinessEntity-
Address rowguid] DEFAULT (newid()),
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_BusinessEntityAddress_ModifiedDate]
DEFAULT (getdate())
) ON [PRIMARY]
ALTER TABLE [Person].[BusinessEntityAddress] ADD CONSTRAINT [PK BusinessEntityAddress -
BusinessEntityID AddressID AddressTypeID] PRIMARY KEY CLUSTERED ([BusinessEntityID],
[AddressID], [AddressTypeID]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX BusinessEntityAddress AddressID] ON [Person].[Business-
EntityAddress] ([AddressID]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX BusinessEntityAddress AddressTypeID] ON
[Person].[BusinessEntityAddress] ([AddressTypeID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK BusinessEntityAddress rowguid] ON
[Person].[BusinessEntityAddress] ([rowguid]) ON [PRIMARY]
ALTER TABLE [Person].[BusinessEntityAddress] ADD CONSTRAINT [FK BusinessEntityAddress -
Address AddressID] FOREIGN KEY ([AddressID]) REFERENCES [Person].[Address] ([Address-
TD1)
GO
ALTER TABLE [Person].[BusinessEntityAddress] ADD CONSTRAINT [FK BusinessEntityAddress -
AddressType AddressTypeID] FOREIGN KEY ([AddressTypeID]) REFERENCES [Person].[Address-
Type] ([AddressTypeID])
GO
ALTER TABLE [Person].[BusinessEntityAddress] ADD CONSTRAINT [FK BusinessEntityAddress -
BusinessEntity BusinessEntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES
[Person].[BusinessEntity] ([BusinessEntityID])
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
```

```
EXEC sp addextendedproperty N'MS Description', N'Primary key. Foreign key to
Address.AddressID.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityAddress', 'COLUMN',
N'AddressID'
EXEC sp addextendedproperty N'MS Description', N'Primary key. Foreign key to Address-
Type.AddressTypeID.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityAddress', 'COLUMN',
N'AddressTvpeID'
EXEC sp addextendedproperty N'MS Description', N'Primary key. Foreign key to Business-
Entity.BusinessEntityID.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityAddress',
'COLUMN', N'BusinessEntityID'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityAddress', 'COLUMN', N'Modified-
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'
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 Business-
EntityAddress 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 Business-
EntityAddress BusinessEntityID AddressID AddressTypeID'
GO
EXEC sp addextendedproperty N'MS Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityAddress',
'INDEX', N'AK BusinessEntityAddress rowguid'
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Person', 'TABLE', N'BusinessEntityAddress', 'INDEX', N'IX_BusinessEntityAddress_-
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Person', 'TABLE', N'BusinessEntityAddress', 'INDEX', N'IX BusinessEntityAddress -
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
```

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKEFKE	BusinessEntityID Primary key. Foreign key to Business- Entity.BusinessEntityID.	int	4	NOT NULL	
P <mark>/2</mark> .f.	PersonID Primary key. Foreign key to Person.BusinessEntityID.	int	4	NOT NULL	
P	ContactTypeID Primary key. Foreign key to Contact- Type.ContactTypeID.	int	4	NOT NULL	
rii.	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL	(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

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

Foreign Keys

Name	Columns
FK_BusinessEntityContact_BusinessEntity_Business- EntityID Foreign key constraint referencing Business- Entity.BusinessEntityID.	BusinessEntityID->[Person].[BusinessEntity].[Business-EntityID]
FK_BusinessEntityContact_ContactType_ContactTypeID Foreign key constraint referencing ContactType.Contact-TypeID.	ContactTypeID->[Person].[ContactType].[ContactTypeID]
FK_BusinessEntityContact_Person_PersonID Foreign key constraint referencing Person.BusinessEntity-ID.	PersonID->[Person].[BusinessEntityID]

```
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_ModifiedDate]
DEFAULT (getdate())
) ON [PRIMARY]
ALTER TABLE [Person].[BusinessEntityContact] ADD CONSTRAINT [PK BusinessEntityContact -
BusinessEntityID PersonID ContactTypeID] PRIMARY KEY CLUSTERED ([BusinessEntityID],
[PersonID], [ContactTypeID]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX BusinessEntityContact ContactTypeID] ON
[Person].[BusinessEntityContact] ([ContactTypeID]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX BusinessEntityContact PersonID] ON [Person].[Business-
EntityContact] ([PersonID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK BusinessEntityContact rowguid] ON
[Person].[BusinessEntityContact] ([rowguid]) ON [PRIMARY]
ALTER TABLE [Person].[BusinessEntityContact] ADD CONSTRAINT [FK BusinessEntityContact -
BusinessEntity BusinessEntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES
[Person].[BusinessEntity] ([BusinessEntityID])
ALTER TABLE [Person].[BusinessEntityContact] ADD CONSTRAINT [FK BusinessEntityContact -
ContactType ContactTypeID] FOREIGN KEY ([ContactTypeID]) REFERENCES [Person].[Contact-
Type] ([ContactTypeID])
GO
ALTER TABLE [Person].[BusinessEntityContact] ADD CONSTRAINT [FK BusinessEntityContact -
Person PersonID] FOREIGN KEY ([PersonID]) REFERENCES [Person]. [Person] ([BusinessEntity-
ID1)
GO
EXEC sp addextendedproperty N'MS Description', N'Cross-reference table mapping stores,
vendors, and employees to people, 'SCHEMA', N'Person', 'TABLE', N'BusinessEntity-
Contact', NULL, NULL
GO
```

```
EXEC sp addextendedproperty N'MS Description', N'Primary key. Foreign key to Business-
Entity. BusinessEntityID.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityContact',
'COLUMN', N'BusinessEntityID'
EXEC sp addextendedproperty N'MS Description', N'Primary key. Foreign key to Contact-
Type.ContactTypeID.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityContact', 'COLUMN',
N'ContactTvpeID'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityContact', 'COLUMN', N'Modified-
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 Business-
EntityContact 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 Business-
EntityContact BusinessEntityID PersonID ContactTypeID'
GO
EXEC sp addextendedproperty N'MS Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Person', 'TABLE', N'BusinessEntityContact',
'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_-
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Person', 'TABLE', N'BusinessEntityContact', 'INDEX', N'IX BusinessEntityContact -
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
```

[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	17:24:40 2017. október 19., csütörtök	
Last Modified	17:24:51 2017. október 19., csütörtök	

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
PK <mark>P</mark> G	ContactTypeID Primary key for ContactType records.	int	4	NOT NULL	1 - 1	
.th	Name Contact type description.	[dbo].[Name]	100	NOT NULL		
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

Key	Name	Key Columns	Unique
PK G	PK_ContactType_ContactTypeID Primary key (clustered) constraint	ContactTypeID	True
	AK_ContactType_Name Unique nonclustered index.	Name	True

```
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]
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'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Person', 'TABLE', N'ContactType', 'COLUMN', N'ModifiedDate'
EXEC sp_addextendedproperty N'MS_Description', N'Contact type description.', 'SCHEMA',
N'Person', 'TABLE', N'ContactType', 'COLUMN', N'Name'
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_Contact-
GO
EXEC sp addextendedproperty N'MS Description', N'Unique nonclustered index.', 'SCHEMA',
N'Person', 'TABLE', N'ContactType', 'INDEX', N'AK ContactType Name'
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a primary
key constraint.', 'SCHEMA', N'Person', 'TABLE', N'ContactType', 'INDEX', N'PK Contact-
Type_ContactTypeID'
GO
```

[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	17:24:40 2017. október 19., csütörtök	
Last Modified	17:24:52 2017. október 19., csütörtök	

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PK C	CountryRegionCode ISO standard code for countries and regions.	nvarchar(3)	6	NOT NULL	
.	Name Country or region name.	[dbo].[Name]	100	NOT NULL	
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Key	Name	Key Columns	Unique
PK C	PK_CountryRegion_CountryRegionCode Primary key (clustered) constraint	CountryRegionCode	True
	AK_CountryRegion_Name Unique nonclustered index.	Name	True

```
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]
```

```
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK CountryRegion Name] ON [Person].[CountryRegion]
([Name]) ON [PRIMARY]
EXEC sp addextendedproperty N'MS Description', N'Lookup table containing the ISO
standard codes for countries and regions.', 'SCHEMA', N'Person', 'TABLE', N'Country-
Region', 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'CountryRegion-
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'
EXEC sp addextendedproperty N'MS Description', N'Country or region name.', 'SCHEMA',
N'Person', 'TABLE', N'CountryRegion', 'COLUMN', N'Name'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'CountryRegion', 'CONSTRAINT', N'DF Country-
Region 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'
EXEC sp addextendedproperty N'MS Description', N'Clustered index created by a primary
key constraint.', 'SCHEMA', N'Person', 'TABLE', N'CountryRegion', 'INDEX', N'PK_Country-
Region CountryRegionCode'
GO
```

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullabilit y	Identity	Default
PKC FK	BusinessEntityID Primary key. Person associated with this email address. Foreign key to Person.BusinessEntityID	int	4	NOT NULL		
P <mark>//</mark> C	EmailAddressID Primary key. ID of this email address.	int	4	NOT NULL	1 - 1	
. 	EmailAddress E-mail address for the person.	nvarchar(50)	100	NULL allowed		
	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL		(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

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

Foreign Keys

Name	Columns
FK_EmailAddress_Person_BusinessEntityID Foreign key constraint referencing Person.BusinessEntity-ID.	BusinessEntityID->[Person].[Person].[BusinessEntityID]

```
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,
[rowquid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF EmailAddress rowquid]
DEFAULT (newid()),
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF EmailAddress ModifiedDate] DEFAULT
(getdate())
) ON [PRIMARY]
ALTER TABLE [Person].[EmailAddress] ADD CONSTRAINT [PK EmailAddress BusinessEntityID -
EmailAddressID] PRIMARY KEY CLUSTERED ([BusinessEntityID], [EmailAddressID]) ON
CREATE NONCLUSTERED INDEX [IX EmailAddress_EmailAddress] ON [Person].[EmailAddress]
([EmailAddress]) ON [PRIMARY]
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.',
'SCHEMA', N'Person', 'TABLE', N'EmailAddress', NULL, NULL
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'
EXEC sp addextendedproperty N'MS Description', N'E-mail address for the person.',
'SCHEMA', N'Person', 'TABLE', N'EmailAddress', 'COLUMN', N'EmailAddress'
EXEC sp addextendedproperty N'MS Description', N'Primary key. ID of this email
address.', 'SCHEMA', N'Person', 'TABLE', N'EmailAddress', 'COLUMN', N'EmailAddressID'
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Person', 'TABLE', N'EmailAddress', 'COLUMN', N'ModifiedDate'
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'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'EmailAddress', 'CONSTRAINT', N'DF Email-
Address ModifiedDate
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of NEWID()',
```

```
"SCHEMA', N'Person', 'TABLE', N'EmailAddress', 'CONSTRAINT', N'DF_EmailAddress_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_EmailAddress_Business-
EntityID_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_Email-
Address_BusinessEntityID_EmailAddressID'

GO
```

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKC FKD	BusinessEntityID	int	4	NOT NULL	
	PasswordHash Password for the e-mail account.	varchar(128)	128	NOT NULL	
	PasswordSalt Random value concatenated with the password string before the password is hashed.	varchar(10)	10	NOT NULL	
	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL	(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Key	Name	Key Columns	Unique
PKP C	PK_Password_BusinessEntityID Primary key (clustered) constraint	BusinessEntityID	True

Foreign Keys

Name	Columns
FK_Password_Person_BusinessEntityID Foreign key constraint referencing Person.BusinessEntity-	BusinessEntityID->[Person].[Person].[BusinessEntityID]

ID.

```
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,
[rowquid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF Password rowquid]
DEFAULT (newid()),
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF Password ModifiedDate] DEFAULT
(getdate())
) ON [PRIMARY]
ALTER TABLE [Person].[Password] ADD CONSTRAINT [PK Password_BusinessEntityID] PRIMARY
KEY CLUSTERED ([BusinessEntityID]) ON [PRIMARY]
ALTER TABLE [Person].[Password] ADD CONSTRAINT [FK Password Person BusinessEntityID]
FOREIGN KEY ([BusinessEntityID]) REFERENCES [Person].[Person] ([BusinessEntityID])
EXEC sp addextendedproperty N'MS Description', N'One way hashed authentication
information', 'SCHEMA', N'Person', 'TABLE', N'Password', NULL, NULL
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Person', 'TABLE', N'Password', 'COLUMN', N'ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Password for the e-mail account.',
'SCHEMA', N'Person', 'TABLE', N'Password', 'COLUMN', N'PasswordHash'
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'
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'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'Password', 'CONSTRAINT', N'DF Password -
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of NEWID()',
'SCHEMA', N'Person', 'TABLE', N'Password', 'CONSTRAINT', N'DF Password rowquid'
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'
EXEC sp addextendedproperty N'MS Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Person', 'TABLE', N'Password', 'CONSTRAINT', N'PK_Password BusinessEntity-
TD'
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			

Project > localhost > User databases > AdventureWorks2016 > Tables > Person.Password

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:29:01 2021. március 21., vasárnap

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullabi lity	Default
PKPFKP C	BusinessEntityID Primary key for Person records.	int	4	NOT NULL	
III	PersonType Primary type of person: SC = Store Contact, IN = Individual (retail) customer, SP = Sales person, EM = Employee (non-sales), VC = Vendor contact, GC = General contact	nchar(2)	4	NOT NULL	
	NameStyle 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.	[dbo].[NameStyle]	1	NOT NULL	((0))
	Title A courtesy title. For example, Mr. or Ms.	nvarchar(8)	16	NULL allowed	
.	FirstName First name of the person.	[dbo].[Name]	100	NOT NULL	
.	MiddleName Middle name or middle initial of the person.	[dbo].[Name]	100	NULL allowed	
.	LastName Last name of the person.	[dbo].[Name]	100	NOT NULL	
	Suffix Surname suffix. For example, Sr. or Jr.	nvarchar(10)	20	NULL allowed	
[II]	EmailPromotion 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.	int	4	NOT NULL	((0))

.	AdditionalContactInfo Additional contact information about the person stored in xml format.	xml([Person].[Additional- ContactInfoSchema- Collection])	max	NULL allowed	
∴ (4)	Demographics Personal information such as hobbies, and income collected from online shoppers. Used for sales analysis.	xml([Person].[IndividualSurvey- SchemaCollection])	max	NULL allowed	
#	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL	(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdat e())
F	PetID	int	4	NULL allowed	

Indexes

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

Triggers

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

Check Constraints

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

Foreign Keys

Name	Columns
FK_Person_BusinessEntity_BusinessEntityID Foreign key constraint referencing Business- Entity.BusinessEntityID.	BusinessEntityID->[Person].[BusinessEntity].[Business- EntityID]
FK_Person_PetID	PetID->[Person].[Pet].[PetID]

```
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].[AdditionalContactInfoSchema-
Collection]) 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()),
[PetID] [int] NULL
) 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</pre>
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 (([Email-
Promotion]>=(0) AND [EmailPromotion]<=(2)))</pre>
ALTER TABLE [Person]. [Person] ADD CONSTRAINT [CK Person PersonType] CHECK (([Person-
Type] IS NULL OR upper([PersonType])='GC' OR upper([PersonType])='SP' OR upper([Person-
Type])='EM' OR upper([PersonType])='IN' OR upper([PersonType])='VC' OR upper([Person-
Type])='SC'))
GO
ALTER TABLE [Person].[Person] ADD CONSTRAINT [PK Person BusinessEntityID] PRIMARY KEY
CLUSTERED ([BusinessEntityID]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX Person LastName FirstName MiddleName] ON
[Person].[Person] ([LastName], [FirstName], [MiddleName]) ON [PRIMARY]
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
CREATE XML INDEX [XMLVALUE Person Demographics]
ON [Person].[Person] ([Demographics])
USING XML INDEX [PXML Person Demographics]
FOR VALUE
ALTER TABLE [Person].[Person] ADD CONSTRAINT [FK Person BusinessEntity BusinessEntity-
ID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [Person]. [BusinessEntity] ([Business-
EntityID])
ALTER TABLE [Person].[Person] ADD CONSTRAINT [FK Person PetID] FOREIGN KEY ([PetID])
REFERENCES [Person].[Pet] ([PetID])
GO
EXEC sp addextendedproperty N'MS Description', N'Human beings involved with Adventure-
Works: 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'
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'
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 Adventure-
Works, 2 = Contact does wish to receive e-mail promotions from AdventureWorks and
selected partners. ', 'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'Email-
Promotion'
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'
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'
EXEC sp addextendedproperty N'MS Description', N'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.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'Name-
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'
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'
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'
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'
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'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of 0',
'SCHEMA', N'Person', 'TABLE', N'Person', 'CONSTRAINT', N'DF Person EmailPromotion'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'Person', 'CONSTRAINT', N'DF Person Modified-
Date'
GO
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of 0',
'SCHEMA', N'Person', 'TABLE', N'Person', 'CONSTRAINT', N'DF Person NameStyle'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of NEWID()',
'SCHEMA', N'Person', 'TABLE', N'Person', 'CONSTRAINT', N'DF Person rowguid'
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'
EXEC sp addextendedproperty N'MS Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Person', 'TABLE', N'Person', 'CONSTRAINT', N'PK Person BusinessEntityID'
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'
```

```
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', 'TABLE', N'Person', 'INDEX', N'INDEX', N'SCHEMA', N'Person', 'TABLE', N'Person', 'TABLE', N'Person', 'INDEX', N'INDEX', N'SCHEMA', N'PERSON', 'TABLE', N'PERSON', 'INDEX', N'SCHEMA', N'PERSON', 'INDEX', N'SCHEMA',
```

[Person].[BusinessEntity]

[Person].[Pet]

[dbo].[Name]

[dbo].[NameStyle]

Person

[Person].[AdditionalContactInfoSchemaCollection]

[Person].[IndividualSurveySchemaCollection]

Used By

[HumanResources].[Employee]

[Person].[BusinessEntityContact]

[Person].[EmailAddress]

[Person].[Password]

[Person].[PersonPhone]

[Person].[Pet]

[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]. [usp Get Employee Managers]

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKC FK	BusinessEntityID Business entity identification number. Foreign key to Person.BusinessEntityID.	int	4	NOT NULL	
₽ % 	PhoneNumber Telephone number identification number.	[dbo].[Phone]	50	NOT NULL	
PKC FKD	PhoneNumberTypeID Kind of phone number. Foreign key to PhoneNumberType.PhoneNumberTypeID.	int	4	NOT NULL	
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Key	Name	Key Columns	Unique
P∳ <mark>C</mark>	PK_PersonPhone_BusinessEntityID_PhoneNumber_PhoneNumberTypeID	BusinessEntity- ID, Phone- Number, Phone- NumberTypeID	True
	IX_PersonPhone_PhoneNumber Nonclustered index.	PhoneNumber	

Foreign Keys

Name	Columns
FK_PersonPhone_Person_BusinessEntityID Foreign key constraint referencing Person.BusinessEntity-ID.	BusinessEntityID->[Person].[Person].[BusinessEntityID]

FK_PersonPhone_PhoneNumberType_PhoneNumber-TypeID	PhoneNumberTypeID->[Person].[PhoneNumber- Type].[PhoneNumberTypeID]
Foreign key constraint referencing PhoneNumber-	<i>31 11</i>

```
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]
CREATE NONCLUSTERED INDEX [IX PersonPhone PhoneNumber] ON [Person].[PersonPhone]
([PhoneNumber]) ON [PRIMARY]
ALTER TABLE [Person].[PersonPhone] ADD CONSTRAINT [FK_PersonPhone_Person_BusinessEntity-
ID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [Person]. [Person] ([BusinessEntityID])
ALTER TABLE [Person].[PersonPhone] ADD CONSTRAINT [FK PersonPhone PhoneNumberType Phone-
NumberTypeID] FOREIGN KEY ([PhoneNumberTypeID]) REFERENCES [Person].[PhoneNumberType]
([PhoneNumberTypeID])
EXEC sp addextendedproperty N'MS Description', N'Telephone number and type of a
person.', 'SCHEMA', N'Person', 'TABLE', N'PersonPhone', NULL, NULL
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'
EXEC sp addextendedproperty N'MS Description', N'Telephone number identification
number.', 'SCHEMA', N'Person', 'TABLE', N'PersonPhone', 'COLUMN', N'PhoneNumber'
EXEC sp addextendedproperty N'MS Description', N'Kind of phone number. Foreign key to
PhoneNumberType.PhoneNumberTypeID.', 'SCHEMA', N'Person', 'TABLE', N'PersonPhone',
'COLUMN', N'PhoneNumberTypeID'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'PersonPhone', 'CONSTRAINT', N'DF Person-
Phone ModifiedDate'
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'
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_PersonPhone_Business-
EntityID_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_Person-
Phone_BusinessEntityID_PhoneNumber_PhoneNumberTypeID'
GO
```

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

Used By

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

[Person].[Pet]

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	0
Created	17:29:01 2021. március 21., vasárnap
Last Modified	17:29:01 2021. március 21., vasárnap

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PKC	PetID	int	4	NOT NULL
	Name	nvarchar(25)	50	NULL allowed
	Color	nvarchar(15)	30	NULL allowed
	Туре	nvarchar(15)	30	NULL allowed
	Breed	nvarchar(25)	50	NULL allowed
F	OwnerID	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
PK2 C	PK_Pet	PetID	True

Foreign Keys

Name	Columns
FK_Pet_OwnerID	OwnerID->[Person].[Person].[BusinessEntityID]

```
CREATE TABLE [Person].[Pet]

(
[PetID] [int] NOT NULL,

[Name] [nvarchar] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,

[Color] [nvarchar] (15) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,

[Type] [nvarchar] (15) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,

[Breed] [nvarchar] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,

[OwnerID] [int] NOT NULL
```

```
ON [PRIMARY]

GO

ALTER TABLE [Person].[Pet] ADD CONSTRAINT [PK_Pet] PRIMARY KEY CLUSTERED ([PetID]) ON [PRIMARY]

GO

ALTER TABLE [Person].[Pet] ADD CONSTRAINT [FK_Pet_OwnerID] FOREIGN KEY ([OwnerID]) REFERENCES [Person].[Person] ([BusinessEntityID])

GO
```

[Person].[Person]

Person

Used By

[Person].[Person]

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
PK P G	PhoneNumberTypeID Primary key for telephone number type records.	int	4	NOT NULL	1 - 1	
	Name Name of the telephone number type	[dbo].[Name]	100	NOT NULL		
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

Key	Name	Key Columns	Unique
PK	PK_PhoneNumberType_PhoneNumberTypeID Primary key (clustered) constraint	PhoneNumberTypeID	True

```
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_PhoneNumberTypeID] PRIMARY KEY CLUSTERED ([PhoneNumberTypeID]) ON [PRIMARY]
```

```
GO
EXEC sp addextendedproperty N'MS Description', N'Type of phone number of a person.',
'SCHEMA', N'Person', 'TABLE', N'PhoneNumberType', NULL, NULL
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Person', 'TABLE', N'PhoneNumberType', 'COLUMN', N'ModifiedDate'
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'PhoneNumber-
TypeID'
GO
{\tt EXEC} \  \, {\tt sp\_addextended} property \  \, {\tt N'MS\_Description'}, \  \, {\tt N'Default} \  \, {\tt constraint} \  \, {\tt value} \  \, {\tt of}
GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'PhoneNumberType', 'CONSTRAINT', N'DF_Phone-
NumberType_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 Phone-
NumberType PhoneNumberTypeID'
GΟ
```

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:52 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullabilit y	Identity	Default
PK <mark>P</mark> C	StateProvinceID Primary key for StateProvince records.	int	4	NOT NULL	1 - 1	
.	StateProvinceCode ISO standard state or province code.	nchar(3)	6	NOT NULL		
FK/P	CountryRegionCode ISO standard country or region code. Foreign key to Country- Region.CountryRegionCode.	nvarchar(3)	6	NOT NULL		
	IsOnlyStateProvinceFlag 0 = StateProvinceCode exists. 1 = StateProvinceCode unavailable, using CountryRegionCode.	[dbo].[Flag]	1	NOT NULL		((1))
.	Name State or province description.	[dbo].[Name]	100	NOT NULL		
F/P	TerritoryID ID of the territory in which the state or province is located. Foreign key to Sales- Territory.SalesTerritoryID.	int	4	NOT NULL		
#	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL		(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

Key	Name	Key Columns	Unique
PK C	PK_StateProvince_StateProvinceID Primary key (clustered) constraint	StateProvince- ID	True
	AK_StateProvince_Name Unique nonclustered index.	Name	True
	AK_StateProvince_StateProvinceCode_CountryRegionCode Unique nonclustered index.	StateProvince- Code, Country- RegionCode	True
	AK_StateProvince_rowguid Unique nonclustered index. Used to support replication samples.	rowguid	True

Foreign Keys

Name	Columns
FK_StateProvince_CountryRegion_CountryRegionCode Foreign key constraint referencing Country-Region.CountryRegionCode.	CountryRegionCode->[Person].[CountryRegion].[Country-RegionCode]
FK_StateProvince_SalesTerritory_TerritoryID Foreign key constraint referencing SalesTerritory.Territory-ID.	TerritoryID->[Sales].[SalesTerritory].[TerritoryID]

```
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 IsOnly-
StateProvinceFlag] 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]
ALTER TABLE [Person].[StateProvince] ADD CONSTRAINT [PK StateProvince StateProvinceID]
PRIMARY KEY CLUSTERED ([StateProvinceID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK StateProvince Name] ON [Person].[StateProvince]
([Name]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK StateProvince rowguid] ON [Person].[StateProvince]
([rowguid]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK StateProvince StateProvinceCode CountryRegionCode]
ON [Person].[StateProvince] ([StateProvinceCode], [CountryRegionCode]) ON [PRIMARY]
ALTER TABLE [Person].[StateProvince] ADD CONSTRAINT [FK StateProvince CountryRegion -
```

```
CountryRegionCode] FOREIGN KEY ([CountryRegionCode]) REFERENCES [Person].[Country-
Region] ([CountryRegionCode])
ALTER TABLE [Person].[StateProvince] ADD CONSTRAINT [FK StateProvince SalesTerritory -
TerritoryID] FOREIGN KEY ([TerritoryID]) REFERENCES [Sales].[SalesTerritory]
([TerritoryID])
EXEC sp addextendedproperty N'MS Description', N'State and province lookup table.',
'SCHEMA', N'Person', 'TABLE', N'StateProvince', NULL, NULL
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'
EXEC sp addextendedproperty N'MS Description', N'0 = StateProvinceCode exists. 1 =
StateProvinceCode unavailable, using CountryRegionCode.', 'SCHEMA', N'Person', 'TABLE',
N'StateProvince', 'COLUMN', N'IsOnlyStateProvinceFlag'
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Person', 'TABLE', N'StateProvince', 'COLUMN', N'ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'State or province description.',
'SCHEMA', N'Person', 'TABLE', N'StateProvince', 'COLUMN', N'Name'
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'
EXEC sp_addextendedproperty N'MS_Description', N'ISO standard state or province code.',
'SCHEMA', N'Person', 'TABLE', N'StateProvince', 'COLUMN', N'StateProvinceCode'
EXEC sp addextendedproperty N'MS Description', N'Primary key for StateProvince
records.', 'SCHEMA', N'Person', 'TABLE', N'StateProvince', 'COLUMN', N'StateProvinceID'
EXEC sp addextendedproperty N'MS Description', N'ID of the territory in which the state
or province is located. Foreign key to SalesTerritory.SalesTerritoryID.', 'SCHEMA',
N'Person', 'TABLE', N'StateProvince', 'COLUMN', N'TerritoryID'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of 1 (TRUE)',
'SCHEMA', N'Person', 'TABLE', N'StateProvince', 'CONSTRAINT', N'DF StateProvince IsOnly-
StateProvinceFlag'
{\tt EXEC} \  \, {\tt sp\_addextended property} \  \, {\tt N'MS\_Description'}, \  \, {\tt N'Default} \  \, {\tt constraint} \  \, {\tt value} \  \, {\tt of}
GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'StateProvince', 'CONSTRAINT', N'DF State-
Province ModifiedDate'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of NEWID()',
'SCHEMA', N'Person', 'TABLE', N'StateProvince', 'CONSTRAINT', N'DF State-
Province rowguid'
{\tt EXEC} \  \, {\tt sp\_addextended} property \  \, {\tt N'MS\_Description'}, \  \, {\tt N'Foreign} \  \, {\tt key} \  \, {\tt constraint} \  \, {\tt referencing}
CountryRegion.CountryRegionCode.', 'SCHEMA', N'Person', 'TABLE', N'StateProvince',
'CONSTRAINT', N'FK StateProvince CountryRegion CountryRegionCode'
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'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Person', 'TABLE', N'StateProvince', 'CONSTRAINT', N'PK StateProvince State-
```

```
ProvinceID'

GO

EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.', 'SCHEMA', 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.', 'SCHEMA', N'Person', 'TABLE', N'StateProvince', 'INDEX', N'AK_StateProvince_StateProvinceCode_-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_State-Province_StateProvinceID'

GO

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_State-Province_StateProvinceID'

GO
```

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

Used By

Person

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
Р	BillOfMaterialsID Primary key for BillOfMaterials records.	int	4	NOT NULL	1 - 1	
/	ProductAssemblyID Parent product identification number. Foreign key to Product.ProductID.	int	4	NULL allowed		
F/P	ComponentID Component identification number. Foreign key to Product.ProductID.	int	4	NOT NULL		
/	StartDate Date the component started being used in the assembly item.	datetime	8	NOT NULL		(getdate())
	EndDate Date the component stopped being used in the assembly item.	datetime	8	NULL allowed		
F 19	UnitMeasureCode Standard code identifying the unit of measure for the quantity.	nchar(3)	6	NOT NULL		
	BOMLevel Indicates the depth the component is from its parent (AssemblyID).	smallint	2	NOT NULL		
	PerAssemblyQty Quantity of the component needed to create the assembly.	decimal(8,2)	5	NOT NULL		((1.00))
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

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

Check Constraints

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

Foreign Keys

Name	Columns
FK_BillOfMaterials_Product_ComponentID Foreign key constraint referencing Product.ComponentID.	ComponentID->[Production].[Product].[ProductID]
FK_BillOfMaterials_Product_ProductAssemblyID Foreign key constraint referencing Product.Product- AssemblyID.	ProductAssemblyID->[Production].[Product].[ProductID]
FK_BillOfMaterials_UnitMeasure_UnitMeasureCode Foreign key constraint referencing UnitMeasure.Unit- MeasureCode.	UnitMeasureCode->[Production].[UnitMeasure].[Unit-MeasureCode]

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 PerAssembly-
Qty] DEFAULT ((1.00)),
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF BillOfMaterials ModifiedDate] DEFAULT
(getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Production].[BillOfMaterials] ADD CONSTRAINT [CK BillOfMaterials BOMLevel]
CHECK (([ProductAssemblyID] IS NULL AND [BOMLevel]=(0) AND [PerAssemblyQty]=(1.00) OR
[ProductAssemblyID] IS NOT NULL AND [BOMLevel] >= (1)))
ALTER TABLE [Production].[BillOfMaterials] ADD CONSTRAINT [CK BillOfMaterials EndDate]
CHECK (([EndDate]>[StartDate] OR [EndDate] IS NULL))
ALTER TABLE [Production].[BillofMaterials] ADD CONSTRAINT [CK BillofMaterials Per-
AssemblyQty] CHECK (([PerAssemblyQty]>=(1.00)))
ALTER TABLE [Production].[BillOfMaterials] ADD CONSTRAINT [CK BillOfMaterials Product-
AssemblyID] CHECK (([ProductAssemblyID] <> [ComponentID]))
ALTER TABLE [Production].[BillOfMaterials] ADD CONSTRAINT [PK BillOfMaterials BillOf-
MaterialsID] PRIMARY KEY NONCLUSTERED ([BillOfMaterialsID]) ON [PRIMARY]
CREATE UNIQUE CLUSTERED INDEX [AK BillOfMaterials ProductAssemblyID_ComponentID_Start-
Date] ON [Production].[BillOfMaterials] ([ProductAssemblyID], [ComponentID], [Start-
Date]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX BillOfMaterials UnitMeasureCode] ON [Production].[BillOf-
Materials] ([UnitMeasureCode]) ON [PRIMARY]
ALTER TABLE [Production].[BillOfMaterials] ADD CONSTRAINT [FK BillOfMaterials Product -
ComponentID] FOREIGN KEY ([ComponentID]) REFERENCES [Production].[Product] ([Product-
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].[Unit-
Measure] ([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
EXEC sp addextendedproperty N'MS Description', N'Primary key for BillOfMaterials
records.', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials', 'COLUMN', N'BillOf-
MaterialsID'
```

```
EXEC sp addextendedproperty N'MS Description', N'Indicates the depth the component is
from its parent (AssemblyID).', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials',
'COLUMN', N'BOMLevel'
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'
EXEC sp addextendedproperty N'MS Description', N'Date the component stopped being used
in the assembly item.', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials', 'COLUMN',
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials', 'COLUMN', N'Modified-
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'
EXEC sp addextendedproperty N'MS Description', N'Parent product identification number.
Foreign key to Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'BillOf-
Materials', 'COLUMN', N'ProductAssemblyID'
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'
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 [ProductAssemblyID]
IS NULL AND [BOMLevel] = (0) AND [PerAssemblyQty] = (1) OR [ProductAssemblyID] IS NOT
NULL AND [BOMLevel] >= (1)', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials',
'CONSTRAINT', N'CK_BillOfMaterials_BOMLevel'
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'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [PerAssemblyQty] >=
(1.00)', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials', 'CONSTRAINT', N'CK Bill-
OfMaterials PerAssemblyQty'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [ProductAssemblyID]
<> [ComponentID]', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials', 'CONSTRAINT',
N'CK BillOfMaterials ProductAssemblyID'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'BillofMaterials', 'CONSTRAINT', N'DF -
BillOfMaterials ModifiedDate'
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'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'BillOfMaterials', 'CONSTRAINT', N'DF -
BillOfMaterials StartDate'
```

```
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'
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 BillOf-
Materials 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'
EXEC sp addextendedproperty N'MS Description', N'Nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'BillOfMaterials', 'INDEX', N'IX BillOfMaterials UnitMeasure-
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
```

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PK G	CultureID Primary key for Culture records.	nchar(6)	12	NOT NULL	
.	Name Culture description.	[dbo].[Name]	100	NOT NULL	
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

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

```
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])
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'
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'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Culture', 'CONSTRAINT', N'DF Culture -
ModifiedDate'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Production', 'TABLE', N'Culture', 'CONSTRAINT', N'PK_Culture_CultureID'
EXEC sp addextendedproperty N'MS Description', N'Unique nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'Culture', 'INDEX', N'AK Culture Name'
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
```

[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	AW2016FullTextCatalog
Full Text Key Index	PK_Document_DocumentNode
Row Count (~)	13
Created	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Comp uted	Max Length (Bytes	Nullabi lity	Full Text Indexe d	Langu age	Defaul t
P % .fi.	DocumentNode Primary key for Document records.	hierarchyid		892	NOT NULL			
ф	DocumentLevel Depth in the document hierarchy.	smallint	True	2	NULL allowe d			
	Title Title of the document.	nvarchar(50)		100	NOT NULL			
F / P	Owner Employee who controls the document. Foreign key to Employee.Business- EntityID	int		4	NOT NULL			
	FolderFlag 0 = This is a folder, 1 = This is a document.	bit		1	NOT NULL			((0))
ф	FileName File name of the document	nvarchar(400)		800	NOT NULL			
	FileExtension File extension indicating the document type. For example, .doc or .txt.	nvarchar(8)		16	NOT NULL			
ф	Revision Revision number of the document.	nchar(5)		10	NOT NULL			

	ChangeNumber Engineering change approval number.	int	4	NOT NULL			((0))
	Status 1 = Pending approval, 2 = Approved, 3 = Obsolete	tinyint	1	NOT NULL			
	DocumentSummary Document abstract.	nvarchar(max)	max	NULL allowe d	True	1033	
	Document Complete document.	varbinary(max)	max	NULL allowe d	True	1033	
(2)	rowguid ROWGUIDCOL number uniquely identifying the record. Required for File- Stream.	uniqueidentifier	16	NOT NULL			(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL			(getdat e())

Computed columns

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

Indexes

Key	Name	Key Columns	Unique
PK <mark>P</mark> C	PK_Document_DocumentNode Primary key (clustered) constraint	DocumentNode	True
	AK_Document_DocumentLevel_DocumentNode Unique nonclustered index.	DocumentLevel, DocumentNode	True
	AK_Document_rowguid Unique nonclustered index. Used to support FileStream.	rowguid	True
	UQDocumentF73921F744672977	rowguid	True
	IX_Document_FileName_Revision Unique nonclustered index.	FileName, Revision	

Check Constraints

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

Foreign Keys

Name	Columns
FK_Document_Employee_Owner Foreign key constraint referencing Employee.Business- EntityID.	Owner->[HumanResources].[Employee].[BusinessEntity-ID]

```
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,
[rowquid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF Document rowquid]
DEFAULT (newid()),
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_Document_ModifiedDate] DEFAULT
(getdate())
) ON [PRIMARY]
ALTER TABLE [Production].[Document] ADD CONSTRAINT [CK Document Status] CHECK
(([Status]>=(1) AND [Status]<=(3)))
ALTER TABLE [Production].[Document] ADD CONSTRAINT [PK Document DocumentNode] PRIMARY
KEY CLUSTERED ([DocumentNode]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK Document DocumentLevel DocumentNode] ON
[Production].[Document] ([DocumentLevel], [DocumentNode]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX Document FileName Revision] ON [Production].[Document]
([FileName], [Revision]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK Document rowguid] ON [Production].[Document]
([rowguid]) ON [PRIMARY]
ALTER TABLE [Production].[Document] ADD CONSTRAINT [UQ__Document__F73921F744672977]
UNIQUE NONCLUSTERED ([rowguid]) ON [PRIMARY]
ALTER TABLE [Production].[Document] ADD CONSTRAINT [FK Document Employee Owner] FOREIGN
KEY ([Owner]) REFERENCES [HumanResources].[Employee] ([BusinessEntityID])
EXEC sp addextendedproperty N'MS Description', N'Product maintenance documents.',
'SCHEMA', N'Production', 'TABLE', N'Document', NULL, NULL
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'
EXEC sp_addextendedproperty N'MS_Description', N'Depth in the document hierarchy.',
'SCHEMA, N'Production', 'TABLE', N'Document', 'COLUMN', N'DocumentLevel'
EXEC sp addextendedproperty N'MS Description', N'Primary key for Document records.',
'SCHEMA, N'Production', 'TABLE', N'Document', 'COLUMN', N'DocumentNode'
EXEC sp addextendedproperty N'MS_Description', N'Document abstract.', 'SCHEMA',
N'Production', 'TABLE', N'Document', 'COLUMN', N'DocumentSummary'
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'
EXEC sp addextendedproperty N'MS Description', N'File name of the document', 'SCHEMA',
N'Production', 'TABLE', N'Document', 'COLUMN', N'FileName'
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'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'ModifiedDate'
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'
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
= Obsolete', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'Status'
EXEC sp addextendedproperty N'MS Description', N'Title of the document.', 'SCHEMA',
N'Production', 'TABLE', N'Document', 'COLUMN', N'Title'
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 Change-
Number'
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'
EXEC sp addextendedproperty N'MS Description', N'Foreign key constraint referencing
```

```
Employee.BusinessEntityID.', 'SCHEMA', N'Production', 'TABLE', N'Document',
'CONSTRAINT', N'FK_Document_Employee_Owner'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Production', 'TABLE', N'Document', 'CONSTRAINT', N'PK Document Document-
Node!
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'
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'
CREATE FULLTEXT INDEX ON [Production].[Document] KEY INDEX [PK Document DocumentNode]
ON [AW2016FullTextCatalog]
GO
ALTER FULLTEXT INDEX ON [Production].[Document] ADD ([DocumentSummary] LANGUAGE 1033)
ALTER FULLTEXT INDEX ON [Production].[Document] ADD ([Document] TYPE COLUMN [File-
Extension] LANGUAGE 1033)
GO
```

[HumanResources].[Employee] Production

Used By

[Production].[ProductDocument] AW2016FullTextCatalog

[Production].[Illustration]

MS_Description

Bicycle assembly diagrams.

Properties

Property	Value
Row Count (~)	5
Created	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
PK	IllustrationID Primary key for Illustration records.	int	4	NOT NULL	1 - 1	
	Diagram Illustrations used in manufacturing instructions. Stored as XML.	xml	max	NULL allowed		
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

Key	Name	Key Columns	Unique
PK P C	PK_Illustration_IllustrationID Primary key (clustered) constraint	IllustrationID	True

```
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]

GO

ALTER TABLE [Production].[Illustration] ADD CONSTRAINT [PK_Illustration_IllustrationID]

PRIMARY KEY CLUSTERED ([IllustrationID]) ON [PRIMARY]

GO
```

```
EXEC sp addextendedproperty N'MS Description', N'Bicycle assembly diagrams.', 'SCHEMA',
N'Production', 'TABLE', N'Illustration', NULL, NULL
EXEC sp_addextendedproperty N'MS_Description', N'Illustrations used in manufacturing
instructions. Stored as XML.', 'SCHEMA', N'Production', 'TABLE', N'Illustration',
'COLUMN', N'Diagram'
EXEC sp addextendedproperty N'MS Description', N'Primary key for Illustration
records.', 'SCHEMA', N'Production', 'TABLE', N'Illustration', 'COLUMN', N'Illustration-
ID'
GO
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'Illustration', 'COLUMN', N'ModifiedDate'
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
GO
```

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	17:24:40 2017. október 19., csütörtök		
Last Modified	17:24:51 2017. október 19., csütörtök		

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
PK	LocationID Primary key for Location records.	smallint	2	NOT NULL	1 - 1	
.	Name Location description.	[dbo].[Name]	100	NOT NULL		
	CostRate Standard hourly cost of the manufacturing location.	smallmoney	4	NOT NULL		((0.00))
	Availability Work capacity (in hours) of the manufacturing location.	decimal(8,2)	5	NOT NULL		((0.00))
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

Key	Name	Key Columns	Unique
PK G	PK_Location_LocationID Primary key (clustered) constraint	LocationID	True
	AK_Location_Name Unique nonclustered index.	Name	True

Check Constraints

Name	On Column	Constraint	
CK_Location_Availability	Availability	([Availability]>=(0.00))	

Check constraint [Availability] >= (0.00)		
CK_Location_CostRate Check constraint [CostRate] >= (0.00)	CostRate	([CostRate]>=(0.00))

```
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)))
ALTER TABLE [Production].[Location] ADD CONSTRAINT [CK Location CostRate] CHECK (([Cost-
Rate]>=(0.00))
ALTER TABLE [Production]. [Location] ADD CONSTRAINT [PK Location LocationID] PRIMARY KEY
CLUSTERED ([LocationID]) ON [PRIMARY]
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
EXEC sp addextendedproperty N'MS Description', N'Work capacity (in hours) of the
manufacturing location.', 'SCHEMA', N'Production', 'TABLE', N'Location', 'COLUMN',
N'Availability'
EXEC sp addextendedproperty N'MS Description', N'Standard hourly cost of the
manufacturing location.', 'SCHEMA', N'Production', 'TABLE', N'Location', 'COLUMN',
N'CostRate'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Location records.',
'SCHEMA', N'Production', 'TABLE', N'Location', 'COLUMN', N'LocationID'
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'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [Availability] >=
(0.00)', 'SCHEMA', N'Production', 'TABLE', N'Location', 'CONSTRAINT', N'CK Location -
Availability'
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'
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'
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'
GΟ
```

[dbo].[Name] Production

Used By

[Production].[ProductInventory]

[Production].[WorkOrderRouting]

I [Production].[Product]

MS_Description

Products sold or used in the manfacturing of sold products.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	504
Created	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:52 2017. október 19., csütörtök

Columns

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

FK	WeightUnitMeasureCode Unit of measure for Weight column.	nchar(3)	6	NULL allowed	
■	Weight Product weight.	decimal(8,2)	5	NULL allowed	
	DaysToManufacture Number of days required to manufacture the product.	int	4	NOT NULL	
	ProductLine R = Road, M = Mountain, T = Touring, S = Standard	nchar(2)	4	NULL allowed	
⊞	Class H = High, M = Medium, L = Low	nchar(2)	4	NULL allowed	
	Style W = Womens, M = Mens, U = Universal	nchar(2)	4	NULL allowed	
F	ProductSubcategoryID Product is a member of this product subcategory. Foreign key to ProductSubCategory.ProductSub-CategoryID.	int	4	NULL allowed	
FK	ProductModelID Product is a member of this product model. Foreign key to Product- Model.ProductModelID.	int	4	NULL allowed	
	SellStartDate Date the product was available for sale.	datetime	8	NOT NULL	
	SellEndDate Date the product was no longer available for sale.	datetime	8	NULL allowed	
	DiscontinuedDate Date the product was discontinued.	datetime	8	NULL allowed	
π.	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL	(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Key	Name	Key Columns	Unique
PK C	PK_Product_ProductID Primary key (clustered) constraint	ProductID	True
	AK_Product_Name Unique nonclustered index.	Name	True
	AK_Product_ProductNumber Unique nonclustered index.	ProductNumber	True
	AK_Product_rowguid Unique nonclustered index. Used to support replication samples.	rowguid	True

Check Constraints

Name	On Column	Constraint
CK_Product_Class Check constraint [Class]='h' OR [Class]='m' OR [Class]='l' OR [Class]='H' OR [Class]='H' OR [Class]='H' 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_DaysToManufacture Check constraint [DaysToManufacture] >= (0)	DaysTo- Manufacture	([DaysTo- Manufacture]>=(0))
CK_Product_ListPrice Check constraint [ListPrice] >= (0.00)	ListPrice	([List- Price]>=(0.00))
CK_Product_ProductLine Check constraint [ProductLine]='r' OR [ProductLine]='m' OR [ProductLine]='t' OR [ProductLine]='s' OR [ProductLine]='M' OR [ProductLine]='M' OR [ProductLine]='T' OR [ProductLine]='S' OR [ProductLine] IS NULL	ProductLine	(upper([Product- Line])='R' OR upper([Product- Line])='M' OR upper([Product- Line])='T' OR upper([Product- Line])='S' OR [ProductLine] IS NULL)
CK_Product_ReorderPoint Check constraint [ReorderPoint] > (0)	ReorderPoint	([Reorder- Point]>(0))
CK_Product_SafetyStockLevel Check constraint [SafetyStockLevel] > (0)	SafetyStock- Level	([SafetyStock- Level]>(0))
CK_Product_SellEndDate Check constraint [SellEndDate] >= [SellStartDate] OR [SellEndDate] IS NULL		([SellEnd- Date]>=[SellStart- Date] OR [Sell- EndDate] IS NULL)
CK_Product_StandardCost Check constraint [SafetyStockLevel] > (0)	StandardCost	([Standard- Cost]>=(0.00))
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)
CK_Product_Weight Check constraint [Weight] > (0.00)	Weight	([Weight]>(0.00))

Foreign Keys

Name	Columns
FK_Product_ProductModel_ProductModelID Foreign key constraint referencing ProductModel.Product- ModelID.	ProductModelID->[Production].[ProductModel].[Product-ModelID]
FK_Product_ProductSubcategory_ProductSubcategoryID Foreign key constraint referencing Product-Subcategory.ProductSubcategoryID.	ProductSubcategoryID->[Production].[Product-Subcategory].[ProductSubcategoryID]
FK_Product_UnitMeasure_SizeUnitMeasureCode Foreign key constraint referencing UnitMeasure.Unit-	SizeUnitMeasureCode->[Production].[UnitMeasure].[Unit-

MeasureCode.	MeasureCode]
FK_Product_UnitMeasure_WeightUnitMeasureCode Foreign key constraint referencing UnitMeasure.Unit-MeasureCode.	WeightUnitMeasureCode->[Production].[Unit-Measure].[UnitMeasureCode]

```
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,
[rowquid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF Product rowquid]
DEFAULT (newid()),
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF Product ModifiedDate] DEFAULT
(getdate())
) ON [PRIMARY]
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))
ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK Product DaysToManufacture] CHECK
(([DaysToManufacture]>=(0)))
ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK Product ListPrice] CHECK (([List-
Price] >= (0.00)))
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))
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)))
ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK Product SellEndDate] CHECK
(([SellEndDate]>=[SellStartDate] OR [SellEndDate] IS NULL))
ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK Product StandardCost] CHECK
(([StandardCost]>=(0.00)))
GΩ
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))
ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK Product Weight] CHECK
(([Weight]>(0.00)))
ALTER TABLE [Production].[Product] ADD CONSTRAINT [PK Product ProductID] PRIMARY KEY
CLUSTERED ([ProductID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK Product Name] ON [Production].[Product] ([Name])
ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK Product ProductNumber] ON [Production].[Product]
([ProductNumber]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK Product rowquid] ON [Production].[Product]
([rowguid]) ON [PRIMARY]
ALTER TABLE [Production].[Product] ADD CONSTRAINT [FK Product ProductModel ProductModel-
ID] FOREIGN KEY ([ProductModelID]) REFERENCES [Production].[ProductModel] ([Product-
ModelID1)
ALTER TABLE [Production].[Product] ADD CONSTRAINT [FK Product ProductSubcategory -
ProductSubcategoryID] FOREIGN KEY ([ProductSubcategoryID]) REFERENCES [Production].[ProductSubcategory] ([ProductSubcategoryID])
ALTER TABLE [Production].[Product] ADD CONSTRAINT [FK Product UnitMeasure SizeUnit-
MeasureCode] FOREIGN KEY ([SizeUnitMeasureCode]) REFERENCES [Production].[UnitMeasure]
([UnitMeasureCode])
ALTER TABLE [Production].[Product] ADD CONSTRAINT [FK Product UnitMeasure WeightUnit-
MeasureCode] FOREIGN KEY ([WeightUnitMeasureCode]) REFERENCES [Production].[Unit-
Measure] ([UnitMeasureCode])
EXEC sp_addextendedproperty N'MS_Description', N'Products sold or used in the
manfacturing of sold products.', 'SCHEMA', N'Production', 'TABLE', N'Product', NULL,
EXEC sp_addextendedproperty N'MS_Description', N'H = High, M = Medium, L = Low',
'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Class'
EXEC sp addextendedproperty N'MS Description', N'Product color.', 'SCHEMA',
N'Production', 'TABLE', N'Product', 'COLUMN', N'Color'
EXEC sp addextendedproperty N'MS Description', N'Number of days required to manufacture
the product.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'DaysTo-
Manufacture'
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'
EXEC sp addextendedproperty N'MS Description', N'Selling price.', 'SCHEMA',
N'Production', 'TABLE', N'Product', 'COLUMN', N'ListPrice'
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'
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'
EXEC sp addextendedproperty N'MS Description', N'Primary key for Product records.',
'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'ProductID'
EXEC sp_addextendedproperty N'MS_Description', N'R = Road, M = Mountain, T = Touring, S
= Standard', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'ProductLine'
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'
EXEC sp addextendedproperty N'MS Description', N'Unique product identification
number.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'ProductNumber'
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'
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'
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'
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'SellEnd-
Date'
GO
EXEC sp addextendedproperty N'MS Description', N'Date the product was available for
sale.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'SellStartDate'
EXEC sp addextendedproperty N'MS Description', N'Product size.', 'SCHEMA',
N'Production', 'TABLE', N'Product', 'COLUMN', N'Size'
EXEC sp_addextendedproperty N'MS_Description', N'Unit of measure for Size column.',
'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'SizeUnitMeasureCode'
```

```
EXEC sp_addextendedproperty N'MS_Description', N'Standard cost of the product.',
'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'StandardCost
EXEC sp_addextendedproperty N'MS_Description', N'W = Womens, M = Mens, U = Universal',
'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Style'
EXEC sp addextendedproperty N'MS Description', N'Product weight.', 'SCHEMA',
N'Production', 'TABLE', N'Product', 'COLUMN', N'Weight'
EXEC sp_addextendedproperty N'MS_Description', N'Unit of measure for Weight column.',
'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'WeightUnitMeasureCode'
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 [DaysToManufacture]
>= (0)', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK Product Days-
ToManufacture'
GO
EXEC sp addextendedproperty N'MS Description', N'Check constraint [ListPrice] >=
(0.00), 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK_Product_List-
Price!
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [ProductLine]=''r''
OR [ProductLine]=''m'' OR [ProductLine]=''t'' OR [ProductLine]=''s'' OR [Product-
Line]=''R'' OR [ProductLine]=''M'' OR [ProductLine]=''T'' OR [ProductLine]=''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 Reorder-
Point'
GO
EXEC sp addextendedproperty N'MS Description', N'Check constraint [SafetyStockLevel] >
(0)', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK Product Safety-
StockLevel'
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 Standard-
Cost'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Style]=''u'' OR [Style]=''m'' OR [Style]=''W'' OR [Style]=''W'' OR
[Style] IS NULL', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK -
Product Style'
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Weight] > (0.00)',
'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK Product Weight'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 1', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'DF_Product_FinishedGoods-
Flag'
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'
CO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of NEWID()',
'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'DF_Product_rowguid'
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'
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'
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'
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'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'PK_Product_ProductID'
EXEC sp addextendedproperty N'MS Description', N'Unique nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'Product', 'INDEX', N'AK Product Name'
EXEC sp addextendedproperty N'MS Description', N'Unique nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'Product', 'INDEX', N'AK Product ProductNumber'
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'
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 -
Product ID!
```

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
₽ <mark>%</mark> G	ProductCategoryID Primary key for ProductCategory records.	int	4	NOT NULL	1 - 1	
#	Name Category description.	[dbo].[Name]	100	NOT NULL		
.	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL		(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

Key	Name	Key Columns	Unique
PK <mark>P</mark> G	PK_ProductCategory_ProductCategoryID Primary key (clustered) constraint	Product- CategoryID	True
	AK_ProductCategory_Name Unique nonclustered index.	Name	True
	AK_ProductCategory_rowguid Unique nonclustered index. Used to support replication samples.	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]
ALTER TABLE [Production].[ProductCategory] ADD CONSTRAINT [PK ProductCategory Product-
CategoryID] PRIMARY KEY CLUSTERED ([ProductCategoryID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK ProductCategory Name] ON [Production].[Product-
Category] ([Name]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK ProductCategory rowquid] ON [Production].[Product-
Category] ([rowquid]) ON [PRIMARY]
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'Modified-
Date'
GO
EXEC sp addextendedproperty N'MS Description', N'Category description.', 'SCHEMA',
N'Production', 'TABLE', N'ProductCategory', 'COLUMN', N'Name'
EXEC sp addextendedproperty N'MS Description', N'Primary key for ProductCategory
records.', 'SCHEMA', N'Production', 'TABLE', N'ProductCategory', 'COLUMN', N'Product-
CategoryID'
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'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductCategory', 'CONSTRAINT', N'DF_-
ProductCategory ModifiedDate'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
NEWID() )', 'SCHEMA', N'Production', 'TABLE', N'ProductCategory', 'CONSTRAINT', N'DF_-
ProductCategory rowguid'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Production', 'TABLE', N'ProductCategory', 'CONSTRAINT', N'PK Product-
Category ProductCategoryID'
EXEC sp addextendedproperty N'MS Description', N'Unique nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'ProductCategory', 'INDEX', N'AK ProductCategory Name'
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'
```

Uses
[dbo].[Name]
Production
Used By

Project > localhost > User databases > AdventureWorks2016 > Tables > Production.ProductCategory

[Production].[ProductSubcategory]

[Production].[ProductCostHistory]

MS_Description

Changes in the cost of a product over time.

Properties

Property	Value
Row Count (~)	395
Created	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKO C FKO	ProductID Product identification number. Foreign key to Product.ProductID	int	4	NOT NULL	
PK P C	StartDate Product cost start date.	datetime	8	NOT NULL	
	EndDate Product cost end date.	datetime	8	NULL allowed	
	StandardCost Standard cost of the product.	money	8	NOT NULL	
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Key	Name	Key Columns	Unique
PK	PK_ProductCostHistory_ProductID_StartDate Primary key (clustered) constraint	ProductID, StartDate	True

Check Constraints

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

Foreign Keys

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

```
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]
ALTER TABLE [Production].[ProductCostHistory] ADD CONSTRAINT [CK ProductCostHistory End-
Date] CHECK (([EndDate]>=[StartDate] OR [EndDate] IS NULL))
ALTER TABLE [Production].[ProductCostHistory] ADD CONSTRAINT [CK ProductCostHistory -
StandardCost] CHECK (([StandardCost]>=(0.00)))
ALTER TABLE [Production].[ProductCostHistory] ADD CONSTRAINT [PK ProductCostHistory -
ProductID StartDate] PRIMARY KEY CLUSTERED ([ProductID], [StartDate]) ON [PRIMARY]
ALTER TABLE [Production].[ProductCostHistory] ADD CONSTRAINT [FK ProductCostHistory -
Product ProductID] FOREIGN KEY ([ProductID]) REFERENCES [Production].[Product]
([ProductID])
EXEC sp addextendedproperty N'MS Description', N'Changes in the cost of a product over
time.', 'SCHEMA', N'Production', 'TABLE', N'ProductCostHistory', NULL, NULL
EXEC sp addextendedproperty N'MS Description', N'Product cost end date.', 'SCHEMA',
N'Production', 'TABLE', N'ProductCostHistory', 'COLUMN', N'EndDate'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductCostHistory', 'COLUMN',
N'ModifiedDate'
{\tt EXEC} \  \, {\tt sp\_addextended property} \  \, {\tt N'MS\_Description'}, \  \, {\tt N'Product} \  \, {\tt identification} \  \, {\tt number}. \  \, {\tt Foreign}
key to Product.ProductID', 'SCHEMA', N'Production', 'TABLE', N'ProductCostHistory',
'COLUMN', N'ProductID'
EXEC sp_addextendedproperty N'MS_Description', N'Standard cost of the product.',
'SCHEMA', N'Production', 'TABLE', N'ProductCostHistory', 'COLUMN', N'StandardCost'
EXEC sp addextendedproperty N'MS Description', N'Product cost start date.', 'SCHEMA',
N'Production', 'TABLE', N'ProductCostHistory', 'COLUMN', N'StartDate'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [EndDate] >= [Start-
Date] OR [EndDate] IS NULL', 'SCHEMA', N'Production', 'TABLE', N'ProductCostHistory',
'CONSTRAINT', N'CK ProductCostHistory EndDate'
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_ProductCost-
History_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
```

[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	17:24:40 2017. október 19., csütörtök	
Last Modified	17:24:52 2017. október 19., csütörtök	

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
₽ <mark>%</mark> G	ProductDescriptionID Primary key for ProductDescription records.	int	4	NOT NULL	1 - 1	
	Description Description of the product.	nvarchar(400)	800	NOT NULL		
#	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL		(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

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

```
CREATE TABLE [Production].[ProductDescription]

(
[ProductDescriptionID] [int] NOT NULL IDENTITY(1, 1),

[Description] [nvarchar] (400) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
```

```
[rowquid] [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]
CREATE UNIQUE NONCLUSTERED INDEX [AK ProductDescription rowguid] ON
[Production].[ProductDescription] ([rowguid]) ON [PRIMARY]
EXEC sp_addextendedproperty N'MS_Description', N'Product descriptions in several
languages.', 'SCHEMA', N'Production', 'TABLE', N'ProductDescription', NULL, NULL
EXEC sp addextendedproperty N'MS Description', N'Description of the product.',
'SCHEMA', N'Production', 'TABLE', N'ProductDescription', 'COLUMN', N'Description'
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductDescription', 'COLUMN',
N'ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Primary key for ProductDescription
records.', 'SCHEMA', N'Production', 'TABLE', N'ProductDescription', 'COLUMN', N'Product-
DescriptionID'
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'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductDescription', 'CONSTRAINT',
N'DF ProductDescription ModifiedDate'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of NEWID()',
'SCHEMA', N'Production', 'TABLE', N'ProductDescription', 'CONSTRAINT', N'DF Product-
Description rowguid'
EXEC sp addextendedproperty N'MS Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Production', 'TABLE', N'ProductDescription', 'CONSTRAINT', N'PK Product-
Description ProductDescriptionID
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Production', 'TABLE', N'ProductDescription',
'INDEX', N'AK ProductDescription rowguid'
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a primary
key constraint.', 'SCHEMA', N'Production', 'TABLE', N'ProductDescription',
N'PK ProductDescription ProductDescriptionID'
```

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	17:24:40 2017. október 19., csütörtök	
Last Modified	17:24:51 2017. október 19., csütörtök	

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKC FKG	ProductID Product identification number. Foreign key to Product.ProductID.	int	4	NOT NULL	
PK C FK	DocumentNode Document identification number. Foreign key to Document.DocumentNode.	hierarchyid	892	NOT NULL	
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Key	Name	Key Columns	Unique
PKP C	PK_ProductDocument_ProductID_DocumentNode Primary key (clustered) constraint	ProductID, Document- Node	True

Foreign Keys

Name	Columns
FK_ProductDocument_DocumentNode Foreign key constraint referencing Document.Document- Node.	DocumentNode->[Production].[Document].[Document-Node]
FK_ProductDocument_Product_ProductID Foreign key constraint referencing Product.ProductID.	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]
ALTER TABLE [Production].[ProductDocument] ADD CONSTRAINT [PK ProductDocument Product-
ID DocumentNode] PRIMARY KEY CLUSTERED ([ProductID], [DocumentNode]) ON [PRIMARY]
ALTER TABLE [Production].[ProductDocument] ADD CONSTRAINT [FK ProductDocument Document -
DocumentNode] FOREIGN KEY ([DocumentNode]) REFERENCES [Production].[Document]
([DocumentNodel)
GO
ALTER TABLE [Production].[ProductDocument] ADD CONSTRAINT [FK ProductDocument Product -
ProductID] FOREIGN KEY ([ProductID]) REFERENCES [Production].[Product] ([ProductID])
EXEC sp addextendedproperty N'MS Description', N'Cross-reference table mapping products
to related product documents.', 'SCHEMA', N'Production', 'TABLE', N'ProductDocument',
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'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductDocument', 'COLUMN', N'Modified-
GO
EXEC sp addextendedproperty N'MS Description', N'Product identification number. Foreign
key to Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'ProductDocument',
'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_Product-
Document 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
```

[Production].[Document]

 $Project > localhost > User\ databases > Adventure Works 2016 > Tables > Production. Product Document$

[Production].[Product]
Production

[Production].[ProductInventory]

MS_Description

Product inventory information.

Properties

Property	Value	
Collation	SQL_Latin1_General_CP1_CI_AS	
Row Count (~)	1069	
Created	17:24:40 2017. október 19., csütörtök	
Last Modified	17:24:51 2017. október 19., csütörtök	

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKO FKO	ProductID Product identification number. Foreign key to Product.ProductID.	int	4	NOT NULL	
PKP C FKP	LocationID Inventory location identification number. Foreign key to Location.LocationID.	smallint	2	NOT NULL	
■	Shelf Storage compartment within an inventory location.	nvarchar(10)	20	NOT NULL	
	Bin Storage container on a shelf in an inventory location.	tinyint	1	NOT NULL	
	Quantity Quantity of products in the inventory location.	smallint	2	NOT NULL	((0))
	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL	(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Key	Name	Key Columns	Unique
PK	PK_ProductInventory_ProductID_LocationID Primary key (clustered) constraint	ProductID, LocationID	True

Check Constraints

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

Foreign Keys

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

```
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)),
[rowquid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF Product-
Inventory rowguid] DEFAULT (newid()),
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF ProductInventory ModifiedDate]
DEFAULT (getdate())
ON [PRIMARY]
ALTER TABLE [Production]. [ProductInventory] ADD CONSTRAINT [CK ProductInventory Bin]
CHECK (([Bin]>=(0) AND [Bin]<=(100)))
ALTER TABLE [Production].[ProductInventory] ADD CONSTRAINT [CK ProductInventory Shelf]
CHECK (([Shelf] like '[A-Za-z]' OR [Shelf]='N/A'))
ALTER TABLE [Production].[ProductInventory] ADD CONSTRAINT [PK ProductInventory Product-
ID LocationID] PRIMARY KEY CLUSTERED ([ProductID], [LocationID]) ON [PRIMARY]
ALTER TABLE [Production].[ProductInventory] ADD CONSTRAINT [FK ProductInventory -
Location LocationID] FOREIGN KEY ([LocationID]) REFERENCES [Production].[Location]
([LocationID])
ALTER TABLE [Production].[ProductInventory] ADD CONSTRAINT [FK_ProductInventory_-
Product ProductID] FOREIGN KEY ([ProductID]) REFERENCES [Production].[Product]
([ProductID])
EXEC sp_addextendedproperty N'MS_Description', N'Product inventory information.',
'SCHEMA', N'Production', 'TABLE', N'ProductInventory', NULL, NULL
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'Modified-
GO
EXEC sp addextendedproperty N'MS Description', N'Product identification number. Foreign
key to Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'ProductInventory',
'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'rowquid'
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 Product-
Inventory rowquid'
GO
{\tt EXEC} \  \, {\tt sp\_addextended} property \  \, {\tt N'MS\_Description'}, \  \, {\tt N'Foreign} \  \, {\tt key} \  \, {\tt constraint} \  \, {\tt referencing}
Location.LocationID.', 'SCHEMA', N'Production', 'TABLE', N'ProductInventory',
'CONSTRAINT', N'FK ProductInventory Location LocationID'
GO
'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 Product-
Inventory_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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKO C FKO	ProductID Product identification number. Foreign key to Product.ProductID	int	4	NOT NULL	
PK P C	StartDate List price start date.	datetime	8	NOT NULL	
	EndDate List price end date	datetime	8	NULL allowed	
	ListPrice Product list price.	money	8	NOT NULL	
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Key	Name	Key Columns	Unique
PK	PK_ProductListPriceHistory_ProductID_StartDate Primary key (clustered) constraint	ProductID, StartDate	True

Check Constraints

Name	On Column	Constraint
CK_ProductListPriceHistory_EndDate Check constraint [EndDate] >= [StartDate] OR [EndDate] IS NULL		([EndDate]>=[StartDate] OR [EndDate] IS NULL)
CK_ProductListPriceHistory_ListPrice Check constraint [ListPrice] > (0.00)	ListPrice	([ListPrice]>(0.00))

Foreign Keys

Name	Columns
FK_ProductListPriceHistory_Product_ProductID Foreign key constraint referencing Product.ProductID.	ProductID->[Production].[Product].[ProductID]

```
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]
ALTER TABLE [Production].[ProductListPriceHistory] ADD CONSTRAINT [CK ProductListPrice-
History EndDate] CHECK (([EndDate]>=[StartDate] OR [EndDate] IS NULL))
ALTER TABLE [Production].[ProductListPriceHistory] ADD CONSTRAINT [CK ProductListPrice-
History ListPrice] CHECK (([ListPrice]>(0.00)))
ALTER TABLE [Production].[ProductListPriceHistory] ADD CONSTRAINT [PK ProductListPrice-
History ProductID StartDate] PRIMARY KEY CLUSTERED ([ProductID], [StartDate]) ON
[PRIMARY]
ALTER TABLE [Production].[ProductListPriceHistory] ADD CONSTRAINT [FK ProductListPrice-
History 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
EXEC sp addextendedproperty N'MS Description', N'List price end date', 'SCHEMA',
N'Production', 'TABLE', N'ProductListPriceHistory', 'COLUMN', N'EndDate'
EXEC sp addextendedproperty N'MS Description', N'Product list price.', 'SCHEMA',
N'Production', 'TABLE', N'ProductListPriceHistory', 'COLUMN', N'ListPrice'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductListPriceHistory', 'COLUMN',
N'ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Product identification number. Foreign
key to Product.ProductID', 'SCHEMA', N'Production', 'TABLE', N'ProductListPrice-
History', 'COLUMN', N'ProductID'
EXEC sp addextendedproperty N'MS Description', N'List price start date.', 'SCHEMA',
N'Production', 'TABLE', N'ProductListPriceHistory', 'COLUMN', N'StartDate'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [EndDate] >= [Start-
Date] OR [EndDate] IS NULL', 'SCHEMA', N'Production', 'TABLE', N'ProductListPrice-
History', 'CONSTRAINT', N'CK_ProductListPriceHistory_EndDate'
```

```
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [ListPrice] >
(0.00)', 'SCHEMA', N'Production', 'TABLE', N'ProductListPriceHistory', 'CONSTRAINT',
N'CK ProductListPriceHistory ListPrice'
{\tt EXEC} \  \, {\tt sp\_addextended} \\ {\tt property} \  \, {\tt N'MS\_Description'}, \  \, {\tt N'Default} \  \, {\tt constraint} \  \, {\tt value} \  \, {\tt of} \\
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductListPriceHistory', 'CONSTRAINT',
N'DF ProductListPriceHistory ModifiedDate'
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'
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'ProductListPriceHistory',
'INDEX', N'PK ProductListPriceHistory ProductID StartDate'
GO
```

[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	17:24:40 2017. október 19., csütörtök		
Last Modified	17:24:52 2017. október 19., csütörtök		

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullabil ity	Identity	Default
PK <mark>P</mark> G	ProductModelID Primary key for ProductModel records.	int	4	NOT NULL	1 - 1	
ı.	Name Product model description.	[dbo].[Name]	100	NOT NULL		
ı.	CatalogDescription Detailed product catalog information in xml format.	xml([Production].[Product- DescriptionSchema- Collection])	max	NULL allowed		
. 	Instructions Manufacturing instructions in xml format.	xml([Production].[Manu- InstructionsSchema- Collection])	max	NULL allowed		
ф.	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL		(newid()
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate ())

Indexes

Key	Name	Key Columns	Туре	Unique	XML Type
PK#C	PK_ProductModel_ProductModelID Primary key (clustered) constraint	ProductModel- ID		True	
	AK_ProductModel_Name Unique nonclustered index.	Name		True	

AK_ProductModel_rowguid Unique nonclustered index. Used to support replication samples.	rowguid		True	
PXML_ProductModel_CatalogDescription Primary XML index.	Catalog- Description	xml		Primary
PXML_ProductModel_Instructions Primary XML index.	Instructions	xml		Primary

```
CREATE TABLE [Production].[ProductModel]
[ProductModelID] [int] NOT NULL IDENTITY(1, 1),
[Name] [dbo].[Name] NOT NULL,
[CatalogDescription] [xml] (CONTENT [Production].[ProductDescriptionSchemaCollection])
NULL,
[Instructions] [xml] (CONTENT [Production]. [ManuInstructionsSchemaCollection]) NULL,
[rowquid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF ProductModel rowquid]
DEFAULT (newid()),
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF ProductModel ModifiedDate] DEFAULT
(getdate())
) ON [PRIMARY]
ALTER TABLE [Production].[ProductModel] ADD CONSTRAINT [PK ProductModel ProductModelID]
PRIMARY KEY CLUSTERED ([ProductModelID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK ProductModel Name] ON [Production].[ProductModel]
([Name]) ON [PRIMARY]
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])
CREATE PRIMARY XML INDEX [PXML ProductModel Instructions]
ON [Production].[ProductModel] ([Instructions])
EXEC sp addextendedproperty N'MS Description', N'Product model classification.',
'SCHEMA', 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'Catalog-
Description'
GO
EXEC sp addextendedproperty N'MS Description', N'Manufacturing instructions in xml
format. , 'SCHEMA', N'Production, 'TABLE', N'ProductModel', 'COLUMN', N'Instructions'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductModel', 'COLUMN', N'ModifiedDate'
GO
EXEC sp addextendedproperty N'MS Description', N'Product model description.', 'SCHEMA',
N'Production', 'TABLE', N'ProductModel', 'COLUMN', N'Name'
EXEC sp addextendedproperty N'MS Description', N'Primary key for ProductModel
```

```
records.', 'SCHEMA', N'Production', 'TABLE', N'ProductModel', 'COLUMN', N'ProductModel-
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'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductModel', 'CONSTRAINT', N'DF -
ProductModel ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of NEWID()',
'SCHEMA', N'Production', 'TABLE', N'ProductModel', 'CONSTRAINT', N'DF Product-
Model rowguid'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Production', 'TABLE', N'ProductModel', 'CONSTRAINT', N'PK ProductModel -
ProductModelID'
EXEC sp addextendedproperty N'MS Description', N'Unique nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'ProductModel', 'INDEX', N'AK ProductModel Name'
GO
{\tt EXEC} \  \, {\tt sp\_addextended} property \  \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index. \ Used \ to \  \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt Used \ to} \, \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt Used \ to} \, \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt Used \ to} \, \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt Used \ to} \, \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt Used \ to} \, \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt Used \ to} \, \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'Unique \ nonclustered \ index.} \, \, {\tt N'MS\_Description', N'MS\_Description', N'MS\_Description', N'MS\_Description', N'MS\_Description', N'MS\_Description', N'MS\_Des
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
```

[dbo].[Name]

Production

[Production].[ManuInstructionsSchemaCollection]

[Production].[ProductDescriptionSchemaCollection]

Used By

[Production].[Product]

[Production].[ProductModelIllustration]

[Product Model Product Description Culture]

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKC FKB	ProductModelID Primary key. Foreign key to Product- Model.ProductModelID.	int	4	NOT NULL	
PKC FKD	IllustrationID Primary key. Foreign key to Illustration.IllustrationID.	int	4	NOT NULL	
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Key	Name	Key Columns	Unique
PK#C	PK_ProductModelIllustration_ProductModelID_IllustrationID Primary key (clustered) constraint	ProductModelID, IllustrationID	True

Foreign Keys

Name	Columns	
FK_ProductModelIllustration_Illustration_IllustrationID Foreign key constraint referencing Illustration.Illustration-ID.	IllustrationID->[Production].[Illustration].[IllustrationID]	
FK_ProductModelIllustration_ProductModel_Product-ModelID Foreign key constraint referencing ProductModel.Product-ModelID.	ProductModelID->[Production].[ProductModel].[Product-ModelID]	

```
CREATE TABLE [Production].[ProductModelIllustration]
[ProductModelID] [int] NOT NULL,
[IllustrationID] [int] NOT NULL,
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF ProductModelIllustration Modified-
Date] DEFAULT (getdate())
) ON [PRIMARY]
ALTER TABLE [Production].[ProductModelIllustration] ADD CONSTRAINT [PK ProductModel-
Illustration ProductModelID IllustrationID] PRIMARY KEY CLUSTERED ([ProductModelID],
[IllustrationID]) ON [PRIMARY]
ALTER TABLE [Production].[ProductModelIllustration] ADD CONSTRAINT [FK ProductModel-
Illustration Illustration IllustrationID] FOREIGN KEY ([IllustrationID]) REFERENCES
[Production].[Illustration] ([IllustrationID])
ALTER TABLE [Production].[ProductModelIllustration] ADD CONSTRAINT [FK ProductModel-
Illustration ProductModel ProductModelID] FOREIGN KEY ([ProductModelID]) REFERENCES
[Production].[ProductModelID])
EXEC sp addextendedproperty N'MS Description', N'Cross-reference table mapping product
models and illustrations.', 'SCHEMA', N'Production', 'TABLE', N'ProductModel-
Illustration', NULL, NULL
EXEC sp addextendedproperty N'MS Description', N'Primary key. Foreign key to
Illustration.IllustrationID.', 'SCHEMA', N'Production', 'TABLE', N'ProductModel-
Illustration', 'COLUMN', N'IllustrationID'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductModelIllustration', 'COLUMN',
N'ModifiedDate'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Foreign key to Product-
Model.ProductModelID.', 'SCHEMA', N'Production', 'TABLE', N'ProductModelIllustration',
'COLUMN', N'ProductModelID'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductModelIllustration',
'CONSTRAINT', N'DF_ProductModelIllustration_ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Foreign key constraint referencing
Illustration.IllustrationID.', 'SCHEMA', N'Production', 'TABLE', N'ProductModel-
Illustration', 'CONSTRAINT', N'FK ProductModelIllustration Illustration IllustrationID'
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
ProductModel.ProductModelID.', 'SCHEMA', N'Production', 'TABLE', N'ProductModel-Illustration', 'CONSTRAINT', N'FK_ProductModelIllustration_ProductModel_ProductModelID'
EXEC sp addextendedproperty N'MS Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Production', 'TABLE', N'ProductModelIllustration', 'CONSTRAINT', N'PK_-ProductModelIllustration_ProductModelID_IllustrationID'
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'
```

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:52 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKP C FKP	ProductModelID Primary key. Foreign key to Product- Model.ProductModelID.	int	4	NOT NULL	
PK C FK	ProductDescriptionID Primary key. Foreign key to Product- Description.ProductDescriptionID.	int	4	NOT NULL	
PKP C FKP	CultureID Culture identification number. Foreign key to Culture.CultureID.	nchar(6)	12	NOT NULL	
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Key	Name	Key Columns	Unique
PKP C	PK_ProductModelProductDescriptionCulture_ProductModelID_ProductDescriptionID_CultureID Primary key (clustered) constraint	ProductModel- ID, Product- DescriptionID, CultureID	True

Foreign Keys

Name	Columns
FK_ProductModelProductDescriptionCulture_Culture CultureID Foreign key constraint referencing Culture.CultureID.	CultureID->[Production].[Culture].[CultureID]
FK_ProductModelProductDescriptionCulture_Product- Description_ProductDescriptionID	ProductDescriptionID->[Production].[Product-

Foreign key constraint referencing Product- Description.ProductDescriptionID.	Description].[ProductDescriptionID]
FK_ProductModelProductDescriptionCulture_Product-Model_ProductModelID Foreign key constraint referencing ProductModel.Product-ModelID.	ProductModelID->[Production].[ProductModel].[Product-ModelID]

```
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]
ALTER TABLE [Production].[ProductModelProductDescriptionCulture] ADD CONSTRAINT [PK -
ProductModelProductDescriptionCulture ProductModelID ProductDescriptionID CultureID
PRIMARY KEY CLUSTERED ([ProductModelID], [ProductDescriptionID], [CultureID]) ON
GO
ALTER TABLE [Production].[ProductModelProductDescriptionCulture] ADD CONSTRAINT [FK -
ProductModelProductDescriptionCulture Culture CultureID] FOREIGN KEY ([CultureID])
REFERENCES [Production].[Culture] ([CultureID])
ALTER TABLE [Production].[ProductModelProductDescriptionCulture] ADD CONSTRAINT [FK -
ProductModelProductDescriptionCulture ProductDescription ProductDescriptionID] FOREIGN
KEY ([ProductDescriptionID]) REFERENCES [Production].[ProductDescription] ([Product-
DescriptionID])
ALTER TABLE [Production].[ProductModelProductDescriptionCulture] ADD CONSTRAINT [FK -
ProductModelProductDescriptionCulture ProductModel ProductModelID] FOREIGN KEY
([ProductModelID]) REFERENCES [Production].[ProductModel] ([ProductModelID])
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
EXEC sp addextendedproperty N'MS Description', N'Culture identification number. Foreign
key to Culture.CultureID.', 'SCHEMA', N'Production', 'TABLE', N'ProductModelProduct-
DescriptionCulture', 'COLUMN', N'CultureID'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductModelProductDescriptionCulture',
'COLUMN', N'ModifiedDate'
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'
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'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductModelProductDescriptionCulture',
```

```
'CONSTRAINT', N'DF ProductModelProductDescriptionCulture_ModifiedDate'
EXEC sp_addextendedproperty N'MS_Description', N'Foreign key constraint referencing
Culture_CultureID.', 'SCHEMA', N Production', 'TABLE', N'ProductModelProductDescription-
Culture', 'CONSTRAINT', N'FK ProductModelProductDescriptionCulture_Culture_CultureID'
EXEC sp addextendedproperty N'MS Description', N'Foreign key constraint referencing
ProductDescription.ProductDescriptionID.', 'SCHEMA', N'Production', 'TABLE', N'Product-ModelProductDescriptionCulture', 'CONSTRAINT', N'FK_ProductModelProductDescription-
Culture ProductDescription ProductDescriptionID'
GO
EXEC sp addextendedproperty N'MS Description', N'Foreign key constraint referencing
ProductModel.ProductModelID.', 'SCHEMA', N'Production', 'TABLE', N'ProductModelProductDescriptionCulture', 'CONSTRAINT', N'FK_ProductModelProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDescriptionCulture_ProductDes
Model_ProductModelID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Production', 'TABLE', N'ProductModelProductDescriptionCulture',
'CONSTRAINT', N'PK ProductModelProductDescriptionCulture ProductModelID Product-
DescriptionID CultureID'
EXEC sp addextendedproperty N'MS Description', N'Clustered index created by a primary
key constraint.', 'SCHEMA', N'Production', 'TABLE', N'ProductModelProductDescription-Culture', 'INDEX', N'PK_ProductModelProductDescriptionCulture_ProductModelID_Product-
DescriptionID CultureID'
GO
```

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
P⊁C	ProductPhotoID Primary key for ProductPhoto records.	int	4	NOT NULL	1 - 1	
	ThumbNailPhoto Small image of the product.	varbinary(max)	max	NULL allowed		
	ThumbnailPhotoFileName Small image file name.	nvarchar(50)	100	NULL allowed		
	LargePhoto Large image of the product.	varbinary(max)	max	NULL allowed		
	LargePhotoFileName Large image file name.	nvarchar(50)	100	NULL allowed		
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

Key	Name	Key Columns	Unique
PKP G	PK_ProductPhoto_ProductPhotoID Primary key (clustered) constraint	ProductPhotoID	True

```
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]
ALTER TABLE [Production].[ProductPhoto] ADD CONSTRAINT [PK ProductPhoto ProductPhotoID]
PRIMARY KEY CLUSTERED ([ProductPhotoID]) ON [PRIMARY]
EXEC sp addextendedproperty N'MS Description', N'Product images.', 'SCHEMA',
N'Production', 'TABLE', N'ProductPhoto', NULL, NULL
EXEC sp addextendedproperty N'MS Description', N'Large image of the product.',
'SCHEMA', N'Production', 'TABLE', N'ProductPhoto', 'COLUMN', N'LargePhoto'
EXEC sp addextendedproperty N'MS Description', N'Large image file name.', 'SCHEMA',
N'Production', 'TABLE', N'ProductPhoto', 'COLUMN', N'LargePhotoFileName'
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductPhoto', 'COLUMN', N'ModifiedDate'
GO
EXEC sp addextendedproperty N'MS Description', N'Primary key for ProductPhoto
records.', 'SCHEMA', N'Production', 'TABLE', N'ProductPhoto', 'COLUMN', N'ProductPhoto-
TD'
GO
EXEC sp addextendedproperty N'MS Description', N'Small image of the product.',
'SCHEMA', N'Production', 'TABLE', N'ProductPhoto', 'COLUMN', N'ThumbNailPhoto'
EXEC sp addextendedproperty N'MS Description', N'Small image file name.', 'SCHEMA',
N'Production', 'TABLE', N'ProductPhoto', 'COLUMN', N'ThumbnailPhotoFileName'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductPhoto', 'CONSTRAINT', N'DF -
ProductPhoto ModifiedDate'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Production', 'TABLE', N'ProductPhoto', 'CONSTRAINT', N'PK ProductPhoto
ProductPhotoID'
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'
```

Production

Used By

[Production].[ProductProductPhoto]

[ProductProductPhoto]

MS_Description

Cross-reference table mapping products and product photos.

Properties

Property	Value
Неар	True
Row Count (~)	504
Created	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKO FKO	ProductID Product identification number. Foreign key to Product.ProductID.	int	4	NOT NULL	
PK [®]	ProductPhotoID Product photo identification number. Foreign key to ProductPhoto.ProductPhotoID.	int	4	NOT NULL	
	Primary 0 = Photo is not the principal image. 1 = Photo is the principal image.	[dbo].[Flag]	1	NOT NULL	((0))
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Ke	Name	Key Columns	Unique
PK	PK_ProductProductPhoto_ProductID_ProductPhotoID Primary key (clustered) constraint	ProductID, Product- PhotoID	True

Foreign Keys

Name	Columns
FK_ProductProductPhoto_Product_ProductID Foreign key constraint referencing Product.ProductID.	ProductID->[Production].[Product].[ProductID]
FK_ProductProductPhoto_ProductPhotoID Foreign key constraint referencing ProductPhoto.Product-PhotoID.	ProductPhotoID->[Production].[ProductPhoto].[Product-PhotoID]

```
CREATE TABLE [Production].[ProductProductPhoto]
[ProductID] [int] NOT NULL,
[ProductPhotoID] [int] NOT NULL,
[Primary] [dbo].[Flag] NOT NULL CONSTRAINT [DF ProductProductPhoto Primary] DEFAULT
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF ProductProductPhoto ModifiedDate]
DEFAULT (getdate())
) ON [PRIMARY]
ALTER TABLE [Production].[ProductProductPhoto] ADD CONSTRAINT [PK ProductProductPhoto -
ProductID ProductPhotoID] PRIMARY KEY NONCLUSTERED ([ProductID], [ProductPhotoID]) ON
[PRIMARY]
ALTER TABLE [Production].[ProductProductPhoto] ADD CONSTRAINT [FK ProductProductPhoto -
Product ProductID] FOREIGN KEY ([ProductID]) REFERENCES [Production].[Product]
([ProductID])
ALTER TABLE [Production].[ProductProductPhoto] ADD CONSTRAINT [FK ProductProductPhoto -
ProductPhoto ProductPhotoID] FOREIGN KEY ([ProductPhotoID]) REFERENCES
[Production].[ProductPhoto] ([ProductPhotoID])
EXEC sp_addextendedproperty N'MS_Description', N'Cross-reference table mapping products
and product photos.', 'SCHEMA', N'Production', 'TABLE', N'ProductProductPhoto', 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'
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'ProductProduct-
Photo', 'COLUMN', N'Primary'
EXEC sp addextendedproperty N'MS Description', N'Product identification number. Foreign
key to Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'ProductProductPhoto',
'COLUMN', N'ProductID'
N'ProductProductPhoto', 'COLUMN', N'ProductPhotoID'
{\tt EXEC} \  \, {\tt sp\_addextended} property \  \, {\tt N'MS\_Description'}, \  \, {\tt N'Default} \  \, {\tt constraint} \  \, {\tt value} \  \, {\tt of}
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductProductPhoto', 'CONSTRAINT',
N'DF_ProductProductPhoto ModifiedDate'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0
(FALSE), 'SCHEMA', N'Production, 'TABLE', N'ProductProductPhoto', 'CONSTRAINT', N'DF -
ProductProductPhoto Primary'
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'
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 D'
```

```
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Production', 'TABLE', N'ProductProductPhoto', 'CONSTRAINT', N'PK_Product-
ProductPhoto_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
```

[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	AW2016FullTextCatalog
Full Text Key Index	PK_ProductReview_ProductReviewID
Row Count (~)	4
Created	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullabi lity	Full Text Indexe d	Langu age	ldentit y	Defaul t
PK <mark>P</mark> C	ProductReviewID Primary key for Product- Review records.	int	4	NOT NULL			1 - 1	
F 19	ProductID Product identification number. Foreign key to Product.ProductID.	int	4	NOT NULL				
.	ReviewerName Name of the reviewer.	[dbo].[Name]	100	NOT NULL				
	ReviewDate Date review was submitted.	datetime	8	NOT NULL				(getdat e())
	EmailAddress Reviewer's e-mail address.	nvarchar(50)	100	NOT NULL				
⊞	Rating Product rating given by the reviewer. Scale is 1 to 5 with 5 as the highest rating.	int	4	NOT NULL				
т.	Comments Reviewer's comments	nvarchar(3850)	7700	NULL allowe d	True	1033		
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL				(getdat e())

Indexes

Key	Name	Key Columns	Included Columns	Unique
PK C	PK_ProductReview_ProductReviewID Primary key (clustered) constraint	ProductReviewID		True
	IX_ProductReview_ProductID_Name Nonclustered index.	ProductID, ReviewerName	Comments	

Check Constraints

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

Foreign Keys

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

```
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]
ALTER TABLE [Production].[ProductReview] ADD CONSTRAINT [CK ProductReview Rating] CHECK
(([Rating]>=(1) AND [Rating]<=(5)))
ALTER TABLE [Production].[ProductReview] ADD CONSTRAINT [PK ProductReview ProductReview-
ID] PRIMARY KEY CLUSTERED ([ProductReviewID]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX ProductReview ProductID Name] ON [Production].[Product-
Review] ([ProductID], [ReviewerName]) INCLUDE ([Comments]) ON [PRIMARY]
ALTER TABLE [Production].[ProductReview] ADD CONSTRAINT [FK ProductReview Product -
ProductID] FOREIGN KEY ([ProductID]) REFERENCES [Production].[Product] ([ProductID])
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'
EXEC sp addextendedproperty N'MS Description', N'Reviewer''s e-mail address.',
'SCHEMA', N'Production', 'TABLE', N'ProductReview', 'COLUMN', N'EmailAddress'
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'ProductReview',
'COLUMN', N'ProductID'
records.', 'SCHEMA', N'Production', 'TABLE', N'ProductReview', 'COLUMN', N'Product-ReviewID'
EXEC sp addextendedproperty N'MS Description', N'Primary key for ProductReview
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'
EXEC sp addextendedproperty N'MS Description', N'Date review was submitted.', 'SCHEMA',
N'Production', 'TABLE', N'ProductReview', 'COLUMN', N'ReviewDate'
EXEC sp addextendedproperty N'MS Description', N'Name of the reviewer.', 'SCHEMA',
N'Production', 'TABLE', N'ProductReview', 'COLUMN', N'ReviewerName'
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'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductReview', 'CONSTRAINT', N'DF -
ProductReview ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductReview', 'CONSTRAINT', N'DF -
ProductReview ReviewDate'
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'
EXEC sp addextendedproperty N'MS Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Production', 'TABLE', N'ProductReview', 'CONSTRAINT', N'PK ProductReview -
ProductReviewID'
EXEC sp addextendedproperty N'MS Description', N'Nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'ProductReview', 'INDEX', N'IX ProductReview ProductID Name'
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'
CREATE FULLTEXT INDEX ON [Production].[ProductReview] KEY INDEX [PK ProductReview -
ProductReviewID] ON [AW2016FullTextCatalog]
ALTER FULLTEXT INDEX ON [Production].[ProductReview] ADD ([Comments] LANGUAGE 1033)
```

GO		
Uses		
[Production].[Product] [dbo].[Name] Production		
Production		
Used By		

AW2016FullTextCatalog

[Production].[ProductSubcategory]

MS_Description

Product subcategories. See ProductCategory table.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	37
Created	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
PK P G	ProductSubcategoryID Primary key for Product- Subcategory records.	int	4	NOT NULL	1 - 1	
F/P	ProductCategoryID Product category identification number. Foreign key to Product- Category.ProductCategoryID.	int	4	NOT NULL		
ı.	Name Subcategory description.	[dbo].[Name]	100	NOT NULL		
.t.	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL		(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

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

Foreign Keys

Name	Columns
FK_ProductSubcategory_ProductCategoryID Foreign key constraint referencing Product- Category.ProductCategoryID.	ProductCategoryID->[Production].[Product-Category].[ProductCategoryID]

```
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 Product-
Subcategory_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]
CREATE UNIQUE NONCLUSTERED INDEX [AK ProductSubcategory_Name] ON [Production].[Product-
Subcategory] ([Name]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK ProductSubcategory rowquid] ON
[Production].[ProductSubcategory] ([rowguid]) ON [PRIMARY]
ALTER TABLE [Production].[ProductSubcategory] ADD CONSTRAINT [FK ProductSubcategory -
ProductCategory ProductCategoryID] FOREIGN KEY ([ProductCategoryID]) REFERENCES
[Production].[ProductCategory] ([ProductCategoryID])
EXEC sp addextendedproperty N'MS Description', N'Product subcategories. See Product-
Category table.', 'SCHEMA', N'Production', 'TABLE', N'ProductSubcategory', NULL, NULL
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'ProductSubcategory', 'COLUMN',
N'ModifiedDate'
EXEC sp_addextendedproperty N'MS_Description', N'Subcategory description.', 'SCHEMA',
N'Production', 'TABLE', N'ProductSubcategory', 'COLUMN', N'Name'
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'Product-
SubcategoryID'
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'
```

```
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ProductSubcategory', 'CONSTRAINT',
N'DF ProductSubcategory ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of NEWID()',
'SCHEMA', N'Production', 'TABLE', N'ProductSubcategory', 'CONSTRAINT', N'DF Product-
Subcategory rowguid'
GO
EXEC sp addextendedproperty N'MS Description', N'Foreign key constraint referencing
ProductCategory.ProductCategoryID.', 'SCHEMA', N'Production', 'TABLE', N'Product-
Subcategory', 'CONSTRAINT', N'FK ProductSubcategory ProductCategory ProductCategoryID'
EXEC sp addextendedproperty N'MS Description', N'Primary key (clustered) constraint',
'SCHEMA, N'Production', 'TABLE', N'ProductSubcategory', 'CONSTRAINT', N'PK_Product-
Subcategory ProductSubcategoryID
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'ProductSubcategory', 'INDEX', N'AK ProductSubcategory Name'
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'
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'
```

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
PK <mark>P</mark> G	ScrapReasonID Primary key for ScrapReason records.	smallint	2	NOT NULL	1 - 1	
<u>.</u>	Name Failure description.	[dbo].[Name]	100	NOT NULL		
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

Key	Name	Key Columns	Unique
PK G	PK_ScrapReason_ScrapReasonID Primary key (clustered) constraint	ScrapReasonID	True
	AK_ScrapReason_Name Unique nonclustered index.	Name	True

```
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]
CREATE UNIQUE NONCLUSTERED INDEX [AK ScrapReason Name] ON [Production].[ScrapReason]
([Name]) ON [PRIMARY]
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'ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Failure description.', 'SCHEMA',
N'Production', 'TABLE', N'ScrapReason', 'COLUMN', N'Name'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for ScrapReason records.',
'SCHEMA, N'Production', 'TABLE', N'ScrapReason', 'COLUMN', N'ScrapReasonID'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'ScrapReason', 'CONSTRAINT', N'DF_Scrap-
Reason ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Primary key (clustered) constraint',
'SCHEMA, N'Production', 'TABLE', N'ScrapReason', 'CONSTRAINT', N'PK ScrapReason Scrap-
GO
EXEC sp addextendedproperty N'MS Description', N'Unique nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'ScrapReason', 'INDEX', N'AK ScrapReason Name'
EXEC sp addextendedproperty N'MS Description', N'Clustered index created by a primary
key constraint.', 'SCHEMA', N'Production', 'TABLE', N'ScrapReason', 'INDEX', N'PK Scrap-
Reason ScrapReasonID'
```

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
PK <mark>P</mark> C	TransactionID Primary key for Transaction- History records.	int	4	NOT NULL	100000 - 1	
FKP	ProductID Product identification number. Foreign key to Product.ProductID.	int	4	NOT NULL		
ф	ReferenceOrderID Purchase order, sales order, or work order identification number.	int	4	NOT NULL		
Α.	ReferenceOrderLineID Line number associated with the purchase order, sales order, or work order.	int	4	NOT NULL		((0))
	TransactionDate Date and time of the transaction.	datetime	8	NOT NULL		(getdate())
	TransactionType W = WorkOrder, S = SalesOrder, P = PurchaseOrder	nchar(1)	2	NOT NULL		
	Quantity Product quantity.	int	4	NOT NULL		
	ActualCost Product cost.	money	8	NOT NULL		
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

Key	Name	Key Columns	Unique
PK	PK_TransactionHistory_TransactionID Primary key (clustered) constraint	TransactionID	True
	IX_TransactionHistory_ProductID Nonclustered index.	ProductID	
	IX_TransactionHistory_ReferenceOrderID_ReferenceOrderLineID Nonclustered index.	Reference- OrderID, Reference- OrderLineID	

Check Constraints

Name	On Column	Constraint
CK_TransactionHistory_TransactionType Check constraint [TransactionType]='p' OR [TransactionType]='s' OR [TransactionType]='w' OR [TransactionType]='P' OR [TransactionType]='S' OR [TransactionType]='W')	TransactionType	(upper([Transacti onType])='P' OR upper([Transacti onType])='S' OR upper([Transacti onType])='W')

Foreign Keys

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

```
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 ReferenceOrder-
LineID] DEFAULT ((0)),
[TransactionDate] [datetime] NOT NULL CONSTRAINT [DF TransactionHistory Transaction-
Date] 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]
ALTER TABLE [Production].[TransactionHistory] ADD CONSTRAINT [CK_TransactionHistory_-TransactionType] CHECK ((upper([TransactionType])='P' OR upper([TransactionType])='S'
OR upper([TransactionType])='W'))
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]
CREATE NONCLUSTERED INDEX [IX TransactionHistory ReferenceOrderID ReferenceOrderLineID]
ON [Production].[TransactionHistory] ([ReferenceOrderID], [ReferenceOrderLineID]) ON
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
EXEC sp_addextendedproperty N'MS_Description', N'Product cost.', 'SCHEMA',
N'Production', 'TABLE', N'TransactionHistory', 'COLUMN', N'ActualCost'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistory', 'COLUMN',
N'ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Product identification number. Foreign
key to Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistory',
'COLUMN', N'ProductID'
EXEC sp addextendedproperty N'MS Description', N'Product quantity.', 'SCHEMA',
N'Production', 'TABLE', N'TransactionHistory', 'COLUMN', N'Quantity'
EXEC sp addextendedproperty N'MS Description', N'Purchase order, sales order, or work
order identification number.', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistory',
'COLUMN', N'ReferenceOrderID'
GO
{\tt EXEC} \ {\tt sp\_addextended property} \ {\tt 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'TransactionDate'
EXEC sp addextendedproperty N'MS Description', N'Primary key for TransactionHistory
records.', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistory', 'COLUMN',
N'TransactionID'
EXEC sp addextendedproperty N'MS Description', N'W = WorkOrder, S = SalesOrder, P =
PurchaseOrder', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistory', 'COLUMN',
N'TransactionType'
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 TransactionHistory -
TransactionType'
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_Transaction-
```

```
History 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_Transaction-
History_TransactionID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'TransactionHistory', 'INDEX', N'IX_TransactionHistory_Product-
GO
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistory', 'INDEX', N'IX_TransactionHistory_-
ReferenceOrderID ReferenceOrderLineID'
GO
N'PK TransactionHistory TransactionID'
GO
```

[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	17:24:40 2017. október 19., csütörtök	
Last Modified	17:24:48 2017. október 19., csütörtök	

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PK P G	TransactionID Primary key for TransactionHistoryArchive records.	int	4	NOT NULL	
. 	ProductID Product identification number. Foreign key to Product.ProductID.	int	4	NOT NULL	
. 	ReferenceOrderID Purchase order, sales order, or work order identification number.	int	4	NOT NULL	
<u>.</u>	ReferenceOrderLineID Line number associated with the purchase order, sales order, or work order.	int	4	NOT NULL	((0))
	TransactionDate Date and time of the transaction.	datetime	8	NOT NULL	(getdate())
	TransactionType W = Work Order, S = Sales Order, P = Purchase Order	nchar(1)	2	NOT NULL	
	Quantity Product quantity.	int	4	NOT NULL	
	ActualCost Product cost.	money	8	NOT NULL	
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Ke	Name	Key Columns	Unique
PK	PK_TransactionHistoryArchive_TransactionID Primary key (clustered) constraint	TransactionID	True

IX_TransactionHistoryArchive_ProductID Nonclustered index.	ProductID
IX_TransactionHistoryArchive_ReferenceOrderID_ReferenceOrderLineID Nonclustered index.	Reference- OrderID, 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'	TransactionType	(upper([Transacti onType])='P' OR upper([Transacti onType])='S' OR upper([Transacti onType])='W')

```
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'))
ALTER TABLE [Production].[TransactionHistoryArchive] ADD CONSTRAINT [PK Transaction-
HistoryArchive TransactionID] PRIMARY KEY CLUSTERED ([TransactionID]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX TransactionHistoryArchive ProductID] ON
[Production].[TransactionHistoryArchive] ([ProductID]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX TransactionHistoryArchive ReferenceOrderID ReferenceOrder-
LineID] ON [Production].[TransactionHistoryArchive] ([ReferenceOrderID], [Reference-
OrderLineID]) ON [PRIMARY]
EXEC sp addextendedproperty N'MS Description', N'Transactions for previous years.',
'SCHEMA', N'Production', 'TABLE', N'TransactionHistoryArchive', NULL, NULL
EXEC sp addextendedproperty N'MS Description', N'Product cost.', 'SCHEMA',
N'Production', 'TABLE', N'TransactionHistoryArchive', 'COLUMN', N'ActualCost'
```

```
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistoryArchive', 'COLUMN',
N'ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Product identification number. Foreign
key to Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistory-
Archive', 'COLUMN', N'ProductID'
EXEC sp addextendedproperty N'MS Description', N'Product quantity.', 'SCHEMA',
N'Production', 'TABLE', N'TransactionHistoryArchive', 'COLUMN', N'Quantity'
EXEC sp addextendedproperty N'MS Description', N'Purchase order, sales order, or work
order identification number.', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistory-
Archive', 'COLUMN', N'ReferenceOrderID'
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'
EXEC sp addextendedproperty N'MS Description', N'Date and time of the transaction.',
'SCHEMA', N'Production', 'TABLE', N'TransactionHistoryArchive', 'COLUMN', N'Transaction-
Date'
EXEC sp addextendedproperty N'MS Description', N'Primary key for TransactionHistory-
Archive records.', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistoryArchive',
'COLUMN', N'TransactionID'
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'
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'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistoryArchive',
'CONSTRAINT', N'DF TransactionHistoryArchive_ModifiedDate'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0',
'SCHEMA, N'Production', 'TABLE', N'TransactionHistoryArchive', 'CONSTRAINT', N'DF -
TransactionHistoryArchive ReferenceOrderLineID'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'TransactionHistoryArchive',
'CONSTRAINT', N'DF_TransactionHistoryArchive_TransactionDate'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Production', 'TABLE', N'TransactionHistoryArchive', 'CONSTRAINT', N'PK -
TransactionHistoryArchive TransactionID'
EXEC sp addextendedproperty N'MS Description', N'Nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'TransactionHistoryArchive', 'INDEX', N'IX TransactionHistory-
Archive ProductID'
EXEC sp addextendedproperty N'MS Description', N'Nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'TransactionHistoryArchive', 'INDEX', N'IX TransactionHistory-
Archive ReferenceOrderID ReferenceOrderLineID'
```

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

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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PK	UnitMeasureCode Primary key.	nchar(3)	6	NOT NULL	
.	Name Unit of measure description.	[dbo].[Name]	100	NOT NULL	
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

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

```
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_UnitMeasureCode]

PRIMARY KEY CLUSTERED ([UnitMeasureCode]) ON [PRIMARY]
```

```
GO
CREATE UNIQUE NONCLUSTERED INDEX [AK UnitMeasure Name] ON [Production].[UnitMeasure]
([Name]) ON [PRIMARY]
EXEC sp addextendedproperty N'MS Description', N'Unit of measure lookup table.',
'SCHEMA', N'Production', 'TABLE', N'UnitMeasure', NULL, NULL
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Production', 'TABLE', N'UnitMeasure', 'COLUMN', N'ModifiedDate'
GO
EXEC sp addextendedproperty N'MS Description', N'Unit of measure description.',
'SCHEMA', N'Production', 'TABLE', N'UnitMeasure', 'COLUMN', N'Name'
EXEC sp addextendedproperty N'MS Description', N'Primary key.', 'SCHEMA',
N'Production', 'TABLE', N'UnitMeasure', 'COLUMN', N'UnitMeasureCode'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'UnitMeasure', 'CONSTRAINT', N'DF_Unit-
Measure ModifiedDate
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Production', 'TABLE', N'UnitMeasure', 'CONSTRAINT', N'PK UnitMeasure Unit-
MeasureCode'
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'UnitMeasure', 'INDEX', N'AK UnitMeasure Name'
EXEC sp addextendedproperty N'MS Description', N'Clustered index created by a primary
key constraint.', 'SCHEMA', N'Production', 'TABLE', N'UnitMeasure', 'INDEX', N'PK Unit-
Measure UnitMeasureCode'
GO
```

[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	17:24:41 2017. október 19., csütörtök
Last Modified	17:24:52 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Compute d	Max Length (Bytes)	Nullabilit y	Identity	Default
PK <mark>P</mark> C	WorkOrderID Primary key for WorkOrder records.	int		4	NOT NULL	1 - 1	
<u>.</u>	ProductID Product identification number. Foreign key to Product.ProductID.	int		4	NOT NULL		
	OrderQty Product quantity to build.	int		4	NOT NULL		
	StockedQty Quantity built and put in inventory.	int	True	4	NOT NULL		
⊞	ScrappedQty Quantity that failed inspection.	smallint		2	NOT NULL		
	StartDate Work order start date.	datetime		8	NOT NULL		
	EndDate Work order end date.	datetime		8	NULL allowed		
	DueDate Work order due date.	datetime		8	NOT NULL		
FK®	ScrapReasonID Reason for inspection failure.	smallint		2	NULL allowed		
	ModifiedDate Date and time the record was last updated.	datetime		8	NOT NULL		(getdate())

Computed columns

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

Indexes

Key	Name	Key Columns	Unique
PK G	PK_WorkOrder_WorkOrderID Primary key (clustered) constraint	WorkOrderID	True
	IX_WorkOrder_ProductID Nonclustered index.	ProductID	
	IX_WorkOrder_ScrapReasonID Nonclustered index.	ScrapReasonID	

Triggers

Name	ANSI Nulls On	Quoted Identifier On	On
iWorkOrder AFTER INSERT trigger that inserts a row in the Transaction- History table.	True	True	After Insert
uWorkOrder AFTER UPDATE trigger that inserts a row in the Transaction-History table, updates ModifiedDate in the WorkOrder table.	True	True	After Update

Check Constraints

Name	On Column	Constraint
CK_WorkOrder_EndDate Check constraint [EndDate] >= [StartDate] OR [EndDate] IS NULL		([EndDate]>=[Start- Date] OR [EndDate] IS NULL)
CK_WorkOrder_OrderQty Check constraint [OrderQty] > (0)	OrderQty	([OrderQty]>(0))
CK_WorkOrder_ScrappedQty Check constraint [ScrappedQty] >= (0)	ScrappedQty	([ScrappedQty]>=(0))

Foreign Keys

Name	Columns
FK_WorkOrder_Product_ProductID Foreign key constraint referencing Product.ProductID.	ProductID->[Production].[Product].[ProductID]
FK_WorkOrder_ScrapReason_ScrapReasonID Foreign key constraint referencing ScrapReason.Scrap- ReasonID.	ScrapReasonID->[Production].[ScrapReason].[Scrap-ReasonID]

```
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]
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])
           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 (([End-
Date]>=[StartDate] OR [EndDate] IS NULL))
```

```
GO
ALTER TABLE [Production].[WorkOrder] ADD CONSTRAINT [CK WorkOrder OrderQty] CHECK
(([OrderQty]>(0)))
ALTER TABLE [Production].[WorkOrder] ADD CONSTRAINT [CK WorkOrder ScrappedQty] CHECK
(([ScrappedQty]>=(0)))
ALTER TABLE [Production].[WorkOrder] ADD CONSTRAINT [PK WorkOrder WorkOrderID] PRIMARY
KEY CLUSTERED ([WorkOrderID]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX_WorkOrder_ProductID] ON [Production].[WorkOrder]
([ProductID]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX WorkOrder ScrapReasonID] ON [Production].[WorkOrder]
([ScrapReasonID]) ON [PRIMARY]
ALTER TABLE [Production].[WorkOrder] ADD CONSTRAINT [FK WorkOrder Product ProductID]
FOREIGN KEY ([ProductID]) REFERENCES [Production].[Product] ([ProductID])
ALTER TABLE [Production]. [WorkOrder] ADD CONSTRAINT [FK WorkOrder ScrapReason Scrap-
ReasonID] FOREIGN KEY ([ScrapReasonID]) REFERENCES [Production].[ScrapReason] ([Scrap-
EXEC sp addextendedproperty N'MS Description', N'Manufacturing work orders.', 'SCHEMA',
N'Production', 'TABLE', N'WorkOrder', NULL, NULL
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'
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'
EXEC sp addextendedproperty N'MS Description', N'Product identification number. Foreign
key to Product.ProductID.', 'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'COLUMN',
N'ProductID'
EXEC sp addextendedproperty N'MS Description', N'Quantity that failed inspection.',
'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'COLUMN', N'ScrappedQty'
EXEC sp_addextendedproperty N'MS_Description', N'Reason for inspection failure.',
'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'COLUMN', N'ScrapReasonID'
EXEC sp addextendedproperty N'MS Description', N'Work order start date.', 'SCHEMA',
N'Production', 'TABLE', N'WorkOrder', 'COLUMN', N'StartDate'
EXEC sp addextendedproperty N'MS_Description', N'Quantity built and put in inventory.',
'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'COLUMN', N'StockedQty'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for WorkOrder records.',
'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'COLUMN', N'WorkOrderID'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [EndDate] >= [Start-
Date] 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 OrderQty'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [ScrappedQty] >=
(0)', 'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'CONSTRAINT', N'CK WorkOrder -
ScrappedQty'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'CONSTRAINT', N'DF Work-
Order ModifiedDate'
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'
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'
EXEC sp addextendedproperty N'MS Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'CONSTRAINT', N'PK WorkOrder WorkOrder-
ID'
GO
EXEC sp addextendedproperty N'MS Description', N'Nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'WorkOrder', 'INDEX', N'IX WorkOrder ProductID'
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Production', 'TABLE', N'WorkOrder', 'INDEX', N'IX WorkOrder ScrapReasonID'
EXEC sp addextendedproperty N'MS Description', N'Clustered index created by a primary
key constraint.', 'SCHEMA', N'Production', 'TABLE', N'WorkOrder', 'INDEX', N'PK Work-
Order WorkOrderID'
EXEC sp addextendedproperty N'MS Description', N'AFTER INSERT trigger that inserts a
row in the TransactionHistory table.', 'SCHEMA', N'Production', 'TABLE', N'WorkOrder',
'TRIGGER', N'iWorkOrder'
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
```

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

Used By

[Production].[WorkOrderRouting]

[Production].[WorkOrderRouting]

MS_Description

Work order details.

Properties

Property	Value
Row Count (~)	67131
Created	17:24:41 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKO FKO	WorkOrderID Primary key. Foreign key to Work- Order.WorkOrderID.	int	4	NOT NULL	
PKC .ft.	ProductID Primary key. Foreign key to Product.ProductID.	int	4	NOT NULL	
PK <mark>P</mark> C	OperationSequence Primary key. Indicates the manufacturing process sequence.	smallint	2	NOT NULL	
F	LocationID Manufacturing location where the part is processed. Foreign key to Location.LocationID.	smallint	2	NOT NULL	
	ScheduledStartDate Planned manufacturing start date.	datetime	8	NOT NULL	
	ScheduledEndDate Planned manufacturing end date.	datetime	8	NOT NULL	
	ActualStartDate Actual start date.	datetime	8	NULL allowed	
	ActualEndDate Actual end date.	datetime	8	NULL allowed	
III	ActualResourceHrs Number of manufacturing hours used.	decimal(9,4)	5	NULL allowed	
III	PlannedCost Estimated manufacturing cost.	money	8	NOT NULL	
Ħ	ActualCost Actual manufacturing cost.	money	8	NULL allowed	
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Key	Name	Key Columns	Unique
₽ <mark>≯</mark> C	PK_WorkOrderRouting_WorkOrderID_ProductID_OperationSequence Primary key (clustered) constraint	WorkOrderID, ProductID, Operation- Sequence	True
	IX_WorkOrderRouting_ProductID Nonclustered index.	ProductID	

Check Constraints

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

Foreign Keys

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

```
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]
ALTER TABLE [Production]. [WorkOrderRouting] ADD CONSTRAINT [CK WorkOrderRouting Actual-
Cost] CHECK (([ActualCost]>(0.00)))
ALTER TABLE [Production]. [WorkOrderRouting] ADD CONSTRAINT [CK WorkOrderRouting Actual-
EndDate] CHECK (([ActualEndDate]>=[ActualStartDate] OR [ActualEndDate] IS NULL OR
[ActualStartDate] IS NULL))
ALTER TABLE [Production].[WorkOrderRouting] ADD CONSTRAINT [CK WorkOrderRouting Actual-
ResourceHrs] CHECK (([ActualResourceHrs]>=(0.0000)))
ALTER TABLE [Production]. [WorkOrderRouting] ADD CONSTRAINT [CK WorkOrderRouting Planned-
Cost] CHECK (([PlannedCost]>(0.00)))
ALTER TABLE [Production].[WorkOrderRouting] ADD CONSTRAINT [CK WorkOrderRouting -
ScheduledEndDate] CHECK (([ScheduledEndDate]>=[ScheduledStartDate]))
ALTER TABLE [Production]. [WorkOrderRouting] ADD CONSTRAINT [PK WorkOrderRouting Work-
OrderID ProductID OperationSequence] PRIMARY KEY CLUSTERED ([WorkOrderID], [Product-
ID], [OperationSequence]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX WorkOrderRouting ProductID] ON [Production].[WorkOrder-
Routing] ([ProductID]) ON [PRIMARY]
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
EXEC sp addextendedproperty N'MS Description', N'Actual manufacturing cost.', 'SCHEMA',
N'Production', 'TABLE', N'WorkOrderRouting', 'COLUMN', N'ActualCost'
EXEC sp_addextendedproperty N'MS_Description', N'Actual end date.', 'SCHEMA',
N'Production', 'TABLE', N'WorkOrderRouting', 'COLUMN', N'ActualEndDate'
EXEC sp addextendedproperty N'MS Description', N'Number of manufacturing hours used.',
'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting', 'COLUMN', N'ActualResourceHrs'
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'
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
```

```
updated.', 'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting', 'COLUMN', N'Modified-
Date!
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Indicates the
manufacturing process sequence.', 'SCHEMA', N'Production', 'TABLE', N'WorkOrder-
Routing', 'COLUMN', N'OperationSequence'
EXEC sp addextendedproperty N'MS Description', N'Estimated manufacturing cost.',
'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting', 'COLUMN', N'PlannedCost'
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'
EXEC sp addextendedproperty N'MS Description', N'Planned manufacturing start date.',
'SCHEMA, N'Production', 'TABLE', N'WorkOrderRouting', 'COLUMN', N'ScheduledStartDate'
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 Work-
OrderRouting_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_WorkOrderRouting_Actual-
EndDate'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [ActualResourceHrs]
>= (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 Work-
OrderRouting PlannedCost'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [ScheduledEndDate] >=
[ScheduledStartDate]', 'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting',
'CONSTRAINT', N'CK WorkOrderRouting ScheduledEndDate'
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'
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'
EXEC sp addextendedproperty N'MS Description', N'Primary key (clustered) constraint',
'SCHEMAT, N'Production', 'TABLE', N'WorkOrderRouting', 'CONSTRAINT', N'PK_WorkOrder-
Routing WorkOrderID ProductID OperationSequence'
```

```
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'WorkOrderRouting', 'INDEX', N'IX_WorkOrderRouting_ProductID'

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
```

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKPFKP C	ProductID Primary key. Foreign key to Product.ProductID.	int	4	NOT NULL	
P <mark>/2</mark> .ft. F/ ³	BusinessEntityID Primary key. Foreign key to Vendor.BusinessEntityID.	int	4	NOT NULL	
iii	AverageLeadTime The average span of time (in days) between placing an order with the vendor and receiving the purchased product.	int	4	NOT NULL	
	StandardPrice The vendor's usual selling price.	money	8	NOT NULL	
	LastReceiptCost The selling price when last purchased.	money	8	NULL allowed	
	LastReceiptDate Date the product was last received by the vendor.	datetime	8	NULL allowed	
III	MinOrderQty The maximum quantity that should be ordered.	int	4	NOT NULL	
iii	MaxOrderQty The minimum quantity that should be ordered.	int	4	NOT NULL	
	OnOrderQty The quantity currently on order.	int	4	NULL allowed	
.i 🥍	UnitMeasureCode The product's unit of measure.	nchar(3)	6	NOT NULL	
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

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

Check Constraints

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

Foreign Keys

Name	Columns
FK_ProductVendor_Product_ProductID Foreign key constraint referencing Product.ProductID.	ProductID->[Production].[Product].[ProductID]
FK_ProductVendor_UnitMeasure_UnitMeasureCode Foreign key constraint referencing UnitMeasure.Unit- MeasureCode.	UnitMeasureCode->[Production].[UnitMeasure].[Unit-MeasureCode]
FK_ProductVendor_Vendor_BusinessEntityID Foreign key constraint referencing Vendor.BusinessEntity-ID.	BusinessEntityID->[Purchasing].[Vendor].[BusinessEntity-ID]

```
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] [nchar] (3) COLLATE SQL Latin1 General CP1 CI AS NOT NULL,
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF ProductVendor ModifiedDate] DEFAULT
(getdate())
) ON [PRIMARY]
ALTER TABLE [Purchasing].[ProductVendor] ADD CONSTRAINT [CK ProductVendor AverageLead-
Time] CHECK (([AverageLeadTime]>=(1)))
ALTER TABLE [Purchasing].[ProductVendor] ADD CONSTRAINT [CK ProductVendor LastReceipt-
Cost] CHECK (([LastReceiptCost]>(0.00)))
ALTER TABLE [Purchasing].[ProductVendor] ADD CONSTRAINT [CK ProductVendor MaxOrderQty]
CHECK (([MaxOrderOty]>=(1)))
ALTER TABLE [Purchasing].[ProductVendor] ADD CONSTRAINT [CK ProductVendor MinOrderQty]
CHECK (([MinOrderQty]>=(1)))
ALTER TABLE [Purchasing].[ProductVendor] ADD CONSTRAINT [CK ProductVendor OnOrderQty]
CHECK (([OnOrderQty]>=(0)))
ALTER TABLE [Purchasing].[ProductVendor] ADD CONSTRAINT [CK ProductVendor Standard-
Price] CHECK (([StandardPrice]>(0.00)))
ALTER TABLE [Purchasing].[ProductVendor] ADD CONSTRAINT [PK ProductVendor ProductID -
BusinessEntityID] PRIMARY KEY CLUSTERED ([ProductID], [BusinessEntityID]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX ProductVendor BusinessEntityID] ON [Purchasing].[Product-
Vendor] ([BusinessEntityID]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX ProductVendor UnitMeasureCode] ON [Purchasing].[Product-
Vendor] ([UnitMeasureCode]) ON [PRIMARY]
ALTER TABLE [Purchasing].[ProductVendor] ADD CONSTRAINT [FK ProductVendor Product -
ProductID] FOREIGN KEY ([ProductID]) REFERENCES [Production].[Product] ([ProductID])
ALTER TABLE [Purchasing].[ProductVendor] ADD CONSTRAINT [FK ProductVendor UnitMeasure -
UnitMeasureCode] FOREIGN KEY ([UnitMeasureCode]) REFERENCES [Production]. [UnitMeasure]
([UnitMeasureCode])
ALTER TABLE [Purchasing].[ProductVendor] ADD CONSTRAINT [FK ProductVendor Vendor -
BusinessEntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [Purchasing].[Vendor]
([BusinessEntityID])
EXEC sp addextendedproperty N'MS Description', N'Cross-reference table mapping vendors
with the products they supply.', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor',
NULL, NULL
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'
EXEC sp addextendedproperty N'MS Description', N'Primary key. Foreign key to
Vendor.BusinessEntityID.', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor',
'COLUMN', N'BusinessEntityID'
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'Last-
ReceiptDate'
GO
EXEC sp addextendedproperty N'MS Description', N'The minimum quantity that should be
ordered.', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'COLUMN', N'MaxOrderQty'
EXEC sp addextendedproperty N'MS Description', N'The maximum quantity that should be
ordered.', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'COLUMN', N'MinOrderQty'
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'
EXEC sp addextendedproperty N'MS Description', N'Primary key. Foreign key to
Product.ProductID.', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'COLUMN',
N'ProductID'
EXEC sp addextendedproperty N'MS Description', N'The vendor''s usual selling price.',
'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'COLUMN', N'StandardPrice'
EXEC sp_addextendedproperty N'MS_Description', N'The product''s unit of measure.',
'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'COLUMN', N'UnitMeasureCode
EXEC sp addextendedproperty N'MS Description', N'Check constraint [AverageLeadTime] >=
(1)', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'CONSTRAINT', N'CK Product-
Vendor AverageLeadTime'
GO
EXEC sp addextendedproperty N'MS Description', N'Check constraint [LastReceiptCost] >
(0.00), 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'CONSTRAINT', N'CK Product-
Vendor 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 ProductVendor On-
GO
EXEC sp addextendedproperty N'MS Description', N'Check constraint [StandardPrice] >
(0.00)', 'SCHEMA', N'Purchasing', 'TABLE', N'ProductVendor', 'CONSTRAINT', N'CK Product-
Vendor StandardPrice'
GO
{\tt EXEC} \  \, {\tt sp\_addextended} property \  \, {\tt N'MS\_Description'}, \  \, {\tt N'Default} \  \, {\tt constraint} \  \, {\tt value} \  \, {\tt 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'
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'
EXEC sp addextendedproperty N'MS Description', N'Primary key (clustered) constraint',
'SCHEMAT, N'Purchasing', 'TABLE', N'ProductVendor', 'CONSTRAINT', N'PK ProductVendor -
ProductID_BusinessEntityID'
EXEC sp addextendedproperty N'MS Description', N'Nonclustered index.', 'SCHEMA',
N'Purchasing', 'TABLE', N'ProductVendor', 'INDEX', N'IX ProductVendor BusinessEntityID'
GO
EXEC sp addextendedproperty N'MS Description', N'Nonclustered index.', 'SCHEMA',
N'Purchasing', 'TABLE', N'ProductVendor', 'INDEX', N'IX_ProductVendor_UnitMeasureCode'
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
```

[Production].[Product]
[Production].[UnitMeasure]
[Purchasing].[Vendor]
Purchasing

[Purchasing].[PurchaseOrderDetail]

MS_Description

 $Individual\ products\ associated\ with\ a\ specific\ purchase\ order.\ See\ Purchase\ Order Header.$

Properties

Property	Value
Row Count (~)	8845
Created	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:52 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Comput ed	Max Length (Bytes)	Nullabilit y	Identity	Default
PKO FKO	PurchaseOrderID Primary key. Foreign key to PurchaseOrder- Header.PurchaseOrderID.	int		4	NOT NULL		
P <mark>/</mark> C	PurchaseOrderDetailID Primary key. One line number per purchased product.	int		4	NOT NULL	1 - 1	
	DueDate Date the product is expected to be received.	datetime		8	NOT NULL		
	OrderQty Quantity ordered.	smallint		2	NOT NULL		
<u></u>	ProductID Product identification number. Foreign key to Product.ProductID.	int		4	NOT NULL		
■	UnitPrice Vendor's selling price of a single product.	money		8	NOT NULL		
	LineTotal Per product subtotal. Computed as OrderQty * UnitPrice.	money	True	8	NOT NULL		
██	ReceivedQty Quantity actually received from the vendor.	decimal(8,2)		5	NOT NULL		
██	RejectedQty Quantity rejected during inspection.	decimal(8,2)		5	NOT NULL		
	StockedQty Quantity accepted into inventory. Computed as	decimal(9,2)	True	5	NOT NULL		

ReceivedQty - RejectedQty	-				
ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())	

Computed columns

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

Indexes

Key	Name	Key Columns	Unique
P⊁C	PK_PurchaseOrderDetail_PurchaseOrderID_PurchaseOrderDetailID Primary key (clustered) constraint	PurchaseOrder- ID, Purchase- OrderDetailID	True
	IX_PurchaseOrderDetail_ProductID Nonclustered index.	ProductID	

Triggers

Name	ANSI Nulls On	Quoted Identifier On	On
iPurchaseOrderDetail AFTER INSERT trigger that inserts a row in the Transaction- History table and updates the PurchaseOrderHeader.Sub- Total column.	True	True	After Insert
uPurchaseOrderDetail AFTER UPDATE trigger that inserts a row in the Transaction-History table, updates ModifiedDate in PurchaseOrderDetail and updates the PurchaseOrderHeader.SubTotal column.	True	True	After Update

Check Constraints

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

Foreign Keys

Name	Columns
FK_PurchaseOrderDetail_Product_ProductID Foreign key constraint referencing Product.ProductID.	ProductID->[Production].[Product].[ProductID]
FK_PurchaseOrderDetail_PurchaseOrderHeader PurchaseOrderID Foreign key constraint referencing PurchaseOrder- Header.PurchaseOrderID.	PurchaseOrderID->[Purchasing].[PurchaseOrder-Header].[PurchaseOrderID]

```
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]
            , 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])
        {\tt WHERE} \ \ [{\tt PurchaseOrderHeader}] \ . \ [{\tt PurchaseOrderID}] \ \ {\tt IN} \ \ ({\tt 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-
Detail1
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
            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 PurchaseOrderDetail -
OrderQty] CHECK (([OrderQty]>(0)))
ALTER TABLE [Purchasing].[PurchaseOrderDetail] ADD CONSTRAINT [CK PurchaseOrderDetail -
ReceivedQty] CHECK (([ReceivedQty]>=(0.00)))
ALTER TABLE [Purchasing].[PurchaseOrderDetail] ADD CONSTRAINT [CK PurchaseOrderDetail -
RejectedQty] CHECK (([RejectedQty]>=(0.00)))
ALTER TABLE [Purchasing].[PurchaseOrderDetail] ADD CONSTRAINT [CK PurchaseOrderDetail -
UnitPrice] CHECK (([UnitPrice]>=(0.00)))
ALTER TABLE [Purchasing].[PurchaseOrderDetail] ADD CONSTRAINT [PK PurchaseOrderDetail -
PurchaseOrderID PurchaseOrderDetailID] PRIMARY KEY CLUSTERED ([PurchaseOrderID],
[PurchaseOrderDetailID]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX PurchaseOrderDetail ProductID] ON [Purchasing].[Purchase-
OrderDetail] ([ProductID]) ON [PRIMARY]
ALTER TABLE [Purchasing].[PurchaseOrderDetail] ADD CONSTRAINT [FK PurchaseOrderDetail -
Product ProductID] FOREIGN KEY ([ProductID]) REFERENCES [Production].[Product]
([ProductID])
GO
ALTER TABLE [Purchasing].[PurchaseOrderDetail] ADD CONSTRAINT [FK PurchaseOrderDetail -
PurchaseOrderHeader PurchaseOrderID] FOREIGN KEY ([PurchaseOrderID]) REFERENCES
[Purchasing].[PurchaseOrderHeader] ([PurchaseOrderID])
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
EXEC sp_addextendedproperty N'MS_Description', N'Date the product is expected to be
received.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderDetail', 'COLUMN', N'Due-
Date!
GO
{\tt EXEC} \ {\tt sp\_addextended} property \ {\tt N'MS\_Description', \ N'Per} \ product \ subtotal. \ {\tt Computed} \ as
OrderQty * UnitPrice.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderDetail',
'COLUMN', N'LineTotal'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderDetail', 'COLUMN',
N'ModifiedDate'
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'PurchaseOrderDetail',
'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 Purchase-
OrderHeader.PurchaseOrderID.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrder-
Detail', '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'
EXEC sp_addextendedproperty N'MS_Description', N'Quantity rejected during inspection.',
'SCHEMA, N'Purchasing', 'TABLE', N'PurchaseOrderDetail', 'COLUMN', N'RejectedQty'
EXEC sp addextendedproperty N'MS Description', N'Quantity accepted into inventory.
Computed as ReceivedQty - RejectedQty.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase-
OrderDetail', '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'Unit-
Price!
GO
EXEC sp addextendedproperty N'MS Description', N'Check constraint [OrderQty] > (0)',
'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderDetail', 'CONSTRAINT', N'CK Purchase-
OrderDetail 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 PurchaseOrder-
TD'
EXEC sp addextendedproperty N'MS Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderDetail', 'CONSTRAINT', N'PK Purchase-
OrderDetail PurchaseOrderID PurchaseOrderDetailID'
EXEC sp addextendedproperty N'MS Description', N'Nonclustered index.', 'SCHEMA',
N'Purchasing', 'TABLE', N'PurchaseOrderDetail', 'INDEX', N'IX PurchaseOrderDetail -
ProductID'
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'
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'
```

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

[Purchasing].[PurchaseOrderHeader]

MS_Description

General purchase order information. See PurchaseOrderDetail.

Properties

Property	Value
Row Count (~)	4012
Created	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:52 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Persist ed	Compu ted	Max Length (Bytes)	Nullabil ity	Identity	Default
P/ <mark>C</mark>	PurchaseOrderID Primary key.	int			4	NOT NULL	1 - 1	
	RevisionNumber Incremental number to track changes to the purchase order over time.	tinyint			1	NOT NULL		((0))
■	Status Order current status. 1 = Pending; 2 = Approved; 3 = Rejected; 4 = Complete	tinyint			1	NOT NULL		((1))
FKP	EmployeeID Employee who created the purchase order. Foreign key to Employee.Business- EntityID.	int			4	NOT NULL		
F P	VendorID Vendor with whom the purchase order is placed. Foreign key to Vendor.BusinessEntity-ID.	int			4	NOT NULL		
F	ShipMethodID Shipping method. Foreign key to Ship- Method.ShipMethodID.	int			4	NOT NULL		
	OrderDate Purchase order creation date.	datetime			8	NOT NULL		(getdate ())
	ShipDate Estimated shipment date from the vendor.	datetime			8	NULL allowed		

SubTotal Purchase order subtotal. Computed as SUM(PurchaseOrder- Detail.LineTotal)for the appropriate Purchase- OrderID.	money			8	NOT NULL	((0.00))
TaxAmt Tax amount.	money			8	NOT NULL	((0.00))
Freight Shipping cost.	money			8	NOT NULL	((0.00))
TotalDue Total due to vendor. Computed as Subtotal + TaxAmt + Freight.	money	True	True	8	NOT NULL	
ModifiedDate Date and time the record was last updated.	datetime			8	NOT NULL	(getdate ())

Computed columns

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

Indexes

Key	Name	Key Columns	Unique
PK G	PK_PurchaseOrderHeader_PurchaseOrderID Primary key (clustered) constraint	PurchaseOrderID	True
	IX_PurchaseOrderHeader_EmployeeID Nonclustered index.	EmployeeID	
	IX_PurchaseOrderHeader_VendorID Nonclustered index.	VendorID	

Triggers

Name	ANSI Nulls On	Quoted Identifier On	On
uPurchaseOrderHeader AFTER UPDATE trigger that updates the RevisionNumber and ModifiedDate columns in the PurchaseOrderHeader table.	True	True	After Update

Check Constraints

Name	On Column	Constraint
CK_PurchaseOrderHeader_Freight Check constraint [Freight] >= (0.00)	Freight	([Freight]>=(0.00))
CK_PurchaseOrderHeader_ShipDate Check constraint [ShipDate] >= [OrderDate] OR [ShipDate] IS NULL		([ShipDate]>=[Order- Date] OR [ShipDate] IS

		NULL)
CK_PurchaseOrderHeader_Status Check constraint [Status] BETWEEN (1) AND (4)	Status	([Status]>=(1) AND [Status]<=(4))
CK_PurchaseOrderHeader_SubTotal Check constraint [SubTotal] >= (0.00)	SubTotal	([SubTotal]>=(0.00))
CK_PurchaseOrderHeader_TaxAmt Check constraint [TaxAmt] >= (0.00)	TaxAmt	([TaxAmt]>=(0.00))

Foreign Keys

Name	Columns
FK_PurchaseOrderHeader_Employee_EmployeeID Foreign key constraint referencing Employee.Employee-ID.	EmployeeID->[HumanResources].[Employee].[Business-EntityID]
FK_PurchaseOrderHeader_ShipMethod_ShipMethodID Foreign key constraint referencing ShipMethod.Ship- MethodID.	ShipMethodID->[Purchasing].[ShipMethod].[ShipMethod-ID]
FK_PurchaseOrderHeader_Vendor_VendorID Foreign key constraint referencing Vendor.VendorID.	VendorID->[Purchasing].[Vendor].[BusinessEntityID]

```
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].[PurchaseOrder-
Header]
AFTER UPDATE AS
   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 PurchaseOrderHeader -
Freight] CHECK (([Freight]>=(0.00)))
ALTER TABLE [Purchasing].[PurchaseOrderHeader] ADD CONSTRAINT [CK PurchaseOrderHeader -
ShipDate] CHECK (([ShipDate]>=[OrderDate] OR [ShipDate] IS NULL))
ALTER TABLE [Purchasing].[PurchaseOrderHeader] ADD CONSTRAINT [CK PurchaseOrderHeader -
Status] CHECK (([Status]>=(1) AND [Status]<=(4)))
ALTER TABLE [Purchasing].[PurchaseOrderHeader] ADD CONSTRAINT [CK PurchaseOrderHeader -
SubTotal] CHECK (([SubTotal]>=(0.00)))
ALTER TABLE [Purchasing].[PurchaseOrderHeader] ADD CONSTRAINT [CK PurchaseOrderHeader -
TaxAmt] CHECK (([TaxAmt]>=(0.00)))
ALTER TABLE [Purchasing].[PurchaseOrderHeader] ADD CONSTRAINT [PK PurchaseOrderHeader -
PurchaseOrderID] PRIMARY KEY CLUSTERED ([PurchaseOrderID]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX PurchaseOrderHeader EmployeeID] ON [Purchasing].[Purchase-
OrderHeader] ([EmployeeID]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX PurchaseOrderHeader VendorID] ON [Purchasing].[Purchase-
OrderHeader] ([VendorID]) ON [PRIMARY]
ALTER TABLE [Purchasing].[PurchaseOrderHeader] ADD CONSTRAINT [FK_PurchaseOrderHeader_-
Employee_EmployeeID] FOREIGN KEY ([EmployeeID]) REFERENCES [HumanResources].[Employee]
```

```
([BusinessEntityID])
ALTER TABLE [Purchasing].[PurchaseOrderHeader] ADD CONSTRAINT [FK PurchaseOrderHeader -
ShipMethod ShipMethodID] FOREIGN KEY ([ShipMethodID]) REFERENCES [Purchasing].[Ship-
Method] ([ShipMethodID])
ALTER TABLE [Purchasing].[PurchaseOrderHeader] ADD CONSTRAINT [FK PurchaseOrderHeader -
Vendor VendorID] FOREIGN KEY ([VendorID]) REFERENCES [Purchasing].[Vendor] ([Business-
EntityID])
GO
{\tt EXEC} \ {\tt sp\_addextended} property \ {\tt N'MS\_Description'}, \ {\tt N'General} \ purchase \ {\tt order} \ {\tt 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'
EXEC sp_addextendedproperty N'MS_Description', N'Shipping cost.', 'SCHEMA',
N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'COLUMN', N'Freight'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'COLUMN',
N'ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Purchase order creation date.',
'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'COLUMN', N'OrderDate'
EXEC sp addextendedproperty N'MS Description', N'Primary key.', 'SCHEMA',
N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'COLUMN', N'PurchaseOrderID'
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'
EXEC sp addextendedproperty N'MS Description', N'Estimated shipment date from the
vendor., 'SCHEMA', N'Purchasing, 'TABLE', N'PurchaseOrderHeader', 'COLUMN', N'Ship-
EXEC sp addextendedproperty N'MS Description', N'Shipping method. Foreign key to Ship-
Method.ShipMethodID.', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader',
'COLUMN', N'ShipMethodID'
EXEC sp addextendedproperty N'MS Description', N'Order current status. 1 = Pending; 2 =
Approved; 3 = Rejected; 4 = Complete', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase-
OrderHeader', 'COLUMN', N'Status'
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'
EXEC sp addextendedproperty N'MS Description', N'Tax amount.', 'SCHEMA', N'Purchasing',
'TABLE', N'PurchaseOrderHeader', 'COLUMN', N'TaxAmt'
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
EXEC sp_addextendedproperty N'MS_Description', N'Vendor with whom the purchase order is
placed. Foreign key to Vendor. Business Entity ID. ', '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 Purchase-
OrderHeader Freight'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [ShipDate] >= [Order-
Date] OR [ShipDate] IS NULL', 'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader',
'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 Purchase-
OrderHeader TaxAmt'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'CONSTRAINT', N'DF Purchase-
OrderHeader 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 Purchase-
OrderHeader RevisionNumber'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 1',
'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'CONSTRAINT', N'DF Purchase-
OrderHeader Status'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA, N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'CONSTRAINT', N'DF Purchase-
OrderHeader SubTotal'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'CONSTRAINT', N'DF Purchase-
OrderHeader 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'
```

```
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Purchasing', 'TABLE', N'PurchaseOrderHeader', 'CONSTRAINT', N'PK_Purchase-
OrderHeader_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'uPurchaseOrderHeader'
GO
```

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
PK P G	ShipMethodID Primary key for ShipMethod records.	int	4	NOT NULL	1 - 1	
±.	Name Shipping company name.	[dbo].[Name]	100	NOT NULL		
=	ShipBase Minimum shipping charge.	money	8	NOT NULL		((0.00))
=	ShipRate Shipping charge per pound.	money	8	NOT NULL		((0.00))
ф	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL		(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

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

Check Constraints

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

```
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)),
[rowquid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF ShipMethod rowquid]
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)))
ALTER TABLE [Purchasing].[ShipMethod] ADD CONSTRAINT [CK ShipMethod ShipRate] CHECK
(([ShipRate]>(0.00)))
ALTER TABLE [Purchasing]. [ShipMethod] ADD CONSTRAINT [PK ShipMethod ShipMethodID]
PRIMARY KEY CLUSTERED ([ShipMethodID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK ShipMethod Name] ON [Purchasing].[ShipMethod]
([Name]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK ShipMethod rowquid] ON [Purchasing].[ShipMethod]
([rowguid]) ON [PRIMARY]
EXEC sp addextendedproperty N'MS Description', N'Shipping company lookup table.',
'SCHEMA', N'Purchasing', 'TABLE', N'ShipMethod', NULL, NULL
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Purchasing', 'TABLE', N'ShipMethod', 'COLUMN', N'ModifiedDate'
GO
EXEC sp addextendedproperty N'MS Description', N'Shipping company name.', 'SCHEMA',
N'Purchasing', 'TABLE', N'ShipMethod', 'COLUMN', N'Name'
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'
EXEC sp addextendedproperty N'MS Description', N'Minimum shipping charge.', 'SCHEMA',
N'Purchasing', 'TABLE', N'ShipMethod', 'COLUMN', N'ShipBase'
EXEC sp addextendedproperty N'MS Description', N'Primary key for ShipMethod records.',
'SCHEMA', N'Purchasing', 'TABLE', N'ShipMethod', 'COLUMN', N'ShipMethodID'
```

```
GO
EXEC sp addextendedproperty N'MS Description', N'Shipping charge per pound.', 'SCHEMA',
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 ShipMethod Ship-
GO
EXEC sp addextendedproperty N'MS Description', N'Check constraint [ShipRate] > (0.00)',
'SCHEMA', N'Purchasing', 'TABLE', N'ShipMethod', 'CONSTRAINT', N'CK ShipMethod Ship-
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 ShipMethod rowguid'
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 Ship-
MethodID'
EXEC sp addextendedproperty N'MS Description', N'Unique nonclustered index.', 'SCHEMA',
N'Purchasing', 'TABLE', N'ShipMethod', 'INDEX', N'AK ShipMethod Name'
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 Ship-
Method ShipMethodID'
GO
```

[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	17:24:41 2017. október 19., csütörtök	
Last Modified	17:24:52 2017. október 19., csütörtök	

Columns

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

Indexes

Key	Name	Key Columns	Unique
PK G	PK_Vendor_BusinessEntityID Primary key (clustered) constraint	BusinessEntityID	True

AK_Vendor_AccountNumber	AccountNumber	True	
Unique nonclustered index.			

Triggers

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

Check Constraints

Name	On Column	Constraint
CK_Vendor_CreditRating Check constraint [CreditRating] BETWEEN (1) AND (5)	CreditRating	([CreditRating]>=(1) AND [Credit-Rating]<=(5))

Foreign Keys

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

```
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
   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;
ALTER TABLE [Purchasing].[Vendor] ADD CONSTRAINT [CK Vendor CreditRating] CHECK
(([CreditRating]>=(1) AND [CreditRating]<=(5)))
ALTER TABLE [Purchasing].[Vendor] ADD CONSTRAINT [PK Vendor BusinessEntityID] PRIMARY
KEY CLUSTERED ([BusinessEntityID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK Vendor AccountNumber] ON [Purchasing].[Vendor]
([AccountNumber]) ON [PRIMARY]
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'Active-
Flag'
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'
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'
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'
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'
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'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'CONSTRAINT', N'DF Vendor -
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of 1 (TRUE)',
'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'CONSTRAINT', N'DF Vendor PreferredVendor-
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'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'CONSTRAINT', N'PK_Vendor_BusinessEntity-
GO
EXEC sp addextendedproperty N'MS Description', N'Unique nonclustered index.', 'SCHEMA',
N'Purchasing', 'TABLE', N'Vendor', 'INDEX', N'AK Vendor AccountNumber'
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
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'
```

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

Used By

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

Image: [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	17:24:40 2017. október 19., csütörtök	
Last Modified	17:24:51 2017. október 19., csütörtök	

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PK <mark>P</mark> FKP	CountryRegionCode ISO code for countries and regions. Foreign key to CountryRegion.Country- RegionCode.	nvarchar(3)	6	NOT NULL	
P <mark>/</mark> Pth.	CurrencyCode ISO standard currency code. Foreign key to Currency.CurrencyCode.	nchar(3)	6	NOT NULL	
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Key	Name	Key Columns	Unique
PKP C	PK_CountryRegionCurrency_CountryRegionCode_CurrencyCode Primary key (clustered) constraint	CountryRegion- Code, Currency- Code	True
	IX_CountryRegionCurrency_CurrencyCode Nonclustered index.	CurrencyCode	

Foreign Keys

Name	Columns
FK_CountryRegionCurrency_CountryRegion_Country- RegionCode Foreign key constraint referencing Country- Region.CountryRegionCode.	CountryRegionCode->[Person].[CountryRegion].[Country-RegionCode]
FK_CountryRegionCurrency_Currency_CurrencyCode	CurrencyCode->[Sales].[Currency].[CurrencyCode]

```
Foreign key constraint referencing Currency-Code.
```

```
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]
ALTER TABLE [Sales]. [CountryRegionCurrency] ADD CONSTRAINT [PK CountryRegionCurrency -
CountryRegionCode CurrencyCode] PRIMARY KEY CLUSTERED ([CountryRegionCode], [Currency-
Code]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX CountryRegionCurrency CurrencyCode] ON [Sales].[Country-
RegionCurrency] ([CurrencyCode]) ON [PRIMARY]
ALTER TABLE [Sales].[CountryRegionCurrency] ADD CONSTRAINT [FK CountryRegionCurrency -
CountryRegion CountryRegionCode] FOREIGN KEY ([CountryRegionCode]) REFERENCES
[Person].[CountryRegion] ([CountryRegionCode])
ALTER TABLE [Sales]. [CountryRegionCurrency] ADD CONSTRAINT [FK CountryRegionCurrency -
Currency CurrencyCode] FOREIGN KEY ([CurrencyCode]) REFERENCES [Sales].[Currency]
([CurrencyCode])
EXEC sp addextendedproperty N'MS Description', N'Cross-reference table mapping ISO
currency codes to a country or region.', 'SCHEMA', N'Sales', 'TABLE', N'CountryRegion-
Currency', 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'
EXEC sp addextendedproperty N'MS Description', N'ISO standard currency code. Foreign
key to Currency.CurrencyCode.', "SCHEMA', N'Sales', 'TABLE', N'CountryRegionCurrency',
'COLUMN', N'CurrencyCode'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'CountryRegionCurrency', 'COLUMN', N'Modified-
Date'
GO
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'CountryRegionCurrency', 'CONSTRAINT', N'DF -
CountryRegionCurrency ModifiedDate'
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'
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'
EXEC sp addextendedproperty N'MS Description', N'Primary key (clustered) constraint',
'SCHEMAT, N'Sales', 'TABLE', N'CountryRegionCurrency', 'CONSTRAINT', N'PK_CountryRegion-
```

```
Currency_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
```

[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	17:24:40 2017. október 19., csütörtök	
Last Modified	17:24:51 2017. október 19., csütörtök	

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
PK	CreditCardID Primary key for CreditCard records.	int	4	NOT NULL	1 - 1	
	CardType Credit card name.	nvarchar(50)	100	NOT NULL		
.	CardNumber Credit card number.	nvarchar(25)	50	NOT NULL		
	ExpMonth Credit card expiration month.	tinyint	1	NOT NULL		
	ExpYear Credit card expiration year.	smallint	2	NOT NULL		
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

ŀ	C ey	Name	Key Columns	Unique
P	C _C	PK_CreditCard_CreditCardID Primary key (clustered) constraint	CreditCardID	True
		AK_CreditCard_CardNumber Unique nonclustered index.	CardNumber	True

```
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]
ALTER TABLE [Sales].[CreditCard] ADD CONSTRAINT [PK CreditCard CreditCardID] PRIMARY
KEY CLUSTERED ([CreditCardID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK CreditCard CardNumber] ON [Sales].[CreditCard]
([CardNumber]) ON [PRIMARY]
EXEC sp addextendedproperty N'MS Description', N'Customer credit card information.',
'SCHEMA', N'Sales', 'TABLE', N'CreditCard', NULL, NULL
EXEC sp addextendedproperty N'MS Description', N'Credit card number.', 'SCHEMA',
N'Sales', 'TABLE', N'CreditCard', 'COLUMN', N'CardNumber'
EXEC sp addextendedproperty N'MS Description', N'Credit card name.', 'SCHEMA',
N'Sales, 'TABLE', N'CreditCard, 'COLUMN', N'CardType'
EXEC sp addextendedproperty N'MS Description', N'Primary key for CreditCard records.',
'SCHEMA', N'Sales', 'TABLE', N'CreditCard', 'COLUMN', N'CreditCardID'
EXEC sp addextendedproperty N'MS Description', N'Credit card expiration month.',
'SCHEMA', N'Sales', 'TABLE', N'CreditCard', 'COLUMN', N'ExpMonth'
EXEC sp addextendedproperty N'MS Description', N'Credit card expiration year.',
'SCHEMA', N'Sales', 'TABLE', N'CreditCard', 'COLUMN', N'ExpYear'
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_CreditCard_-
ModifiedDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint',
'SCHEMA, N'Sales', 'TABLE', N'CreditCard', 'CONSTRAINT', N'PK CreditCard CreditCardID'
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index.', 'SCHEMA',
N'Sales, 'TABLE', N'CreditCard', 'INDEX', N'AK CreditCard CardNumber'
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
```

Sales

Used By

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

Image: Sales in Sales in

MS_Description

Lookup table containing standard ISO currencies.

Properties

Property	Value	
Collation SQL_Latin1_General_CP1_CI_AS		
Row Count (~)	105	
Created	17:24:40 2017. október 19., csütörtök	
Last Modified	17:24:51 2017. október 19., csütörtök	

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PK C	CurrencyCode The ISO code for the Currency.	nchar(3)	6	NOT NULL	
.	Name Currency name.	[dbo].[Name]	100	NOT NULL	
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Key	Name	Key Columns	Unique
PK	PK_CurrencyCode Primary key (clustered) constraint	CurrencyCode	True
	AK_Currency_Name Unique nonclustered index.	Name	True

```
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
EXEC sp addextendedproperty N'MS Description', N'Lookup table containing standard ISO
currencies.', 'SCHEMA', N'Sales', 'TABLE', N'Currency', NULL, NULL
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'
EXEC sp addextendedproperty N'MS Description', N'Currency name.', 'SCHEMA', N'Sales',
'TABLE', N'Currency', 'COLUMN', N'Name'
{\tt EXEC} \  \, {\tt sp\_addextended} property \  \, {\tt N'MS\_Description'}, \  \, {\tt N'Default} \  \, {\tt constraint} \  \, {\tt value} \  \, {\tt of}
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Currency', 'CONSTRAINT', N'DF Currency -
ModifiedDate'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Sales', 'TABLE', N'Currency', 'CONSTRAINT', N'PK Currency CurrencyCode'
EXEC sp addextendedproperty N'MS Description', N'Unique nonclustered index.', 'SCHEMA',
N'Sales', 'TABLE', N'Currency', 'INDEX', N'AK Currency Name'
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'
```

[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	17:24:40 2017. október 19., csütörtök	
Last Modified	17:24:51 2017. október 19., csütörtök	

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
PK <mark>P</mark> C	CurrencyRateID Primary key for CurrencyRate records.	int	4	NOT NULL	1 - 1	
.	CurrencyRateDate Date and time the exchange rate was obtained.	datetime	8	NOT NULL		
r F	FromCurrencyCode Exchange rate was converted from this currency code.	nchar(3)	6	NOT NULL		
r P	ToCurrencyCode Exchange rate was converted to this currency code.	nchar(3)	6	NOT NULL		
	AverageRate Average exchange rate for the day.	money	8	NOT NULL		
	EndOfDayRate Final exchange rate for the day.	money	8	NOT NULL		
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

Key	Name	Key Columns	Unique
PK	PK_CurrencyRate_CurrencyRateID Primary key (clustered) constraint	CurrencyRate- ID	True
	AK_CurrencyRate_CurrencyRateDate_FromCurrencyCode_ToCurrencyCode Unique nonclustered index.	CurrencyRate- Date, From- CurrencyCode, ToCurrency-	True

	Code	

Foreign Keys

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

```
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]
ALTER TABLE [Sales].[CurrencyRate] ADD CONSTRAINT [PK CurrencyRate CurrencyRateID]
PRIMARY KEY CLUSTERED ([CurrencyRateID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK CurrencyRate CurrencyRateDate FromCurrencyCode To-
CurrencyCode] ON [Sales].[CurrencyRate] ([CurrencyRateDate], [FromCurrencyCode], [To-
CurrencyCode]) ON [PRIMARY]
ALTER TABLE [Sales].[CurrencyRate] ADD CONSTRAINT [FK CurrencyRate Currency From-
CurrencyCode] FOREIGN KEY ([FromCurrencyCode]) REFERENCES [Sales]. [Currency] ([Currency-
Code])
ALTER TABLE [Sales].[CurrencyRate] ADD CONSTRAINT [FK CurrencyRate Currency ToCurrency-
Code] FOREIGN KEY ([ToCurrencyCode]) REFERENCES [Sales].[Currency] ([CurrencyCode])
EXEC sp addextendedproperty N'MS Description', N'Currency exchange rates.', 'SCHEMA',
N'Sales', 'TABLE', N'CurrencyRate', NULL, NULL
EXEC sp_addextendedproperty N'MS_Description', N'Average exchange rate for the day.',
'SCHEMA', N'Sales', 'TABLE', N'CurrencyRate', 'COLUMN', N'AverageRate'
EXEC sp addextendedproperty N'MS Description', N'Date and time the exchange rate was
obtained.', 'SCHEMA', N'Sales', 'TABLE', N'CurrencyRate', 'COLUMN', N'CurrencyRateDate'
EXEC sp addextendedproperty N'MS Description', N'Primary key for CurrencyRate
records.', 'SCHEMA', N'Sales', 'TABLE', N'CurrencyRate', 'COLUMN', N'CurrencyRateID'
EXEC sp_addextendedproperty N'MS_Description', N'Final exchange rate for the day.',
'SCHEMA', 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'FromCurrency-
Code'
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'ToCurrency-
Code'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'CurrencyRate', 'CONSTRAINT', N'DF_Currency-
Rate 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_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPate_CurrencyPa
RateID'
GO
EXEC sp addextendedproperty N'MS Description', N'Unique nonclustered index.', 'SCHEMA',
N'Sales, 'TABLE', N'CurrencyRate', 'INDEX', N'AK_CurrencyRate_CurrencyRateDate_From-
CurrencyCode 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 Currency-
Rate CurrencyRateID'
GO
```

[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	17:24:40 2017. október 19., csütörtök	
Last Modified	17:24:51 2017. október 19., csütörtök	

Columns

Key	Name	Data Type	Comp uted	Max Length (Bytes	Nullabi lity	ldentit y	Identit y Replic ation	Defaul t
PKP C	CustomerID Primary key.	int		4	NOT NULL	1 - 1	False	
F	PersonID Foreign key to Person.BusinessEntityID	int		4	NULL allowe d			
FK	StoreID Foreign key to Store.BusinessEntityID	int		4	NULL allowe d			
FKP	TerritoryID ID of the territory in which the customer is located. Foreign key to SalesTerritory.Sales- TerritoryID.	int		4	NULL allowe d			
.ft.	AccountNumber Unique number identifying the customer assigned by the accounting system.	varchar(10)	True	10	NOT NULL			
#	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier		16	NOT NULL			(newid())
	ModifiedDate Date and time the record was last updated.	datetime		8	NOT NULL			(getdat e())

Computed columns

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

Indexes

Key	Name	Key Columns	Unique
PK C	PK_Customer_CustomerID Primary key (clustered) constraint	CustomerID	True
	AK_Customer_AccountNumber Unique nonclustered index.	AccountNumber	True
	AK_Customer_rowguid Unique nonclustered index. Used to support replication samples.	rowguid	True
	IX_Customer_TerritoryID Nonclustered index.	TerritoryID	

Foreign Keys

Name	Columns	
FK_Customer_Person_PersonID Foreign key constraint referencing Person.BusinessEntityID.	PersonID->[Person].[BusinessEntityID]	
FK_Customer_SalesTerritory_TerritoryID Foreign key constraint referencing SalesTerritory.TerritoryID.	TerritoryID->[Sales].[SalesTerritory].[TerritoryID]	
FK_Customer_Store_StoreID Foreign key constraint referencing Store.BusinessEntityID.	StoreID->[Sales].[Store].[BusinessEntityID]	

```
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]
ALTER TABLE [Sales].[Customer] ADD CONSTRAINT [PK Customer CustomerID] PRIMARY KEY
CLUSTERED ([CustomerID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK Customer AccountNumber] ON [Sales].[Customer]
([AccountNumber]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK_Customer_rowguid] ON [Sales].[Customer]
```

```
([rowguid]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX Customer TerritoryID] ON [Sales].[Customer] ([Territory-
ID]) ON [PRIMARY]
ALTER TABLE [Sales].[Customer] ADD CONSTRAINT [FK Customer Person PersonID] FOREIGN KEY
([PersonID]) REFERENCES [Person].[Person] ([BusinessEntityID])
ALTER TABLE [Sales].[Customer] ADD CONSTRAINT [FK Customer SalesTerritory TerritoryID]
FOREIGN KEY ([TerritoryID]) REFERENCES [Sales].[SalesTerritory] ([TerritoryID])
ALTER TABLE [Sales].[Customer] ADD CONSTRAINT [FK Customer Store StoreID] FOREIGN KEY
([StoreID]) REFERENCES [Sales].[Store] ([BusinessEntityID])
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
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'
EXEC sp addextendedproperty N'MS Description', N'Primary key.', 'SCHEMA', N'Sales',
'TABLE', N'Customer', 'COLUMN', N'CustomerID'
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.BusinessEntity-
ID', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'COLUMN', N'PersonID'
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'
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'
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'
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'
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'
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'
```

```
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
```

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

Used By

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

I [Sales].[PersonCreditCard]

MS_Description

 $Cross-reference\ table\ mapping\ people\ to\ their\ credit\ card\ information\ in\ the\ Credit\ Card\ table.$

Properties

Property	Value
Row Count (~)	19118
Created	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKC FK	BusinessEntityID Business entity identification number. Foreign key to Person.BusinessEntityID.	int	4	NOT NULL	
PK C FK	CreditCardID Credit card identification number. Foreign key to CreditCard.CreditCardID.	int	4	NOT NULL	
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Key	Name	Key Columns	Unique
PK# C	PK_PersonCreditCard_BusinessEntityID_CreditCardID Primary key (clustered) constraint	BusinessEntityID, CreditCardID	True

Foreign Keys

Name	Columns
FK_PersonCreditCard_CreditCard_CreditCardID Foreign key constraint referencing CreditCard.CreditCard- ID.	CreditCardID->[Sales].[CreditCard].[CreditCardID]
FK_PersonCreditCard_Person_BusinessEntityID Foreign key constraint referencing Person.BusinessEntity-ID.	BusinessEntityID->[Person].[Person].[BusinessEntityID]

```
CREATE TABLE [Sales].[PersonCreditCard]
[BusinessEntityID] [int] NOT NULL,
[CreditCardID] [int] NOT NULL,
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF PersonCreditCard ModifiedDate]
DEFAULT (getdate())
) ON [PRIMARY]
ALTER TABLE [Sales].[PersonCreditCard] ADD CONSTRAINT [PK PersonCreditCard Business-
EntityID CreditCardID] PRIMARY KEY CLUSTERED ([BusinessEntityID], [CreditCardID]) ON
[PRIMARY]
ALTER TABLE [Sales].[PersonCreditCard] ADD CONSTRAINT [FK PersonCreditCard CreditCard -
CreditCardID] FOREIGN KEY ([CreditCardID]) REFERENCES [Sales].[CreditCard] ([CreditCard-
TD1)
ALTER TABLE [Sales].[PersonCreditCard] ADD CONSTRAINT [FK PersonCreditCard Person -
BusinessEntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES [Person].[Person]
([BusinessEntityID])
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
EXEC sp addextendedproperty N'MS Description', N'Business entity identification number.
Foreign key to Person. Business Entity ID.', 'SCHEMA', N'Sales', 'TABLE', N'Person Credit-
Card', 'COLUMN', N'BusinessEntityID'
EXEC sp addextendedproperty N'MS Description', N'Credit card identification number.
Foreign key to CreditCard.CreditCardID.', 'SCHEMA', N'Sales', 'TABLE', N'PersonCredit-
Card', 'COLUMN', N'CreditCardID'
EXEC sp_addextendedproperty N'MS_Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'PersonCreditCard', 'COLUMN', N'ModifiedDate'
GO
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'PersonCreditCard', 'CONSTRAINT', N'DF_Person-
CreditCard 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'
GΩ
```

[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	17:24:40 2017. október 19., csütörtök		
Last Modified	17:24:52 2017. október 19., csütörtök		

Columns

Key	Name	Data Type	Comput ed	Max Length (Bytes)	Nullabili ty	Identity	Default
PKO FKO	SalesOrderID Primary key. Foreign key to SalesOrderHeader.Sales- OrderID.	int		4	NOT NULL		
PK <mark>P</mark> C	SalesOrderDetailID Primary key. One incremental unique number per product sold.	int		4	NOT NULL	1-1	
	CarrierTrackingNumber Shipment tracking number supplied by the shipper.	nvarchar(25)		50	NULL allowed		
III	OrderQty Quantity ordered per product.	smallint		2	NOT NULL		
r <mark>j</mark>	ProductID Product sold to customer. Foreign key to Product.ProductID.	int		4	NOT NULL		
F	SpecialOfferID Promotional code. Foreign key to SpecialOffer.Special- OfferID.	int		4	NOT NULL		
III	UnitPrice Selling price of a single product.	money		8	NOT NULL		
	UnitPriceDiscount Discount amount.	money		8	NOT NULL		((0.0))
	LineTotal Per product subtotal. Computed as UnitPrice * (1 - UnitPriceDiscount) * Order-	numeric(38,6)	True	17	NOT NULL		

	Qty.				
ı.th	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL	(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Computed columns

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

Indexes

Key	Name	Key Columns	Unique
₽ <mark>%</mark> G	PK_SalesOrderDetail_SalesOrderID_SalesOrderDetailID Primary key (clustered) constraint	SalesOrderID, SalesOrder- DetailID	True
	AK_SalesOrderDetail_rowguid Unique nonclustered index. Used to support replication samples.	rowguid	True
	IX_SalesOrderDetail_ProductID Nonclustered index.	ProductID	

Triggers

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

Check Constraints

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

Foreign Keys

Name	Delete	Columns
FK_SalesOrderDetail_SalesOrderHeader_SalesOrderID Foreign key constraint referencing SalesOrderHeader.PurchaseOrderID.	Cascade	SalesOrderID- >[Sales].[SalesOrder- Header].[SalesOrderID]
FK_SalesOrderDetail_SpecialOfferProduct_SpecialOfferIDProductID Foreign key constraint referencing SpecialOfferProduct.SpecialOfferIDProduct-ID.		SpecialOfferID- >[Sales].[SpecialOffer- Product].[SpecialOffer- ID] ProductID- >[Sales].[SpecialOffer- Product].[ProductID]

```
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))),
[rowquid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF SalesOrder-
Detail_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
   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].[SalesOrderID];
            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].[SalesOrder-
Detail].[SalesOrderID])
       WHERE [Sales].[SalesOrderHeader].[SalesOrderID] IN (SELECT inserted.[SalesOrder-
ID] FROM inserted);
        UPDATE [Person].[Person]
        SET [Demographics].modify('declare default element namespace
            "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Individual-
Survey";
            replace value of (/IndividualSurvey/TotalPurchaseYTD)[1]
            with data(/IndividualSurvey/TotalPurchaseYTD)[1] - sql:column("deleted.Line-
Total")')
        FROM deleted
            INNER JOIN [Sales].[SalesOrderHeader]
            ON deleted.[SalesOrderID] = [Sales].[SalesOrderHeader].[SalesOrderID]
            INNER JOIN [Sales].[Customer]
```

```
ON [Sales].[Customer].[CustomerID] = [Sales].[SalesOrderHeader].[Customer-
TD1
                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)))
ALTER TABLE [Sales].[SalesOrderDetail] ADD CONSTRAINT [CK SalesOrderDetail UnitPrice]
CHECK (([UnitPrice]>=(0.00)))
ALTER TABLE [Sales].[SalesOrderDetail] ADD CONSTRAINT [CK SalesOrderDetail UnitPrice-
Discount] CHECK (([UnitPriceDiscount]>=(0.00)))
ALTER TABLE [Sales].[SalesOrderDetail] ADD CONSTRAINT [PK SalesOrderDetail SalesOrder-
ID SalesOrderDetailID] PRIMARY KEY CLUSTERED ([SalesOrderID], [SalesOrderDetailID]) ON
[PRIMARY]
CREATE NONCLUSTERED INDEX [IX SalesOrderDetail ProductID] ON [Sales].[SalesOrderDetail]
([ProductID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK_SalesOrderDetail_rowguid] ON [Sales].[SalesOrder-
Detail] ([rowguid]) ON [PRIMARY]
ALTER TABLE [Sales].[SalesOrderDetail] ADD CONSTRAINT [FK SalesOrderDetail SalesOrder-
Header SalesOrderID] FOREIGN KEY ([SalesOrderID]) REFERENCES [Sales].[SalesOrderHeader]
([SalesOrderID]) ON DELETE CASCADE
GO
ALTER TABLE [Sales].[SalesOrderDetail] ADD CONSTRAINT [FK SalesOrderDetail SpecialOffer-
Product_SpecialOfferIDProductID] FOREIGN KEY ([SpecialOfferID], [ProductID]) REFERENCES
[Sales].[SpecialOfferProduct] ([SpecialOfferID], [ProductID])
{\tt EXEC} \  \, {\tt sp\_addextended} property \  \, {\tt N'MS\_Description'}, \  \, {\tt N'Individual} \  \, products \  \, {\tt associated} \  \, {\tt with} \  \, {\tt associated} \  \, {\tt associated} \  \, {\tt with} \  \, {\tt associated} \  \, {\tt associated} \  \, {\tt associated} \  \, {\tt with} \  \, {\tt associated} \ 
specific sales order. See SalesOrderHeader.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrder-
Detail', 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'Carrier-
TrackingNumber'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Per product subtotal. Computed as Unit-
Price * (1 - UnitPriceDiscount) * OrderQty.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrder-
Detail', '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'ModifiedDate'
GO
EXEC sp addextendedproperty N'MS Description', N'Quantity ordered per product.',
'SCHEMA, N'Sales', 'TABLE', N'SalesOrderDetail', 'COLUMN', N'OrderQty
EXEC sp addextendedproperty N'MS Description', N'Product sold to customer. Foreign key
to Product.ProductID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail', 'COLUMN',
N'Product ID'
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'
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'
EXEC sp addextendedproperty N'MS Description', N'Primary key. Foreign key to SalesOrder-
Header.SalesOrderID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail', 'COLUMN',
N'SalesOrderID'
EXEC sp addextendedproperty N'MS Description', N'Promotional code. Foreign key to
SpecialOffer.SpecialOfferID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail',
'COLUMN', N'SpecialOfferID'
EXEC sp addextendedproperty N'MS Description', N'Selling price of a single product.',
'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail', 'COLUMN', N'UnitPrice'
EXEC sp addextendedproperty N'MS Description', N'Discount amount.', 'SCHEMA', N'Sales',
'TABLE', N'SalesOrderDetail', 'COLUMN', N'UnitPriceDiscount'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [OrderQty] > (0)',
'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail', 'CONSTRAINT', N'CK SalesOrderDetail -
EXEC sp addextendedproperty N'MS Description', N'Check constraint [UnitPrice] >=
(0.00), 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail', 'CONSTRAINT', N'CK Sales-
OrderDetail UnitPrice'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [UnitPriceDiscount]
>= (0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail', 'CONSTRAINT', N'CK Sales-
OrderDetail UnitPriceDiscount'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail', 'CONSTRAINT', N'DF Sales-
OrderDetail ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of NEWID()',
'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail', 'CONSTRAINT', N'DF SalesOrder-
Detail rowquid'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of 0.0',
'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail', 'CONSTRAINT', N'DF SalesOrderDetail -
UnitPriceDiscount'
EXEC sp addextendedproperty N'MS Description', N'Foreign key constraint referencing
SalesOrderHeader.PurchaseOrderID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail',
'CONSTRAINT', N'FK SalesOrderDetail SalesOrderHeader SalesOrderID'
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'
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 rowquid'
EXEC sp addextendedproperty N'MS Description', N'Nonclustered index.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesOrderDetail', 'INDEX', N'IX SalesOrderDetail ProductID'
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a primary
key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderDetail', 'INDEX', N'PK_Sales-
OrderDetail SalesOrderID SalesOrderDetailID'
EXEC sp addextendedproperty N'MS Description', N'AFTER INSERT, DELETE, UPDATE trigger
that inserts a row in the TransactionHistory table, updates ModifiedDate in SalesOrder-
Detail and updates the SalesOrderHeader.SubTotal column.', 'SCHEMA', N'Sales', 'TABLE',
N'SalesOrderDetail', 'TRIGGER', N'iduSalesOrderDetail'
GO
```

[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	17:24:40 2017. október 19., csütörtök		
Last Modified	17:24:52 2017. október 19., csütörtök		

Columns

Key	Name	Data Type	Comp	Max Lengt h (Bytes	Nullab ility	Identit y	Identit y Replic ation	Defaul t
PKP C	SalesOrderID Primary key.	int		4	NOT NULL	1 - 1	False	
	RevisionNumber Incremental number to track changes to the sales order over time.	tinyint		1	NOT NULL			((0))
	OrderDate Dates the sales order was created.	datetime		8	NOT NULL			(getda te())
	DueDate Date the order is due to the customer.	datetime		8	NOT NULL			
	ShipDate Date the order was shipped to the customer.	datetime		8	NULL allowe d			
	Status Order current status. 1 = In process; 2 = Approved; 3 = Backordered; 4 = Rejected; 5 = Shipped; 6 = Cancelled	tinyint		1	NOT NULL			((1))
	OnlineOrderFlag 0 = Order placed by sales person. 1 = Order placed online by customer.	[dbo].[Flag]		1	NOT NULL			((1))
ф	SalesOrderNumber Unique sales order identification number.	nvarchar(25)	True	50	NOT NULL			

	PurchaseOrderNumber Customer purchase order number reference.	[dbo].[OrderNumber]	50	NULL allowe d		
	AccountNumber Financial accounting number reference.	[dbo].[Account- Number]	30	NULL allowe d		
÷.	CustomerID Customer identification number. Foreign key to Customer.Business- EntityID.	int	4	NOT NULL		
F P	SalesPersonID Sales person who created the sales order. Foreign key to Sales- Person.BusinessEntity- ID.	int	4	NULL allowe d		
F 199	TerritoryID Territory in which the sale was made. Foreign key to Sales- Territory.SalesTerritory-ID.	int	4	NULL allowe d		
F/P	BillToAddressID Customer billing address. Foreign key to Address.AddressID.	int	4	NOT NULL		
F/P	ShipToAddressID Customer shipping address. Foreign key to Address.AddressID.	int	4	NOT NULL		
F	ShipMethodID Shipping method. Foreign key to Ship- Method.ShipMethodID.	int	4	NOT NULL		
F	CreditCardID Credit card identification number. Foreign key to CreditCard.CreditCard- ID.	int	4	NULL allowe d		
	CreditCardApproval- Code Approval code provided by the credit card company.	varchar(15)	15	NULL allowe d		
F	CurrencyRateID Currency exchange rate used. Foreign key to CurrencyRate.Currency- RateID.	int	4	NULL allowe d		
III	SubTotal Sales subtotal. Computed as SUM(SalesOrder- Detail.LineTotal)for the appropriate SalesOrder- ID.	money	8	NOT NULL		((0.00)
	TaxAmt Tax amount.	money	8	NOT NULL		((0.00)
■	Freight	money	8	NOT		((0.00)

	Shipping cost.				NULL)
	TotalDue Total due from customer. Computed as Subtotal + TaxAmt + Freight.	money	True	8	NOT NULL		
	Comment Sales representative comments.	nvarchar(128)		256	NULL allowe d		
ı.	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier		16	NOT NULL		(newid ())
	ModifiedDate Date and time the record was last updated.	datetime		8	NOT NULL		(getda te())

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 <mark>P</mark> C	PK_SalesOrderHeader_SalesOrderID Primary key (clustered) constraint	SalesOrderID	True
	AK_SalesOrderHeader_SalesOrderNumber Unique nonclustered index.	SalesOrder- Number	True
	AK_SalesOrderHeader_rowguid Unique nonclustered index. Used to support replication samples.	rowguid	True
	IX_SalesOrderHeader_CustomerID Nonclustered index.	CustomerID	
	IX_SalesOrderHeader_SalesPersonID Nonclustered index.	SalesPersonID	

Triggers

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

Check Constraints

Name	On Column	Constraint
CK_SalesOrderHeader_DueDate Check constraint [DueDate] >= [OrderDate]		([DueDate]>=[Order- Date])
CK_SalesOrderHeader_Freight Check constraint [Freight] >= (0.00)	Freight	([Freight]>=(0.00))
CK_SalesOrderHeader_ShipDate Check constraint [ShipDate] >= [OrderDate] OR [ShipDate] IS NULL		([ShipDate]>=[Order- Date] OR [ShipDate] IS NULL)
CK_SalesOrderHeader_Status Check constraint [Status] BETWEEN (0) AND (8)	Status	([Status]>=(0) AND [Status]<=(8))
CK_SalesOrderHeader_SubTotal Check constraint [SubTotal] >= (0.00)	SubTotal	([SubTotal]>=(0.00))
CK_SalesOrderHeader_TaxAmt Check constraint [TaxAmt] >= (0.00)	TaxAmt	([TaxAmt]>=(0.00))

Foreign Keys

Name	Columns
FK_SalesOrderHeader_Address_BillToAddressID Foreign key constraint referencing Address.AddressID.	BillToAddressID->[Person].[Address].[AddressID]
FK_SalesOrderHeader_Address_ShipToAddressID Foreign key constraint referencing Address.AddressID.	ShipToAddressID->[Person].[Address].[AddressID]
FK_SalesOrderHeader_CreditCard_CreditCardID Foreign key constraint referencing CreditCard.CreditCard-ID.	CreditCardID->[Sales].[CreditCard].[CreditCardID]
FK_SalesOrderHeader_CurrencyRate_CurrencyRateID Foreign key constraint referencing Currency-Rate.CurrencyRateID.	CurrencyRateID->[Sales].[CurrencyRate].[CurrencyRate-ID]
FK_SalesOrderHeader_Customer_CustomerID Foreign key constraint referencing Customer.CustomerID.	CustomerID->[Sales].[Customer].[CustomerID]
FK_SalesOrderHeader_SalesPerson_SalesPersonID Foreign key constraint referencing SalesPerson.Sales-PersonID.	SalesPersonID->[Sales].[SalesPerson].[BusinessEntityID]
FK_SalesOrderHeader_SalesTerritory_TerritoryID Foreign key constraint referencing SalesTerritory.Territory-ID.	TerritoryID->[Sales].[SalesTerritory].[TerritoryID]
FK_SalesOrderHeader_ShipMethod_ShipMethodID Foreign key constraint referencing ShipMethod.Ship- MethodID.	ShipMethodID->[Purchasing].[ShipMethod].[ShipMethod-ID]

```
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,
[rowquid] [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
            \begin{tabular}{ll} WHERE & [Sales].[SalesOrderHeader].[SalesOrderID] & IN \\ \end{tabular}
               (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;
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [CK SalesOrderHeader DueDate]
CHECK (([DueDate]>=[OrderDate]))
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [CK SalesOrderHeader Freight]
CHECK (([Freight]>=(0.00)))
```

```
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [CK SalesOrderHeader ShipDate]
CHECK (([ShipDate]>=[OrderDate] OR [ShipDate] IS NULL))
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [CK SalesOrderHeader Status]
CHECK (([Status]>=(0) AND [Status]<=(8)))</pre>
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [CK SalesOrderHeader SubTotal]
CHECK (([SubTotal]>=(0.00)))
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [CK SalesOrderHeader TaxAmt]
CHECK (([TaxAmt]>=(0.00)))
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [PK SalesOrderHeader SalesOrder-
ID] PRIMARY KEY CLUSTERED ([SalesOrderID]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX SalesOrderHeader CustomerID] ON [Sales].[SalesOrder-
Header] ([CustomerID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK SalesOrderHeader rowguid] ON [Sales].[SalesOrder-
Header] ([rowguid]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK SalesOrderHeader SalesOrderNumber] ON
[Sales].[SalesOrderHeader] ([SalesOrderNumber]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX SalesOrderHeader SalesPersonID] ON [Sales].[SalesOrder-
Header] ([SalesPersonID]) ON [PRIMARY]
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [FK SalesOrderHeader Address Bill-
ToAddressID] FOREIGN KEY ([BillToAddressID]) REFERENCES [Person].[Address] ([Address-
GO
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [FK SalesOrderHeader Address Ship-
ToAddressID] FOREIGN KEY ([ShipToAddressID]) REFERENCES [Person].[Address] ([Address-
GO
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [FK SalesOrderHeader CreditCard -
CreditCardID] FOREIGN KEY ([CreditCardID]) REFERENCES [Sales].[CreditCard] ([CreditCard-
GO
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [FK SalesOrderHeader Currency-
Rate CurrencyRateID] FOREIGN KEY ([CurrencyRateID]) REFERENCES [Sales].[CurrencyRate]
([CurrencyRateID])
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [FK SalesOrderHeader Customer -
CustomerID] FOREIGN KEY ([CustomerID]) REFERENCES [Sales].[Customer] ([CustomerID])
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [FK SalesOrderHeader SalesPerson -
SalesPersonID] FOREIGN KEY ([SalesPersonID]) REFERENCES [Sales].[SalesPerson]
([BusinessEntityID])
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [FK SalesOrderHeader Sales-
Territory TerritoryID] FOREIGN KEY ([TerritoryID]) REFERENCES [Sales].[SalesTerritory]
([TerritoryID])
ALTER TABLE [Sales].[SalesOrderHeader] ADD CONSTRAINT [FK SalesOrderHeader ShipMethod -
ShipMethodID] FOREIGN KEY ([ShipMethodID]) REFERENCES [Purchasing].[ShipMethod] ([Ship-
MethodID1)
EXEC sp addextendedproperty N'MS Description', N'General sales order information.',
```

```
'SCHEMA', 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-
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.',
'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'COLUMN', N'Comment'
EXEC sp addextendedproperty N'MS Description', N'Approval code provided by the credit
card company.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'COLUMN', N'Credit-
CardApprovalCode'
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'
EXEC sp addextendedproperty N'MS Description', N'Currency exchange rate used. Foreign
key to CurrencyRate.CurrencyRateID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader',
'COLUMN', N'CurrencyRateID'
EXEC sp addextendedproperty N'MS Description', N'Customer identification number.
Foreign key to Customer.BusinessEntityID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrder-
Header', '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'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'COLUMN', N'ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'0 = Order placed by sales person. 1 =
Order placed online by customer.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader',
'COLUMN', N'OnlineOrderFlag'
EXEC sp addextendedproperty N'MS Description', N'Dates the sales order was created.',
'SCHEMA, N'Sales', 'TABLE', N'SalesOrderHeader', 'COLUMN', N'OrderDate'
EXEC sp addextendedproperty N'MS Description', N'Customer purchase order number
reference. ', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'COLUMN', N'Purchase-
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'rowquid'
EXEC sp addextendedproperty N'MS Description', N'Primary key.', 'SCHEMA', N'Sales',
'TABLE', N'SalesOrderHeader', 'COLUMN', N'SalesOrderID'
```

```
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'
EXEC sp addextendedproperty N'MS Description', N'Date the order was shipped to the
customer.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'COLUMN', N'ShipDate'
EXEC sp addextendedproperty N'MS_Description', N'Shipping method. Foreign key to Ship-
Method.ShipMethodID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'COLUMN',
N'ShipMethodID'
EXEC sp_addextendedproperty N'MS_Description', N'Customer shipping address. Foreign key
to Address.AddressID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'COLUMN',
N'ShipToAddressID'
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'
EXEC sp addextendedproperty N'MS Description', N'Sales subtotal. Computed as SUM(Sales-
OrderDetail.LineTotal) for the appropriate SalesOrderID.', 'SCHEMA', N'Sales', 'TABLE',
N'SalesOrderHeader', 'COLUMN', N'SubTotal'
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'Sales-
OrderHeader', '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] >= [Order-
Date]', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'CK SalesOrder-
Header 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] >= [Order-
Date] OR [ShipDate] IS NULL', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader',
'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 Sales-
OrderHeader Status'
GO
EXEC sp addextendedproperty N'MS Description', N'Check constraint [SubTotal] >=
(0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'CK Sales-
OrderHeader 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'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'DF Sales-
OrderHeader ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of 1 (TRUE)',
'SCHEMAT, N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'DF SalesOrderHeader
OnlineOrderFlag'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'DF Sales-
OrderHeader OrderDate'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of 0',
'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'DF SalesOrderHeader -
RevisionNumber'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of NEWID()',
'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'DF SalesOrder-
Header rowguid'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of 1',
'SCHEMA, N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'DF SalesOrderHeader -
Status'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of 0.0',
'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'DF SalesOrderHeader -
SubTotal'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of 0.0',
'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeader', 'CONSTRAINT', N'DF SalesOrderHeader -
TaxAmt'
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'
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'
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'
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'
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'
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'
```

```
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'
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'
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'
EXEC sp addextendedproperty N'MS Description', N'Unique nonclustered index.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesOrderHeader', 'INDEX', N'AK SalesOrderHeader SalesOrder-
GO
EXEC sp addextendedproperty N'MS Description', N'Nonclustered index.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesOrderHeader', 'INDEX', N'IX SalesOrderHeader CustomerID'
EXEC sp addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Sales, 'TABLE', N'SalesOrderHeader', 'INDEX', N'IX SalesOrderHeader SalesPersonID'
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_Sales-
OrderHeader 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 Sales-
YTD column in the SalesPerson and SalesTerritory tables.', 'SCHEMA', N'Sales', 'TABLE',
N'SalesOrderHeader', 'TRIGGER', N'uSalesOrderHeader'
GΩ
```

[Person].[Address]

[Purchasing].[ShipMethod]

[Sales].[CreditCard]

[Sales].[CurrencyRate]

[Sales].[Customer]

[Sales].[SalesPerson]

[Sales].[SalesTerritory]

[dbo].[AccountNumber]

[abo].[rtocountrame

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKP FKP	SalesOrderID Primary key. Foreign key to SalesOrder- Header.SalesOrderID.	int	4	NOT NULL	
PKC FKG	SalesReasonID Primary key. Foreign key to Sales- Reason.SalesReasonID.	int	4	NOT NULL	
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Key	Name	Key Columns	Unique
PK	PK_SalesOrderHeaderSalesReason_SalesOrderID_SalesReasonID Primary key (clustered) constraint	SalesOrderID, SalesReasonID	True

Foreign Keys

Name	Delete	Columns
FK_SalesOrderHeaderSalesReason_SalesOrderHeader_SalesOrderID Foreign key constraint referencing SalesOrderHeader.SalesOrderID.	Cascade	SalesOrderID- >[Sales].[SalesOrder- Header].[SalesOrderID]
FK_SalesOrderHeaderSalesReason_SalesReasonID Foreign key constraint referencing SalesReason.SalesReasonID.		SalesReasonID- >[Sales].[Sales- Reason].[SalesReason- ID]

```
CREATE TABLE [Sales].[SalesOrderHeaderSalesReason]
[SalesOrderID] [int] NOT NULL,
[SalesReasonID] [int] NOT NULL,
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF_SalesOrderHeaderSalesReason_Modified-
Date] DEFAULT (getdate())
) ON [PRIMARY]
ALTER TABLE [Sales].[SalesOrderHeaderSalesReason] ADD CONSTRAINT [PK SalesOrderHeader-
SalesReason SalesOrderID SalesReasonID] PRIMARY KEY CLUSTERED ([SalesOrderID], [Sales-
ReasonID]) ON [PRIMARY]
ALTER TABLE [Sales].[SalesOrderHeaderSalesReason] ADD CONSTRAINT [FK SalesOrderHeader-
SalesReason SalesOrderHeader SalesOrderID] FOREIGN KEY ([SalesOrderID]) REFERENCES
[Sales].[SalesOrderHeader] ([SalesOrderID]) ON DELETE CASCADE
ALTER TABLE [Sales].[SalesOrderHeaderSalesReason] ADD CONSTRAINT [FK SalesOrderHeader-
SalesReason SalesReason SalesReasonID] FOREIGN KEY ([SalesReasonID]) REFERENCES
[Sales].[SalesReason] ([SalesReasonID])
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
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeaderSalesReason', 'COLUMN',
N'ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Primary key. Foreign key to SalesOrder-
Header.SalesOrderID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeaderSalesReason',
'COLUMN', N'SalesOrderID'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Foreign key to Sales-
Reason.SalesReasonID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeaderSalesReason',
'COLUMN', N'SalesReasonID'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeaderSalesReason', 'CONSTRAINT',
N'DF SalesOrderHeaderSalesReason ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Foreign key constraint referencing
SalesOrderHeader.SalesOrderID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeaderSales-
Reason', 'CONSTRAINT', N'FK SalesOrderHeaderSalesReason SalesOrderHeader SalesOrderID'
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'
EXEC sp addextendedproperty N'MS_Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeaderSalesReason', 'CONSTRAINT', N'PK Sales-
OrderHeaderSalesReason SalesOrderID SalesReasonID'
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a primary
key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'SalesOrderHeaderSalesReason', 'INDEX',
N'PK SalesOrderHeaderSalesReason SalesOrderID SalesReasonID'
GO
```

[Sales].[SalesOrderHeader] [Sales].[SalesReason] Sales

■ [Sales].[SalesPerson]

MS_Description

Sales representative current information.

Properties

Property	Value
Row Count (~)	17
Created	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKO FKO	BusinessEntityID Primary key for SalesPerson records. Foreign key to Employee.BusinessEntity-ID	int	4	NOT NULL	
F	TerritoryID Territory currently assigned to. Foreign key to SalesTerritory.SalesTerritoryID.	int	4	NULL allowed	
■	SalesQuota Projected yearly sales.	money	8	NULL allowed	
	Bonus Bonus due if quota is met.	money	8	NOT NULL	((0.00))
	CommissionPct Commision percent received per sale.	smallmoney	4	NOT NULL	((0.00))
	SalesYTD Sales total year to date.	money	8	NOT NULL	((0.00))
	SalesLastYear Sales total of previous year.	money	8	NOT NULL	((0.00))
ф	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL	(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Key	Name	Key Columns	Unique
PK <mark>P</mark> C	PK_SalesPerson_BusinessEntityID Primary key (clustered) constraint	BusinessEntity- ID	True

AK_SalesPerson_rowguid	rowguid	True
Unique nonclustered index. Used to support replication samples.		

Check Constraints

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

Foreign Keys

Name	Columns
FK_SalesPerson_Employee_BusinessEntityID Foreign key constraint referencing Employee.Employee-ID.	BusinessEntityID->[Human- Resources].[Employee].[BusinessEntityID]
FK_SalesPerson_SalesTerritory_TerritoryID Foreign key constraint referencing SalesTerritory.Territory-ID.	TerritoryID->[Sales].[SalesTerritory].[TerritoryID]

```
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]
ALTER TABLE [Sales].[SalesPerson] ADD CONSTRAINT [CK SalesPerson Bonus] CHECK
(([Bonus]>=(0.00)))
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)))
ALTER TABLE [Sales].[SalesPerson] ADD CONSTRAINT [CK SalesPerson SalesQuota] CHECK
(([SalesQuota]>(0.00)))
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]
CREATE UNIQUE NONCLUSTERED INDEX [AK SalesPerson rowquid] ON [Sales].[SalesPerson]
([rowguid]) ON [PRIMARY]
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])
EXEC sp addextendedproperty N'MS Description', N'Sales representative current
information.', 'SCHEMA', N'Sales, 'TABLE', N'SalesPerson', NULL, NULL
EXEC sp addextendedproperty N'MS Description', N'Bonus due if quota is met.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesPerson', 'COLUMN', N'Bonus'
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'
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'
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'
EXEC sp_addextendedproperty N'MS_Description', N'Sales total of previous year.',
'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'COLUMN', N'SalesLastYear'
EXEC sp addextendedproperty N'MS_Description', N'Projected yearly sales.', 'SCHEMA',
N'Sales, 'TABLE', N'SalesPerson, 'COLUMN', N'SalesQuota'
EXEC sp addextendedproperty N'MS Description', N'Sales total year to date.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesPerson', 'COLUMN', N'SalesYTD'
EXEC sp addextendedproperty N'MS Description', N'Territory currently assigned to.
Foreign key to SalesTerritory.SalesTerritoryID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales-
Person', 'COLUMN', N'TerritoryID'
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [Bonus] >= (0.00)',
'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'CONSTRAINT', N'CK SalesPerson Bonus'
```

```
GO
EXEC sp addextendedproperty N'MS_Description', N'Check constraint [CommissionPct] >=
(0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'CONSTRAINT', N'CK SalesPerson -
CommissionPct'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [SalesLastYear] >=
(0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'CONSTRAINT', N'CK SalesPerson -
SalesLastYear'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [SalesQuota] >
(0.00), 'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'CONSTRAINT', N'CK SalesPerson -
SalesQuota'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [SalesYTD] >=
(0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'CONSTRAINT', N'CK SalesPerson -
SalesYTD'
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 Commission-
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_SalesPerson_rowguid'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'CONSTRAINT', N'DF SalesPerson SalesLast-
Year'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Sales', 'TABLE', N'SalesPerson', 'CONSTRAINT', N'DF SalesPerson SalesYTD'
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 SalesPerson Business-
EntityID'
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 Sales-
Person BusinessEntityID'
GΟ
```

[HumanResources].[Employee] [Sales].[SalesTerritory] Sales

Used By

[Sales].[SalesOrderHeader]
[Sales].[SalesPersonQuotaHistory]
[Sales].[SalesTerritoryHistory]
[Sales].[Store]
[Sales].[vSalesPerson]
[Sales].[vSalesPersonSalesByFiscalYears]

□ [Sales].[SalesPersonQuotaHistory]

MS_Description

Sales performance tracking.

Properties

Property	Value
Row Count (~)	163
Created	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKC FKD	BusinessEntityID Sales person identification number. Foreign key to SalesPerson.Business- EntityID.	int	4	NOT NULL	
PKP C	QuotaDate Sales quota date.	datetime	8	NOT NULL	
	SalesQuota Sales quota amount.	money	8	NOT NULL	
ф	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL	(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

K	Cey	Name	Key Columns	Unique
Pi	·C	PK_SalesPersonQuotaHistory_BusinessEntityID_QuotaDate Primary key (clustered) constraint	BusinessEntity- ID, QuotaDate	True
		AK_SalesPersonQuotaHistory_rowguid Unique nonclustered index. Used to support replication samples.	rowguid	True

Check Constraints

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

Foreign Keys

Name	Columns
FK_SalesPersonQuotaHistory_SalesPerson_Business- EntityID Foreign key constraint referencing SalesPerson.Sales- PersonID.	BusinessEntityID->[Sales].[SalesPerson].[BusinessEntity-ID]

```
CREATE TABLE [Sales].[SalesPersonQuotaHistory]
[BusinessEntityID] [int] NOT NULL,
[QuotaDate] [datetime] NOT NULL,
[SalesQuota] [money] NOT NULL,
[rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF SalesPersonQuota-
History rowguid] DEFAULT (newid()),
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF SalesPersonQuotaHistory ModifiedDate]
DEFAULT (getdate())
) ON [PRIMARY]
GO
ALTER TABLE [Sales].[SalesPersonQuotaHistory] ADD CONSTRAINT [CK SalesPersonQuota-
History SalesQuota] CHECK (([SalesQuota]>(0.00)))
ALTER TABLE [Sales].[SalesPersonQuotaHistory] ADD CONSTRAINT [PK SalesPersonQuota-
History BusinessEntityID QuotaDate] PRIMARY KEY CLUSTERED ([BusinessEntityID], [Quota-
Date]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK SalesPersonQuotaHistory rowguid] ON [Sales].[Sales-
PersonQuotaHistory] ([rowquid]) ON [PRIMARY]
ALTER TABLE [Sales].[SalesPersonQuotaHistory] ADD CONSTRAINT [FK SalesPersonQuota-
History_SalesPerson_BusinessEntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES
[Sales].[SalesPerson] ([BusinessEntityID])
EXEC sp addextendedproperty N'MS Description', N'Sales performance tracking.',
'SCHEMA', N'Sales', 'TABLE', N'SalesPersonQuotaHistory', NULL, NULL
EXEC sp_addextendedproperty N'MS_Description', N'Sales person identification number.
Foreign key to SalesPerson.BusinessEntityID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales-
PersonQuotaHistory', 'COLUMN', N'BusinessEntityID'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'SalesPersonQuotaHistory', 'COLUMN',
N'ModifiedDate'
EXEC sp_addextendedproperty N'MS_Description', N'Sales quota date.', 'SCHEMA',
N'Sales, 'TABLE', N'SalesPersonQuotaHistory', 'COLUMN', N'QuotaDate'
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'rowquid'
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'SalesPersonQuotaHistory',
'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 SalesPerson-
QuotaHistory BusinessEntityID QuotaDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'SalesPersonQuotaHistory',
'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
```

[Sales].[SalesPerson] Sales

Image: Sales | S

MS_Description

Lookup table of customer purchase reasons.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	10
Created	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
PK P G	SalesReasonID Primary key for SalesReason records.	int	4	NOT NULL	1 - 1	
	Name Sales reason description.	[dbo].[Name]	100	NOT NULL		
	ReasonType Category the sales reason belongs to.	[dbo].[Name]	100	NOT NULL		
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

Key	Name	Key Columns	Unique
PK	PK_SalesReason_SalesReasonID Primary key (clustered) constraint	SalesReasonID	True

```
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]
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'
EXEC sp addextendedproperty N'MS Description', N'Sales reason description.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesReason', 'COLUMN', N'Name'
EXEC sp addextendedproperty N'MS Description', N'Category the sales reason belongs
to.', 'SCHEMA', N'Sales', 'TABLE', N'SalesReason', 'COLUMN', N'ReasonType'
EXEC sp addextendedproperty N'MS Description', N'Primary key for SalesReason records.',
'SCHEMA', N'Sales', 'TABLE', N'SalesReason', 'COLUMN', N'SalesReasonID'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', TABLE', N'SalesReason', 'CONSTRAINT', N'DF Sales-
Reason ModifiedDate'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Sales', 'TABLE', N'SalesReason', 'CONSTRAINT', N'PK_SalesReason_SalesReason-
TD'
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index created by a primary
key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'SalesReason', 'INDEX', N'PK Sales-
Reason SalesReasonID'
```

[dbo].[Name] Sales

Used By

[Sales].[SalesOrderHeaderSalesReason]

Image: Sales Tax Rate Image: Image: Image: Image: Sales Tax Rate Image: Tax Rate Image:

MS_Description

Tax rate lookup table.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
Row Count (~)	29
Created	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullabilit y	Identity	Default
P <mark>/</mark> C	SalesTaxRateID Primary key for SalesTaxRate records.	int	4	NOT NULL	1 - 1	
FKP	StateProvinceID State, province, or country/region the sales tax applies to.	int	4	NOT NULL		
# []	TaxType 1 = Tax applied to retail transactions, 2 = Tax applied to wholesale transactions, 3 = Tax applied to all sales (retail and wholesale) transactions.	tinyint	1	NOT NULL		
	TaxRate Tax rate amount.	smallmoney	4	NOT NULL		((0.00))
	Name Tax rate description.	[dbo].[Name]	100	NOT NULL		
#	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL		(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

Key	Name	Key Columns	Unique
PK C	PK_SalesTaxRate_SalesTaxRateID	SalesTaxRate-	True

Primary key (clustered) constraint		ID	
AK_SalesTaxRate_StateProvinceID_ Unique nonclustered index.	ТахТуре	StateProvince- ID, TaxType	True
AK_SalesTaxRate_rowguid Unique nonclustered index. Used to	support replication samples.	rowguid	True

Check Constraints

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

Foreign Keys

Name	Columns
FK_SalesTaxRate_StateProvince_StateProvinceID Foreign key constraint referencing StateProvince.State- ProvinceID.	StateProvinceID->[Person].[StateProvince].[State-ProvinceID]

```
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 SalesTaxRate rowguid]
DEFAULT (newid()),
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF SalesTaxRate ModifiedDate] DEFAULT
(getdate())
) ON [PRIMARY]
ALTER TABLE [Sales].[SalesTaxRate] ADD CONSTRAINT [CK SalesTaxRate TaxType] CHECK
(([TaxType]>=(1) AND [TaxType]<=(3)))
ALTER TABLE [Sales].[SalesTaxRate] ADD CONSTRAINT [PK SalesTaxRate SalesTaxRateID]
PRIMARY KEY CLUSTERED ([SalesTaxRateID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK SalesTaxRate rowguid] ON [Sales].[SalesTaxRate]
([rowguid]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK SalesTaxRate StateProvinceID TaxType] ON
[Sales].[SalesTaxRate] ([StateProvinceID], [TaxType]) ON [PRIMARY]
ALTER TABLE [Sales].[SalesTaxRate] ADD CONSTRAINT [FK SalesTaxRate StateProvince State-
ProvinceID] FOREIGN KEY ([StateProvinceID]) REFERENCES [Person].[StateProvince] ([State-
ProvinceID])
EXEC sp_addextendedproperty N'MS_Description', N'Tax rate lookup table.', 'SCHEMA',
```

```
N'Sales', 'TABLE', N'SalesTaxRate', NULL, NULL
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTaxRate', 'COLUMN', N'ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Tax rate description.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesTaxRate', 'COLUMN', N'Name'
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'
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'State-
ProvinceID'
EXEC sp addextendedproperty N'MS Description', N'Tax rate amount.', 'SCHEMA', N'Sales',
'TABLE', N'SalesTaxRate', 'COLUMN', N'TaxRate'
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'SalesTaxRate',
'COLUMN', N'TaxType'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [TaxType] BETWEEN (1)
AND (3), 'SCHEMA', N'Sales', 'TABLE', N'SalesTaxRate', 'CONSTRAINT', N'CK SalesTax-
Rate TaxType'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'SalesTaxRate', 'CONSTRAINT', N'DF SalesTax-
Rate ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of NEWID()',
'SCHEMAT, N'Sales', 'TABLE', N'SalesTaxRate', 'CONSTRAINT', N'DF SalesTaxRate rowguid'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA', N'Sales', 'TABLE', N'SalesTaxRate', 'CONSTRAINT', N'DF_SalesTaxRate_TaxRate'
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'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Sales', 'TABLE', N'SalesTaxRate', 'CONSTRAINT', N'PK_SalesTaxRate_SalesTax-
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'
EXEC sp addextendedproperty N'MS Description', N'Unique nonclustered index.', 'SCHEMA',
N'Sales', 'TABLE', N'SalesTaxRate', 'INDEX', N'AK_SalesTaxRate_StateProvinceID TaxType'
EXEC sp addextendedproperty N'MS Description', N'Clustered index created by a primary
key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTaxRate', 'INDEX', N'PK SalesTax-
Rate_SalesTaxRateID'
```

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
PK <mark>P</mark> C	TerritoryID Primary key for SalesTerritory records.	int	4	NOT NULL	1 - 1	
<u>.</u>	Name Sales territory description	[dbo].[Name]	100	NOT NULL		
F/P	CountryRegionCode ISO standard country or region code. Foreign key to Country- Region.CountryRegionCode.	nvarchar(3)	6	NOT NULL		
	Group Geographic area to which the sales territory belong.	nvarchar(50)	100	NOT NULL		
■	SalesYTD Sales in the territory year to date.	money	8	NOT NULL		((0.00))
	SalesLastYear Sales in the territory the previous year.	money	8	NOT NULL		((0.00))
	CostYTD Business costs in the territory year to date.	money	8	NOT NULL		((0.00))
	CostLastYear Business costs in the territory the previous year.	money	8	NOT NULL		((0.00))
. 	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL		(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

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

Check Constraints

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

Foreign Keys

Name	Columns
FK_SalesTerritory_CountryRegion_CountryRegionCode Foreign key constraint referencing Country- Region.CountryRegionCode.	CountryRegionCode->[Person].[CountryRegion].[Country-RegionCode]

```
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_SalesTerritory_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)))
ALTER TABLE [Sales].[SalesTerritory] ADD CONSTRAINT [CK SalesTerritory CostYTD] CHECK
(([CostYTD]>=(0.00)))
ALTER TABLE [Sales].[SalesTerritory] ADD CONSTRAINT [CK SalesTerritory SalesLastYear]
CHECK (([SalesLastYear]>=(0.00)))
ALTER TABLE [Sales].[SalesTerritory] ADD CONSTRAINT [CK SalesTerritory SalesYTD] CHECK
(([SalesYTD]>=(0.00)))
ALTER TABLE [Sales].[SalesTerritory] ADD CONSTRAINT [PK SalesTerritory TerritoryID]
PRIMARY KEY CLUSTERED ([TerritoryID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK SalesTerritory Name] ON [Sales].[SalesTerritory]
([Name]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK SalesTerritory rowquid] ON [Sales].[Sales-
Territory] ([rowguid]) ON [PRIMARY]
ALTER TABLE [Sales].[SalesTerritory] ADD CONSTRAINT [FK SalesTerritory CountryRegion -
CountryRegionCode] FOREIGN KEY ([CountryRegionCode]) REFERENCES [Person].[Country-
Region] ([CountryRegionCode])
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'CostLast-
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'
EXEC sp addextendedproperty N'MS Description', N'ISO standard country or region code.
Foreign key to CountryRegion.CountryRegionCode. ', 'SCHEMA', N'Sales', 'TABLE', N'Sales-
Territory', 'COLUMN', N'CountryRegionCode'
EXEC sp addextendedproperty N'MS Description', N'Geographic area to which the sales
territory belong.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'COLUMN', N'Group'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'COLUMN', N'ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Sales territory description',
'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'COLUMN', N'Name
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'
EXEC sp addextendedproperty N'MS Description', N'Sales in the territory the previous
year.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'COLUMN', N'SalesLastYear'
EXEC sp addextendedproperty N'MS Description', N'Sales in the territory year to date.',
'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'COLUMN', N'SalesYTD'
```

```
EXEC sp addextendedproperty N'MS Description', N'Primary key for SalesTerritory
records.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'COLUMN', N'TerritoryID'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [CostLastYear] >=
(0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'CK Sales-
Territory_CostLastYear'
EXEC sp addextendedproperty N'MS_Description', N'Check constraint [CostYTD] >= (0.00)',
'SCHEMA, N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'CK SalesTerritory Cost-
YTD'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [SalesLastYear] >=
(0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'CK Sales-
Territory SalesLastYear'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [SalesYTD] >=
(0.00)', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'CK Sales-
Territory SalesYTD'
{\tt EXEC} \  \, {\tt sp} \  \, {\tt addextended property} \  \, {\tt N'MS\_Description'}, \  \, {\tt N'Default} \  \, {\tt constraint} \  \, {\tt value} \  \, {\tt of} \  \, {\tt 0.0'}, \\
'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'DF SalesTerritory Cost-
LastYear'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of 0.0',
'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'DF SalesTerritory Cost-
YTD'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'DF Sales-
Territory_ModifiedDate'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of NEWID()',
'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'DF Sales-
Territory_rowguid'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of 0.0',
'SCHEMAT, N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'DF SalesTerritory Sales-
LastYear'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA, N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'DF SalesTerritory Sales-
YTD'
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'
EXEC sp addextendedproperty N'MS Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory', 'CONSTRAINT', N'PK SalesTerritory -
TerritoryID'
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 rowquid'
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_Sales-
Territory 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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKC FKD	BusinessEntityID Primary key. The sales rep. Foreign key to SalesPerson.BusinessEntityID.	int	4	NOT NULL	
PKP C FKP	TerritoryID Primary key. Territory identification number. Foreign key to Sales- Territory.SalesTerritoryID.	int	4	NOT NULL	
PK <mark>P</mark> C	StartDate Primary key. Date the sales representive started work in the territory.	datetime	8	NOT NULL	
	EndDate Date the sales representative left work in the territory.	datetime	8	NULL allowed	
#	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL	(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

Key	Name	Key Columns	Unique
₽ <mark>≯</mark> G	PK_SalesTerritoryHistory_BusinessEntityID_StartDate_TerritoryID Primary key (clustered) constraint	BusinessEntity- ID, StartDate, TerritoryID	True
	AK_SalesTerritoryHistory_rowguid Unique nonclustered index. Used to support replication samples.	rowguid	True

Check Constraints

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

Foreign Keys

Name	Columns
FK_SalesTerritoryHistory_SalesPerson_BusinessEntityID Foreign key constraint referencing SalesPerson.Sales-PersonID.	BusinessEntityID->[Sales].[SalesPerson].[BusinessEntity-ID]
FK_SalesTerritoryHistory_SalesTerritory_TerritoryID Foreign key constraint referencing SalesTerritory.Territory-ID.	TerritoryID->[Sales].[SalesTerritory].[TerritoryID]

```
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 SalesTerritory-
History_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))
ALTER TABLE [Sales].[SalesTerritoryHistory] ADD CONSTRAINT [PK SalesTerritoryHistory -
BusinessEntityID StartDate TerritoryID] PRIMARY KEY CLUSTERED ([BusinessEntityID],
[StartDate], [TerritoryID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK SalesTerritoryHistory rowguid] ON [Sales].[Sales-
TerritoryHistory] ([rowguid]) ON [PRIMARY]
ALTER TABLE [Sales].[SalesTerritoryHistory] ADD CONSTRAINT [FK SalesTerritoryHistory -
SalesPerson BusinessEntityID] FOREIGN KEY ([BusinessEntityID]) REFERENCES
[Sales].[SalesPerson] ([BusinessEntityID])
ALTER TABLE [Sales].[SalesTerritoryHistory] ADD CONSTRAINT [FK SalesTerritoryHistory -
SalesTerritory_TerritoryID] FOREIGN KEY ([TerritoryID]) REFERENCES [Sales].[Sales-
Territory] ([TerritoryID])
EXEC sp_addextendedproperty N'MS_Description', N'Sales representative transfers to
other sales territories.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritoryHistory', NULL,
NULL
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. The sales rep. Foreign
```

```
key to SalesPerson.BusinessEntityID.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritory-
History', 'COLUMN', N'BusinessEntityID'
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'
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritoryHistory', 'COLUMN', N'Modified-
Date'
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'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key. Date the sales
representive started work in the territory.', 'SCHEMA', N'Sales', 'TABLE', N'Sales-
TerritoryHistory', 'COLUMN', N'StartDate'
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'
EXEC sp addextendedproperty N'MS_Description', N'Check constraint [EndDate] >= [Start-
Date] OR [EndDate] IS NULL', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritoryHistory',
'CONSTRAINT', N'CK SalesTerritoryHistory EndDate'
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritoryHistory', 'CONSTRAINT', N'DF -
SalesTerritoryHistory ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of NEWID()',
'SCHEMA, N'Sales', 'TABLE', N'SalesTerritoryHistory', 'CONSTRAINT', N'DF Sales-
TerritoryHistory rowguid'
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'
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'
EXEC sp addextendedproperty N'MS Description', N'Primary key (clustered) constraint',
'SCHEMA, N'Sales', 'TABLE', N'SalesTerritoryHistory', 'CONSTRAINT', N'PK Sales-
TerritoryHistory_BusinessEntityID_StartDate TerritoryID'
EXEC sp addextendedproperty N'MS Description', N'Unique nonclustered index. Used to
support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'SalesTerritoryHistory',
'INDEX', N'AK SalesTerritoryHistory rowguid'
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'
```

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

III [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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
PK <mark>P</mark> C	ShoppingCartItemID Primary key for ShoppingCartItem records.	int	4	NOT NULL	1 - 1	
. 	ShoppingCartID Shopping cart identification number.	nvarchar(50)	100	NOT NULL		
	Quantity Product quantity ordered.	int	4	NOT NULL		((1))
F P	ProductID Product ordered. Foreign key to Product.ProductID.	int	4	NOT NULL		
	DateCreated Date the time the record was created.	datetime	8	NOT NULL		(getdate())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

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

Check Constraints

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

Foreign Keys

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

```
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)))
ALTER TABLE [Sales]. [ShoppingCartItem] ADD CONSTRAINT [PK ShoppingCartItem ShoppingCart-
ItemID] PRIMARY KEY CLUSTERED ([ShoppingCartItemID]) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [IX ShoppingCartItem ShoppingCartID ProductID] ON
[Sales].[ShoppingCartItem] ([ShoppingCartID], [ProductID]) ON [PRIMARY]
ALTER TABLE [Sales].[ShoppingCartItem] ADD CONSTRAINT [FK ShoppingCartItem Product -
ProductID] FOREIGN KEY ([ProductID]) REFERENCES [Production].[Product] ([ProductID])
EXEC sp addextendedproperty N'MS Description', N'Contains online customer orders until
the order is submitted or cancelled.', 'SCHEMA', N'Sales', 'TABLE', N'ShoppingCart-
Item', NULL, NULL
EXEC sp addextendedproperty N'MS Description', N'Date the time the record was
created.', 'SCHEMA', N'Sales', 'TABLE', N'ShoppingCartItem', 'COLUMN', N'DateCreated'
GO
EXEC sp addextendedproperty N'MS Description', N'Date and time the record was last
updated.', 'SCHEMA', N'Sales', 'TABLE', N'ShoppingCartItem', 'COLUMN', N'ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Product ordered. Foreign key to
Product.ProductID.', 'SCHEMA', N'Sales', 'TABLE', N'ShoppingCartItem', 'COLUMN',
N'ProductID'
EXEC sp_addextendedproperty N'MS_Description', N'Product quantity ordered.', 'SCHEMA',
```

```
N'Sales', 'TABLE', N'ShoppingCartItem', 'COLUMN', N'Quantity'
EXEC sp addextendedproperty N'MS Description', N'Shopping cart identification number.',
'SCHEMA', N'Sales', 'TABLE', N'ShoppingCartItem', 'COLUMN', N'ShoppingCartID'
EXEC sp addextendedproperty N'MS Description', N'Primary key for ShoppingCartItem
records.', 'SCHEMA', N'Sales', 'TABLE', N'ShoppingCartItem', 'COLUMN', N'ShoppingCart-
ItemID'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [Quantity] >= (1)',
'SCHEMA', N'Sales', 'TABLE', N'ShoppingCartItem', 'CONSTRAINT', N'CK ShoppingCartItem -
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'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of
GETDATE()', 'SCHEMA', N'Sales', TABLE', N'ShoppingCartItem', 'CONSTRAINT', N'DF_-
ShoppingCartItem ModifiedDate'
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of 1',
'SCHEMA, N'Sales', 'TABLE', N'ShoppingCartItem', 'CONSTRAINT', N'DF_ShoppingCartItem_-
Quantity'
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'
EXEC sp addextendedproperty N'MS Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Sales', 'TABLE', N'ShoppingCartItem', 'CONSTRAINT', N'PK_ShoppingCartItem_-ShoppingCartItemID'
EXEC sp addextendedproperty N'MS Description', N'Nonclustered index.', 'SCHEMA',
N'Sales, 'TABLE', N'ShoppingCartItem', 'INDEX', N'IX_ShoppingCartItem_ShoppingCartID_-
ProductID'
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
```

[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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity	Default
PK P C	SpecialOfferID Primary key for SpecialOffer records.	int	4	NOT NULL	1 - 1	
	Description Discount description.	nvarchar(255)	510	NOT NULL		
	DiscountPct Discount precentage.	smallmoney	4	NOT NULL		((0.00))
	Type Discount type category.	nvarchar(50)	100	NOT NULL		
	Category Group the discount applies to such as Reseller or Customer.	nvarchar(50)	100	NOT NULL		
	StartDate Discount start date.	datetime	8	NOT NULL		
	EndDate Discount end date.	datetime	8	NOT NULL		
	MinQty Minimum discount percent allowed.	int	4	NOT NULL		((0))
■	MaxQty Maximum discount percent allowed.	int	4	NULL allowed		
ф	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL		(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL		(getdate())

Indexes

Key	Name	Key Columns	Unique
PK C	PK_SpecialOffer_SpecialOfferID Primary key (clustered) constraint	SpecialOfferID	True
	AK_SpecialOffer_rowguid Unique nonclustered index. Used to support replication samples.	rowguid	True

Check Constraints

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

```
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)))
ALTER TABLE [Sales].[SpecialOffer] ADD CONSTRAINT [CK_SpecialOffer_EndDate] CHECK
(([EndDate]>=[StartDate]))
ALTER TABLE [Sales].[SpecialOffer] ADD CONSTRAINT [CK SpecialOffer MaxQty] CHECK (([Max-
Qty] >= (0))
ALTER TABLE [Sales].[SpecialOffer] ADD CONSTRAINT [CK SpecialOffer MinQty] CHECK (([Min-
Qty] >= (0))
```

```
ALTER TABLE [Sales].[SpecialOffer] ADD CONSTRAINT [PK SpecialOffer SpecialOfferID]
PRIMARY KEY CLUSTERED ([SpecialOfferID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK SpecialOffer rowguid] ON [Sales].[SpecialOffer]
([rowquid]) ON [PRIMARY]
EXEC sp addextendedproperty N'MS Description', N'Sale discounts lookup table.',
'SCHEMA', N'Sales', 'TABLE', N'SpecialOffer', NULL, NULL
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'
EXEC sp addextendedproperty N'MS Description', N'Discount precentage.', 'SCHEMA',
N'Sales', 'TABLE', N'SpecialOffer', 'COLUMN', N'DiscountPct'
EXEC sp addextendedproperty N'MS Description', N'Discount end date.', 'SCHEMA',
N'Sales', 'TABLE', N'SpecialOffer', 'COLUMN', N'EndDate'
EXEC sp_addextendedproperty N'MS_Description', N'Maximum discount percent allowed.',
'SCHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'COLUMN', N'MaxQty'
EXEC sp addextendedproperty N'MS Description', N'Minimum discount percent allowed.',
'SCHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'COLUMN', N'MinQty'
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'
EXEC sp addextendedproperty N'MS Description', N'Discount start date.', 'SCHEMA',
N'Sales', 'TABLE', N'SpecialOffer', 'COLUMN', N'StartDate'
EXEC sp addextendedproperty N'MS Description', N'Discount type category.', 'SCHEMA',
N'Sales', 'TABLE', N'SpecialOffer', 'COLUMN', N'Type'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [DiscountPct] >=
(0.00), 'SCHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'CONSTRAINT', N'CK SpecialOffer -
DiscountPct'
EXEC sp addextendedproperty N'MS Description', N'Check constraint [EndDate] >= [Start-
Date]', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'CONSTRAINT', N'CK SpecialOffer -
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [MaxQty] >= (0)',
'SCHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'CONSTRAINT', N'CK SpecialOffer MaxQty'
EXEC sp_addextendedproperty N'MS_Description', N'Check constraint [MinQty] >= (0)',
'SCHEMA, N'Sales', 'TABLE', N'SpecialOffer', 'CONSTRAINT', N'CK_SpecialOffer_MinQty'
GO
EXEC sp addextendedproperty N'MS Description', N'Default constraint value of 0.0',
```

```
'SCHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'CONSTRAINT', N'DF SpecialOffer Discount-
Pct. '
GO
EXEC sp_addextendedproperty N'MS_Description', N'Default constraint value of 0.0',
'SCHEMA, N'Sales', 'TABLE', N'SpecialOffer', 'CONSTRAINT', N'DF_SpecialOffer_MinQty'
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_SpecialOffer_rowguid'
EXEC sp_addextendedproperty N'MS_Description', N'Primary key (clustered) constraint',
'SCHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'CONSTRAINT', N'PK_SpecialOffer_Special-
OfferID'
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'
EXEC sp addextendedproperty N'MS Description', N'Clustered index created by a primary
key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOffer', 'INDEX', N'PK Special-
Offer SpecialOfferID'
GO
```

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	17:24:40 2017. október 19., csütörtök
Last Modified	17:24:51 2017. október 19., csütörtök

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Default
PKEFKE	SpecialOfferID Primary key for SpecialOfferProduct records.	int	4	NOT NULL	
PKC	ProductID Product identification number. Foreign key to Product.ProductID.	int	4	NOT NULL	
.	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL	(newid())
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdate())

Indexes

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

Foreign Keys

Name	Columns	
FK_SpecialOfferProduct_Product_ProductID	ProductID->[Production].[Product].[ProductID]	

Foreign key constraint referencing Product.ProductID.	
FK_SpecialOfferProduct_SpecialOffer_SpecialOfferID Foreign key constraint referencing SpecialOffer.Special-OfferID.	SpecialOfferID->[Sales].[SpecialOffer].[SpecialOfferID]

```
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]
ALTER TABLE [Sales].[SpecialOfferProduct] ADD CONSTRAINT [PK SpecialOfferProduct -
SpecialOfferID ProductID] PRIMARY KEY CLUSTERED ([SpecialOfferID], [ProductID]) ON
CREATE NONCLUSTERED INDEX [IX SpecialOfferProduct ProductID] ON [Sales].[SpecialOffer-
Product] ([ProductID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK SpecialOfferProduct rowquid] ON [Sales]. [Special-
OfferProduct] ([rowguid]) ON [PRIMARY]
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].[Special-
Offer] ([SpecialOfferID])
GO
EXEC sp addextendedproperty N'MS Description', N'Cross-reference table mapping products
to special offer discounts.', 'SCHEMA', N'Sales', 'TABLE', N'SpecialOfferProduct',
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'SpecialOfferProduct',
'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 SpecialOffer-
Product rowquid'
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 SpecialOffer-
Product 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 rowquid'
EXEC sp_addextendedproperty N'MS_Description', N'Nonclustered index.', 'SCHEMA',
N'Sales, 'TABLE', N'SpecialOfferProduct', 'INDEX', N'IX_SpecialOfferProduct_ProductID'
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
```

[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	17:24:40 2017. október 19., csütörtök	
Last Modified	17:24:51 2017. október 19., csütörtök	

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullabil ity	Default
PKO FKO	BusinessEntityID Primary key. Foreign key to Customer.BusinessEntityID.	int	4	NOT NULL	
	Name Name of the store.	[dbo].[Name]	100	NOT NULL	
F 19	SalesPersonID ID of the sales person assigned to the customer. Foreign key to Sales-Person.BusinessEntityID.	int	4	NULL allowed	
ф	Demographics Demographic informationg about the store such as the number of employees, annual sales and store type.	xml([Sales].[StoreSurvey- SchemaCollection])	max	NULL allowed	
#	rowguid ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.	uniqueidentifier	16	NOT NULL	(newid()
	ModifiedDate Date and time the record was last updated.	datetime	8	NOT NULL	(getdat e())

Indexes

Key	Name	Key Columns	Туре	Unique	XML Type
PK	PK_Store_BusinessEntityID Primary key (clustered) constraint	Business- EntityID		True	
	AK_Store_rowguid Unique nonclustered index. Used to support replication	rowguid		True	

samples.			
IX_Store_SalesPersonID Nonclustered index.	SalesPerson- ID		
PXML_Store_Demographics Primary XML index.	Demographic s	xml	Primary

Foreign Keys

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

```
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]
GO
ALTER TABLE [Sales].[Store] ADD CONSTRAINT [PK Store BusinessEntityID] PRIMARY KEY
CLUSTERED ([BusinessEntityID]) ON [PRIMARY]
CREATE UNIQUE NONCLUSTERED INDEX [AK_Store_rowguid] ON [Sales].[Store] ([rowguid]) ON
[PRIMARY]
CREATE NONCLUSTERED INDEX [IX Store SalesPersonID] ON [Sales].[Store] ([SalesPersonID])
ON [PRIMARY]
CREATE PRIMARY XML INDEX [PXML Store Demographics]
ON [Sales].[Store] ([Demographics])
ALTER TABLE [Sales].[Store] ADD CONSTRAINT [FK_Store_BusinessEntity_BusinessEntityID]
FOREIGN KEY ([BusinessEntityID]) REFERENCES [Person].[BusinessEntity] ([BusinessEntity-
ID])
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
EXEC sp addextendedproperty N'MS Description', N'Primary key. Foreign key to
```

```
Customer.BusinessEntityID.', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'COLUMN',
N'BusinessEntityID'
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'
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'
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'
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'
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 rowquid'
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'
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 rowquid'
EXEC sp addextendedproperty N'MS Description', N'Nonclustered index.', 'SCHEMA',
N'Sales', 'TABLE', N'Store', 'INDEX', N'IX Store SalesPersonID'
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 Business-
EntityID'
EXEC sp addextendedproperty N'MS Description', N'Primary XML index.', 'SCHEMA',
N'Sales', 'TABLE', N'Store', 'INDEX', N'PXML Store Demographics'
GO
```

[Person].[BusinessEntity]

[Sales].[SalesPerson]

[dbo].[Name]

Sales

[Sales].[StoreSurveySchemaCollection]

Used By

[Sales].[Customer]

[Sales].[vStoreWithAddresses]

[Sales].[vStoreWithContacts]

[Sales].[vStoreWithDemographics]

[dbo].[ufnGetContactInformation]

■ Views

Objects

Name

HumanResources.vEmployee

Employee names and addresses.

HumanResources.vEmployeeDepartment

Returns employee name, title, and current department.

HumanResources.vEmployeeDepartmentHistory

Returns employee name and current and previous departments.

HumanResources.vJobCandidate

Job candidate names and resumes.

HumanResources.vJobCandidateEducation

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.

HumanResources.vJobCandidateEmployment

Displays the content from each employement 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.

Person.vAdditionalContactInfo

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

Person.vStateProvinceCountryRegion

Joins StateProvince table with CountryRegion table.

Production.vProductAndDescription

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

Product Ion. v Product Model Catalog Description

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

Production.vProductModelInstructions

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

Purchasing.vVendorWithAddresses

Vendor (company) names and addresses .

Purchasing.vVendorWithContacts

Vendor (company) names and the names of vendor employees to contact.

Sales.vIndividualCustomer

Individual customers (names and addresses) that purchase Adventure Works Cycles products online.

Sales.vPersonDemographics

Displays the content from each element in the xml column Demographics for each customer in the Person Person table

Sales.vSalesPerson

Sales representiatives (names and addresses) and their sales-related information.

Sales.vSalesPersonSalesByFiscalYears

Uses PIVOT to return aggregated sales information for each sales representative.

Sales.vStoreWithAddresses

Stores (including store addresses) that sell Adventure Works Cycles products to consumers.

Sales.vStoreWithContacts

Stores (including store contacts) that sell Adventure Works Cycles products to consumers.

Sales.vStoreWithDemographics
Stores (including demographics) that sell Adventure Works Cycles products to consumers.

[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	17:24:52 2017. október 19., csütörtök
Last Modified	17:29:01 2021. március 21., vasárnap

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

```
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.',
'SCHEMA', N'HumanResources', 'VIEW', N'vEmployee', NULL, NULL
GO
```

```
[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	17:24:52 2017. október 19., csütörtök
Last Modified	17:29:01 2021. március 21., vasárnap

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

```
CREATE VIEW [HumanResources].[vEmployeeDepartment]

AS

SELECT

e.[BusinessEntityID]
,p.[Title]
,p.[FirstName]
,p.[MiddleName]
,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
```

[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	17:24:52 2017. október 19., csütörtök
Last Modified	17:29:01 2021. március 21., vasárnap

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

```
CREATE VIEW [HumanResources].[vEmployeeDepartmentHistory]

AS

SELECT

e.[BusinessEntityID]

,p.[Title]

,p.[FirstName]

,p.[MiddleName]

,p.[LastName]

,p.[Suffix]
```

Project > localhost > User databases > AdventureWorks2016 > Views > HumanResources.vEmployeeDepartment-History

```
,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'vEmployeeDepartment-
History', 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	17:24:52 2017. október 19., csütörtök
Last Modified	17:24:52 2017. október 19., csütörtök

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	

```
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.',
'SCHEMA', N'HumanResources', 'VIEW', N'vJobCandidate', NULL, NULL
GO
```

[HumanResources].[JobCandidate] HumanResources

[HumanResources].[vJobCandidateEducation]

MS_Description

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.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Created	17:24:52 2017. október 19., csütörtök
Last Modified	17:24:52 2017. október 19., csütörtök

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	

```
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. 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'vJobCandidateEducation', NULL, NULL
GO
```

[HumanResources].[JobCandidate] HumanResources

[HumanResources].[vJobCandidateEmployment]

MS_Description

Displays the content from each employement 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	17:24:52 2017. október 19., csütörtök
Last Modified	17:24:52 2017. október 19., csütörtök

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	

```
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
employement history 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'vJobCandidateEmployment', NULL,
GO
```

[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	17:24:52 2017. október 19., csütörtök
Last Modified	17:29:01 2021. március 21., vasárnap

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-
Provincel
    ,[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
\verb|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-
Regionl
    ,[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
\verb|ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";|
       declare namespace
(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;
EXEC sp addextendedproperty N'MS Description', N'Displays the contact name and content
from each element in the xml column AdditionalContactInfo for that person.', 'SCHEMA',
N'Person', 'VIEW', N'vAdditionalContactInfo', NULL, NULL
GO
```

[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	17:24:52 2017. október 19., csütörtök
Last Modified	17:24:52 2017. október 19., csütörtök

Columns

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

Indexes

Key	Name	Key Columns	Unique
<i>></i>	IX_vStateProvinceCountryRegion Clustered index on the view vStateProvinceCountryRegion.	StateProvinceID, CountryRegion- Code	True

```
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]
EXEC sp addextendedproperty N'MS Description', N'Joins StateProvince table with Country-
Region table.', 'SCHEMA', N'Person', 'VIEW', N'vStateProvinceCountryRegion', NULL, NULL
EXEC sp_addextendedproperty N'MS_Description', N'Clustered index on the view vState-ProvinceCountryRegion.', 'SCHEMA', N'Person', 'VIEW', N'vStateProvinceCountryRegion',
'INDEX', N'IX vStateProvinceCountryRegion'
```

[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	17:24:52 2017. október 19., csütörtök
Last Modified	17:24:52 2017. október 19., csütörtök

Columns

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

Indexes

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

```
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].[vProductAnd-
Description] ([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 vProduct-
AndDescription.', 'SCHEMA', N'Production', 'VIEW', N'vProductAndDescription', 'INDEX',
N'IX vProductAndDescription'
GO
```

[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.Product-Model table that has catalog data.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Created	17:24:52 2017. október 19., csütörtök
Last Modified	17:24:52 2017. október 19., csütörtök

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	

```
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";
        (/pl:ProductDescription/pl:Summary/html:p) [1]', 'nvarchar(max)') AS [Summary]
    , [CatalogDescription].value(N'declare namespace
pl="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";
        (/pl:ProductDescription/pl:Manufacturer/pl: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/ProductModelWarrAndMain";
        (/p1:ProductDescription/p1:Features/wm:Warranty/wm:WarrantyPeriod) [1]',
'nvarchar(256)') AS [WarrantyPeriod]
    , [CatalogDescription].value(N'declare namespace
pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
        declare namespace wm="http://schemas.microsoft.com/sqlserver/2004/07/adventure-
works/ProductModelWarrAndMain";
        (/pl:ProductDescription/pl: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/ProductModelWarrAndMain";
        (/p1:ProductDescription/p1:Features/wm:Maintenance/wm:NoOfYears) [1]',
'nvarchar(256)') AS [NoOfYears]
    ,[CatalogDescription].value(N'declare namespace
pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
```

```
declare namespace wm="http://schemas.microsoft.com/sqlserver/2004/07/adventure-
works/ProductModelWarrAndMain";
             (/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 [Picture-
Anglel
      ,[CatalogDescription].value(N'declare namespace
\verb|pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-pl="http://schemas.microsoft.com/sqlserver
Description";
             (/pl:ProductDescription/pl:Picture/pl:Size)[1]', 'nvarchar(256)') AS [Picture-
Size]
      ,[CatalogDescription].value(N'declare namespace
p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
             (/pl:ProductDescription/pl:Picture/pl:ProductPhotoID)[1]', 'nvarchar(256)') AS
[ProductPhotoID]
      ,[CatalogDescription].value(N'declare namespace
pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description";
            (/pl:ProductDescription/pl: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";
             (/pl:ProductDescription/pl:Specifications/ProductLine)[1]', 'nvarchar(256)') AS
```

Project > localhost > User databases > AdventureWorks2016 > Views > Production.vProductModelCatalog-Description

```
[ProductLine]
    ,[CatalogDescription].value(N'declare namespace
pl="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
pl="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;
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'vProductModelCatalog-
Description', NULL, NULL
```

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.ProductModel table that has manufacturing instructions.

Properties

Property	Value
Collation	SQL_Latin1_General_CP1_CI_AS
ANSI Nulls On	True
Quoted Identifier On	True
Created	17:24:52 2017. október 19., csütörtök
Last Modified	17:24:52 2017. október 19., csütörtök

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	

```
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 [MachineHours]
           , [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
\verb|"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/ProductModelManu-works/
Instructions";
            /root/Location') MfgInstructions(ref)
CROSS APPLY [MfqInstructions].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.ProductModel table that has manufacturing instructions.', 'SCHEMA', N'Production', 'VIEW', N'vProductModel-
Instructions', NULL, NULL
GO
```

[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	17:24:52 2017. október 19., csütörtök
Last Modified	17:24:52 2017. október 19., csütörtök

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

```
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
```

[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	17:24:52 2017. október 19., csütörtök
Last Modified	17:29:01 2021. március 21., vasárnap

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

```
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'vVendorWith-
Contacts', NULL, NULL
```

[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	17:24:52 2017. október 19., csütörtök
Last Modified	17:29:01 2021. március 21., vasárnap

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

```
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
```

[Person].[Address] [Person].[AddressType] [Person].[BusinessEntityAddress] [Person].[CountryRegion]

[Person].[EmailAddress]

[Person].[Person]

[Person].[PersonPhone]

[Person].[PhoneNumberType]

[Person].[StateProvince]

[Sales].[Customer]

[dbo].[Name]

[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	17:24:52 2017. október 19., csütörtök
Last Modified	17:29:01 2021. március 21., vasárnap

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

```
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 [DateFirstPurchase]
    , 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;
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
```

[Person].[Person] Sales

[Sales].[vSalesPerson]

MS_Description

Sales representiatives (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	17:24:52 2017. október 19., csütörtök
Last Modified	17:29:01 2021. március 21., vasárnap

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

```
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];
EXEC sp_addextendedproperty N'MS_Description', N'Sales representiatives (names and
addresses) and their sales-related information.', 'SCHEMA', N'Sales', 'VIEW', N'vSales-
```

```
Person', NULL, NULL
GO
```

[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	17:24:52 2017. október 19., csütörtök	
Last Modified	17:29:01 2021. március 21., vasárnap	

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

```
CREATE VIEW [Sales].[vSalesPersonSalesByFiscalYears]

AS

SELECT

pvt.[SalesPersonID]

,pvt.[JobTitle]

,pvt.[JobTitle]

,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'vSalesPerson-
SalesByFiscalYears', NULL, NULL
```

[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	17:24:52 2017. október 19., csütörtök	
Last Modified	17:24:52 2017. október 19., csütörtök	

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

```
CREATE VIEW [Sales].[vStoreWithAddresses] AS

SELECT

s.[BusinessEntityID]
,s.[Name]
,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
```

[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	17:24:52 2017. október 19., csütörtök	
Last Modified	17:29:01 2021. március 21., vasárnap	

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

```
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'v-
StoreWithContacts', NULL, NULL
```

[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	17:24:52 2017. október 19., csütörtök	
Last Modified	17:24:52 2017. október 19., csütörtök	

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

```
"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'v-
StoreWithDemographics', NULL, NULL
```

[Sales].[Store] [dbo].[Name] Sales

Stored Procedures

Objects

Name

dbo.uspGetBillOfMaterials

Stored procedure using a recursive query to return a multi-level bill of material for the specified ProductID.

dbo.uspGetEmployeeManagers

Stored procedure using a recursive query to return the direct and indirect managers of the specified employee.

dbo.uspGetManagerEmployees

Stored procedure using a recursive query to return the direct and indirect employees of the specified manager.

dbo.uspGetWhereUsedProductID

Stored procedure using a recursive query to return all components or assemblies that directly or indirectly use the specified ProductID.

dbo.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. Should be executed from within the scope of a CATCH block otherwise it will return without inserting error information.

dbo.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.

dbo.uspSearchCandidateResumes

Human Resources. usp Update Employee Hire Info

Updates the Employee table and inserts a new row in the EmployeePayHistory table with the values specified in the input parameters.

Human Resources. usp Update Employee Login

Updates the Employee table with the values specified in the input parameters for the given BusinessEntityID.

HumanResources.uspUpdateEmployeePersonalInfo

Updates the Employee table with the values specified in the input parameters for the given EmployeeID.

[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 Input parameter for the stored procedure uspGetBillOfMaterials. Enter a valid ProductID from the Production.Product table.	int	4
@CheckDate Input parameter for the stored procedure uspGetBillOfMaterials used to eliminate components not used after that date. Enter a valid date.	datetime	8

```
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)
        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;
GΟ
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.', 'SCHEMA',
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'
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'@Start-
ProductID'
GO
```

[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 Input parameter for the stored procedure uspGetEmployeeManagers. Enter a valid BusinessEntityID from the HumanResources.Employee table.	int	4

```
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
       SELECT e.[BusinessEntityID], e.[OrganizationNode], p.[FirstName], p.[LastName],
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.[LastName],
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].[First-
Name], [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.', 'SCHEMA',
N'dbo', 'PROCEDURE', N'uspGetEmployeeManagers', NULL, NULL
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'uspGetEmployeeManagers',
'PARAMETER', N'@BusinessEntityID'
```

[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 Input parameter for the stored procedure uspGetManagerEmployees. Enter a valid BusinessEntityID of the manager from the HumanResources.Employee table.	int	4

```
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
       SELECT e.[BusinessEntityID], e.[OrganizationNode], p.[FirstName], p.[LastName],
{\tt 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.[LastName],
[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 'ManagerLast-
Name',
        [EMP cte].[BusinessEntityID], [EMP cte].[FirstName], [EMP cte].[LastName] --
Outer select from the \mathtt{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.', 'SCHEMA',
N'dbo', 'PROCEDURE', N'uspGetManagerEmployees', NULL, NULL
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'
```

[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 Input parameter for the stored procedure uspGetWhereUsedProductID. Enter a valid ProductID from the Production.Product table.	int	4
@CheckDate Input parameter for the stored procedure uspGetWhereUsedProductID used to eliminate components not used after that date. Enter a valid date.	datetime	8

```
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], [PerAssembly-
Qty], [StandardCost], [ListPrice], [BOMLevel], [RecursionLevel]) -- CTE name and
columns
       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)
```

```
UNTON ATIT
        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'uspGetWhereUsedProductID',
'PARAMETER', N'@CheckDate'
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'uspGetWhereUsedProductID', 'PARAMETER',
N'@StartProductID'
GΩ
```

[Production].[BillOfMaterials] [Production].[Product]

Project > localhost > User databases > AdventureWorks2016 > Programmability > Stored Procedures > dbo.usp-LogError

[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 Output parameter for the stored procedure uspLogError. Contains the ErrorLogID value corresponding to the row inserted by uspLogError in the ErrorLog table.	int	4	Out

```
-- 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
                                -- by uspLogError in the ErrorLog table
AS
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 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.', 'SCHEMA', N'dbo', 'PROCEDURE', N'uspLog-
Error', NULL, NULL
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
```

Project > localhost > User databases > AdventureWorks2016 > Programmability > Stored Procedures > dbo.usp-LogError

Uses

[dbo].[ErrorLog] [dbo].[uspPrintError]

Used By

[HumanResources].[uspUpdateEmployeeHireInfo] [HumanResources].[uspUpdateEmployeeLogin] [HumanResources].[uspUpdateEmployeePersonalInfo] Project > localhost > User databases > AdventureWorks2016 > Programmability > Stored Procedures > dbo.usp-PrintError

[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]
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;
GΟ
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
```

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

```
--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
```

[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 Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a valid BusinessEntityID from the Employee table.	int	4
@JobTitle Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a title for the employee.	nvarchar(50)	100
@HireDate Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a hire date for the employee.	datetime	8
@RateChangeDate Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the date the rate changed for the employee.	datetime	8
@Rate Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the new rate for the employee.	money	8
@PayFrequency Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the pay frequency for the employee.	tinyint	1
@CurrentFlag Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the current flag for the employee.	Flag	1

```
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'uspUpdateEmployeeHireInfo',
NULL, NULL
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'uspUpdateEmployeeHireInfo',
'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'
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'@HireDate'
```

```
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'@JobTitle'
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'
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'
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
```

[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 Input parameter for the stored procedure uspUpdateEmployeeLogin. Enter a valid EmployeeID from the Employee table.	int	4
@OrganizationNode Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a valid ManagerID for the employee.	hierarchyid	892
@LoginID Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a valid login for the employee.	nvarchar(256)	512
@JobTitle Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a title for the employee.	nvarchar(50)	100
@HireDate Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a hire date for the employee.	datetime	8
@CurrentFlag Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the current flag for the employee.	Flag	1

```
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
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'
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'
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'
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'
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
```

[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 Input parameter for the stored procedure uspUpdateEmployeePersonalInfo. Enter a valid BusinessEntityID from the HumanResources.Employee table.	int	4
@NationalIDNumber Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a national ID for the employee.	nvarchar(15)	30
@BirthDate Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a birth date for the employee.	datetime	8
@MaritalStatus Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a marital status for the employee.	nchar	1
@Gender Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a gender for the employee.	nchar	1

```
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'Human-
Resources', 'PROCEDURE', N'uspUpdateEmployeePersonalInfo', NULL, NULL
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'@Birth-
Date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the stored
procedure uspUpdateEmployeePersonalInfo. Enter a valid BusinessEntityID from the Human-
Resources.Employee table.', 'SCHEMA', N'HumanResources', 'PROCEDURE', N'uspUpdate-
EmployeePersonalInfo', 'PARAMETER', N'@BusinessEntityID'
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'
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'
GΩ
```

[HumanResources].[Employee] [dbo].[uspLogError] HumanResources

■ Table-valued Functions

Objects

Name

dbo.ufnGetContactInformation

Table value function returning the first name, last name, job title and contact type for a given contact.

[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 Input parameter for the table value function ufnGetContactInformation. Enter a valid PersonID from the Person.Contact table.	int	4

```
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:
{\tt EXEC} \  \, {\tt sp\_addextended} property \  \, {\tt N'MS\_Description'}, \  \, {\tt N'Table} \  \, {\tt value} \  \, {\tt function} \  \, {\tt returning} \  \, {\tt the}
first name, last name, job title and contact type for a given contact.', 'SCHEMA',
N'dbo', 'FUNCTION', N'ufnGetContactInformation', NULL, NULL
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'
```

 $\label{local-project} Project > local host > User \ databases > Adventure Works 2016 > Programma bility > Functions > Table-valued Functions > dbo.ufn Get Contact Information$

GO

Uses

[HumanResources].[Employee]
[Person].[BusinessEntityContact]
[Person].[ContactType]
[Person].[Person]
[Purchasing].[Vendor]

[Sales].[Customer]

[Sales].[Store]



Scalar-valued Functions

Objects

Name

dbo.ufnGetAccountingEndDate

Scalar function used in the uSalesOrderHeader trigger to set the starting account date.

dbo.ufnGetAccountingStartDate

Scalar function used in the uSalesOrderHeader trigger to set the ending account date.

dbo.ufnGetDocumentStatusText

Scalar function returning the text representation of the Status column in the Document table.

dbo.ufnGetProductDealerPrice

Scalar function returning the dealer price for a given product on a particular order date.

dbo.ufnGetProductListPrice

Scalar function returning the list price for a given product on a particular order date.

dbo.ufnGetProductStandardCost

Scalar function returning the standard cost for a given product on a particular order date.

dbo.ufnGetPurchaseOrderStatusText

Scalar function returning the text representation of the Status column in the PurchaseOrderHeader table.

dbo.ufnGetSalesOrderStatusText

Scalar function returning the text representation of the Status column in the SalesOrderHeader table.

dbo.ufnGetStock

Scalar function returning the quantity of inventory in LocationID 6 (Miscellaneous Storage)for a specified ProductID.

dbo.ufnLeadingZeros

Scalar function used by the Sales. Customer table to help set the account number.

[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

```
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

```
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 Input parameter for the scalar function ufnGetDocumentStatusText. Enter a valid integer.	tinyint	1

```
CREATE FUNCTION [dbo].[ufnGetDocumentStatusText](@Status [tinyint])
RETURNS [nvarchar] (16)
-- Returns the sales order status text representation for the status value.
   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
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'
```

Project > localhost > User databases >	AdventureWorks2016 >	Programmability >	Functions >	Scalar-valued
Functions > dbo.ufnGetDocumentStatu	sText			

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 Input parameter for the scalar function ufnGetProductDealerPrice. Enter a valid ProductID from the Production.Product table.	int	4
@OrderDate Input parameter for the scalar function ufnGetProductDealerPrice. Enter a valid order date.	datetime	8

```
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'ufnGetProductDealerPrice', 'PARAMETER', N'@ProductID'

GO
```

Uses

[Production].[Product]
[Production].[ProductListPriceHistory]

[fix [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 Input parameter for the scalar function ufnGetProductListPrice. Enter a valid ProductID from the Production.Product table.	int	4
@OrderDate Input parameter for the scalar function ufnGetProductListPrice. Enter a valid order date.	datetime	8

```
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;
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
EXEC sp_addextendedproperty N'MS_Description', N'Input parameter for the scalar
function ufnGetProductListPrice. Enter a valid order date.', 'SCHEMA', N'dbo',
```

Project > localhost > User databases > AdventureWorks2016 > Programmability > Functions > Scalar-valued Functions > dbo.ufnGetProductListPrice

```
'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 Input parameter for the scalar function ufnGetProductStandardCost. Enter a valid ProductID from the Production.Product table.	int	4
@OrderDate Input parameter for the scalar function ufnGetProductStandardCost. Enter a valid order date.	datetime	8

```
CREATE FUNCTION [dbo].[ufnGetProductStandardCost](@ProductID [int], @OrderDate
[datetime])
RETURNS [money]
-- Returns the standard cost for the product on a specific date.
    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;
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
```

Project > localhost > User databases > AdventureWorks2016 > Programmability > Functions > Scalar-valued Functions > dbo.ufnGetProductStandardCost

```
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'ufnGetProductStandardCost', '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 Input parameter for the scalar function ufnGetPurchaseOrdertStatusText. Enter a valid integer.	tinyint	1

```
CREATE FUNCTION [dbo].[ufnGetPurchaseOrderStatusText] (@Status [tinyint])
RETURNS [nvarchar] (15)
-- Returns the sales order status text representation for the status value.
   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
EXEC sp addextendedproperty N'MS Description', N'Input parameter for the scalar
function ufnGetPurchaseOrdertStaTusText. Enter a valid integer.', 'SCHEMA', N'dbo',
```

```
'FUNCTION', N'ufnGetPurchaseOrderStatusText', 'PARAMETER', N'@Status'
GO
```

[f]x [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 Input parameter for the scalar function ufnGetSalesOrderStatusText. Enter a valid integer.	tinyint	1

```
CREATE FUNCTION [dbo].[ufnGetSalesOrderStatusText](@Status [tinyint])
RETURNS [nvarchar] (15)
-- Returns the sales order status text representation for the status value.
   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
```

```
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 Input parameter for the scalar function ufnGetStock. Enter a valid ProductID from the Production.ProductInventory table.	int	4

```
CREATE FUNCTION [dbo].[ufnGetStock](@ProductID [int])
RETURNS [int]
-- Returns the stock level for the product. This function is used internally only
    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 ProductID.',
'SCHEMA', N'dbo', 'FUNCTION', N'ufnGetStock', NULL, NULL
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		

 $\label{local-project} Project > local host > User \ databases > Adventure Works 2016 > Programma bility > Functions > Scalar-valued Functions > dbo.ufn Get Stock$

[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	int	4
Input parameter for the scalar function ufnLeadingZeros. Enter a valid integer.		

```
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
EXEC sp addextendedproperty N'MS Description', N'Input parameter for the scalar
function ufnLeadingZeros. Enter a valid integer.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn-
LeadingZeros', 'PARAMETER', N'@Value'
```

Project > localhost > User databases > AdventureWorks2016 > Programmability > Functions > Scalar-valued Functions > dbo.ufnLeadingZeros	
Used By	

[Sales].[Customer]

Database Triggers

Objects

Name

ddlDatabaseTriggerLog

Database trigger to audit all of the DDL changes made to the AdventureWorks 2016 database.

ddlDatabaseTriggerLog

MS_Description

Database trigger to audit all of the DDL changes made to the AdventureWorks 2016 database.

Properties

Property	Value
ANSI Nulls On	True
Quoted Identifier On	True
Disabled	True

```
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;
       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
EXEC sp_addextendedproperty N'MS_Description', N'Database trigger to audit all of the
DDL changes made to the AdventureWorks 2016 database.', 'TRIGGER', N'ddlDatabaseTrigger-
Log', NULL, NULL, NULL, NULL
GO
```

iii User-Defined Data Types

Objects

Name	
dbo.AccountNumber	
dbo.Flag	
dbo.Name	
dbo.NameStyle	
dbo.OrderNumber	
dbo.Phone	

[dbo].[AccountNumber]

Properties

Property	Value
Nullability	NULL allowed
Base Type Name	nvarchar
Length	15

SQL Script

CREATE TYPE [dbo].[AccountNumber] FROM nvarchar (15) NULL
GO

Used By

[Purchasing].[Vendor] [Sales].[SalesOrderHeader]

iii [dbo].[Flag]

Properties

Property	Value
Nullability	NOT NULL
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
Nullability	NULL allowed
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]

Project > localhost > User databases > AdventureWorks2016 > Programmability > Types > User-Defined Data Types > dbo.Name

[Production].[vProductAndDescription]

[Production].[vProductModelCatalogDescription]

[Production].[vProductModelInstructions]

[Purchasing].[vVendorWithAddresses]

[Purchasing].[vVendorWithContacts]

[Sales].[vIndividualCustomer]

[Sales].[vSalesPerson]

[Sales].[vSalesPersonSalesByFiscalYears]

[Sales].[vStoreWithAddresses]

[Sales].[vStoreWithContacts]

[Sales].[vStoreWithDemographics]

iii [dbo].[NameStyle]

Properties

Property	Value
Nullability	NOT NULL
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
Nullability	NULL allowed
Base Type Name	nvarchar
Length	25

SQL Script

CREATE TYPE [dbo].[OrderNumber] FROM nvarchar (25) NULL
GO

Used By

[Sales].[SalesOrderHeader]

iii [dbo].[Phone]

Properties

Property	Value
Nullability	NULL allowed
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]

A XML Schema Collections

Objects

Name

HumanResources.HRResumeSchemaCollection

Collection of XML schemas for the Resume column in the HumanResources. Job Candidate table.

Person. Additional Contact Info Schema Collection

Collection of XML schemas for the AdditionalContactInfo column in the Person. Contact table.

Person.IndividualSurveySchemaCollection

Collection of XML schemas for the Demographics column in the Person. Person table.

${\bf Production. ManuInstructions Schema Collection}$

Collection of XML schemas for the Instructions column in the Production. Product Model table.

Production.ProductDescriptionSchemaCollection

Collection of XML schemas for the CatalogDescription column in the Production ProductModel table.

Sales.StoreSurveySchemaCollection

Collection of XML schemas for the Demographics column in the Sales. Store table.

[HumanResources].[HRResumeSchemaCollection]

MS_Description

Collection of XML schemas for the Resume column in the HumanResources.JobCandidate table.

Dependent Columns

• [HumanResources].[JobCandidate].[Resume]

```
CREATE XML SCHEMA COLLECTION [HumanResources].[HRResumeSchemaCollection]
AS N'<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:t="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume" target-
Namespace="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: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: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.GPAAlternate" 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: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: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">
```

Project > localhost > User databases > AdventureWorks2016 > Programmability > Types > XML Schema Collections > HumanResources.HRResumeSchemaCollection

```
<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>
</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'HumanResources',
'XML SCHEMA COLLECTION', N'HRResumeSchemaCollection', 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].[AdditionalContactInfo]
- [Person].[AdditionalContactInfo]

```
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/Contact-
 <xsd:element name="AdditionalContactInfo">
   <xsd:complexType mixed="true">
     <xsd:complexContent mixed="true">
       <xsd:restriction base="xsd:anyType">
         <xsd:sequence>
          <xsd:anv
http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes" min-
Occurs="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"</pre>
xmlns:t="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:anv
namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes"
        </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"</pre>
```

```
xmlns:t="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes"
targetNamespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact-
Types" elementFormDefault="qualified">
  <xsd:element name="eMail" type="t:eMailType" />
 <xsd:element name="facsimileTelephoneNumber" type="t:phoneNumberType" />
 <xsd:element name="homePostalAddress" type="t:addressType" />
  <xsd:element name="internationaliSDNNumber" 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" min-</pre>
Occurs="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" min-</pre>
Occurs="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" min-</pre>
Occurs="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-</pre>
works/ContactTypes" 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].[Demographics]
- [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-
{\tt Survey"\ targetNamespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likely-likel
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-</pre>
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" />
```

Project > localhost > User databases > AdventureWorks2016 > Programmability > Types > XML Schema Collections > Person.IndividualSurveySchemaCollection

```
</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]
- [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/ProductModel-
ManuInstructions" 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" maxOccurs="unbounded"</pre>
                    </xsd:sequence>
                    <xsd:attribute name="LocationID" type="xsd:integer" use="required"</pre>
                    <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">
```

Project > localhost > User databases > AdventureWorks2016 > Programmability > Types > XML Schema Collections > Production.ManuInstructionsSchemaCollection

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]
- [Production].[ProductModel].[CatalogDescription]

SQL Script

Author: dnagykrisztina

```
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" targetNamespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-
works/ProductModelWarrAndMain" 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"</pre>
xmlns:ns1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
WarrAndMain" xmlns:t="http://schemas.microsoft.com/sqlserver/2004/07/adventure-
works/ProductModelDescription" target-
Namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModel-
Description" elementFormDefault="qualified">
 <xsd:import namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-</pre>
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-</pre>
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-</pre>
Occurs="unbounded" />
          <xsd:element name="Picture" type="t:Picture" minOccurs="0" max-</pre>
Occurs="unbounded" />
          <xsd:element name="Category" type="t:Category" minOccurs="0" max-</pre>
Occurs="unbounded" />
         <xsd:element name="Specifications" type="t:Specifications" minOccurs="0" max-</pre>
Occurs="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" min-</pre>
Occurs="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]
- [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/Store-
Survey" 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: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>'
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.AW2016FullTextCatalog



AW2016FullTextCatalog

Properties

Property	Value
Owner	dbo
Default	True
Accent Sensitive	True

Tables

- **JobCandidate**
- Document
- ProductReview

SQL Script

```
CREATE FULLTEXT CATALOG [AW2016FullTextCatalog]
WITH ACCENT_SENSITIVITY = ON
AS DEFAULT
AUTHORIZATION [dbo]
CREATE FULLTEXT INDEX ON [HumanResources].[JobCandidate] KEY INDEX [PK JobCandidate Job-
CandidateID] ON [AW2016FullTextCatalog]
CREATE FULLTEXT INDEX ON [Production].[Document] KEY INDEX [PK_Document_DocumentNode]
ON [AW2016FullTextCatalog]
CREATE FULLTEXT INDEX ON [Production].[ProductReview] KEY INDEX [PK ProductReview -
ProductReviewID] ON [AW2016FullTextCatalog]
```

Uses

[HumanResources].[JobCandidate] [Production].[Document]

[Production].[ProductReview]



Objects

Name	
dbo	
guest	



Properties

Property	Value
Туре	WindowsUser
Default Schema	dbo

Database Level Permissions

Туре	Action
CONNECT	Grant

SQL Script

GO



Properties

Property	Value
Туре	SqlUser
Default Schema	guest

SQL Script

GO			

La 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

db_datawriter

Properties

Property	Value
Owner	dbo

db_ddladmin

Properties

Property	Value
Owner	dbo

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

Objects

Name

HumanResources

Contains objects related to employees and departments.

Person

Contains objects related to names and addresses of customers, vendors, and employees

Production

Contains objects related to products, inventory, and manufacturing.

Purchasing

Contains objects related to vendors and purchase orders.

Sales

Contains objects related to customers, sales orders, and sales territories.

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, OCCUPANTIAL OCCUPANTIAL
```

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].[Pet]

[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

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]

[Production].[vProductModelInstructions]

[Production]. [ManuInstructionsSchema Collection]

[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]. [Purchase Order Detail]

[Purchasing].[PurchaseOrderHeader]

[Purchasing].[ShipMethod]

[Purchasing].[Vendor]

[Purchasing].[vVendorWithAddresses]

[Purchasing].[vVendorWithContacts]

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, OULL, OULL,
```

Used By

[Sales]. [Country Region Currency]

[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]. [Special Offer Product]

[Sales].[Store]

[Sales].[vIndividualCustomer]

[Sales]. [vPersonDemographics]

[Sales].[vSalesPerson]

[Sales].[vSalesPersonSalesByFiscalYears]

[Sales].[vStoreWithAddresses]

[Sales].[vStoreWithContacts]
[Sales].[vStoreWithDemographics]
[Sales].[StoreSurveySchemaCollection]