localhost Documentation

AdventureWorks2016

|  |  |
| --- | --- |
| Server | localhost |
| Author | dnagykrisztina |
| Created | 2021. március 21., vasárnap 18:40:53 |
| File Path | C:\Users\dnagykrisztina\Desktop\Szakdolgozat\Redgate-MSSQL\Dokumentáció\documenter\_word-2021-03-21T18-40-53.docx |

# Table of Contents

[Table of Contents 2](#_Toc256000000)

[localhost 7](#_Toc256000001)



[User databases 9](#_Toc256000002)



[Adventure­Works2016 Database 10](#_Toc256000003)



[Tables 13](#_Toc256000004)



[[dbo].[AWBuild­Version] 17](#_Toc256000005)



[[dbo].[Database­Log] 19](#_Toc256000006)



[[dbo].[Error­Log] 21](#_Toc256000007)



[[Human­Resources].[Department] 24](#_Toc256000008)



[[Human­Resources].[Employee] 26](#_Toc256000009)



[[Human­Resources].[Employee­Department­History] 34](#_Toc256000010)



[[Human­Resources].[Employee­Pay­History] 38](#_Toc256000011)



[[Human­Resources].[Job­Candidate] 41](#_Toc256000012)



[[Human­Resources].[Shift] 44](#_Toc256000013)



[[Person].[Address] 47](#_Toc256000014)



[[Person].[Address­Type] 51](#_Toc256000015)



[[Person].[Business­Entity] 54](#_Toc256000016)



[[Person].[Business­Entity­Address] 56](#_Toc256000017)



[[Person].[Business­Entity­Contact] 60](#_Toc256000018)



[[Person].[Contact­Type] 64](#_Toc256000019)



[[Person].[Country­Region] 66](#_Toc256000020)



[[Person].[Email­Address] 68](#_Toc256000021)



[[Person].[Password] 71](#_Toc256000022)



[[Person].[Person] 74](#_Toc256000023)



[[Person].[Person­Phone] 82](#_Toc256000024)



[[Person].[Pet] 85](#_Toc256000025)



[[Person].[Phone­Number­Type] 87](#_Toc256000026)



[[Person].[State­Province] 89](#_Toc256000027)



[[Production].[Bill­Of­Materials] 93](#_Toc256000028)



[[Production].[Culture] 98](#_Toc256000029)



[[Production].[Document] 100](#_Toc256000030)



[[Production].[Illustration] 105](#_Toc256000031)



[[Production].[Location] 107](#_Toc256000032)



[[Production].[Product] 110](#_Toc256000033)



[[Production].[Product­Category] 119](#_Toc256000034)



[[Production].[Product­Cost­History] 122](#_Toc256000035)



[[Production].[Product­Description] 125](#_Toc256000036)



[[Production].[Product­Document] 128](#_Toc256000037)



[[Production].[Product­Inventory] 131](#_Toc256000038)



[[Production].[Product­List­Price­History] 135](#_Toc256000039)



[[Production].[Product­Model] 138](#_Toc256000040)



[[Production].[Product­Model­Illustration] 141](#_Toc256000041)



[[Production].[Product­Model­Product­Description­Culture] 144](#_Toc256000042)



[[Production].[Product­Photo] 147](#_Toc256000043)



[[Production].[Product­Product­Photo] 149](#_Toc256000044)



[[Production].[Product­Review] 152](#_Toc256000045)



[[Production].[Product­Subcategory] 156](#_Toc256000046)



[[Production].[Scrap­Reason] 159](#_Toc256000047)



[[Production].[Transaction­History] 161](#_Toc256000048)



[[Production].[Transaction­History­Archive] 165](#_Toc256000049)



[[Production].[Unit­Measure] 169](#_Toc256000050)



[[Production].[Work­Order] 171](#_Toc256000051)



[[Production].[Work­Order­Routing] 177](#_Toc256000052)



[[Purchasing].[Product­Vendor] 182](#_Toc256000053)



[[Purchasing].[Purchase­Order­Detail] 187](#_Toc256000054)



[[Purchasing].[Purchase­Order­Header] 195](#_Toc256000055)



[[Purchasing].[Ship­Method] 202](#_Toc256000056)



[[Purchasing].[Vendor] 205](#_Toc256000057)



[[Sales].[Country­Region­Currency] 210](#_Toc256000058)



[[Sales].[Credit­Card] 213](#_Toc256000059)



[[Sales].[Currency] 216](#_Toc256000060)



[[Sales].[Currency­Rate] 218](#_Toc256000061)



[[Sales].[Customer] 221](#_Toc256000062)



[[Sales].[Person­Credit­Card] 225](#_Toc256000063)



[[Sales].[Sales­Order­Detail] 228](#_Toc256000064)



[[Sales].[Sales­Order­Header] 235](#_Toc256000065)



[[Sales].[Sales­Order­Header­Sales­Reason] 246](#_Toc256000066)



[[Sales].[Sales­Person] 249](#_Toc256000067)



[[Sales].[Sales­Person­Quota­History] 254](#_Toc256000068)



[[Sales].[Sales­Reason] 257](#_Toc256000069)



[[Sales].[Sales­Tax­Rate] 259](#_Toc256000070)



[[Sales].[Sales­Territory] 263](#_Toc256000071)



[[Sales].[Sales­Territory­History] 268](#_Toc256000072)



[[Sales].[Shopping­Cart­Item] 272](#_Toc256000073)



[[Sales].[Special­Offer] 275](#_Toc256000074)



[[Sales].[Special­Offer­Product] 279](#_Toc256000075)



[[Sales].[Store] 282](#_Toc256000076)



[Views 286](#_Toc256000077)



[[Human­Resources].[v­Employee] 288](#_Toc256000078)



[[Human­Resources].[v­Employee­Department] 291](#_Toc256000079)



[[Human­Resources].[v­Employee­Department­History] 293](#_Toc256000080)



[[Human­Resources].[v­Job­Candidate] 295](#_Toc256000081)



[[Human­Resources].[v­Job­Candidate­Education] 298](#_Toc256000082)



[[Human­Resources].[v­Job­Candidate­Employment] 300](#_Toc256000083)



[[Person].[v­Additional­Contact­Info] 302](#_Toc256000084)



[[Person].[v­State­Province­Country­Region] 305](#_Toc256000085)



[[Production].[v­Product­And­Description] 307](#_Toc256000086)



[[Production].[v­Product­Model­Catalog­Description] 309](#_Toc256000087)



[[Production].[v­Product­Model­Instructions] 313](#_Toc256000088)



[[Purchasing].[v­Vendor­With­Addresses] 315](#_Toc256000089)



[[Purchasing].[v­Vendor­With­Contacts] 317](#_Toc256000090)



[[Sales].[v­Individual­Customer] 319](#_Toc256000091)



[[Sales].[v­Person­Demographics] 322](#_Toc256000092)



[[Sales].[v­Sales­Person] 324](#_Toc256000093)



[[Sales].[v­Sales­Person­Sales­By­Fiscal­Years] 327](#_Toc256000094)



[[Sales].[v­Store­With­Addresses] 329](#_Toc256000095)



[[Sales].[v­Store­With­Contacts] 331](#_Toc256000096)



[[Sales].[v­Store­With­Demographics] 333](#_Toc256000097)



[Stored Procedures 335](#_Toc256000098)



[[dbo].[usp­Get­Bill­Of­Materials] 336](#_Toc256000099)



[[dbo].[usp­Get­Employee­Managers] 338](#_Toc256000100)



[[dbo].[usp­Get­Manager­Employees] 340](#_Toc256000101)



[[dbo].[usp­Get­Where­Used­Product­ID] 342](#_Toc256000102)



[[dbo].[usp­Log­Error] 344](#_Toc256000103)



[[dbo].[usp­Print­Error] 347](#_Toc256000104)



[[dbo].[usp­Search­Candidate­Resumes] 348](#_Toc256000105)



[[Human­Resources].[usp­Update­Employee­Hire­Info] 350](#_Toc256000106)



[[Human­Resources].[usp­Update­Employee­Login] 353](#_Toc256000107)



[[Human­Resources].[usp­Update­Employee­Personal­Info] 355](#_Toc256000108)



[Table-valued Functions 357](#_Toc256000109)



[[dbo].[ufn­Get­Contact­Information] 358](#_Toc256000110)



[Scalar-valued Functions 361](#_Toc256000111)



[[dbo].[ufn­Get­Accounting­End­Date] 362](#_Toc256000112)



[[dbo].[ufn­Get­Accounting­Start­Date] 363](#_Toc256000113)



[[dbo].[ufn­Get­Document­Status­Text] 364](#_Toc256000114)



[[dbo].[ufn­Get­Product­Dealer­Price] 366](#_Toc256000115)



[[dbo].[ufn­Get­Product­List­Price] 368](#_Toc256000116)



[[dbo].[ufn­Get­Product­Standard­Cost] 370](#_Toc256000117)



[[dbo].[ufn­Get­Purchase­Order­Status­Text] 372](#_Toc256000118)



[[dbo].[ufn­Get­Sales­Order­Status­Text] 374](#_Toc256000119)



[[dbo].[ufn­Get­Stock] 376](#_Toc256000120)



[[dbo].[ufn­Leading­Zeros] 378](#_Toc256000121)



[Database Triggers 380](#_Toc256000122)



[ddl­Database­Trigger­Log 381](#_Toc256000123)



[User-Defined Data Types 383](#_Toc256000124)



[[dbo].[Account­Number] 384](#_Toc256000125)



[[dbo].[Flag] 385](#_Toc256000126)



[[dbo].[Name] 386](#_Toc256000127)



[[dbo].[Name­Style] 388](#_Toc256000128)



[[dbo].[Order­Number] 389](#_Toc256000129)



[[dbo].[Phone] 390](#_Toc256000130)



[XML Schema Collections 391](#_Toc256000131)



[[Human­Resources].[HRResume­Schema­Collection] 392](#_Toc256000132)



[[Person].[Additional­Contact­Info­Schema­Collection] 396](#_Toc256000133)



[[Person].[Individual­Survey­Schema­Collection] 399](#_Toc256000134)



[[Production].[Manu­Instructions­Schema­Collection] 401](#_Toc256000135)



[[Production].[Product­Description­Schema­Collection] 403](#_Toc256000136)



[[Sales].[Store­Survey­Schema­Collection] 406](#_Toc256000137)



[Full Text Catalogs 408](#_Toc256000138)



[AW2016Full­Text­Catalog 409](#_Toc256000139)



[Users 410](#_Toc256000140)



[dbo 411](#_Toc256000141)



[guest 412](#_Toc256000142)



[Database Roles 413](#_Toc256000143)



[db\_accessadmin 413](#_Toc256000144)



[db\_backupoperator 413](#_Toc256000145)



[db\_datareader 414](#_Toc256000146)



[db\_datawriter 414](#_Toc256000147)



[db\_ddladmin 414](#_Toc256000148)



[db\_denydatareader 415](#_Toc256000149)



[db\_denydatawriter 415](#_Toc256000150)



[db\_owner 415](#_Toc256000151)



[db\_securityadmin 415](#_Toc256000152)



[public 416](#_Toc256000153)



[Schemas 417](#_Toc256000154)



[Human­Resources 418](#_Toc256000155)



[Person 419](#_Toc256000156)



[Production 420](#_Toc256000157)



[Purchasing 422](#_Toc256000158)



[Sales 423](#_Toc256000159)



|  |
| --- |
| localhost |

Databases (1)

* [Adventure­Works2016](#3YtA7ToWRtTERJ33CCELIhGpcrY=)



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 |
| Default language | English |
| Network packet size | 4096 |
| Default fulltext language LCID | 1033 |
| Two-digit year cutoff | 2049 |
| Remote login timeout | 10 |
| Cursor threshold | -1 |
| Max text replication size | 65536 |
| Parallelism cost threshold | 5 |
| Max degree of parallelism | 0 |
| Min server memory | 16 |
| Max server memory | 2147483647 |
| Scan for startup procs | False |
| Transform noise words | False |
| CLR enabled | False |
| Blocked process threshold | 0 |
| Filestream access level | False |
| Optimize for ad hoc workloads | False |
| CLR strict security | True |

|  |
| --- |
| User databases |

Databases (1)

* [Adventure­Works2016](#3YtA7ToWRtTERJ33CCELIhGpcrY=)



|  |
| --- |
| 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 |
| 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 | Type | Size | Maxsize | Autogrowth | File Name |
| Adventure­Works2016\_­Data | Data | 207,63 MB | unlimited | 16,00 MB | /var/opt/mssql/data/Adventure­Works2016\_­Data.mdf |
| Adventure­Works2016\_­Log | Log | 2,00 MB | 2048,00 GB | 16,00 MB | /var/opt/mssql/data/Adventure­Works2016\_­Log.ldf |

|  |
| --- |
| Tables |

Objects

|  |
| --- |
| Name |
| [dbo.AWBuild­Version](#35GkMcSMKI2EkwGsvgX+wAYiViI=) Current version number of the Adventure­Works 2016 sample database. |
| [dbo.Database­Log](#CnKY1YtkVKFieznXxLjA9MoKmgE=) Audit table tracking all DDL changes made to the Adventure­Works database. Data is captured by the database trigger ddl­Database­Trigger­Log. |
| [dbo.Error­Log](#XCrMUsfgKMjh8aFAM1/ck+QFKjQ=) 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](#TH5R06keykSJYajBUWfR3phTY8s=) Lookup table containing the departments within the Adventure Works Cycles company. |
| [Human­Resources.Employee](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=) Employee information such as salary, department, and title. |
| [Human­Resources.Employee­Department­History](#tDjny2Q9sWvuqg97wWzd145nXkA=) Employee department transfers. |
| [Human­Resources.Employee­Pay­History](#WzREUJ9VRXbYWerHGwiGJinYM0A=) Employee pay history. |
| [Human­Resources.Job­Candidate](#WVRXAhzXWcgHQbdlg570h3Fv6QI=) Résumés submitted to Human Resources by job applicants. |
| [Human­Resources.Shift](#oOQXice18uca5/e1dhJkYUmlNt0=) Work shift lookup table. |
| [Person.Address](#YHZkNb8Ts04ByOupF/OAfoeI2qk=) Street address information for customers, employees, and vendors. |
| [Person.Address­Type](#sC112VucbUsBFhIBrJnCpuBv1os=) Types of addresses stored in the Address table. |
| [Person.Business­Entity](#I7t6VoTbrv7x5ZQ/kE/EZs1npdg=) Source of the ID that connects vendors, customers, and employees with address and contact information. |
| [Person.Business­Entity­Address](#HK6CehnfZ4YYh7ckP8cx74El9pY=) Cross-reference table mapping customers, vendors, and employees to their addresses. |
| [Person.Business­Entity­Contact](#9Zj2hWeFVzGQn5EUQXP6lLMtAZI=) Cross-reference table mapping stores, vendors, and employees to people |
| [Person.Contact­Type](#uyTva1snlxlEidD4b0wN1HQzMw4=) Lookup table containing the types of business entity contacts. |
| [Person.Country­Region](#efOGmVlBeofSPEY2jc0F91joV9Q=) Lookup table containing the ISO standard codes for countries and regions. |
| [Person.Email­Address](#3EL9ymWvglf8ADJZjk1nQ8j/hLY=) Where to send a person email. |
| [Person.Password](#z1Ylcf/ZAQ6li+sOkPO88E0ccrw=) One way hashed authentication information |
| [Person.Person](#TZe4JItyjmK+s/tUWju0b9W3HXU=) Human beings involved with Adventure­Works: employees, customer contacts, and vendor contacts. |
| [Person.Person­Phone](#NyzXUZBpBs0rFeuVHGsZlpmRP0Y=) Telephone number and type of a person. |
| [Person.Pet](#etnWX6SZkHYAYL8t5N4pzt7zXKE=) |
| [Person.Phone­Number­Type](#05nCels02cq7i+PWSDjMFGEmWFU=) Type of phone number of a person. |
| [Person.State­Province](#fYuToAkdHMSws7pXek8EIcFTfvY=) State and province lookup table. |
| [Production.Bill­Of­Materials](#m9P+2olmZFXOowkUY18+Ri7YHx4=) Items required to make bicycles and bicycle subassemblies. It identifies the heirarchical relationship between a parent product and its components. |
| [Production.Culture](#D4g0rs1owPY1TNtNA3F8BcetcxU=) Lookup table containing the languages in which some Adventure­Works data is stored. |
| [Production.Document](#v5/qynDVNwkOWZdAe2xATpOnRWk=) Product maintenance documents. |
| [Production.Illustration](#q4hmTMYciVWtuNDgg2tVsn/4PAw=) Bicycle assembly diagrams. |
| [Production.Location](#ciHD61a29Ub1OJEWhZ1zarNhvcs=) Product inventory and manufacturing locations. |
| [Production.Product](#mO6TpkyW7q4W/OMRHISuYfWNCME=) Products sold or used in the manfacturing of sold products. |
| [Production.Product­Category](#upfoEFG3EP8VWKmQ3NJKJ7V9JTs=) High-level product categorization. |
| [Production.Product­Cost­History](#V4hNECfDVL8rNsab6KfgsgmqeHE=) Changes in the cost of a product over time. |
| [Production.Product­Description](#4iEXAFuDYt/4TBUTP9DP8dTGazY=) Product descriptions in several languages. |
| [Production.Product­Document](#hFy0+8CYqgHPJaiGh0/YlUFf+eI=) Cross-reference table mapping products to related product documents. |
| [Production.Product­Inventory](#JnGLivmNHPEgSuzGYyO0WefiqDw=) Product inventory information. |
| [Production.Product­List­Price­History](#lSXw7btC5MVIn+g9cscCbIFpkU0=) Changes in the list price of a product over time. |
| [Production.Product­Model](#n9VdHBX3Ji5OWfmMzrHHD+J8HSM=) Product model classification. |
| [Production.Product­Model­Illustration](#ha0ZY5wgktKInuoqdKFBVRbeJaI=) Cross-reference table mapping product models and illustrations. |
| [Production.Product­Model­Product­Description­Culture](#Kkf63jUYCrQKelqjjcG51JBQDNI=) Cross-reference table mapping product descriptions and the language the description is written in. |
| [Production.Product­Photo](#wYFkOtQB0E8+2MDoAl84kiyzc9A=) Product images. |
| [Production.Product­Product­Photo](#40voS36ihwbm/xtmbxsgEQceBOQ=) Cross-reference table mapping products and product photos. |
| [Production.Product­Review](#Z1+PLnel1CKvek/LdN60/nVl5cI=) Customer reviews of products they have purchased. |
| [Production.Product­Subcategory](#WthFJ9hmewdhBB85kY7NAjtek7Y=) Product subcategories. See Product­Category table. |
| [Production.Scrap­Reason](#0dK3gJTjqlelVRIns1VtvoCHDlU=) Manufacturing failure reasons lookup table. |
| [Production.Transaction­History](#qEI6rAj76QhivliqVRIQ5mR7wrA=) Record of each purchase order, sales order, or work order transaction year to date. |
| [Production.Transaction­History­Archive](#mWvwPJkOXR0q1NT8gm4kJjfJ/aA=) Transactions for previous years. |
| [Production.Unit­Measure](#11scPpeMkfJepgAP6eC9jOIFK+Y=) Unit of measure lookup table. |
| [Production.Work­Order](#BSRICvfPko6shp7o8HSu003oCno=) Manufacturing work orders. |
| [Production.Work­Order­Routing](#wxYgGkEd/o/k/roV2tY7ZCLH+qE=) Work order details. |
| [Purchasing.Product­Vendor](#igWcZ2fEqwA+AlsF0xQn66oTGAU=) Cross-reference table mapping vendors with the products they supply. |
| [Purchasing.Purchase­Order­Detail](#dsxg+I54iAv4FByATvQGW4l6Cco=) Individual products associated with a specific purchase order. See Purchase­Order­Header. |
| [Purchasing.Purchase­Order­Header](#wVItRHlooKKFqOhuItRUJuA60Fs=) General purchase order information. See Purchase­Order­Detail. |
| [Purchasing.Ship­Method](#Xt5ka3ip5ljwxYo+n5TjCRZypoY=) Shipping company lookup table. |
| [Purchasing.Vendor](#yfc9dLywcQtxtbztLjV6J3bb0tM=) Companies from whom Adventure Works Cycles purchases parts or other goods. |
| [Sales.Country­Region­Currency](#ZTDvJcpGJmjH9tZPEMXFxv/sN0o=) Cross-reference table mapping ISO currency codes to a country or region. |
| [Sales.Credit­Card](#Juoic8jJ3a0e1KCG+cj8LixVRbQ=) Customer credit card information. |
| [Sales.Currency](#FEyILkTLRQRZ4zkAPe+AjFF8vg8=) Lookup table containing standard ISO currencies. |
| [Sales.Currency­Rate](#DbnMf0tD82NB+BGbYOX9jWqScts=) Currency exchange rates. |
| [Sales.Customer](#UK53WWAjbLvEqOKY0KByf769x9M=) Current customer information. Also see the Person and Store tables. |
| [Sales.Person­Credit­Card](#HOEWmD+eQ6y/+8bq8hkiLEiBBjU=) Cross-reference table mapping people to their credit card information in the Credit­Card table. |
| [Sales.Sales­Order­Detail](#OkvyE7uLBgi7y2P/yCWzRf5c0wE=) Individual products associated with a specific sales order. See Sales­Order­Header. |
| [Sales.Sales­Order­Header](#J0oqEfXKE3WFfJ7PoPfs1QUbO+c=) General sales order information. |
| [Sales.Sales­Order­Header­Sales­Reason](#8Csb5Ws8nFAO1qWhqs6WCgPvmp4=) Cross-reference table mapping sales orders to sales reason codes. |
| [Sales.Sales­Person](#hKN58zHn4ffskJv6ldcUqjf87do=) Sales representative current information. |
| [Sales.Sales­Person­Quota­History](#ZEp3m1cn9szNTztS7qyCt1Ng+u8=) Sales performance tracking. |
| [Sales.Sales­Reason](#a3VrLxK/Cx1h+4jZ069Ywnkt+DM=) Lookup table of customer purchase reasons. |
| [Sales.Sales­Tax­Rate](#qRl1u9t1FSaWq4TfyLPf6H9xl4k=) Tax rate lookup table. |
| [Sales.Sales­Territory](#hvNbyL+2AhjcTkzEdMCOV3mg8XQ=) Sales territory lookup table. |
| [Sales.Sales­Territory­History](#STLuHwwNIM3t3qtp2pV+eqOT5iE=) Sales representative transfers to other sales territories. |
| [Sales.Shopping­Cart­Item](#usOp9/vzTEpC3eW1PfV8jtNCfNQ=) Contains online customer orders until the order is submitted or cancelled. |
| [Sales.Special­Offer](#H28uErAJ2leiaB1YxFBqojYITRE=) Sale discounts lookup table. |
| [Sales.Special­Offer­Product](#8e4RMrihzXJLXw2gPdnNJ+2ijQ0=) Cross-reference table mapping products to special offer discounts. |
| [Sales.Store](#o4yE5vuw4wTVa38E7sxNcXx3KhA=) Customers (resellers) of Adventure Works products. |

|  |
| --- |
| [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 |
|  | 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\_­AWBuild­Version\_­System­Information­ID Primary key (clustered) constraint | System­Information­ID | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[AWBuild­Version]  (  [System­Information­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Database Version] [nvarchar] (25) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Version­Date] [datetime] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­AWBuild­Version\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[AWBuild­Version] ADD CONSTRAINT [PK\_­AWBuild­Version\_­System­Information­ID] PRIMARY KEY CLUSTERED ([System­Information­ID]) ON [PRIMARY]  GO  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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for AWBuild­Version records.', 'SCHEMA', N'dbo', 'TABLE', N'AWBuild­Version', 'COLUMN', N'System­Information­ID'  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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'dbo', 'TABLE', N'AWBuild­Version', 'CONSTRAINT', N'DF\_­AWBuild­Version\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'dbo', 'TABLE', N'AWBuild­Version', 'CONSTRAINT', N'PK\_­AWBuild­Version\_­System­Information­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'dbo', 'TABLE', N'AWBuild­Version', 'INDEX', N'PK\_­AWBuild­Version\_­System­Information­ID'  GO |

|  |
| --- |
| [dbo].[Database­Log] |

MS\_­Description

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

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Heap | True |
| Row Count (~) | 1597 |
| Created | 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 |
|  | 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\_­Database­Log\_­Database­Log­ID Primary key (nonclustered) constraint | Database­Log­ID | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Database­Log]  (  [Database­Log­ID] [int] NOT NULL IDENTITY(1, 1),  [Post­Time] [datetime] NOT NULL,  [Database­User] [sys].[sysname] NOT NULL,  [Event] [sys].[sysname] NOT NULL,  [Schema] [sys].[sysname] NULL,  [Object] [sys].[sysname] NULL,  [TSQL] [nvarchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Xml­Event] [xml] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[Database­Log] ADD CONSTRAINT [PK\_­Database­Log\_­Database­Log­ID] PRIMARY KEY NONCLUSTERED ([Database­Log­ID]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Audit table tracking all DDL changes made to the Adventure­Works database. Data is captured by the database trigger ddl­Database­Trigger­Log.', 'SCHEMA', N'dbo', 'TABLE', N'Database­Log', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Database­Log records.', 'SCHEMA', N'dbo', 'TABLE', N'Database­Log', 'COLUMN', N'Database­Log­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The user who implemented the DDL change.', 'SCHEMA', N'dbo', 'TABLE', N'Database­Log', 'COLUMN', N'Database­User'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The type of DDL statement that was executed.', 'SCHEMA', N'dbo', 'TABLE', N'Database­Log', 'COLUMN', N'Event'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The object that was changed by the DDL statment.', 'SCHEMA', N'dbo', 'TABLE', N'Database­Log', 'COLUMN', N'Object'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The date and time the DDL change occurred.', 'SCHEMA', N'dbo', 'TABLE', N'Database­Log', 'COLUMN', N'Post­Time'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The schema to which the changed object belongs.', 'SCHEMA', N'dbo', 'TABLE', N'Database­Log', 'COLUMN', N'Schema'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The exact Transact-SQL statement that was executed.', 'SCHEMA', N'dbo', 'TABLE', N'Database­Log', 'COLUMN', N'TSQL'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The raw XML data generated by database trigger.', 'SCHEMA', N'dbo', 'TABLE', N'Database­Log', 'COLUMN', N'Xml­Event'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (nonclustered) constraint', 'SCHEMA', N'dbo', 'TABLE', N'Database­Log', 'CONSTRAINT', N'PK\_­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) | Nullability | Identity | Default |
|  | 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\_­Error­Log\_­Error­Log­ID Primary key (clustered) constraint | Error­Log­ID | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Error­Log]  (  [Error­Log­ID] [int] NOT NULL IDENTITY(1, 1),  [Error­Time] [datetime] NOT NULL CONSTRAINT [DF\_­Error­Log\_­Error­Time] DEFAULT (getdate()),  [User­Name] [sys].[sysname] NOT NULL,  [Error­Number] [int] NOT NULL,  [Error­Severity] [int] NULL,  [Error­State] [int] NULL,  [Error­Procedure] [nvarchar] (126) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Error­Line] [int] NULL,  [Error­Message] [nvarchar] (4000) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[Error­Log] ADD CONSTRAINT [PK\_­Error­Log\_­Error­Log­ID] PRIMARY KEY CLUSTERED ([Error­Log­ID]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Audit table tracking errors in the the Adventure­Works database that are caught by the CATCH block of a TRY...CATCH construct. Data is inserted by stored procedure dbo.usp­Log­Error when it is executed from inside the CATCH block of a TRY...CATCH construct.', 'SCHEMA', N'dbo', 'TABLE', N'Error­Log', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The line number at which the error occurred.', 'SCHEMA', N'dbo', 'TABLE', N'Error­Log', 'COLUMN', N'Error­Line'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Error­Log records.', 'SCHEMA', N'dbo', 'TABLE', N'Error­Log', 'COLUMN', N'Error­Log­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The message text of the error that occurred.', 'SCHEMA', N'dbo', 'TABLE', N'Error­Log', 'COLUMN', N'Error­Message'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The error number of the error that occurred.', 'SCHEMA', N'dbo', 'TABLE', N'Error­Log', 'COLUMN', N'Error­Number'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The name of the stored procedure or trigger where the error occurred.', 'SCHEMA', N'dbo', 'TABLE', N'Error­Log', 'COLUMN', N'Error­Procedure'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The severity of the error that occurred.', 'SCHEMA', N'dbo', 'TABLE', N'Error­Log', 'COLUMN', N'Error­Severity'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The state number of the error that occurred.', 'SCHEMA', N'dbo', 'TABLE', N'Error­Log', 'COLUMN', N'Error­State'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The date and time at which the error occurred.', 'SCHEMA', N'dbo', 'TABLE', N'Error­Log', 'COLUMN', N'Error­Time'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The user who executed the batch in which the error occurred.', 'SCHEMA', N'dbo', 'TABLE', N'Error­Log', 'COLUMN', N'User­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'dbo', 'TABLE', N'Error­Log', 'CONSTRAINT', N'DF\_­Error­Log\_­Error­Time'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'dbo', 'TABLE', N'Error­Log', 'CONSTRAINT', N'PK\_­Error­Log\_­Error­Log­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'dbo', 'TABLE', N'Error­Log', 'INDEX', N'PK\_­Error­Log\_­Error­Log­ID'  GO |

Used By

[[dbo].[usp­Log­Error]](#5QeOhszr4MkHJhy/bzXlVS2S9sg=)

|  |
| --- |
| [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 |
|  | Department­ID Primary key for Department records. | smallint | 2 | NOT NULL | 1 - 1 |  |
|  | Name Name of the department. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NOT NULL |  |  |
|  | Group­Name Name of the group to which the department belongs. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NOT NULL |  |  |
|  | Modified­Date Date and time the record was last updated. | datetime | 8 | NOT NULL |  | (getdate()) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Department\_­Department­ID Primary key (clustered) constraint | Department­ID | True |
|  | AK\_­Department\_­Name Unique nonclustered index. | Name | True |

SQL Script

|  |
| --- |
| CREATE TABLE [Human­Resources].[Department]  (  [Department­ID] [smallint] NOT NULL IDENTITY(1, 1),  [Name] [dbo].[Name] NOT NULL,  [Group­Name] [dbo].[Name] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Department\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Human­Resources].[Department] ADD CONSTRAINT [PK\_­Department\_­Department­ID] PRIMARY KEY CLUSTERED ([Department­ID]) ON [PRIMARY]  GO  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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Name of the group to which the department belongs.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Department', 'COLUMN', N'Group­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Department', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Name of the department.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Department', 'COLUMN', N'Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Human­Resources', 'TABLE', N'Department', 'CONSTRAINT', N'DF\_­Department\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Human­Resources', 'TABLE', N'Department', 'CONSTRAINT', N'PK\_­Department\_­Department­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Department', 'INDEX', N'AK\_­Department\_­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Department', 'INDEX', N'PK\_­Department\_­Department­ID'  GO |

Uses

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Human­Resources](#wRNBi9QOObVjEO+Wxjr4GvvwBBs=)

Used By

[[Human­Resources].[Employee­Department­History]](#tDjny2Q9sWvuqg97wWzd145nXkA=)

[[Human­Resources].[v­Employee­Department]](#JhbW+DYcE93jITPkfvVrtIWEHzE=)

[[Human­Resources].[v­Employee­Department­History]](#Ppb9efXEcRDxKIGzLbHZkOB+Nto=)

|  |
| --- |
| [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 | Computed | Max Length (Bytes) | Nullability | Default |
|  | Business­Entity­ID Primary key for Employee records. Foreign key to Business­Entity.Business­Entity­ID. | int |  | 4 | NOT NULL |  |
|  | National­IDNumber Unique national identification number such as a social security number. | nvarchar(15) |  | 30 | NOT NULL |  |
|  | 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 bargaining. 1 = Salaried, exempt from collective bargaining. | [[dbo].[Flag]](#j88jjDtwCbK8khz/m8jxryH/SFk=) |  | 1 | NOT NULL | ((1)) |
|  | 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]](#j88jjDtwCbK8khz/m8jxryH/SFk=) |  | 1 | 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()) |

Computed columns

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

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Employee\_­Business­Entity­ID Primary key (clustered) constraint | Business­Entity­ID | True |
|  | AK\_­Employee\_­Login­ID Unique nonclustered index. | Login­ID | True |
|  | AK\_­Employee\_­National­IDNumber Unique nonclustered index. | National­IDNumber | True |
|  | AK\_­Employee\_rowguid Unique nonclustered index. Used to support replication samples. | rowguid | True |
|  | IX\_­Employee\_­Organization­Level\_­Organization­Node Unique nonclustered index. | Organization­Level, Organization­Node |  |
|  | IX\_­Employee\_­Organization­Node Unique nonclustered index. | Organization­Node |  |

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 | ([Birth­Date]>='1930-01-01' AND [Birth­Date]<=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 | ([Hire­Date]>='1996-07-01' AND [Hire­Date]<=dateadd(day,(1),getdate())) |
| CK\_­Employee\_­Marital­Status Check constraint [Marital­Status]='s' OR [Marital­Status]='m' OR [Marital­Status]='S' OR [Marital­Status]='M' | Marital­Status | (upper([Marital­Status])='S' OR upper([Marital­Status])='M') |
| CK\_­Employee\_­Sick­Leave­Hours Check constraint [Sick­Leave­Hours] >= (0) AND [Sick­Leave­Hours] <= (120) | Sick­Leave­Hours | ([Sick­Leave­Hours]>=(0) AND [Sick­Leave­Hours]<=(120)) |
| CK\_­Employee\_­Vacation­Hours Check constraint [Vacation­Hours] >= (-40) AND [Vacation­Hours] <= (240) | Vacation­Hours | ([Vacation­Hours]>=((-40)) AND [Vacation­Hours]<=(240)) |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Employee\_­Person\_­Business­Entity­ID Foreign key constraint referencing Person.Business­Entity­ID. | Business­Entity­ID->[[Person].[Person].[Business­Entity­ID]](#TZe4JItyjmK+s/tUWju0b9W3HXU=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Human­Resources].[Employee]  (  [Business­Entity­ID] [int] NOT NULL,  [National­IDNumber] [nvarchar] (15) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Login­ID] [nvarchar] (256) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Organization­Node] [sys].[hierarchyid] NULL,  [Organization­Level] AS ([Organization­Node].[Get­Level]()),  [Job­Title] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Birth­Date] [date] NOT NULL,  [Marital­Status] [nchar] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Gender] [nchar] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Hire­Date] [date] NOT NULL,  [Salaried­Flag] [dbo].[Flag] NOT NULL CONSTRAINT [DF\_­Employee\_­Salaried­Flag] DEFAULT ((1)),  [Vacation­Hours] [smallint] NOT NULL CONSTRAINT [DF\_­Employee\_­Vacation­Hours] DEFAULT ((0)),  [Sick­Leave­Hours] [smallint] NOT NULL CONSTRAINT [DF\_­Employee\_­Sick­Leave­Hours] DEFAULT ((0)),  [Current­Flag] [dbo].[Flag] NOT NULL CONSTRAINT [DF\_­Employee\_­Current­Flag] DEFAULT ((1)),  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Employee\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Employee\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  CREATE TRIGGER [Human­Resources].[d­Employee] ON [Human­Resources].[Employee]  INSTEAD OF DELETE NOT FOR REPLICATION AS  BEGIN  DECLARE @Count int;  SET @Count = @@ROWCOUNT;  IF @Count = 0  RETURN;  SET NOCOUNT ON;  BEGIN  RAISERROR  (N'Employees cannot be deleted. They can only be marked as not current.', -- Message  10, -- Severity.  1); -- State.  -- Rollback any active or uncommittable transactions  IF @@TRANCOUNT > 0  BEGIN  ROLLBACK TRANSACTION;  END  END;  END;  GO  ALTER TABLE [Human­Resources].[Employee] ADD CONSTRAINT [CK\_­Employee\_­Birth­Date] CHECK (([Birth­Date]>='1930-01-01' AND [Birth­Date]<=dateadd(year,(-18),getdate())))  GO  ALTER TABLE [Human­Resources].[Employee] ADD CONSTRAINT [CK\_­Employee\_­Gender] CHECK ((upper([Gender])='F' OR upper([Gender])='M'))  GO  ALTER TABLE [Human­Resources].[Employee] ADD CONSTRAINT [CK\_­Employee\_­Hire­Date] CHECK (([Hire­Date]>='1996-07-01' AND [Hire­Date]<=dateadd(day,(1),getdate())))  GO  ALTER TABLE [Human­Resources].[Employee] ADD CONSTRAINT [CK\_­Employee\_­Marital­Status] CHECK ((upper([Marital­Status])='S' OR upper([Marital­Status])='M'))  GO  ALTER TABLE [Human­Resources].[Employee] ADD CONSTRAINT [CK\_­Employee\_­Sick­Leave­Hours] CHECK (([Sick­Leave­Hours]>=(0) AND [Sick­Leave­Hours]<=(120)))  GO  ALTER TABLE [Human­Resources].[Employee] ADD CONSTRAINT [CK\_­Employee\_­Vacation­Hours] CHECK (([Vacation­Hours]>=((-40)) AND [Vacation­Hours]<=(240)))  GO  ALTER TABLE [Human­Resources].[Employee] ADD CONSTRAINT [PK\_­Employee\_­Business­Entity­ID] PRIMARY KEY CLUSTERED ([Business­Entity­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Employee\_­Login­ID] ON [Human­Resources].[Employee] ([Login­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Employee\_­National­IDNumber] ON [Human­Resources].[Employee] ([National­IDNumber]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Employee\_­Organization­Level\_­Organization­Node] ON [Human­Resources].[Employee] ([Organization­Level], [Organization­Node]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Employee\_­Organization­Node] ON [Human­Resources].[Employee] ([Organization­Node]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Employee\_rowguid] ON [Human­Resources].[Employee] ([rowguid]) ON [PRIMARY]  GO  ALTER TABLE [Human­Resources].[Employee] ADD CONSTRAINT [FK\_­Employee\_­Person\_­Business­Entity­ID] FOREIGN KEY ([Business­Entity­ID]) REFERENCES [Person].[Person] ([Business­Entity­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Employee information such as salary, department, and title.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date of birth.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'COLUMN', N'Birth­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Employee records. Foreign key to Business­Entity.Business­Entity­ID.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'COLUMN', N'Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = Inactive, 1 = Active', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'COLUMN', N'Current­Flag'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'M = Male, F = Female', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'COLUMN', N'Gender'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Employee hired on this date.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'COLUMN', N'Hire­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Work title such as Buyer or Sales Representative.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'COLUMN', N'Job­Title'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Network login.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'COLUMN', N'Login­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'M = Married, S = Single', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'COLUMN', N'Marital­Status'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'COLUMN', N'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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Where the employee is located in corporate hierarchy.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'COLUMN', N'Organization­Node'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Job classification. 0 = Hourly, not exempt from collective bargaining. 1 = Salaried, exempt from collective bargaining.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'COLUMN', N'Salaried­Flag'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Number of available sick leave hours.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'COLUMN', N'Sick­Leave­Hours'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Number of available vacation hours.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'COLUMN', N'Vacation­Hours'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Birth­Date] >= ''1930-01-01'' AND [Birth­Date] <= dateadd(year,(-18),GETDATE())', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'CONSTRAINT', N'CK\_­Employee\_­Birth­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Gender]=''f'' OR [Gender]=''m'' OR [Gender]=''F'' OR [Gender]=''M''', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'CONSTRAINT', N'CK\_­Employee\_­Gender'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Hire­Date] >= ''1996-07-01'' AND [Hire­Date] <= dateadd(day,(1),GETDATE())', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'CONSTRAINT', N'CK\_­Employee\_­Hire­Date'  GO  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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Sick­Leave­Hours] >= (0) AND [Sick­Leave­Hours] <= (120)', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'CONSTRAINT', N'CK\_­Employee\_­Sick­Leave­Hours'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Vacation­Hours] >= (-40) AND [Vacation­Hours] <= (240)', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'CONSTRAINT', N'CK\_­Employee\_­Vacation­Hours'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 1', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'CONSTRAINT', N'DF\_­Employee\_­Current­Flag'  GO  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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 1 (TRUE)', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'CONSTRAINT', N'DF\_­Employee\_­Salaried­Flag'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'CONSTRAINT', N'DF\_­Employee\_­Sick­Leave­Hours'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'CONSTRAINT', N'DF\_­Employee\_­Vacation­Hours'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Person.Business­Entity­ID.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'CONSTRAINT', N'FK\_­Employee\_­Person\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'CONSTRAINT', N'PK\_­Employee\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'INDEX', N'AK\_­Employee\_­Login­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'INDEX', N'AK\_­Employee\_­National­IDNumber'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'INDEX', N'AK\_­Employee\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'INDEX', N'IX\_­Employee\_­Organization­Level\_­Organization­Node'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee', 'INDEX', N'IX\_­Employee\_­Organization­Node'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'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]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[[dbo].[Flag]](#j88jjDtwCbK8khz/m8jxryH/SFk=)

[Human­Resources](#wRNBi9QOObVjEO+Wxjr4GvvwBBs=)

Used By

[[Human­Resources].[Employee­Department­History]](#tDjny2Q9sWvuqg97wWzd145nXkA=)

[[Human­Resources].[Employee­Pay­History]](#WzREUJ9VRXbYWerHGwiGJinYM0A=)

[[Human­Resources].[Job­Candidate]](#WVRXAhzXWcgHQbdlg570h3Fv6QI=)

[[Production].[Document]](#v5/qynDVNwkOWZdAe2xATpOnRWk=)

[[Purchasing].[Purchase­Order­Header]](#wVItRHlooKKFqOhuItRUJuA60Fs=)

[[Sales].[Sales­Person]](#hKN58zHn4ffskJv6ldcUqjf87do=)

[[Human­Resources].[v­Employee]](#TVXJJHSO4vPDsqDtNRd5tGQdsO0=)

[[Human­Resources].[v­Employee­Department]](#JhbW+DYcE93jITPkfvVrtIWEHzE=)

[[Human­Resources].[v­Employee­Department­History]](#Ppb9efXEcRDxKIGzLbHZkOB+Nto=)

[[Sales].[v­Sales­Person]](#pOjFS54tNwjD3Yvi8IMnn7dtRuA=)

[[Sales].[v­Sales­Person­Sales­By­Fiscal­Years]](#GfF5ZKdsT1+xL6X5ks+ghnXIfng=)

[[dbo].[usp­Get­Employee­Managers]](#yaIVF+o80FMpJygtFV827IXwYrg=)

[[dbo].[usp­Get­Manager­Employees]](#j0llh4oBM6uN5VRP3n9vBDBGwbc=)

[[Human­Resources].[usp­Update­Employee­Hire­Info]](#6ydkl6OBzxZTt8aQ90AP6hr4cFU=)

[[Human­Resources].[usp­Update­Employee­Login]](#C5oNhWtNH5mLMaKD+WajVUPlL6E=)

[[Human­Resources].[usp­Update­Employee­Personal­Info]](#vCx7xMQcjRyuiOsUmNSNg9QeITY=)

[[dbo].[ufn­Get­Contact­Information]](#MAbUSn/1caUuJ7XN6dczcbTRRyA=)

|  |
| --- |
| [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 |
|  | Business­Entity­ID Employee identification number. Foreign key to Employee.Business­Entity­ID. | int | 4 | NOT NULL |  |
|  | Department­ID Department in which the employee worked including currently. Foreign key to Department.Department­ID. | smallint | 2 | NOT NULL |  |
|  | Shift­ID Identifies which 8-hour shift the employee works. Foreign key to Shift.Shift.ID. | tinyint | 1 | NOT NULL |  |
|  | 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\_­Employee­Department­History\_­Business­Entity­ID\_­Start­Date\_­Department­ID Primary key (clustered) constraint | Business­Entity­ID, 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->[[Human­Resources].[Department].[Department­ID]](#TH5R06keykSJYajBUWfR3phTY8s=) |
| FK\_­Employee­Department­History\_­Employee\_­Business­Entity­ID Foreign key constraint referencing Employee.Employee­ID. | Business­Entity­ID->[[Human­Resources].[Employee].[Business­Entity­ID]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=) |
| FK\_­Employee­Department­History\_­Shift\_­Shift­ID Foreign key constraint referencing Shift.Shift­ID | Shift­ID->[[Human­Resources].[Shift].[Shift­ID]](#oOQXice18uca5/e1dhJkYUmlNt0=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Human­Resources].[Employee­Department­History]  (  [Business­Entity­ID] [int] NOT NULL,  [Department­ID] [smallint] NOT NULL,  [Shift­ID] [tinyint] NOT NULL,  [Start­Date] [date] NOT NULL,  [End­Date] [date] NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Employee­Department­History\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Human­Resources].[Employee­Department­History] ADD CONSTRAINT [CK\_­Employee­Department­History\_­End­Date] CHECK (([End­Date]>=[Start­Date] OR [End­Date] IS NULL))  GO  ALTER TABLE [Human­Resources].[Employee­Department­History] ADD CONSTRAINT [PK\_­Employee­Department­History\_­Business­Entity­ID\_­Start­Date\_­Department­ID] PRIMARY KEY CLUSTERED ([Business­Entity­ID], [Start­Date], [Department­ID], [Shift­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Employee­Department­History\_­Department­ID] ON [Human­Resources].[Employee­Department­History] ([Department­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Employee­Department­History\_­Shift­ID] ON [Human­Resources].[Employee­Department­History] ([Shift­ID]) ON [PRIMARY]  GO  ALTER TABLE [Human­Resources].[Employee­Department­History] ADD CONSTRAINT [FK\_­Employee­Department­History\_­Department\_­Department­ID] FOREIGN KEY ([Department­ID]) REFERENCES [Human­Resources].[Department] ([Department­ID])  GO  ALTER TABLE [Human­Resources].[Employee­Department­History] ADD CONSTRAINT [FK\_­Employee­Department­History\_­Employee\_­Business­Entity­ID] FOREIGN KEY ([Business­Entity­ID]) REFERENCES [Human­Resources].[Employee] ([Business­Entity­ID])  GO  ALTER TABLE [Human­Resources].[Employee­Department­History] ADD CONSTRAINT [FK\_­Employee­Department­History\_­Shift\_­Shift­ID] FOREIGN KEY ([Shift­ID]) REFERENCES [Human­Resources].[Shift] ([Shift­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Employee department transfers.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Department­History', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Employee identification number. Foreign key to Employee.Business­Entity­ID.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Department­History', 'COLUMN', N'Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Department in which the employee worked including currently. Foreign key to Department.Department­ID.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Department­History', 'COLUMN', N'Department­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date the employee left the department. NULL = Current department.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Department­History', 'COLUMN', N'End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Department­History', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Identifies which 8-hour shift the employee works. Foreign key to Shift.Shift.ID.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Department­History', 'COLUMN', N'Shift­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date the employee started work in the department.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Department­History', 'COLUMN', N'Start­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [End­Date] >= [Start­Date] OR [End­Date] IS NUL', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Department­History', 'CONSTRAINT', N'CK\_­Employee­Department­History\_­End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Department­History', 'CONSTRAINT', N'DF\_­Employee­Department­History\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Department.Department­ID.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Department­History', 'CONSTRAINT', N'FK\_­Employee­Department­History\_­Department\_­Department­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Employee.Employee­ID.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Department­History', 'CONSTRAINT', N'FK\_­Employee­Department­History\_­Employee\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Shift.Shift­ID', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Department­History', 'CONSTRAINT', N'FK\_­Employee­Department­History\_­Shift\_­Shift­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Department­History', 'CONSTRAINT', N'PK\_­Employee­Department­History\_­Business­Entity­ID\_­Start­Date\_­Department­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Department­History', 'INDEX', N'IX\_­Employee­Department­History\_­Department­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Department­History', 'INDEX', N'IX\_­Employee­Department­History\_­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]](#TH5R06keykSJYajBUWfR3phTY8s=)

[[Human­Resources].[Employee]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=)

[[Human­Resources].[Shift]](#oOQXice18uca5/e1dhJkYUmlNt0=)

[Human­Resources](#wRNBi9QOObVjEO+Wxjr4GvvwBBs=)

Used By

[[Human­Resources].[v­Employee­Department]](#JhbW+DYcE93jITPkfvVrtIWEHzE=)

[[Human­Resources].[v­Employee­Department­History]](#Ppb9efXEcRDxKIGzLbHZkOB+Nto=)

|  |
| --- |
| [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 |
|  | Business­Entity­ID Employee identification number. Foreign key to Employee.Business­Entity­ID. | int | 4 | NOT NULL |  |
|  | 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 |
|  | 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 [Pay­Frequency]=(1) | Pay­Frequency | ([Pay­Frequency]=(2) OR [Pay­Frequency]=(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.Employee­ID. | Business­Entity­ID->[[Human­Resources].[Employee].[Business­Entity­ID]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Human­Resources].[Employee­Pay­History]  (  [Business­Entity­ID] [int] NOT NULL,  [Rate­Change­Date] [datetime] NOT NULL,  [Rate] [money] NOT NULL,  [Pay­Frequency] [tinyint] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Employee­Pay­History\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Human­Resources].[Employee­Pay­History] ADD CONSTRAINT [CK\_­Employee­Pay­History\_­Pay­Frequency] CHECK (([Pay­Frequency]=(2) OR [Pay­Frequency]=(1)))  GO  ALTER TABLE [Human­Resources].[Employee­Pay­History] ADD CONSTRAINT [CK\_­Employee­Pay­History\_­Rate] CHECK (([Rate]>=(6.50) AND [Rate]<=(200.00)))  GO  ALTER TABLE [Human­Resources].[Employee­Pay­History] ADD CONSTRAINT [PK\_­Employee­Pay­History\_­Business­Entity­ID\_­Rate­Change­Date] PRIMARY KEY CLUSTERED ([Business­Entity­ID], [Rate­Change­Date]) ON [PRIMARY]  GO  ALTER TABLE [Human­Resources].[Employee­Pay­History] ADD CONSTRAINT [FK\_­Employee­Pay­History\_­Employee\_­Business­Entity­ID] FOREIGN KEY ([Business­Entity­ID]) REFERENCES [Human­Resources].[Employee] ([Business­Entity­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Employee pay history.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Pay­History', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Employee identification number. Foreign key to Employee.Business­Entity­ID.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Pay­History', 'COLUMN', N'Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Pay­History', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'1 = Salary received monthly, 2 = Salary received biweekly', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Pay­History', 'COLUMN', N'Pay­Frequency'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Salary hourly rate.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Pay­History', 'COLUMN', N'Rate'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date the change in pay is effective', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Pay­History', 'COLUMN', N'Rate­Change­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Pay­Frequency]=(3) OR [Pay­Frequency]=(2) OR [Pay­Frequency]=(1)', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Pay­History', 'CONSTRAINT', N'CK\_­Employee­Pay­History\_­Pay­Frequency'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Rate] >= (6.50) AND [Rate] <= (200.00)', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Pay­History', 'CONSTRAINT', N'CK\_­Employee­Pay­History\_­Rate'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Pay­History', 'CONSTRAINT', N'DF\_­Employee­Pay­History\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Employee.Employee­ID.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Pay­History', 'CONSTRAINT', N'FK\_­Employee­Pay­History\_­Employee\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Pay­History', 'CONSTRAINT', N'PK\_­Employee­Pay­History\_­Business­Entity­ID\_­Rate­Change­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Employee­Pay­History', 'INDEX', N'PK\_­Employee­Pay­History\_­Business­Entity­ID\_­Rate­Change­Date'  GO |

Uses

[[Human­Resources].[Employee]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=)

[Human­Resources](#wRNBi9QOObVjEO+Wxjr4GvvwBBs=)

Used By

[[Human­Resources].[usp­Update­Employee­Hire­Info]](#6ydkl6OBzxZTt8aQ90AP6hr4cFU=)

|  |
| --- |
| [Human­Resources].[Job­Candidate] |

MS\_­Description

Résumés submitted to Human Resources by job applicants.

Properties

|  |  |
| --- | --- |
| Property | Value |
| Full Text Catalog | [AW2016Full­Text­Catalog](#KQM1/2huuuPy+pMNx4F37D5+p0Y=) |
| 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 Length (Bytes) | Nullability | Full Text Indexed | Language | Identity | Default |
|  | Job­Candidate­ID Primary key for Job­Candidate records. | int | 4 | NOT NULL |  |  | 1 - 1 |  |
|  | Business­Entity­ID Employee identification number if applicant was hired. Foreign key to Employee.Business­Entity­ID. | int | 4 | NULL allowed |  |  |  |  |
|  | Resume Résumé in XML format. | [xml([Human­Resources].[HRResume­Schema­Collection])](#l5jXOsaY8noZjfgneQd+33ebs58=) | max | NULL allowed | True | 1033 |  |  |
|  | Modified­Date Date and time the record was last updated. | datetime | 8 | NOT NULL |  |  |  | (getdate()) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | 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.Employee­ID. | Business­Entity­ID->[[Human­Resources].[Employee].[Business­Entity­ID]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Human­Resources].[Job­Candidate]  (  [Job­Candidate­ID] [int] NOT NULL IDENTITY(1, 1),  [Business­Entity­ID] [int] NULL,  [Resume] [xml] (CONTENT [Human­Resources].[HRResume­Schema­Collection]) NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Job­Candidate\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Human­Resources].[Job­Candidate] ADD CONSTRAINT [PK\_­Job­Candidate\_­Job­Candidate­ID] PRIMARY KEY CLUSTERED ([Job­Candidate­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Job­Candidate\_­Business­Entity­ID] ON [Human­Resources].[Job­Candidate] ([Business­Entity­ID]) ON [PRIMARY]  GO  ALTER TABLE [Human­Resources].[Job­Candidate] ADD CONSTRAINT [FK\_­Job­Candidate\_­Employee\_­Business­Entity­ID] FOREIGN KEY ([Business­Entity­ID]) REFERENCES [Human­Resources].[Employee] ([Business­Entity­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Résumés submitted to Human Resources by job applicants.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Job­Candidate', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Employee identification number if applicant was hired. Foreign key to Employee.Business­Entity­ID.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Job­Candidate', 'COLUMN', N'Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Job­Candidate records.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Job­Candidate', 'COLUMN', N'Job­Candidate­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Job­Candidate', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Résumé in XML format.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Job­Candidate', 'COLUMN', N'Resume'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Human­Resources', 'TABLE', N'Job­Candidate', 'CONSTRAINT', N'DF\_­Job­Candidate\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Employee.Employee­ID.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Job­Candidate', 'CONSTRAINT', N'FK\_­Job­Candidate\_­Employee\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Human­Resources', 'TABLE', N'Job­Candidate', 'CONSTRAINT', N'PK\_­Job­Candidate\_­Job­Candidate­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Job­Candidate', 'INDEX', N'IX\_­Job­Candidate\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Job­Candidate', 'INDEX', N'PK\_­Job­Candidate\_­Job­Candidate­ID'  GO  CREATE FULLTEXT INDEX ON [Human­Resources].[Job­Candidate] KEY INDEX [PK\_­Job­Candidate\_­Job­Candidate­ID] ON [AW2016Full­Text­Catalog]  GO  ALTER FULLTEXT INDEX ON [Human­Resources].[Job­Candidate] ADD ([Resume] LANGUAGE 1033)  GO |

Uses

[[Human­Resources].[Employee]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=)

[Human­Resources](#wRNBi9QOObVjEO+Wxjr4GvvwBBs=)

[[Human­Resources].[HRResume­Schema­Collection]](#l5jXOsaY8noZjfgneQd+33ebs58=)

Used By

[[Human­Resources].[v­Job­Candidate]](#OJ+5x6Pf56todcFiq0tOA8QJbeA=)

[[Human­Resources].[v­Job­Candidate­Education]](#L3qRlBiBKB45IvcTjpNzlK4Zu0c=)

[[Human­Resources].[v­Job­Candidate­Employment]](#xDdPFCC7cBQwDROim4Heve+/Okw=)

[[dbo].[usp­Search­Candidate­Resumes]](#/DT71/7etHi4sXQ3xAYmBMvn9JA=)

[AW2016Full­Text­Catalog](#KQM1/2huuuPy+pMNx4F37D5+p0Y=)

|  |
| --- |
| [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 |
|  | Shift­ID Primary key for Shift records. | tinyint | 1 | NOT NULL | 1 - 1 |  |
|  | Name Shift description. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NOT NULL |  |  |
|  | Start­Time Shift start time. | time | 5 | NOT NULL |  |  |
|  | 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\_­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 |

SQL Script

|  |
| --- |
| CREATE TABLE [Human­Resources].[Shift]  (  [Shift­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Name] [dbo].[Name] NOT NULL,  [Start­Time] [time] NOT NULL,  [End­Time] [time] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Shift\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Human­Resources].[Shift] ADD CONSTRAINT [PK\_­Shift\_­Shift­ID] PRIMARY KEY CLUSTERED ([Shift­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Shift\_­Name] ON [Human­Resources].[Shift] ([Name]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Shift\_­Start­Time\_­End­Time] ON [Human­Resources].[Shift] ([Start­Time], [End­Time]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Work shift lookup table.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Shift', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Shift end time.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Shift', 'COLUMN', N'End­Time'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Shift', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Shift description.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Shift', 'COLUMN', N'Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Shift records.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Shift', 'COLUMN', N'Shift­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Shift start time.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Shift', 'COLUMN', N'Start­Time'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Human­Resources', 'TABLE', N'Shift', 'CONSTRAINT', N'DF\_­Shift\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Human­Resources', 'TABLE', N'Shift', 'CONSTRAINT', N'PK\_­Shift\_­Shift­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Shift', 'INDEX', N'AK\_­Shift\_­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Shift', 'INDEX', N'AK\_­Shift\_­Start­Time\_­End­Time'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Human­Resources', 'TABLE', N'Shift', 'INDEX', N'PK\_­Shift\_­Shift­ID'  GO |

Uses

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Human­Resources](#wRNBi9QOObVjEO+Wxjr4GvvwBBs=)

Used By

[[Human­Resources].[Employee­Department­History]](#tDjny2Q9sWvuqg97wWzd145nXkA=)

[[Human­Resources].[v­Employee­Department­History]](#Ppb9efXEcRDxKIGzLbHZkOB+Nto=)

|  |
| --- |
| [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) | Nullability | Identity | Identity Replication | Default |
|  | Address­ID Primary key for Address records. | int | 4 | NOT NULL | 1 - 1 | False |  |
|  | Address­Line1 First street address line. | nvarchar(60) | 120 | NOT NULL |  |  |  |
|  | Address­Line2 Second street address line. | nvarchar(60) | 120 | NULL allowed |  |  |  |
|  | 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 |  |  |  |
|  | Postal­Code Postal code for the street address. | nvarchar(15) | 30 | NOT NULL |  |  |  |
|  | Spatial­Location Latitude and longitude of this address. | geography | 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 |  |  | (getdate()) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | 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, State­Province­ID, Postal­Code | True |
|  | IX\_­Address\_­State­Province­ID Nonclustered index. | State­Province­ID |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Address\_­State­Province\_­State­Province­ID Foreign key constraint referencing State­Province.State­Province­ID. | State­Province­ID->[[Person].[State­Province].[State­Province­ID]](#fYuToAkdHMSws7pXek8EIcFTfvY=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Person].[Address]  (  [Address­ID] [int] NOT NULL IDENTITY(1, 1) NOT FOR REPLICATION,  [Address­Line1] [nvarchar] (60) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Address­Line2] [nvarchar] (60) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [City] [nvarchar] (30) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [State­Province­ID] [int] NOT NULL,  [Postal­Code] [nvarchar] (15) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Spatial­Location] [sys].[geography] NULL,  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Address\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Address\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Person].[Address] ADD CONSTRAINT [PK\_­Address\_­Address­ID] PRIMARY KEY CLUSTERED ([Address­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [IX\_­Address\_­Address­Line1\_­Address­Line2\_­City\_­State­Province­ID\_­Postal­Code] ON [Person].[Address] ([Address­Line1], [Address­Line2], [City], [State­Province­ID], [Postal­Code]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Address\_rowguid] ON [Person].[Address] ([rowguid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Address\_­State­Province­ID] ON [Person].[Address] ([State­Province­ID]) ON [PRIMARY]  GO  ALTER TABLE [Person].[Address] ADD CONSTRAINT [FK\_­Address\_­State­Province\_­State­Province­ID] FOREIGN KEY ([State­Province­ID]) REFERENCES [Person].[State­Province] ([State­Province­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Street address information for customers, employees, and vendors.', 'SCHEMA', N'Person', 'TABLE', N'Address', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Address records.', 'SCHEMA', N'Person', 'TABLE', N'Address', 'COLUMN', N'Address­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'First street address line.', 'SCHEMA', N'Person', 'TABLE', N'Address', 'COLUMN', N'Address­Line1'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Second street address line.', 'SCHEMA', N'Person', 'TABLE', N'Address', 'COLUMN', N'Address­Line2'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Name of the city.', 'SCHEMA', N'Person', 'TABLE', N'Address', 'COLUMN', N'City'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Person', 'TABLE', N'Address', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Postal code for the street address.', 'SCHEMA', N'Person', 'TABLE', N'Address', 'COLUMN', N'Postal­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Person', 'TABLE', N'Address', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Latitude and longitude of this address.', 'SCHEMA', N'Person', 'TABLE', N'Address', 'COLUMN', N'Spatial­Location'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique identification number for the state or province. Foreign key to State­Province table.', 'SCHEMA', N'Person', 'TABLE', N'Address', 'COLUMN', N'State­Province­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'Address', 'CONSTRAINT', N'DF\_­Address\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Person', 'TABLE', N'Address', 'CONSTRAINT', N'DF\_­Address\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing State­Province.State­Province­ID.', 'SCHEMA', N'Person', 'TABLE', N'Address', 'CONSTRAINT', N'FK\_­Address\_­State­Province\_­State­Province­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Person', 'TABLE', N'Address', 'CONSTRAINT', N'PK\_­Address\_­Address­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Person', 'TABLE', N'Address', 'INDEX', N'AK\_­Address\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Person', 'TABLE', N'Address', 'INDEX', N'IX\_­Address\_­Address­Line1\_­Address­Line2\_­City\_­State­Province­ID\_­Postal­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Person', 'TABLE', N'Address', 'INDEX', N'IX\_­Address\_­State­Province­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'Address', 'INDEX', N'PK\_­Address\_­Address­ID'  GO |

Uses

[[Person].[State­Province]](#fYuToAkdHMSws7pXek8EIcFTfvY=)

[Person](#GSXHhzVRNjGQYtnCQHosa2uthyw=)

Used By

[[Person].[Business­Entity­Address]](#HK6CehnfZ4YYh7ckP8cx74El9pY=)

[[Sales].[Sales­Order­Header]](#J0oqEfXKE3WFfJ7PoPfs1QUbO+c=)

[[Human­Resources].[v­Employee]](#TVXJJHSO4vPDsqDtNRd5tGQdsO0=)

[[Purchasing].[v­Vendor­With­Addresses]](#NdlQ62Q6DZw3GS30J9H9dm/SOqg=)

[[Sales].[v­Individual­Customer]](#WcBS54CxROpNMbufsyU5V6t9Ykw=)

[[Sales].[v­Sales­Person]](#pOjFS54tNwjD3Yvi8IMnn7dtRuA=)

[[Sales].[v­Store­With­Addresses]](#lyHdOUZeccdbPaj16bKzv2RjJKw=)

|  |
| --- |
| [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 |
|  | Address­Type­ID Primary key for Address­Type records. | int | 4 | NOT NULL | 1 - 1 |  |
|  | Name Address type description. For example, Billing, Home, or Shipping. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 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\_­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 |

SQL Script

|  |
| --- |
| CREATE TABLE [Person].[Address­Type]  (  [Address­Type­ID] [int] NOT NULL IDENTITY(1, 1),  [Name] [dbo].[Name] NOT NULL,  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Address­Type\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Address­Type\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Person].[Address­Type] ADD CONSTRAINT [PK\_­Address­Type\_­Address­Type­ID] PRIMARY KEY CLUSTERED ([Address­Type­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Address­Type\_­Name] ON [Person].[Address­Type] ([Name]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Address­Type\_rowguid] ON [Person].[Address­Type] ([rowguid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Types of addresses stored in the Address table. ', 'SCHEMA', N'Person', 'TABLE', N'Address­Type', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Address­Type records.', 'SCHEMA', N'Person', 'TABLE', N'Address­Type', 'COLUMN', N'Address­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Person', 'TABLE', N'Address­Type', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Address type description. For example, Billing, Home, or Shipping.', 'SCHEMA', N'Person', 'TABLE', N'Address­Type', 'COLUMN', N'Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Person', 'TABLE', N'Address­Type', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'Address­Type', 'CONSTRAINT', N'DF\_­Address­Type\_­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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Person', 'TABLE', N'Address­Type', 'CONSTRAINT', N'PK\_­Address­Type\_­Address­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Person', 'TABLE', N'Address­Type', 'INDEX', N'AK\_­Address­Type\_­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Person', 'TABLE', N'Address­Type', 'INDEX', N'AK\_­Address­Type\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'Address­Type', 'INDEX', N'PK\_­Address­Type\_­Address­Type­ID'  GO |

Uses

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Person](#GSXHhzVRNjGQYtnCQHosa2uthyw=)

Used By

[[Person].[Business­Entity­Address]](#HK6CehnfZ4YYh7ckP8cx74El9pY=)

[[Purchasing].[v­Vendor­With­Addresses]](#NdlQ62Q6DZw3GS30J9H9dm/SOqg=)

[[Sales].[v­Individual­Customer]](#WcBS54CxROpNMbufsyU5V6t9Ykw=)

[[Sales].[v­Store­With­Addresses]](#lyHdOUZeccdbPaj16bKzv2RjJKw=)

|  |
| --- |
| [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) | Nullability | Identity | Identity Replication | Default |
|  | Business­Entity­ID Primary key for all customers, vendors, and employees. | int | 4 | NOT NULL | 1 - 1 | False |  |
|  | 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\_­Business­Entity\_­Business­Entity­ID Primary key (clustered) constraint | Business­Entity­ID | True |
|  | AK\_­Business­Entity\_rowguid Unique nonclustered index. Used to support replication samples. | rowguid | True |

SQL Script

|  |
| --- |
| CREATE TABLE [Person].[Business­Entity]  (  [Business­Entity­ID] [int] NOT NULL IDENTITY(1, 1) NOT FOR REPLICATION,  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Business­Entity\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Business­Entity\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Person].[Business­Entity] ADD CONSTRAINT [PK\_­Business­Entity\_­Business­Entity­ID] PRIMARY KEY CLUSTERED ([Business­Entity­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Business­Entity\_rowguid] ON [Person].[Business­Entity] ([rowguid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Source of the ID that connects vendors, customers, and employees with address and contact information.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for all customers, vendors, and employees.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity', 'COLUMN', N'Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity', 'CONSTRAINT', N'DF\_­Business­Entity\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity', 'CONSTRAINT', N'DF\_­Business­Entity\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity', 'CONSTRAINT', N'PK\_­Business­Entity\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity', 'INDEX', N'AK\_­Business­Entity\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity', 'INDEX', N'PK\_­Business­Entity\_­Business­Entity­ID'  GO |

Uses

[Person](#GSXHhzVRNjGQYtnCQHosa2uthyw=)

Used By

[[Person].[Business­Entity­Address]](#HK6CehnfZ4YYh7ckP8cx74El9pY=)

[[Person].[Business­Entity­Contact]](#9Zj2hWeFVzGQn5EUQXP6lLMtAZI=)

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[[Purchasing].[Vendor]](#yfc9dLywcQtxtbztLjV6J3bb0tM=)

[[Sales].[Store]](#o4yE5vuw4wTVa38E7sxNcXx3KhA=)

|  |
| --- |
| [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 |
|  | Business­Entity­ID Primary key. Foreign key to Business­Entity.Business­Entity­ID. | int | 4 | NOT NULL |  |
|  | Address­ID Primary key. Foreign key to Address.Address­ID. | int | 4 | NOT NULL |  |
|  | Address­Type­ID Primary key. Foreign key to Address­Type.Address­Type­ID. | 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\_­Business­Entity­Address\_­Business­Entity­ID\_­Address­ID\_­Address­Type­ID Primary key (clustered) constraint | Business­Entity­ID, 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]](#YHZkNb8Ts04ByOupF/OAfoeI2qk=) |
| FK\_­Business­Entity­Address\_­Address­Type\_­Address­Type­ID Foreign key constraint referencing Address­Type.Address­Type­ID. | Address­Type­ID->[[Person].[Address­Type].[Address­Type­ID]](#sC112VucbUsBFhIBrJnCpuBv1os=) |
| FK\_­Business­Entity­Address\_­Business­Entity\_­Business­Entity­ID Foreign key constraint referencing Business­Entity.Business­Entity­ID. | Business­Entity­ID->[[Person].[Business­Entity].[Business­Entity­ID]](#I7t6VoTbrv7x5ZQ/kE/EZs1npdg=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Person].[Business­Entity­Address]  (  [Business­Entity­ID] [int] NOT NULL,  [Address­ID] [int] NOT NULL,  [Address­Type­ID] [int] NOT NULL,  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Business­Entity­Address\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Business­Entity­Address\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Person].[Business­Entity­Address] ADD CONSTRAINT [PK\_­Business­Entity­Address\_­Business­Entity­ID\_­Address­ID\_­Address­Type­ID] PRIMARY KEY CLUSTERED ([Business­Entity­ID], [Address­ID], [Address­Type­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Business­Entity­Address\_­Address­ID] ON [Person].[Business­Entity­Address] ([Address­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Business­Entity­Address\_­Address­Type­ID] ON [Person].[Business­Entity­Address] ([Address­Type­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Business­Entity­Address\_rowguid] ON [Person].[Business­Entity­Address] ([rowguid]) ON [PRIMARY]  GO  ALTER TABLE [Person].[Business­Entity­Address] ADD CONSTRAINT [FK\_­Business­Entity­Address\_­Address\_­Address­ID] FOREIGN KEY ([Address­ID]) REFERENCES [Person].[Address] ([Address­ID])  GO  ALTER TABLE [Person].[Business­Entity­Address] ADD CONSTRAINT [FK\_­Business­Entity­Address\_­Address­Type\_­Address­Type­ID] FOREIGN KEY ([Address­Type­ID]) REFERENCES [Person].[Address­Type] ([Address­Type­ID])  GO  ALTER TABLE [Person].[Business­Entity­Address] ADD CONSTRAINT [FK\_­Business­Entity­Address\_­Business­Entity\_­Business­Entity­ID] FOREIGN KEY ([Business­Entity­ID]) REFERENCES [Person].[Business­Entity] ([Business­Entity­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Cross-reference table mapping customers, vendors, and employees to their addresses.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Address', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Foreign key to Address.Address­ID.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Address', 'COLUMN', N'Address­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Foreign key to Address­Type.Address­Type­ID.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Address', 'COLUMN', N'Address­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Foreign key to Business­Entity.Business­Entity­ID.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Address', 'COLUMN', N'Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Address', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Address', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Address', 'CONSTRAINT', N'DF\_­Business­Entity­Address\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Address', 'CONSTRAINT', N'DF\_­Business­Entity­Address\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Address.Address­ID.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Address', 'CONSTRAINT', N'FK\_­Business­Entity­Address\_­Address\_­Address­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Address­Type.Address­Type­ID.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Address', 'CONSTRAINT', N'FK\_­Business­Entity­Address\_­Address­Type\_­Address­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Business­Entity.Business­Entity­ID.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Address', 'CONSTRAINT', N'FK\_­Business­Entity­Address\_­Business­Entity\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Address', 'CONSTRAINT', N'PK\_­Business­Entity­Address\_­Business­Entity­ID\_­Address­ID\_­Address­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Address', 'INDEX', N'AK\_­Business­Entity­Address\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Address', 'INDEX', N'IX\_­Business­Entity­Address\_­Address­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Address', 'INDEX', N'IX\_­Business­Entity­Address\_­Address­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Address', 'INDEX', N'PK\_­Business­Entity­Address\_­Business­Entity­ID\_­Address­ID\_­Address­Type­ID'  GO |

Uses

[[Person].[Address]](#YHZkNb8Ts04ByOupF/OAfoeI2qk=)

[[Person].[Address­Type]](#sC112VucbUsBFhIBrJnCpuBv1os=)

[[Person].[Business­Entity]](#I7t6VoTbrv7x5ZQ/kE/EZs1npdg=)

[Person](#GSXHhzVRNjGQYtnCQHosa2uthyw=)

Used By

[[Human­Resources].[v­Employee]](#TVXJJHSO4vPDsqDtNRd5tGQdsO0=)

[[Purchasing].[v­Vendor­With­Addresses]](#NdlQ62Q6DZw3GS30J9H9dm/SOqg=)

[[Sales].[v­Individual­Customer]](#WcBS54CxROpNMbufsyU5V6t9Ykw=)

[[Sales].[v­Sales­Person]](#pOjFS54tNwjD3Yvi8IMnn7dtRuA=)

[[Sales].[v­Store­With­Addresses]](#lyHdOUZeccdbPaj16bKzv2RjJKw=)

|  |
| --- |
| [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 |
|  | Business­Entity­ID Primary key. Foreign key to Business­Entity.Business­Entity­ID. | int | 4 | NOT NULL |  |
|  | Person­ID Primary key. Foreign key to Person.Business­Entity­ID. | int | 4 | NOT NULL |  |
|  | Contact­Type­ID Primary key. Foreign key to Contact­Type.Contact­Type­ID. | 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\_­Business­Entity­Contact\_­Business­Entity­ID\_­Person­ID\_­Contact­Type­ID Primary key (clustered) constraint | Business­Entity­ID, 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\_­Business­Entity­ID Foreign key constraint referencing Business­Entity.Business­Entity­ID. | Business­Entity­ID->[[Person].[Business­Entity].[Business­Entity­ID]](#I7t6VoTbrv7x5ZQ/kE/EZs1npdg=) |
| FK\_­Business­Entity­Contact\_­Contact­Type\_­Contact­Type­ID Foreign key constraint referencing Contact­Type.Contact­Type­ID. | Contact­Type­ID->[[Person].[Contact­Type].[Contact­Type­ID]](#uyTva1snlxlEidD4b0wN1HQzMw4=) |
| FK\_­Business­Entity­Contact\_­Person\_­Person­ID Foreign key constraint referencing Person.Business­Entity­ID. | Person­ID->[[Person].[Person].[Business­Entity­ID]](#TZe4JItyjmK+s/tUWju0b9W3HXU=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Person].[Business­Entity­Contact]  (  [Business­Entity­ID] [int] NOT NULL,  [Person­ID] [int] NOT NULL,  [Contact­Type­ID] [int] NOT NULL,  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Business­Entity­Contact\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Business­Entity­Contact\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Person].[Business­Entity­Contact] ADD CONSTRAINT [PK\_­Business­Entity­Contact\_­Business­Entity­ID\_­Person­ID\_­Contact­Type­ID] PRIMARY KEY CLUSTERED ([Business­Entity­ID], [Person­ID], [Contact­Type­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Business­Entity­Contact\_­Contact­Type­ID] ON [Person].[Business­Entity­Contact] ([Contact­Type­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Business­Entity­Contact\_­Person­ID] ON [Person].[Business­Entity­Contact] ([Person­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Business­Entity­Contact\_rowguid] ON [Person].[Business­Entity­Contact] ([rowguid]) ON [PRIMARY]  GO  ALTER TABLE [Person].[Business­Entity­Contact] ADD CONSTRAINT [FK\_­Business­Entity­Contact\_­Business­Entity\_­Business­Entity­ID] FOREIGN KEY ([Business­Entity­ID]) REFERENCES [Person].[Business­Entity] ([Business­Entity­ID])  GO  ALTER TABLE [Person].[Business­Entity­Contact] ADD CONSTRAINT [FK\_­Business­Entity­Contact\_­Contact­Type\_­Contact­Type­ID] FOREIGN KEY ([Contact­Type­ID]) REFERENCES [Person].[Contact­Type] ([Contact­Type­ID])  GO  ALTER TABLE [Person].[Business­Entity­Contact] ADD CONSTRAINT [FK\_­Business­Entity­Contact\_­Person\_­Person­ID] FOREIGN KEY ([Person­ID]) REFERENCES [Person].[Person] ([Business­Entity­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Cross-reference table mapping stores, vendors, and employees to people', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Contact', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Foreign key to Business­Entity.Business­Entity­ID.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Contact', 'COLUMN', N'Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Foreign key to Contact­Type.Contact­Type­ID.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Contact', 'COLUMN', N'Contact­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Contact', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Foreign key to Person.Business­Entity­ID.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Contact', 'COLUMN', N'Person­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Contact', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Contact', 'CONSTRAINT', N'DF\_­Business­Entity­Contact\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Contact', 'CONSTRAINT', N'DF\_­Business­Entity­Contact\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Business­Entity.Business­Entity­ID.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Contact', 'CONSTRAINT', N'FK\_­Business­Entity­Contact\_­Business­Entity\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Contact­Type.Contact­Type­ID.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Contact', 'CONSTRAINT', N'FK\_­Business­Entity­Contact\_­Contact­Type\_­Contact­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Person.Business­Entity­ID.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Contact', 'CONSTRAINT', N'FK\_­Business­Entity­Contact\_­Person\_­Person­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Contact', 'CONSTRAINT', N'PK\_­Business­Entity­Contact\_­Business­Entity­ID\_­Person­ID\_­Contact­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Person', 'TABLE', N'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\_­Contact­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Contact', 'INDEX', N'IX\_­Business­Entity­Contact\_­Person­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'Business­Entity­Contact', 'INDEX', N'PK\_­Business­Entity­Contact\_­Business­Entity­ID\_­Person­ID\_­Contact­Type­ID'  GO |

Uses

[[Person].[Business­Entity]](#I7t6VoTbrv7x5ZQ/kE/EZs1npdg=)

[[Person].[Contact­Type]](#uyTva1snlxlEidD4b0wN1HQzMw4=)

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[Person](#GSXHhzVRNjGQYtnCQHosa2uthyw=)

Used By

[[Purchasing].[v­Vendor­With­Contacts]](#kSLkxLBh/Shb6NobziJenpOvdFo=)

[[Sales].[v­Store­With­Contacts]](#HCZuGdamTtZV4FxWl+4pACIsexM=)

[[dbo].[ufn­Get­Contact­Information]](#MAbUSn/1caUuJ7XN6dczcbTRRyA=)

|  |
| --- |
| [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 |
|  | Contact­Type­ID Primary key for Contact­Type records. | int | 4 | NOT NULL | 1 - 1 |  |
|  | Name Contact type description. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NOT NULL |  |  |
|  | Modified­Date Date and time the record was last updated. | datetime | 8 | NOT NULL |  | (getdate()) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Contact­Type\_­Contact­Type­ID Primary key (clustered) constraint | Contact­Type­ID | True |
|  | AK\_­Contact­Type\_­Name Unique nonclustered index. | Name | True |

SQL Script

|  |
| --- |
| CREATE TABLE [Person].[Contact­Type]  (  [Contact­Type­ID] [int] NOT NULL IDENTITY(1, 1),  [Name] [dbo].[Name] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Contact­Type\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Person].[Contact­Type] ADD CONSTRAINT [PK\_­Contact­Type\_­Contact­Type­ID] PRIMARY KEY CLUSTERED ([Contact­Type­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Contact­Type\_­Name] ON [Person].[Contact­Type] ([Name]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Lookup table containing the types of business entity contacts.', 'SCHEMA', N'Person', 'TABLE', N'Contact­Type', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Contact­Type records.', 'SCHEMA', N'Person', 'TABLE', N'Contact­Type', 'COLUMN', N'Contact­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Person', 'TABLE', N'Contact­Type', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Contact type description.', 'SCHEMA', N'Person', 'TABLE', N'Contact­Type', 'COLUMN', N'Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'Contact­Type', 'CONSTRAINT', N'DF\_­Contact­Type\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Person', 'TABLE', N'Contact­Type', 'CONSTRAINT', N'PK\_­Contact­Type\_­Contact­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Person', 'TABLE', N'Contact­Type', 'INDEX', N'AK\_­Contact­Type\_­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'Contact­Type', 'INDEX', N'PK\_­Contact­Type\_­Contact­Type­ID'  GO |

Uses

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Person](#GSXHhzVRNjGQYtnCQHosa2uthyw=)

Used By

[[Person].[Business­Entity­Contact]](#9Zj2hWeFVzGQn5EUQXP6lLMtAZI=)

[[Purchasing].[v­Vendor­With­Contacts]](#kSLkxLBh/Shb6NobziJenpOvdFo=)

[[Sales].[v­Store­With­Contacts]](#HCZuGdamTtZV4FxWl+4pACIsexM=)

[[dbo].[ufn­Get­Contact­Information]](#MAbUSn/1caUuJ7XN6dczcbTRRyA=)

|  |
| --- |
| [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 |
|  | Country­Region­Code ISO standard code for countries and regions. | nvarchar(3) | 6 | NOT NULL |  |
|  | Name Country or region name. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NOT NULL |  |
|  | Modified­Date Date and time the record was last updated. | datetime | 8 | NOT NULL | (getdate()) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Country­Region\_­Country­Region­Code Primary key (clustered) constraint | Country­Region­Code | True |
|  | AK\_­Country­Region\_­Name Unique nonclustered index. | Name | True |

SQL Script

|  |
| --- |
| CREATE TABLE [Person].[Country­Region]  (  [Country­Region­Code] [nvarchar] (3) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Name] [dbo].[Name] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Country­Region\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Person].[Country­Region] ADD CONSTRAINT [PK\_­Country­Region\_­Country­Region­Code] PRIMARY KEY CLUSTERED ([Country­Region­Code]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Country­Region\_­Name] ON [Person].[Country­Region] ([Name]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Lookup table containing the ISO standard codes for countries and regions.', 'SCHEMA', N'Person', 'TABLE', N'Country­Region', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ISO standard code for countries and regions.', 'SCHEMA', N'Person', 'TABLE', N'Country­Region', 'COLUMN', N'Country­Region­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Person', 'TABLE', N'Country­Region', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Country or region name.', 'SCHEMA', N'Person', 'TABLE', N'Country­Region', 'COLUMN', N'Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'Country­Region', 'CONSTRAINT', N'DF\_­Country­Region\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Person', 'TABLE', N'Country­Region', 'CONSTRAINT', N'PK\_­Country­Region\_­Country­Region­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Person', 'TABLE', N'Country­Region', 'INDEX', N'AK\_­Country­Region\_­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'Country­Region', 'INDEX', N'PK\_­Country­Region\_­Country­Region­Code'  GO |

Uses

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Person](#GSXHhzVRNjGQYtnCQHosa2uthyw=)

Used By

[[Person].[State­Province]](#fYuToAkdHMSws7pXek8EIcFTfvY=)

[[Sales].[Country­Region­Currency]](#ZTDvJcpGJmjH9tZPEMXFxv/sN0o=)

[[Sales].[Sales­Territory]](#hvNbyL+2AhjcTkzEdMCOV3mg8XQ=)

[[Human­Resources].[v­Employee]](#TVXJJHSO4vPDsqDtNRd5tGQdsO0=)

[[Person].[v­State­Province­Country­Region]](#9RXeFj8QLY2V2kjrQ9Nq0mUcx70=)

[[Purchasing].[v­Vendor­With­Addresses]](#NdlQ62Q6DZw3GS30J9H9dm/SOqg=)

[[Sales].[v­Individual­Customer]](#WcBS54CxROpNMbufsyU5V6t9Ykw=)

[[Sales].[v­Sales­Person]](#pOjFS54tNwjD3Yvi8IMnn7dtRuA=)

[[Sales].[v­Store­With­Addresses]](#lyHdOUZeccdbPaj16bKzv2RjJKw=)

|  |
| --- |
| [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) | Nullability | Identity | Default |
|  | Business­Entity­ID Primary key. Person associated with this email address. Foreign key to Person.Business­Entity­ID | int | 4 | NOT NULL |  |  |
|  | 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\_­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.Business­Entity­ID. | Business­Entity­ID->[[Person].[Person].[Business­Entity­ID]](#TZe4JItyjmK+s/tUWju0b9W3HXU=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Person].[Email­Address]  (  [Business­Entity­ID] [int] NOT NULL,  [Email­Address­ID] [int] NOT NULL IDENTITY(1, 1),  [Email­Address] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Email­Address\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Email­Address\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Person].[Email­Address] ADD CONSTRAINT [PK\_­Email­Address\_­Business­Entity­ID\_­Email­Address­ID] PRIMARY KEY CLUSTERED ([Business­Entity­ID], [Email­Address­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Email­Address\_­Email­Address] ON [Person].[Email­Address] ([Email­Address]) ON [PRIMARY]  GO  ALTER TABLE [Person].[Email­Address] ADD CONSTRAINT [FK\_­Email­Address\_­Person\_­Business­Entity­ID] FOREIGN KEY ([Business­Entity­ID]) REFERENCES [Person].[Person] ([Business­Entity­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Where to send a person email.', 'SCHEMA', N'Person', 'TABLE', N'Email­Address', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Person associated with this email address. Foreign key to Person.Business­Entity­ID', 'SCHEMA', N'Person', 'TABLE', N'Email­Address', 'COLUMN', N'Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'E-mail address for the person.', 'SCHEMA', N'Person', 'TABLE', N'Email­Address', 'COLUMN', N'Email­Address'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. ID of this email address.', 'SCHEMA', N'Person', 'TABLE', N'Email­Address', 'COLUMN', N'Email­Address­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Person', 'TABLE', N'Email­Address', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Person', 'TABLE', N'Email­Address', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'Email­Address', 'CONSTRAINT', N'DF\_­Email­Address\_­Modified­Date'  GO  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\_­Business­Entity­ID\_­Email­Address­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Person', 'TABLE', N'Email­Address', 'INDEX', N'IX\_­Email­Address\_­Email­Address'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'Email­Address', 'INDEX', N'PK\_­Email­Address\_­Business­Entity­ID\_­Email­Address­ID'  GO |

Uses

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[Person](#GSXHhzVRNjGQYtnCQHosa2uthyw=)

Used By

[[Human­Resources].[v­Employee]](#TVXJJHSO4vPDsqDtNRd5tGQdsO0=)

[[Purchasing].[v­Vendor­With­Contacts]](#kSLkxLBh/Shb6NobziJenpOvdFo=)

[[Sales].[v­Individual­Customer]](#WcBS54CxROpNMbufsyU5V6t9Ykw=)

[[Sales].[v­Sales­Person]](#pOjFS54tNwjD3Yvi8IMnn7dtRuA=)

[[Sales].[v­Store­With­Contacts]](#HCZuGdamTtZV4FxWl+4pACIsexM=)

|  |
| --- |
| [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 |
|  | 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 |
|  | PK\_­Password\_­Business­Entity­ID Primary key (clustered) constraint | Business­Entity­ID | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Password\_­Person\_­Business­Entity­ID Foreign key constraint referencing Person.Business­Entity­ID. | Business­Entity­ID->[[Person].[Person].[Business­Entity­ID]](#TZe4JItyjmK+s/tUWju0b9W3HXU=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Person].[Password]  (  [Business­Entity­ID] [int] NOT NULL,  [Password­Hash] [varchar] (128) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Password­Salt] [varchar] (10) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Password\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Password\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Person].[Password] ADD CONSTRAINT [PK\_­Password\_­Business­Entity­ID] PRIMARY KEY CLUSTERED ([Business­Entity­ID]) ON [PRIMARY]  GO  ALTER TABLE [Person].[Password] ADD CONSTRAINT [FK\_­Password\_­Person\_­Business­Entity­ID] FOREIGN KEY ([Business­Entity­ID]) REFERENCES [Person].[Person] ([Business­Entity­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'One way hashed authentication information', 'SCHEMA', N'Person', 'TABLE', N'Password', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Person', 'TABLE', N'Password', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Password for the e-mail account.', 'SCHEMA', N'Person', 'TABLE', N'Password', 'COLUMN', N'Password­Hash'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Random value concatenated with the password string before the password is hashed.', 'SCHEMA', N'Person', 'TABLE', N'Password', 'COLUMN', N'Password­Salt'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Person', 'TABLE', N'Password', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'Password', 'CONSTRAINT', N'DF\_­Password\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Person', 'TABLE', N'Password', 'CONSTRAINT', N'DF\_­Password\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Person.Business­Entity­ID.', 'SCHEMA', N'Person', 'TABLE', N'Password', 'CONSTRAINT', N'FK\_­Password\_­Person\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Person', 'TABLE', N'Password', 'CONSTRAINT', N'PK\_­Password\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'Password', 'INDEX', N'PK\_­Password\_­Business­Entity­ID'  GO |

Uses

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[Person](#GSXHhzVRNjGQYtnCQHosa2uthyw=)

|  |
| --- |
| [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) | Nullability | Default |
|  | Business­Entity­ID Primary key for Person records. | int | 4 | NOT NULL |  |
|  | 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]](#WOmoxZnY6bzyL6GPFaVQqJCk+Sw=) | 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]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NOT NULL |  |
|  | Middle­Name Middle name or middle initial of the person. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NULL allowed |  |
|  | Last­Name Last name of the person. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NOT NULL |  |
|  | Suffix Surname suffix. For example, Sr. or Jr. | nvarchar(10) | 20 | NULL allowed |  |
|  | 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].[Additional­Contact­Info­Schema­Collection])](#KOFxa4KV3R/J1jPMka6b7eLe2ZM=) | max | NULL allowed |  |
| (4) | Demographics Personal information such as hobbies, and income collected from online shoppers. Used for sales analysis. | [xml([Person].[Individual­Survey­Schema­Collection])](#KUt3GRqxMZv6wiyAwm6mIpRjcgc=) | 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 | (getdate()) |
|  | Pet­ID | int | 4 | NULL allowed |  |

Indexes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Key Columns | Type | Unique | XML Type |
|  | PK\_­Person\_­Business­Entity­ID Primary key (clustered) constraint | Business­Entity­ID |  | 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. | Additional­Contact­Info | xml |  | Primary |
|  | PXML\_­Person\_­Demographics Primary XML index. | Demographics | xml |  | Primary |
|  | XMLPATH\_­Person\_­Demographics Secondary XML index for path. | Demographics | xml |  | Secondary |
|  | XMLPROPERTY\_­Person\_­Demographics Secondary XML index for property. | Demographics | xml |  | Secondary |
|  | XMLVALUE\_­Person\_­Demographics Secondary XML index for value. | Demographics | 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 Modified­Date 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 | ([Email­Promotion]>=(0) AND [Email­Promotion]<=(2)) |
| CK\_­Person\_­Person­Type Check constraint [Person­Type] is one of SC, VC, IN, EM or SP. | Person­Type | ([Person­Type] IS NULL OR upper([Person­Type])='GC' OR upper([Person­Type])='SP' OR upper([Person­Type])='EM' OR upper([Person­Type])='IN' OR upper([Person­Type])='VC' OR upper([Person­Type])='SC') |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Person\_­Business­Entity\_­Business­Entity­ID Foreign key constraint referencing Business­Entity.Business­Entity­ID. | Business­Entity­ID->[[Person].[Business­Entity].[Business­Entity­ID]](#I7t6VoTbrv7x5ZQ/kE/EZs1npdg=) |
| FK\_­Person\_­Pet­ID | Pet­ID->[[Person].[Pet].[Pet­ID]](#etnWX6SZkHYAYL8t5N4pzt7zXKE=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Person].[Person]  (  [Business­Entity­ID] [int] NOT NULL,  [Person­Type] [nchar] (2) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Name­Style] [dbo].[Name­Style] NOT NULL CONSTRAINT [DF\_­Person\_­Name­Style] DEFAULT ((0)),  [Title] [nvarchar] (8) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [First­Name] [dbo].[Name] NOT NULL,  [Middle­Name] [dbo].[Name] NULL,  [Last­Name] [dbo].[Name] NOT NULL,  [Suffix] [nvarchar] (10) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Email­Promotion] [int] NOT NULL CONSTRAINT [DF\_­Person\_­Email­Promotion] DEFAULT ((0)),  [Additional­Contact­Info] [xml] (CONTENT [Person].[Additional­Contact­Info­Schema­Collection]) NULL,  [Demographics] [xml] (CONTENT [Person].[Individual­Survey­Schema­Collection]) NULL,  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Person\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Person\_­Modified­Date] DEFAULT (getdate()),  [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 xmlns="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Individual­Survey">  <Total­Purchase­YTD>0.00</Total­Purchase­YTD>  </Individual­Survey>'  FROM inserted  WHERE [Person].[Person].[Business­Entity­ID] = inserted.[Business­Entity­ID]  AND inserted.[Demographics] IS NULL;    UPDATE [Person].[Person]  SET [Demographics].modify(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Individual­Survey";  insert <Total­Purchase­YTD>0.00</Total­Purchase­YTD>  as first  into (/Individual­Survey)[1]')  FROM inserted  WHERE [Person].[Person].[Business­Entity­ID] = inserted.[Business­Entity­ID]  AND inserted.[Demographics] IS NOT NULL  AND inserted.[Demographics].exist(N'declare default element namespace  "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Individual­Survey";  /Individual­Survey/Total­Purchase­YTD') <> 1;  END;  END;  GO  ALTER TABLE [Person].[Person] ADD CONSTRAINT [CK\_­Person\_­Email­Promotion] CHECK (([Email­Promotion]>=(0) AND [Email­Promotion]<=(2)))  GO  ALTER TABLE [Person].[Person] ADD CONSTRAINT [CK\_­Person\_­Person­Type] CHECK (([Person­Type] IS NULL OR upper([Person­Type])='GC' OR upper([Person­Type])='SP' OR upper([Person­Type])='EM' OR upper([Person­Type])='IN' OR upper([Person­Type])='VC' OR upper([Person­Type])='SC'))  GO  ALTER TABLE [Person].[Person] ADD CONSTRAINT [PK\_­Person\_­Business­Entity­ID] PRIMARY KEY CLUSTERED ([Business­Entity­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Person\_­Last­Name\_­First­Name\_­Middle­Name] ON [Person].[Person] ([Last­Name], [First­Name], [Middle­Name]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Person\_rowguid] ON [Person].[Person] ([rowguid]) ON [PRIMARY]  GO  CREATE PRIMARY XML INDEX [PXML\_­Person\_­Add­Contact]  ON [Person].[Person] ([Additional­Contact­Info])  GO  CREATE PRIMARY XML INDEX [PXML\_­Person\_­Demographics]  ON [Person].[Person] ([Demographics])  GO  CREATE XML INDEX [XMLPATH\_­Person\_­Demographics]  ON [Person].[Person] ([Demographics])  USING XML INDEX [PXML\_­Person\_­Demographics]  FOR PATH  GO  CREATE XML INDEX [XMLPROPERTY\_­Person\_­Demographics]  ON [Person].[Person] ([Demographics])  USING XML INDEX [PXML\_­Person\_­Demographics]  FOR PROPERTY  GO  CREATE XML INDEX [XMLVALUE\_­Person\_­Demographics]  ON [Person].[Person] ([Demographics])  USING XML INDEX [PXML\_­Person\_­Demographics]  FOR VALUE  GO  ALTER TABLE [Person].[Person] ADD CONSTRAINT [FK\_­Person\_­Business­Entity\_­Business­Entity­ID] FOREIGN KEY ([Business­Entity­ID]) REFERENCES [Person].[Business­Entity] ([Business­Entity­ID])  GO  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 Adventure­Works: employees, customer contacts, and vendor contacts.', 'SCHEMA', N'Person', 'TABLE', N'Person', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Additional contact information about the person stored in xml format. ', 'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'Additional­Contact­Info'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Person records.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Personal information such as hobbies, and income collected from online shoppers. Used for sales analysis.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'Demographics'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = Contact does not wish to receive e-mail promotions, 1 = Contact does wish to receive e-mail promotions from Adventure­Works, 2 = Contact does wish to receive e-mail promotions from Adventure­Works and selected partners. ', 'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'Email­Promotion'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'First name of the person.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'First­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Last name of the person.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'Last­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Middle name or middle initial of the person.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'Middle­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = The data in First­Name and 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'Name­Style'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary type of person: SC = Store Contact, IN = Individual (retail) customer, SP = Sales person, EM = Employee (non-sales), VC = Vendor contact, GC = General contact', 'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'Person­Type'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Surname suffix. For example, Sr. or Jr.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'Suffix'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'A courtesy title. For example, Mr. or Ms.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'COLUMN', N'Title'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Email­Promotion] >= (0) AND [Email­Promotion] <= (2)', 'SCHEMA', N'Person', 'TABLE', N'Person', 'CONSTRAINT', N'CK\_­Person\_­Email­Promotion'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Person­Type] is one of SC, VC, IN, EM or SP.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'CONSTRAINT', N'CK\_­Person\_­Person­Type'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0', 'SCHEMA', N'Person', 'TABLE', N'Person', 'CONSTRAINT', N'DF\_­Person\_­Email­Promotion'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'Person', 'CONSTRAINT', N'DF\_­Person\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0', 'SCHEMA', N'Person', 'TABLE', N'Person', 'CONSTRAINT', N'DF\_­Person\_­Name­Style'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Person', 'TABLE', N'Person', 'CONSTRAINT', N'DF\_­Person\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Business­Entity.Business­Entity­ID.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'CONSTRAINT', N'FK\_­Person\_­Business­Entity\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Person', 'TABLE', N'Person', 'CONSTRAINT', N'PK\_­Person\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'INDEX', N'AK\_­Person\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'INDEX', N'PK\_­Person\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary XML index.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'INDEX', N'PXML\_­Person\_­Add­Contact'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary XML index.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'INDEX', N'PXML\_­Person\_­Demographics'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Secondary XML index for path.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'INDEX', N'XMLPATH\_­Person\_­Demographics'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Secondary XML index for property.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'INDEX', N'XMLPROPERTY\_­Person\_­Demographics'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Secondary XML index for value.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'INDEX', N'XMLVALUE\_­Person\_­Demographics'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'AFTER INSERT, UPDATE trigger inserting Individual only if the Customer does not exist in the Store table and setting the Modified­Date column in the Person table to the current date.', 'SCHEMA', N'Person', 'TABLE', N'Person', 'TRIGGER', N'iu­Person'  GO |

Uses

[[Person].[Business­Entity]](#I7t6VoTbrv7x5ZQ/kE/EZs1npdg=)

[[Person].[Pet]](#etnWX6SZkHYAYL8t5N4pzt7zXKE=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[[dbo].[Name­Style]](#WOmoxZnY6bzyL6GPFaVQqJCk+Sw=)

[Person](#GSXHhzVRNjGQYtnCQHosa2uthyw=)

[[Person].[Additional­Contact­Info­Schema­Collection]](#KOFxa4KV3R/J1jPMka6b7eLe2ZM=)

[[Person].[Individual­Survey­Schema­Collection]](#KUt3GRqxMZv6wiyAwm6mIpRjcgc=)

Used By

[[Human­Resources].[Employee]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=)

[[Person].[Business­Entity­Contact]](#9Zj2hWeFVzGQn5EUQXP6lLMtAZI=)

[[Person].[Email­Address]](#3EL9ymWvglf8ADJZjk1nQ8j/hLY=)

[[Person].[Password]](#z1Ylcf/ZAQ6li+sOkPO88E0ccrw=)

[[Person].[Person­Phone]](#NyzXUZBpBs0rFeuVHGsZlpmRP0Y=)

[[Person].[Pet]](#etnWX6SZkHYAYL8t5N4pzt7zXKE=)

[[Sales].[Customer]](#UK53WWAjbLvEqOKY0KByf769x9M=)

[[Sales].[Person­Credit­Card]](#HOEWmD+eQ6y/+8bq8hkiLEiBBjU=)

[[Human­Resources].[v­Employee]](#TVXJJHSO4vPDsqDtNRd5tGQdsO0=)

[[Human­Resources].[v­Employee­Department]](#JhbW+DYcE93jITPkfvVrtIWEHzE=)

[[Human­Resources].[v­Employee­Department­History]](#Ppb9efXEcRDxKIGzLbHZkOB+Nto=)

[[Person].[v­Additional­Contact­Info]](#VDCcGn5Gs1OPJIG7Ywjk/m2DaqY=)

[[Purchasing].[v­Vendor­With­Contacts]](#kSLkxLBh/Shb6NobziJenpOvdFo=)

[[Sales].[v­Individual­Customer]](#WcBS54CxROpNMbufsyU5V6t9Ykw=)

[[Sales].[v­Person­Demographics]](#rOUy+E6WBf1AY40+Q9SudpuP9xU=)

[[Sales].[v­Sales­Person]](#pOjFS54tNwjD3Yvi8IMnn7dtRuA=)

[[Sales].[v­Sales­Person­Sales­By­Fiscal­Years]](#GfF5ZKdsT1+xL6X5ks+ghnXIfng=)

[[Sales].[v­Store­With­Contacts]](#HCZuGdamTtZV4FxWl+4pACIsexM=)

[[dbo].[usp­Get­Employee­Managers]](#yaIVF+o80FMpJygtFV827IXwYrg=)

[[dbo].[usp­Get­Manager­Employees]](#j0llh4oBM6uN5VRP3n9vBDBGwbc=)

[[dbo].[ufn­Get­Contact­Information]](#MAbUSn/1caUuJ7XN6dczcbTRRyA=)

|  |
| --- |
| [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 |
|  | Business­Entity­ID Business entity identification number. Foreign key to Person.Business­Entity­ID. | int | 4 | NOT NULL |  |
|  | Phone­Number Telephone number identification number. | [[dbo].[Phone]](#Dz5dpqul0rw8eU0/JuKLh0EO1ek=) | 50 | NOT NULL |  |
|  | 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 |
|  | PK\_­Person­Phone\_­Business­Entity­ID\_­Phone­Number\_­Phone­Number­Type­ID Primary key (clustered) constraint | Business­Entity­ID, Phone­Number, Phone­Number­Type­ID | True |
|  | IX\_­Person­Phone\_­Phone­Number Nonclustered index. | Phone­Number |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Person­Phone\_­Person\_­Business­Entity­ID Foreign key constraint referencing Person.Business­Entity­ID. | Business­Entity­ID->[[Person].[Person].[Business­Entity­ID]](#TZe4JItyjmK+s/tUWju0b9W3HXU=) |
| FK\_­Person­Phone\_­Phone­Number­Type\_­Phone­Number­Type­ID Foreign key constraint referencing Phone­Number­Type.Phone­Number­Type­ID. | Phone­Number­Type­ID->[[Person].[Phone­Number­Type].[Phone­Number­Type­ID]](#05nCels02cq7i+PWSDjMFGEmWFU=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Person].[Person­Phone]  (  [Business­Entity­ID] [int] NOT NULL,  [Phone­Number] [dbo].[Phone] NOT NULL,  [Phone­Number­Type­ID] [int] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Person­Phone\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Person].[Person­Phone] ADD CONSTRAINT [PK\_­Person­Phone\_­Business­Entity­ID\_­Phone­Number\_­Phone­Number­Type­ID] PRIMARY KEY CLUSTERED ([Business­Entity­ID], [Phone­Number], [Phone­Number­Type­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Person­Phone\_­Phone­Number] ON [Person].[Person­Phone] ([Phone­Number]) ON [PRIMARY]  GO  ALTER TABLE [Person].[Person­Phone] ADD CONSTRAINT [FK\_­Person­Phone\_­Person\_­Business­Entity­ID] FOREIGN KEY ([Business­Entity­ID]) REFERENCES [Person].[Person] ([Business­Entity­ID])  GO  ALTER TABLE [Person].[Person­Phone] ADD CONSTRAINT [FK\_­Person­Phone\_­Phone­Number­Type\_­Phone­Number­Type­ID] FOREIGN KEY ([Phone­Number­Type­ID]) REFERENCES [Person].[Phone­Number­Type] ([Phone­Number­Type­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Telephone number and type of a person.', 'SCHEMA', N'Person', 'TABLE', N'Person­Phone', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Business entity identification number. Foreign key to Person.Business­Entity­ID.', 'SCHEMA', N'Person', 'TABLE', N'Person­Phone', 'COLUMN', N'Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Person', 'TABLE', N'Person­Phone', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Telephone number identification number.', 'SCHEMA', N'Person', 'TABLE', N'Person­Phone', 'COLUMN', N'Phone­Number'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Kind of phone number. Foreign key to Phone­Number­Type.Phone­Number­Type­ID.', 'SCHEMA', N'Person', 'TABLE', N'Person­Phone', 'COLUMN', N'Phone­Number­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'Person­Phone', 'CONSTRAINT', N'DF\_­Person­Phone\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Person.Business­Entity­ID.', 'SCHEMA', N'Person', 'TABLE', N'Person­Phone', 'CONSTRAINT', N'FK\_­Person­Phone\_­Person\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Phone­Number­Type.Phone­Number­Type­ID.', 'SCHEMA', N'Person', 'TABLE', N'Person­Phone', 'CONSTRAINT', N'FK\_­Person­Phone\_­Phone­Number­Type\_­Phone­Number­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Person', 'TABLE', N'Person­Phone', 'CONSTRAINT', N'PK\_­Person­Phone\_­Business­Entity­ID\_­Phone­Number\_­Phone­Number­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Person', 'TABLE', N'Person­Phone', 'INDEX', N'IX\_­Person­Phone\_­Phone­Number'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'Person­Phone', 'INDEX', N'PK\_­Person­Phone\_­Business­Entity­ID\_­Phone­Number\_­Phone­Number­Type­ID'  GO |

Uses

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[[Person].[Phone­Number­Type]](#05nCels02cq7i+PWSDjMFGEmWFU=)

[[dbo].[Phone]](#Dz5dpqul0rw8eU0/JuKLh0EO1ek=)

[Person](#GSXHhzVRNjGQYtnCQHosa2uthyw=)

Used By

[[Human­Resources].[v­Employee]](#TVXJJHSO4vPDsqDtNRd5tGQdsO0=)

[[Purchasing].[v­Vendor­With­Contacts]](#kSLkxLBh/Shb6NobziJenpOvdFo=)

[[Sales].[v­Individual­Customer]](#WcBS54CxROpNMbufsyU5V6t9Ykw=)

[[Sales].[v­Sales­Person]](#pOjFS54tNwjD3Yvi8IMnn7dtRuA=)

[[Sales].[v­Store­With­Contacts]](#HCZuGdamTtZV4FxWl+4pACIsexM=)

|  |
| --- |
| [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 |
|  | Pet­ID | int | 4 | NOT NULL |
|  | Name | nvarchar(25) | 50 | NULL allowed |
|  | Color | nvarchar(15) | 30 | NULL allowed |
|  | Type | nvarchar(15) | 30 | NULL allowed |
|  | Breed | nvarchar(25) | 50 | NULL allowed |
|  | Owner­ID | int | 4 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Pet | Pet­ID | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Pet\_­Owner­ID | Owner­ID->[[Person].[Person].[Business­Entity­ID]](#TZe4JItyjmK+s/tUWju0b9W3HXU=) |

SQL Script

|  |
| --- |
| 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 |

Uses

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[Person](#GSXHhzVRNjGQYtnCQHosa2uthyw=)

Used By

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

|  |
| --- |
| [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 |
|  | Phone­Number­Type­ID Primary key for telephone number type records. | int | 4 | NOT NULL | 1 - 1 |  |
|  | Name Name of the telephone number type | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NOT NULL |  |  |
|  | Modified­Date Date and time the record was last updated. | datetime | 8 | NOT NULL |  | (getdate()) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Phone­Number­Type\_­Phone­Number­Type­ID Primary key (clustered) constraint | Phone­Number­Type­ID | True |

SQL Script

|  |
| --- |
| CREATE TABLE [Person].[Phone­Number­Type]  (  [Phone­Number­Type­ID] [int] NOT NULL IDENTITY(1, 1),  [Name] [dbo].[Name] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Phone­Number­Type\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Person].[Phone­Number­Type] ADD CONSTRAINT [PK\_­Phone­Number­Type\_­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  GO  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'Phone­Number­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'Phone­Number­Type', 'CONSTRAINT', N'DF\_­Phone­Number­Type\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Person', 'TABLE', N'Phone­Number­Type', 'CONSTRAINT', N'PK\_­Phone­Number­Type\_­Phone­Number­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Person', 'TABLE', N'Phone­Number­Type', 'INDEX', N'PK\_­Phone­Number­Type\_­Phone­Number­Type­ID'  GO |

Uses

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Person](#GSXHhzVRNjGQYtnCQHosa2uthyw=)

Used By

[[Person].[Person­Phone]](#NyzXUZBpBs0rFeuVHGsZlpmRP0Y=)

[[Human­Resources].[v­Employee]](#TVXJJHSO4vPDsqDtNRd5tGQdsO0=)

[[Purchasing].[v­Vendor­With­Contacts]](#kSLkxLBh/Shb6NobziJenpOvdFo=)

[[Sales].[v­Individual­Customer]](#WcBS54CxROpNMbufsyU5V6t9Ykw=)

[[Sales].[v­Sales­Person]](#pOjFS54tNwjD3Yvi8IMnn7dtRuA=)

[[Sales].[v­Store­With­Contacts]](#HCZuGdamTtZV4FxWl+4pACIsexM=)

|  |
| --- |
| [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) | Nullability | Identity | Default |
|  | State­Province­ID Primary key for State­Province records. | int | 4 | NOT NULL | 1 - 1 |  |
|  | State­Province­Code ISO standard state or province code. | nchar(3) | 6 | NOT NULL |  |  |
|  | Country­Region­Code ISO standard country or region code. Foreign key to Country­Region.Country­Region­Code. | nvarchar(3) | 6 | NOT NULL |  |  |
|  | Is­Only­State­Province­Flag 0 = State­Province­Code exists. 1 = State­Province­Code unavailable, using Country­Region­Code. | [[dbo].[Flag]](#j88jjDtwCbK8khz/m8jxryH/SFk=) | 1 | NOT NULL |  | ((1)) |
|  | Name State or province description. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NOT NULL |  |  |
|  | Territory­ID ID of the territory in which the state or province is located. Foreign key to Sales­Territory.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\_­State­Province\_­State­Province­ID Primary key (clustered) constraint | State­Province­ID | True |
|  | AK\_­State­Province\_­Name Unique nonclustered index. | Name | True |
|  | AK\_­State­Province\_­State­Province­Code\_­Country­Region­Code Unique nonclustered index. | State­Province­Code, Country­Region­Code | 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 Country­Region.Country­Region­Code. | Country­Region­Code->[[Person].[Country­Region].[Country­Region­Code]](#efOGmVlBeofSPEY2jc0F91joV9Q=) |
| FK\_­State­Province\_­Sales­Territory\_­Territory­ID Foreign key constraint referencing Sales­Territory.Territory­ID. | Territory­ID->[[Sales].[Sales­Territory].[Territory­ID]](#hvNbyL+2AhjcTkzEdMCOV3mg8XQ=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Person].[State­Province]  (  [State­Province­ID] [int] NOT NULL IDENTITY(1, 1),  [State­Province­Code] [nchar] (3) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Country­Region­Code] [nvarchar] (3) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Is­Only­State­Province­Flag] [dbo].[Flag] NOT NULL CONSTRAINT [DF\_­State­Province\_­Is­Only­State­Province­Flag] DEFAULT ((1)),  [Name] [dbo].[Name] NOT NULL,  [Territory­ID] [int] NOT NULL,  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­State­Province\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­State­Province\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Person].[State­Province] ADD CONSTRAINT [PK\_­State­Province\_­State­Province­ID] PRIMARY KEY CLUSTERED ([State­Province­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­State­Province\_­Name] ON [Person].[State­Province] ([Name]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­State­Province\_rowguid] ON [Person].[State­Province] ([rowguid]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­State­Province\_­State­Province­Code\_­Country­Region­Code] ON [Person].[State­Province] ([State­Province­Code], [Country­Region­Code]) ON [PRIMARY]  GO  ALTER TABLE [Person].[State­Province] ADD CONSTRAINT [FK\_­State­Province\_­Country­Region\_­Country­Region­Code] FOREIGN KEY ([Country­Region­Code]) REFERENCES [Person].[Country­Region] ([Country­Region­Code])  GO  ALTER TABLE [Person].[State­Province] ADD CONSTRAINT [FK\_­State­Province\_­Sales­Territory\_­Territory­ID] FOREIGN KEY ([Territory­ID]) REFERENCES [Sales].[Sales­Territory] ([Territory­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'State and province lookup table.', 'SCHEMA', N'Person', 'TABLE', N'State­Province', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ISO standard country or region code. Foreign key to Country­Region.Country­Region­Code. ', 'SCHEMA', N'Person', 'TABLE', N'State­Province', 'COLUMN', N'Country­Region­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = State­Province­Code exists. 1 = State­Province­Code unavailable, using Country­Region­Code.', 'SCHEMA', N'Person', 'TABLE', N'State­Province', 'COLUMN', N'Is­Only­State­Province­Flag'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Person', 'TABLE', N'State­Province', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'State or province description.', 'SCHEMA', N'Person', 'TABLE', N'State­Province', 'COLUMN', N'Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Person', 'TABLE', N'State­Province', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ISO standard state or province code.', 'SCHEMA', N'Person', 'TABLE', N'State­Province', 'COLUMN', N'State­Province­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for State­Province records.', 'SCHEMA', N'Person', 'TABLE', N'State­Province', 'COLUMN', N'State­Province­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ID of the territory in which the state or province is located. Foreign key to Sales­Territory.Sales­Territory­ID.', 'SCHEMA', N'Person', 'TABLE', N'State­Province', 'COLUMN', N'Territory­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 1 (TRUE)', 'SCHEMA', N'Person', 'TABLE', N'State­Province', 'CONSTRAINT', N'DF\_­State­Province\_­Is­Only­State­Province­Flag'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Person', 'TABLE', N'State­Province', 'CONSTRAINT', N'DF\_­State­Province\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Person', 'TABLE', N'State­Province', 'CONSTRAINT', N'DF\_­State­Province\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Country­Region.Country­Region­Code.', 'SCHEMA', N'Person', 'TABLE', N'State­Province', 'CONSTRAINT', N'FK\_­State­Province\_­Country­Region\_­Country­Region­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Sales­Territory.Territory­ID.', 'SCHEMA', N'Person', 'TABLE', N'State­Province', 'CONSTRAINT', N'FK\_­State­Province\_­Sales­Territory\_­Territory­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Person', 'TABLE', N'State­Province', 'CONSTRAINT', N'PK\_­State­Province\_­State­Province­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', '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\_­State­Province\_­State­Province­ID'  GO |

Uses

[[Person].[Country­Region]](#efOGmVlBeofSPEY2jc0F91joV9Q=)

[[Sales].[Sales­Territory]](#hvNbyL+2AhjcTkzEdMCOV3mg8XQ=)

[[dbo].[Flag]](#j88jjDtwCbK8khz/m8jxryH/SFk=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Person](#GSXHhzVRNjGQYtnCQHosa2uthyw=)

Used By

[[Person].[Address]](#YHZkNb8Ts04ByOupF/OAfoeI2qk=)

[[Sales].[Sales­Tax­Rate]](#qRl1u9t1FSaWq4TfyLPf6H9xl4k=)

[[Human­Resources].[v­Employee]](#TVXJJHSO4vPDsqDtNRd5tGQdsO0=)

[[Person].[v­State­Province­Country­Region]](#9RXeFj8QLY2V2kjrQ9Nq0mUcx70=)

[[Purchasing].[v­Vendor­With­Addresses]](#NdlQ62Q6DZw3GS30J9H9dm/SOqg=)

[[Sales].[v­Individual­Customer]](#WcBS54CxROpNMbufsyU5V6t9Ykw=)

[[Sales].[v­Sales­Person]](#pOjFS54tNwjD3Yvi8IMnn7dtRuA=)

[[Sales].[v­Store­With­Addresses]](#lyHdOUZeccdbPaj16bKzv2RjJKw=)

|  |
| --- |
| [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 |
|  | 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 |  |  |
|  | 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 |  |  |
|  | 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 |
|  | PK\_­Bill­Of­Materials\_­Bill­Of­Materials­ID Primary key (clustered) constraint | Bill­Of­Materials­ID | True |
|  | AK\_­Bill­Of­Materials\_­Product­Assembly­ID\_­Component­ID\_­Start­Date Clustered index. | Product­Assembly­ID, Component­ID, Start­Date | True |
|  | IX\_­Bill­Of­Materials\_­Unit­Measure­Code Nonclustered index. | Unit­Measure­Code |  |

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) |  | ([Product­Assembly­ID] IS NULL AND [BOMLevel]=(0) AND [Per­Assembly­Qty]=(1.00) OR [Product­Assembly­ID] IS NOT NULL AND [BOMLevel]>=(1)) |
| CK\_­Bill­Of­Materials\_­End­Date Check constraint End­Date] > [Start­Date] OR [End­Date] IS NULL |  | ([End­Date]>[Start­Date] OR [End­Date] IS NULL) |
| CK\_­Bill­Of­Materials\_­Per­Assembly­Qty Check constraint [Per­Assembly­Qty] >= (1.00) | Per­Assembly­Qty | ([Per­Assembly­Qty]>=(1.00)) |
| CK\_­Bill­Of­Materials\_­Product­Assembly­ID Check constraint [Product­Assembly­ID] <> [Component­ID] |  | ([Product­Assembly­ID]<>[Component­ID]) |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Bill­Of­Materials\_­Product\_­Component­ID Foreign key constraint referencing Product.Component­ID. | Component­ID->[[Production].[Product].[Product­ID]](#mO6TpkyW7q4W/OMRHISuYfWNCME=) |
| FK\_­Bill­Of­Materials\_­Product\_­Product­Assembly­ID Foreign key constraint referencing Product.Product­Assembly­ID. | Product­Assembly­ID->[[Production].[Product].[Product­ID]](#mO6TpkyW7q4W/OMRHISuYfWNCME=) |
| FK\_­Bill­Of­Materials\_­Unit­Measure\_­Unit­Measure­Code Foreign key constraint referencing Unit­Measure.Unit­Measure­Code. | Unit­Measure­Code->[[Production].[Unit­Measure].[Unit­Measure­Code]](#11scPpeMkfJepgAP6eC9jOIFK+Y=) |

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\_­Per­Assembly­Qty] 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)))  GO  ALTER TABLE [Production].[Bill­Of­Materials] ADD CONSTRAINT [CK\_­Bill­Of­Materials\_­End­Date] CHECK (([End­Date]>[Start­Date] OR [End­Date] IS NULL))  GO  ALTER TABLE [Production].[Bill­Of­Materials] ADD CONSTRAINT [CK\_­Bill­Of­Materials\_­Per­Assembly­Qty] CHECK (([Per­Assembly­Qty]>=(1.00)))  GO  ALTER TABLE [Production].[Bill­Of­Materials] ADD CONSTRAINT [CK\_­Bill­Of­Materials\_­Product­Assembly­ID] CHECK (([Product­Assembly­ID]<>[Component­ID]))  GO  ALTER TABLE [Production].[Bill­Of­Materials] ADD CONSTRAINT [PK\_­Bill­Of­Materials\_­Bill­Of­Materials­ID] PRIMARY KEY NONCLUSTERED ([Bill­Of­Materials­ID]) ON [PRIMARY]  GO  CREATE UNIQUE CLUSTERED INDEX [AK\_­Bill­Of­Materials\_­Product­Assembly­ID\_­Component­ID\_­Start­Date] ON [Production].[Bill­Of­Materials] ([Product­Assembly­ID], [Component­ID], [Start­Date]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Bill­Of­Materials\_­Unit­Measure­Code] ON [Production].[Bill­Of­Materials] ([Unit­Measure­Code]) ON [PRIMARY]  GO  ALTER TABLE [Production].[Bill­Of­Materials] ADD CONSTRAINT [FK\_­Bill­Of­Materials\_­Product\_­Component­ID] FOREIGN KEY ([Component­ID]) REFERENCES [Production].[Product] ([Product­ID])  GO  ALTER TABLE [Production].[Bill­Of­Materials] ADD CONSTRAINT [FK\_­Bill­Of­Materials\_­Product\_­Product­Assembly­ID] FOREIGN KEY ([Product­Assembly­ID]) REFERENCES [Production].[Product] ([Product­ID])  GO  ALTER TABLE [Production].[Bill­Of­Materials] ADD CONSTRAINT [FK\_­Bill­Of­Materials\_­Unit­Measure\_­Unit­Measure­Code] FOREIGN KEY ([Unit­Measure­Code]) REFERENCES [Production].[Unit­Measure] ([Unit­Measure­Code])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Items required to make bicycles and bicycle subassemblies. It identifies the heirarchical relationship between a parent product and its components.', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Bill­Of­Materials records.', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', 'COLUMN', N'Bill­Of­Materials­ID'  GO  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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Component identification number. Foreign key to Product.Product­ID.', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', 'COLUMN', N'Component­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date the component stopped being used in the assembly item.', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', 'COLUMN', N'End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Quantity of the component needed to create the assembly.', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', 'COLUMN', N'Per­Assembly­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Parent product identification number. Foreign key to Product.Product­ID.', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', 'COLUMN', N'Product­Assembly­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date the component started being used in the assembly item.', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', 'COLUMN', N'Start­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Standard code identifying the unit of measure for the quantity.', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', 'COLUMN', N'Unit­Measure­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint End­Date] > [Start­Date] OR [End­Date] IS NULL', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', 'CONSTRAINT', N'CK\_­Bill­Of­Materials\_­End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Per­Assembly­Qty] >= (1.00)', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', 'CONSTRAINT', N'CK\_­Bill­Of­Materials\_­Per­Assembly­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Product­Assembly­ID] <> [Component­ID]', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', 'CONSTRAINT', N'CK\_­Bill­Of­Materials\_­Product­Assembly­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', 'CONSTRAINT', N'DF\_­Bill­Of­Materials\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 1.0', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', 'CONSTRAINT', N'DF\_­Bill­Of­Materials\_­Per­Assembly­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', 'CONSTRAINT', N'DF\_­Bill­Of­Materials\_­Start­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Product.Component­ID.', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', 'CONSTRAINT', N'FK\_­Bill­Of­Materials\_­Product\_­Component­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Product.Product­Assembly­ID.', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', 'CONSTRAINT', N'FK\_­Bill­Of­Materials\_­Product\_­Product­Assembly­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Unit­Measure.Unit­Measure­Code.', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', 'CONSTRAINT', N'FK\_­Bill­Of­Materials\_­Unit­Measure\_­Unit­Measure­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', 'CONSTRAINT', N'PK\_­Bill­Of­Materials\_­Bill­Of­Materials­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index.', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', 'INDEX', N'AK\_­Bill­Of­Materials\_­Product­Assembly­ID\_­Component­ID\_­Start­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', 'INDEX', N'IX\_­Bill­Of­Materials\_­Unit­Measure­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Bill­Of­Materials', 'INDEX', N'PK\_­Bill­Of­Materials\_­Bill­Of­Materials­ID'  GO |

Uses

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[[Production].[Unit­Measure]](#11scPpeMkfJepgAP6eC9jOIFK+Y=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

Used By

[[dbo].[usp­Get­Bill­Of­Materials]](#C41ile3bJ5ydReR7L8IkkBAf7Ac=)

[[dbo].[usp­Get­Where­Used­Product­ID]](#Nj+nzkRbS2L5x7zZW3glT02uJTA=)

|  |
| --- |
| [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 |
|  | Culture­ID Primary key for Culture records. | nchar(6) | 12 | NOT NULL |  |
|  | Name Culture description. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NOT NULL |  |
|  | Modified­Date Date and time the record was last updated. | datetime | 8 | NOT NULL | (getdate()) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Culture\_­Culture­ID Primary key (clustered) constraint | Culture­ID | True |
|  | AK\_­Culture\_­Name Unique nonclustered index. | Name | True |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Culture]  (  [Culture­ID] [nchar] (6) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Name] [dbo].[Name] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Culture\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Culture] ADD CONSTRAINT [PK\_­Culture\_­Culture­ID] PRIMARY KEY CLUSTERED ([Culture­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Culture\_­Name] ON [Production].[Culture] ([Name]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Lookup table containing the languages in which some Adventure­Works data is stored.', 'SCHEMA', N'Production', 'TABLE', N'Culture', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Culture records.', 'SCHEMA', N'Production', 'TABLE', N'Culture', 'COLUMN', N'Culture­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Culture', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Culture description.', 'SCHEMA', N'Production', 'TABLE', N'Culture', 'COLUMN', N'Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Culture', 'CONSTRAINT', N'DF\_­Culture\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Culture', 'CONSTRAINT', N'PK\_­Culture\_­Culture­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Culture', 'INDEX', N'AK\_­Culture\_­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Culture', 'INDEX', N'PK\_­Culture\_­Culture­ID'  GO |

Uses

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

Used By

[[Production].[Product­Model­Product­Description­Culture]](#Kkf63jUYCrQKelqjjcG51JBQDNI=)

|  |
| --- |
| [Production].[Document] |

MS\_­Description

Product maintenance documents.

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Full Text Catalog | [AW2016Full­Text­Catalog](#KQM1/2huuuPy+pMNx4F37D5+p0Y=) |
| 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 | Computed | Max Length (Bytes) | Nullability | Full Text Indexed | Language | Default |
|  | Document­Node Primary key for Document records. | hierarchyid |  | 892 | NOT NULL |  |  |  |
|  | Document­Level Depth in the document hierarchy. | smallint | True | 2 | NULL allowed |  |  |  |
|  | Title Title of the document. | nvarchar(50) |  | 100 | NOT NULL |  |  |  |
|  | Owner Employee who controls the document. Foreign key to Employee.Business­Entity­ID | int |  | 4 | NOT NULL |  |  |  |
|  | Folder­Flag 0 = This is a folder, 1 = This is a document. | bit |  | 1 | NOT NULL |  |  | ((0)) |
|  | 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 |  |  |  |
|  | 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 allowed | True | 1033 |  |
|  | Document Complete document. | varbinary(max) |  | max | NULL allowed | True | 1033 |  |
| (2) | rowguid ROWGUIDCOL number uniquely identifying the record. Required for File­Stream. | uniqueidentifier |  | 16 | NOT NULL |  |  | (newid()) |
|  | Modified­Date Date and time the record was last updated. | datetime |  | 8 | NOT NULL |  |  | (getdate()) |

Computed columns

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

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | 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 |
|  | UQ\_\_­Document\_\_­F73921F744672977 | 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.Business­Entity­ID. | Owner->[[Human­Resources].[Employee].[Business­Entity­ID]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Document]  (  [Document­Node] [sys].[hierarchyid] NOT NULL,  [Document­Level] AS ([Document­Node].[Get­Level]()),  [Title] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Owner] [int] NOT NULL,  [Folder­Flag] [bit] NOT NULL CONSTRAINT [DF\_­Document\_­Folder­Flag] DEFAULT ((0)),  [File­Name] [nvarchar] (400) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [File­Extension] [nvarchar] (8) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Revision] [nchar] (5) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Change­Number] [int] NOT NULL CONSTRAINT [DF\_­Document\_­Change­Number] DEFAULT ((0)),  [Status] [tinyint] NOT NULL,  [Document­Summary] [nvarchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Document] [varbinary] (max) NULL,  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Document\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Document\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Document] ADD CONSTRAINT [CK\_­Document\_­Status] CHECK (([Status]>=(1) AND [Status]<=(3)))  GO  ALTER TABLE [Production].[Document] ADD CONSTRAINT [PK\_­Document\_­Document­Node] PRIMARY KEY CLUSTERED ([Document­Node]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Document\_­Document­Level\_­Document­Node] ON [Production].[Document] ([Document­Level], [Document­Node]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Document\_­File­Name\_­Revision] ON [Production].[Document] ([File­Name], [Revision]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Document\_rowguid] ON [Production].[Document] ([rowguid]) ON [PRIMARY]  GO  ALTER TABLE [Production].[Document] ADD CONSTRAINT [UQ\_\_­Document\_\_­F73921F744672977] UNIQUE NONCLUSTERED ([rowguid]) ON [PRIMARY]  GO  ALTER TABLE [Production].[Document] ADD CONSTRAINT [FK\_­Document\_­Employee\_­Owner] FOREIGN KEY ([Owner]) REFERENCES [Human­Resources].[Employee] ([Business­Entity­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product maintenance documents.', 'SCHEMA', N'Production', 'TABLE', N'Document', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Engineering change approval number.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'Change­Number'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Complete document.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'Document'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Depth in the document hierarchy.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'Document­Level'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Document records.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'Document­Node'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Document abstract.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'Document­Summary'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'File extension indicating the document type. For example, .doc or .txt.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'File­Extension'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'File name of the document', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'File­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = This is a folder, 1 = This is a document.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'Folder­Flag'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Employee who controls the document. Foreign key to Employee.Business­Entity­ID', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'Owner'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Revision number of the document. ', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'Revision'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Required for File­Stream.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'1 = Pending approval, 2 = Approved, 3 = Obsolete', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'Status'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Title of the document.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'COLUMN', N'Title'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Status] BETWEEN (1) AND (3)', 'SCHEMA', N'Production', 'TABLE', N'Document', 'CONSTRAINT', N'CK\_­Document\_­Status'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0', 'SCHEMA', N'Production', 'TABLE', N'Document', 'CONSTRAINT', N'DF\_­Document\_­Change­Number'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Document', 'CONSTRAINT', N'DF\_­Document\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Production', 'TABLE', N'Document', 'CONSTRAINT', N'DF\_­Document\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Employee.Business­Entity­ID.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'CONSTRAINT', N'FK\_­Document\_­Employee\_­Owner'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Document', 'CONSTRAINT', N'PK\_­Document\_­Document­Node'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'INDEX', N'AK\_­Document\_­Document­Level\_­Document­Node'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support File­Stream.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'INDEX', N'AK\_­Document\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'INDEX', N'IX\_­Document\_­File­Name\_­Revision'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Document', 'INDEX', N'PK\_­Document\_­Document­Node'  GO  CREATE FULLTEXT INDEX ON [Production].[Document] KEY INDEX [PK\_­Document\_­Document­Node] ON [AW2016Full­Text­Catalog]  GO  ALTER FULLTEXT INDEX ON [Production].[Document] ADD ([Document­Summary] LANGUAGE 1033)  GO  ALTER FULLTEXT INDEX ON [Production].[Document] ADD ([Document] TYPE COLUMN [File­Extension] LANGUAGE 1033)  GO |

Uses

[[Human­Resources].[Employee]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

Used By

[[Production].[Product­Document]](#hFy0+8CYqgHPJaiGh0/YlUFf+eI=)

[AW2016Full­Text­Catalog](#KQM1/2huuuPy+pMNx4F37D5+p0Y=)

|  |
| --- |
| [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 |
|  | 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\_­Illustration\_­Illustration­ID Primary key (clustered) constraint | Illustration­ID | True |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Illustration]  (  [Illustration­ID] [int] NOT NULL IDENTITY(1, 1),  [Diagram] [xml] NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Illustration\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  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  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Illustrations used in manufacturing instructions. Stored as XML.', 'SCHEMA', N'Production', 'TABLE', N'Illustration', 'COLUMN', N'Diagram'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Illustration records.', 'SCHEMA', N'Production', 'TABLE', N'Illustration', 'COLUMN', N'Illustration­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Illustration', 'COLUMN', N'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  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Illustration', 'INDEX', N'PK\_­Illustration\_­Illustration­ID'  GO |

Uses

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

Used By

[[Production].[Product­Model­Illustration]](#ha0ZY5wgktKInuoqdKFBVRbeJaI=)

|  |
| --- |
| [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 |
|  | Location­ID Primary key for Location records. | smallint | 2 | NOT NULL | 1 - 1 |  |
|  | Name Location description. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 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\_­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 Check constraint [Availability] >= (0.00) | Availability | ([Availability]>=(0.00)) |
| CK\_­Location\_­Cost­Rate Check constraint [Cost­Rate] >= (0.00) | Cost­Rate | ([Cost­Rate]>=(0.00)) |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Location]  (  [Location­ID] [smallint] NOT NULL IDENTITY(1, 1),  [Name] [dbo].[Name] NOT NULL,  [Cost­Rate] [smallmoney] NOT NULL CONSTRAINT [DF\_­Location\_­Cost­Rate] DEFAULT ((0.00)),  [Availability] [decimal] (8, 2) NOT NULL CONSTRAINT [DF\_­Location\_­Availability] DEFAULT ((0.00)),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Location\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Location] ADD CONSTRAINT [CK\_­Location\_­Availability] CHECK (([Availability]>=(0.00)))  GO  ALTER TABLE [Production].[Location] ADD CONSTRAINT [CK\_­Location\_­Cost­Rate] CHECK (([Cost­Rate]>=(0.00)))  GO  ALTER TABLE [Production].[Location] ADD CONSTRAINT [PK\_­Location\_­Location­ID] PRIMARY KEY CLUSTERED ([Location­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Location\_­Name] ON [Production].[Location] ([Name]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product inventory and manufacturing locations.', 'SCHEMA', N'Production', 'TABLE', N'Location', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Work capacity (in hours) of the manufacturing location.', 'SCHEMA', N'Production', 'TABLE', N'Location', 'COLUMN', N'Availability'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Standard hourly cost of the manufacturing location.', 'SCHEMA', N'Production', 'TABLE', N'Location', 'COLUMN', N'Cost­Rate'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Location records.', 'SCHEMA', N'Production', 'TABLE', N'Location', 'COLUMN', N'Location­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Location', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Location description.', 'SCHEMA', N'Production', 'TABLE', N'Location', 'COLUMN', N'Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Availability] >= (0.00)', 'SCHEMA', N'Production', 'TABLE', N'Location', 'CONSTRAINT', N'CK\_­Location\_­Availability'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Cost­Rate] >= (0.00)', 'SCHEMA', N'Production', 'TABLE', N'Location', 'CONSTRAINT', N'CK\_­Location\_­Cost­Rate'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0.00', 'SCHEMA', N'Production', 'TABLE', N'Location', 'CONSTRAINT', N'DF\_­Location\_­Availability'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0.0', 'SCHEMA', N'Production', 'TABLE', N'Location', 'CONSTRAINT', N'DF\_­Location\_­Cost­Rate'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Location', 'CONSTRAINT', N'DF\_­Location\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Location', 'CONSTRAINT', N'PK\_­Location\_­Location­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Location', 'INDEX', N'AK\_­Location\_­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Location', 'INDEX', N'PK\_­Location\_­Location­ID'  GO |

Uses

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

Used By

[[Production].[Product­Inventory]](#JnGLivmNHPEgSuzGYyO0WefiqDw=)

[[Production].[Work­Order­Routing]](#wxYgGkEd/o/k/roV2tY7ZCLH+qE=)

|  |
| --- |
| [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 |
|  | Product­ID Primary key for Product records. | int | 4 | NOT NULL | 1 - 1 |  |
|  | Name Name of the product. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NOT NULL |  |  |
|  | Product­Number Unique product identification number. | nvarchar(25) | 50 | NOT NULL |  |  |
|  | Make­Flag 0 = Product is purchased, 1 = Product is manufactured in-house. | [[dbo].[Flag]](#j88jjDtwCbK8khz/m8jxryH/SFk=) | 1 | NOT NULL |  | ((1)) |
|  | Finished­Goods­Flag 0 = Product is not a salable item. 1 = Product is salable. | [[dbo].[Flag]](#j88jjDtwCbK8khz/m8jxryH/SFk=) | 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 |  |  |
|  | Size­Unit­Measure­Code Unit of measure for Size column. | nchar(3) | 6 | NULL allowed |  |  |
|  | 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 |  |  |
|  | Product­Subcategory­ID Product is a member of this product subcategory. Foreign key to Product­Sub­Category.Product­Sub­Category­ID. | int | 4 | NULL allowed |  |  |
|  | Product­Model­ID Product is a member of this product model. Foreign key to Product­Model.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 |  |  |
|  | 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\_­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]='M' OR [Class]='L' OR [Class] IS NULL | Class | (upper([Class])='H' OR upper([Class])='M' OR upper([Class])='L' OR [Class] IS NULL) |
| CK\_­Product\_­Days­To­Manufacture Check constraint [Days­To­Manufacture] >= (0) | Days­To­Manufacture | ([Days­To­Manufacture]>=(0)) |
| CK\_­Product\_­List­Price Check constraint [List­Price] >= (0.00) | List­Price | ([List­Price]>=(0.00)) |
| CK\_­Product\_­Product­Line Check constraint [Product­Line]='r' OR [Product­Line]='m' OR [Product­Line]='t' OR [Product­Line]='s' OR [Product­Line]='R' OR [Product­Line]='M' OR [Product­Line]='T' OR [Product­Line]='S' OR [Product­Line] IS NULL | Product­Line | (upper([Product­Line])='R' OR upper([Product­Line])='M' OR upper([Product­Line])='T' OR upper([Product­Line])='S' OR [Product­Line] IS NULL) |
| CK\_­Product\_­Reorder­Point Check constraint [Reorder­Point] > (0) | Reorder­Point | ([Reorder­Point]>(0)) |
| CK\_­Product\_­Safety­Stock­Level Check constraint [Safety­Stock­Level] > (0) | Safety­Stock­Level | ([Safety­Stock­Level]>(0)) |
| CK\_­Product\_­Sell­End­Date Check constraint [Sell­End­Date] >= [Sell­Start­Date] OR [Sell­End­Date] IS NULL |  | ([Sell­End­Date]>=[Sell­Start­Date] OR [Sell­End­Date] IS NULL) |
| CK\_­Product\_­Standard­Cost Check constraint [Safety­Stock­Level] > (0) | Standard­Cost | ([Standard­Cost]>=(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.Product­Model­ID. | Product­Model­ID->[[Production].[Product­Model].[Product­Model­ID]](#n9VdHBX3Ji5OWfmMzrHHD+J8HSM=) |
| FK\_­Product\_­Product­Subcategory\_­Product­Subcategory­ID Foreign key constraint referencing Product­Subcategory.Product­Subcategory­ID. | Product­Subcategory­ID->[[Production].[Product­Subcategory].[Product­Subcategory­ID]](#WthFJ9hmewdhBB85kY7NAjtek7Y=) |
| FK\_­Product\_­Unit­Measure\_­Size­Unit­Measure­Code Foreign key constraint referencing Unit­Measure.Unit­Measure­Code. | Size­Unit­Measure­Code->[[Production].[Unit­Measure].[Unit­Measure­Code]](#11scPpeMkfJepgAP6eC9jOIFK+Y=) |
| FK\_­Product\_­Unit­Measure\_­Weight­Unit­Measure­Code Foreign key constraint referencing Unit­Measure.Unit­Measure­Code. | Weight­Unit­Measure­Code->[[Production].[Unit­Measure].[Unit­Measure­Code]](#11scPpeMkfJepgAP6eC9jOIFK+Y=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Product]  (  [Product­ID] [int] NOT NULL IDENTITY(1, 1),  [Name] [dbo].[Name] NOT NULL,  [Product­Number] [nvarchar] (25) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Make­Flag] [dbo].[Flag] NOT NULL CONSTRAINT [DF\_­Product\_­Make­Flag] DEFAULT ((1)),  [Finished­Goods­Flag] [dbo].[Flag] NOT NULL CONSTRAINT [DF\_­Product\_­Finished­Goods­Flag] DEFAULT ((1)),  [Color] [nvarchar] (15) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Safety­Stock­Level] [smallint] NOT NULL,  [Reorder­Point] [smallint] NOT NULL,  [Standard­Cost] [money] NOT NULL,  [List­Price] [money] NOT NULL,  [Size] [nvarchar] (5) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Size­Unit­Measure­Code] [nchar] (3) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Weight­Unit­Measure­Code] [nchar] (3) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Weight] [decimal] (8, 2) NULL,  [Days­To­Manufacture] [int] NOT NULL,  [Product­Line] [nchar] (2) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Class] [nchar] (2) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Style] [nchar] (2) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Product­Subcategory­ID] [int] NULL,  [Product­Model­ID] [int] NULL,  [Sell­Start­Date] [datetime] NOT NULL,  [Sell­End­Date] [datetime] NULL,  [Discontinued­Date] [datetime] NULL,  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Product\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Product\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK\_­Product\_­Class] CHECK ((upper([Class])='H' OR upper([Class])='M' OR upper([Class])='L' OR [Class] IS NULL))  GO  ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK\_­Product\_­Days­To­Manufacture] CHECK (([Days­To­Manufacture]>=(0)))  GO  ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK\_­Product\_­List­Price] CHECK (([List­Price]>=(0.00)))  GO  ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK\_­Product\_­Product­Line] CHECK ((upper([Product­Line])='R' OR upper([Product­Line])='M' OR upper([Product­Line])='T' OR upper([Product­Line])='S' OR [Product­Line] IS NULL))  GO  ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK\_­Product\_­Reorder­Point] CHECK (([Reorder­Point]>(0)))  GO  ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK\_­Product\_­Safety­Stock­Level] CHECK (([Safety­Stock­Level]>(0)))  GO  ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK\_­Product\_­Sell­End­Date] CHECK (([Sell­End­Date]>=[Sell­Start­Date] OR [Sell­End­Date] IS NULL))  GO  ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK\_­Product\_­Standard­Cost] CHECK (([Standard­Cost]>=(0.00)))  GO  ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK\_­Product\_­Style] CHECK ((upper([Style])='U' OR upper([Style])='M' OR upper([Style])='W' OR [Style] IS NULL))  GO  ALTER TABLE [Production].[Product] ADD CONSTRAINT [CK\_­Product\_­Weight] CHECK (([Weight]>(0.00)))  GO  ALTER TABLE [Production].[Product] ADD CONSTRAINT [PK\_­Product\_­Product­ID] PRIMARY KEY CLUSTERED ([Product­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Product\_­Name] ON [Production].[Product] ([Name]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Product\_­Product­Number] ON [Production].[Product] ([Product­Number]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Product\_rowguid] ON [Production].[Product] ([rowguid]) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product] ADD CONSTRAINT [FK\_­Product\_­Product­Model\_­Product­Model­ID] FOREIGN KEY ([Product­Model­ID]) REFERENCES [Production].[Product­Model] ([Product­Model­ID])  GO  ALTER TABLE [Production].[Product] ADD CONSTRAINT [FK\_­Product\_­Product­Subcategory\_­Product­Subcategory­ID] FOREIGN KEY ([Product­Subcategory­ID]) REFERENCES [Production].[Product­Subcategory] ([Product­Subcategory­ID])  GO  ALTER TABLE [Production].[Product] ADD CONSTRAINT [FK\_­Product\_­Unit­Measure\_­Size­Unit­Measure­Code] FOREIGN KEY ([Size­Unit­Measure­Code]) REFERENCES [Production].[Unit­Measure] ([Unit­Measure­Code])  GO  ALTER TABLE [Production].[Product] ADD CONSTRAINT [FK\_­Product\_­Unit­Measure\_­Weight­Unit­Measure­Code] FOREIGN KEY ([Weight­Unit­Measure­Code]) REFERENCES [Production].[Unit­Measure] ([Unit­Measure­Code])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Products sold or used in the manfacturing of sold products.', 'SCHEMA', N'Production', 'TABLE', N'Product', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'H = High, M = Medium, L = Low', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Class'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product color.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Color'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Number of days required to manufacture the product.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Days­To­Manufacture'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date the product was discontinued.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Discontinued­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = Product is not a salable item. 1 = Product is salable.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Finished­Goods­Flag'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Selling price.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'List­Price'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = Product is purchased, 1 = Product is manufactured in-house.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Make­Flag'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Name of the product.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Product records.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'R = Road, M = Mountain, T = Touring, S = Standard', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Product­Line'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product is a member of this product model. Foreign key to Product­Model.Product­Model­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Product­Model­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique product identification number.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Product­Number'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product is a member of this product subcategory. Foreign key to Product­Sub­Category.Product­Sub­Category­ID. ', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Product­Subcategory­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Inventory level that triggers a purchase order or work order. ', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Reorder­Point'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Minimum inventory quantity. ', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Safety­Stock­Level'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date the product was no longer available for sale.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Sell­End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date the product was available for sale.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Sell­Start­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product size.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Size'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unit of measure for Size column.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Size­Unit­Measure­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Standard cost of the product.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Standard­Cost'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'W = Womens, M = Mens, U = Universal', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Style'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product weight.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Weight'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unit of measure for Weight column.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'COLUMN', N'Weight­Unit­Measure­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Class]=''h'' OR [Class]=''m'' OR [Class]=''l'' OR [Class]=''H'' OR [Class]=''M'' OR [Class]=''L'' OR [Class] IS NULL', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK\_­Product\_­Class'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Days­To­Manufacture] >= (0)', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK\_­Product\_­Days­To­Manufacture'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [List­Price] >= (0.00)', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK\_­Product\_­List­Price'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Product­Line]=''r'' OR [Product­Line]=''m'' OR [Product­Line]=''t'' OR [Product­Line]=''s'' OR [Product­Line]=''R'' OR [Product­Line]=''M'' OR [Product­Line]=''T'' OR [Product­Line]=''S'' OR [Product­Line] IS NULL', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK\_­Product\_­Product­Line'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Reorder­Point] > (0)', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK\_­Product\_­Reorder­Point'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Safety­Stock­Level] > (0)', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK\_­Product\_­Safety­Stock­Level'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Sell­End­Date] >= [Sell­Start­Date] OR [Sell­End­Date] IS NULL', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK\_­Product\_­Sell­End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Safety­Stock­Level] > (0)', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK\_­Product\_­Standard­Cost'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Style]=''u'' OR [Style]=''m'' OR [Style]=''w'' OR [Style]=''U'' OR [Style]=''M'' OR [Style]=''W'' OR [Style] IS NULL', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK\_­Product\_­Style'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Weight] > (0.00)', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'CK\_­Product\_­Weight'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 1', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'DF\_­Product\_­Finished­Goods­Flag'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 1', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'DF\_­Product\_­Make­Flag'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'DF\_­Product\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'DF\_­Product\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Product­Model.Product­Model­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'FK\_­Product\_­Product­Model\_­Product­Model­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Product­Subcategory.Product­Subcategory­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'FK\_­Product\_­Product­Subcategory\_­Product­Subcategory­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Unit­Measure.Unit­Measure­Code.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'FK\_­Product\_­Unit­Measure\_­Size­Unit­Measure­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Unit­Measure.Unit­Measure­Code.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'FK\_­Product\_­Unit­Measure\_­Weight­Unit­Measure­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Product', 'CONSTRAINT', N'PK\_­Product\_­Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'INDEX', N'AK\_­Product\_­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'INDEX', N'AK\_­Product\_­Product­Number'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'INDEX', N'AK\_­Product\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Product', 'INDEX', N'PK\_­Product\_­Product­ID'  GO |

Uses

[[Production].[Product­Model]](#n9VdHBX3Ji5OWfmMzrHHD+J8HSM=)

[[Production].[Product­Subcategory]](#WthFJ9hmewdhBB85kY7NAjtek7Y=)

[[Production].[Unit­Measure]](#11scPpeMkfJepgAP6eC9jOIFK+Y=)

[[dbo].[Flag]](#j88jjDtwCbK8khz/m8jxryH/SFk=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

Used By

[[Production].[Bill­Of­Materials]](#m9P+2olmZFXOowkUY18+Ri7YHx4=)

[[Production].[Product­Cost­History]](#V4hNECfDVL8rNsab6KfgsgmqeHE=)

[[Production].[Product­Document]](#hFy0+8CYqgHPJaiGh0/YlUFf+eI=)

[[Production].[Product­Inventory]](#JnGLivmNHPEgSuzGYyO0WefiqDw=)

[[Production].[Product­List­Price­History]](#lSXw7btC5MVIn+g9cscCbIFpkU0=)

[[Production].[Product­Product­Photo]](#40voS36ihwbm/xtmbxsgEQceBOQ=)

[[Production].[Product­Review]](#Z1+PLnel1CKvek/LdN60/nVl5cI=)

[[Production].[Transaction­History]](#qEI6rAj76QhivliqVRIQ5mR7wrA=)

[[Production].[Work­Order]](#BSRICvfPko6shp7o8HSu003oCno=)

[[Purchasing].[Product­Vendor]](#igWcZ2fEqwA+AlsF0xQn66oTGAU=)

[[Purchasing].[Purchase­Order­Detail]](#dsxg+I54iAv4FByATvQGW4l6Cco=)

[[Sales].[Shopping­Cart­Item]](#usOp9/vzTEpC3eW1PfV8jtNCfNQ=)

[[Sales].[Special­Offer­Product]](#8e4RMrihzXJLXw2gPdnNJ+2ijQ0=)

[[Production].[v­Product­And­Description]](#2wtLw+LERdCNthBlev7/7Ew1+JQ=)

[[dbo].[usp­Get­Bill­Of­Materials]](#C41ile3bJ5ydReR7L8IkkBAf7Ac=)

[[dbo].[usp­Get­Where­Used­Product­ID]](#Nj+nzkRbS2L5x7zZW3glT02uJTA=)

[[dbo].[ufn­Get­Product­Dealer­Price]](#qCHqC6Ssa6gRKHNZ1A5hYnf4CXs=)

[[dbo].[ufn­Get­Product­List­Price]](#8mn5CnDFl2xPNBEKQCwZ9YJdOMk=)

[[dbo].[ufn­Get­Product­Standard­Cost]](#iQHxQM8zJLh+qEeVmjb6xgEGWO4=)

|  |
| --- |
| [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 |
|  | Product­Category­ID Primary key for Product­Category records. | int | 4 | NOT NULL | 1 - 1 |  |
|  | Name Category description. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 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\_­Product­Category\_­Product­Category­ID Primary key (clustered) constraint | Product­Category­ID | 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\_­Product­Category\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Product­Category\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­Category] ADD CONSTRAINT [PK\_­Product­Category\_­Product­Category­ID] PRIMARY KEY CLUSTERED ([Product­Category­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Product­Category\_­Name] ON [Production].[Product­Category] ([Name]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Product­Category\_rowguid] ON [Production].[Product­Category] ([rowguid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'High-level product categorization.', 'SCHEMA', N'Production', 'TABLE', N'Product­Category', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Product­Category', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Category description.', 'SCHEMA', N'Production', 'TABLE', N'Product­Category', 'COLUMN', N'Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Product­Category records.', 'SCHEMA', N'Production', 'TABLE', N'Product­Category', 'COLUMN', N'Product­Category­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Production', 'TABLE', N'Product­Category', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Product­Category', 'CONSTRAINT', N'DF\_­Product­Category\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()()', 'SCHEMA', N'Production', 'TABLE', N'Product­Category', 'CONSTRAINT', N'DF\_­Product­Category\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Product­Category', 'CONSTRAINT', N'PK\_­Product­Category\_­Product­Category­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Product­Category', 'INDEX', N'AK\_­Product­Category\_­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Production', 'TABLE', N'Product­Category', 'INDEX', N'AK\_­Product­Category\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Product­Category', 'INDEX', N'PK\_­Product­Category\_­Product­Category­ID'  GO |

Uses

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

Used By

[[Production].[Product­Subcategory]](#WthFJ9hmewdhBB85kY7NAjtek7Y=)

|  |
| --- |
| [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 |
|  | Product­ID Product identification number. Foreign key to Product.Product­ID | int | 4 | NOT NULL |  |
|  | 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\_­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]>=[Start­Date] OR [End­Date] IS NULL) |
| CK\_­Product­Cost­History\_­Standard­Cost Check constraint [Standard­Cost] >= (0.00) | Standard­Cost | ([Standard­Cost]>=(0.00)) |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Product­Cost­History\_­Product\_­Product­ID Foreign key constraint referencing Product.Product­ID. | Product­ID->[[Production].[Product].[Product­ID]](#mO6TpkyW7q4W/OMRHISuYfWNCME=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Product­Cost­History]  (  [Product­ID] [int] NOT NULL,  [Start­Date] [datetime] NOT NULL,  [End­Date] [datetime] NULL,  [Standard­Cost] [money] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Product­Cost­History\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­Cost­History] ADD CONSTRAINT [CK\_­Product­Cost­History\_­End­Date] CHECK (([End­Date]>=[Start­Date] OR [End­Date] IS NULL))  GO  ALTER TABLE [Production].[Product­Cost­History] ADD CONSTRAINT [CK\_­Product­Cost­History\_­Standard­Cost] CHECK (([Standard­Cost]>=(0.00)))  GO  ALTER TABLE [Production].[Product­Cost­History] ADD CONSTRAINT [PK\_­Product­Cost­History\_­Product­ID\_­Start­Date] PRIMARY KEY CLUSTERED ([Product­ID], [Start­Date]) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­Cost­History] ADD CONSTRAINT [FK\_­Product­Cost­History\_­Product\_­Product­ID] FOREIGN KEY ([Product­ID]) REFERENCES [Production].[Product] ([Product­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Changes in the cost of a product over time.', 'SCHEMA', N'Production', 'TABLE', N'Product­Cost­History', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product cost end date.', 'SCHEMA', N'Production', 'TABLE', N'Product­Cost­History', 'COLUMN', N'End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Product­Cost­History', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product identification number. Foreign key to Product.Product­ID', 'SCHEMA', N'Production', 'TABLE', N'Product­Cost­History', 'COLUMN', N'Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Standard cost of the product.', 'SCHEMA', N'Production', 'TABLE', N'Product­Cost­History', 'COLUMN', N'Standard­Cost'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product cost start date.', 'SCHEMA', N'Production', 'TABLE', N'Product­Cost­History', 'COLUMN', N'Start­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [End­Date] >= [Start­Date] OR [End­Date] IS NULL', 'SCHEMA', N'Production', 'TABLE', N'Product­Cost­History', 'CONSTRAINT', N'CK\_­Product­Cost­History\_­End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Standard­Cost] >= (0.00)', 'SCHEMA', N'Production', 'TABLE', N'Product­Cost­History', 'CONSTRAINT', N'CK\_­Product­Cost­History\_­Standard­Cost'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Product­Cost­History', 'CONSTRAINT', N'DF\_­Product­Cost­History\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Product.Product­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product­Cost­History', 'CONSTRAINT', N'FK\_­Product­Cost­History\_­Product\_­Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Product­Cost­History', 'CONSTRAINT', N'PK\_­Product­Cost­History\_­Product­ID\_­Start­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Product­Cost­History', 'INDEX', N'PK\_­Product­Cost­History\_­Product­ID\_­Start­Date'  GO |

Uses

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

Used By

[[dbo].[ufn­Get­Product­Standard­Cost]](#iQHxQM8zJLh+qEeVmjb6xgEGWO4=)

|  |
| --- |
| [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 |
|  | 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\_­Product­Description\_­Product­Description­ID Primary key (clustered) constraint | Product­Description­ID | True |
|  | AK\_­Product­Description\_rowguid Unique nonclustered index. Used to support replication samples. | rowguid | True |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Product­Description]  (  [Product­Description­ID] [int] NOT NULL IDENTITY(1, 1),  [Description] [nvarchar] (400) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Product­Description\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Product­Description\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­Description] ADD CONSTRAINT [PK\_­Product­Description\_­Product­Description­ID] PRIMARY KEY CLUSTERED ([Product­Description­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Product­Description\_rowguid] ON [Production].[Product­Description] ([rowguid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product descriptions in several languages.', 'SCHEMA', N'Production', 'TABLE', N'Product­Description', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Description of the product.', 'SCHEMA', N'Production', 'TABLE', N'Product­Description', 'COLUMN', N'Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Product­Description', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Product­Description records.', 'SCHEMA', N'Production', 'TABLE', N'Product­Description', 'COLUMN', N'Product­Description­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Production', 'TABLE', N'Product­Description', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Product­Description', 'CONSTRAINT', N'DF\_­Product­Description\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Production', 'TABLE', N'Product­Description', 'CONSTRAINT', N'DF\_­Product­Description\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Product­Description', 'CONSTRAINT', N'PK\_­Product­Description\_­Product­Description­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Production', 'TABLE', N'Product­Description', 'INDEX', N'AK\_­Product­Description\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Product­Description', 'INDEX', N'PK\_­Product­Description\_­Product­Description­ID'  GO |

Uses

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

Used By

[[Production].[Product­Model­Product­Description­Culture]](#Kkf63jUYCrQKelqjjcG51JBQDNI=)

[[Production].[v­Product­And­Description]](#2wtLw+LERdCNthBlev7/7Ew1+JQ=)

|  |
| --- |
| [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 |
|  | Product­ID Product identification number. Foreign key to Product.Product­ID. | int | 4 | NOT NULL |  |
|  | 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 |
|  | PK\_­Product­Document\_­Product­ID\_­Document­Node Primary key (clustered) constraint | Product­ID, Document­Node | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Product­Document\_­Document\_­Document­Node Foreign key constraint referencing Document.Document­Node. | Document­Node->[[Production].[Document].[Document­Node]](#v5/qynDVNwkOWZdAe2xATpOnRWk=) |
| FK\_­Product­Document\_­Product\_­Product­ID Foreign key constraint referencing Product.Product­ID. | Product­ID->[[Production].[Product].[Product­ID]](#mO6TpkyW7q4W/OMRHISuYfWNCME=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Product­Document]  (  [Product­ID] [int] NOT NULL,  [Document­Node] [sys].[hierarchyid] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Product­Document\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­Document] ADD CONSTRAINT [PK\_­Product­Document\_­Product­ID\_­Document­Node] PRIMARY KEY CLUSTERED ([Product­ID], [Document­Node]) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­Document] ADD CONSTRAINT [FK\_­Product­Document\_­Document\_­Document­Node] FOREIGN KEY ([Document­Node]) REFERENCES [Production].[Document] ([Document­Node])  GO  ALTER TABLE [Production].[Product­Document] ADD CONSTRAINT [FK\_­Product­Document\_­Product\_­Product­ID] FOREIGN KEY ([Product­ID]) REFERENCES [Production].[Product] ([Product­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Cross-reference table mapping products to related product documents.', 'SCHEMA', N'Production', 'TABLE', N'Product­Document', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Document identification number. Foreign key to Document.Document­Node.', 'SCHEMA', N'Production', 'TABLE', N'Product­Document', 'COLUMN', N'Document­Node'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Product­Document', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product identification number. Foreign key to Product.Product­ID.', 'SCHEMA', N'Production', 'TABLE', N'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\_­Product­Document\_­Product­ID\_­Document­Node'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Product­Document', 'INDEX', N'PK\_­Product­Document\_­Product­ID\_­Document­Node'  GO |

Uses

[[Production].[Document]](#v5/qynDVNwkOWZdAe2xATpOnRWk=)

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

|  |
| --- |
| [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 |
|  | Product­ID Product identification number. Foreign key to Product.Product­ID. | int | 4 | NOT NULL |  |
|  | 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\_­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]](#ciHD61a29Ub1OJEWhZ1zarNhvcs=) |
| FK\_­Product­Inventory\_­Product\_­Product­ID Foreign key constraint referencing Product.Product­ID. | Product­ID->[[Production].[Product].[Product­ID]](#mO6TpkyW7q4W/OMRHISuYfWNCME=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Product­Inventory]  (  [Product­ID] [int] NOT NULL,  [Location­ID] [smallint] NOT NULL,  [Shelf] [nvarchar] (10) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Bin] [tinyint] NOT NULL,  [Quantity] [smallint] NOT NULL CONSTRAINT [DF\_­Product­Inventory\_­Quantity] DEFAULT ((0)),  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Product­Inventory\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Product­Inventory\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­Inventory] ADD CONSTRAINT [CK\_­Product­Inventory\_­Bin] CHECK (([Bin]>=(0) AND [Bin]<=(100)))  GO  ALTER TABLE [Production].[Product­Inventory] ADD CONSTRAINT [CK\_­Product­Inventory\_­Shelf] CHECK (([Shelf] like '[A-Za-z]' OR [Shelf]='N/A'))  GO  ALTER TABLE [Production].[Product­Inventory] ADD CONSTRAINT [PK\_­Product­Inventory\_­Product­ID\_­Location­ID] PRIMARY KEY CLUSTERED ([Product­ID], [Location­ID]) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­Inventory] ADD CONSTRAINT [FK\_­Product­Inventory\_­Location\_­Location­ID] FOREIGN KEY ([Location­ID]) REFERENCES [Production].[Location] ([Location­ID])  GO  ALTER TABLE [Production].[Product­Inventory] ADD CONSTRAINT [FK\_­Product­Inventory\_­Product\_­Product­ID] FOREIGN KEY ([Product­ID]) REFERENCES [Production].[Product] ([Product­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product inventory information.', 'SCHEMA', N'Production', 'TABLE', N'Product­Inventory', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Storage container on a shelf in an inventory location.', 'SCHEMA', N'Production', 'TABLE', N'Product­Inventory', 'COLUMN', N'Bin'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Inventory location identification number. Foreign key to Location.Location­ID. ', 'SCHEMA', N'Production', 'TABLE', N'Product­Inventory', 'COLUMN', N'Location­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Product­Inventory', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product identification number. Foreign key to Product.Product­ID.', 'SCHEMA', N'Production', 'TABLE', N'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'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Storage compartment within an inventory location.', 'SCHEMA', N'Production', 'TABLE', N'Product­Inventory', 'COLUMN', N'Shelf'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Bin] BETWEEN (0) AND (100)', 'SCHEMA', N'Production', 'TABLE', N'Product­Inventory', 'CONSTRAINT', N'CK\_­Product­Inventory\_­Bin'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Shelf] like ''[A-Za-z]'' OR [Shelf]=''N/A''', 'SCHEMA', N'Production', 'TABLE', N'Product­Inventory', 'CONSTRAINT', N'CK\_­Product­Inventory\_­Shelf'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Product­Inventory', 'CONSTRAINT', N'DF\_­Product­Inventory\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0', 'SCHEMA', N'Production', 'TABLE', N'Product­Inventory', 'CONSTRAINT', N'DF\_­Product­Inventory\_­Quantity'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Production', 'TABLE', N'Product­Inventory', 'CONSTRAINT', N'DF\_­Product­Inventory\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Location.Location­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product­Inventory', 'CONSTRAINT', N'FK\_­Product­Inventory\_­Location\_­Location­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Product.Product­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product­Inventory', 'CONSTRAINT', N'FK\_­Product­Inventory\_­Product\_­Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Product­Inventory', 'CONSTRAINT', N'PK\_­Product­Inventory\_­Product­ID\_­Location­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Product­Inventory', 'INDEX', N'PK\_­Product­Inventory\_­Product­ID\_­Location­ID'  GO |

Uses

[[Production].[Location]](#ciHD61a29Ub1OJEWhZ1zarNhvcs=)

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

Used By

[[dbo].[ufn­Get­Stock]](#uhj1mdMC7NK57d7HxekWpOKGYxI=)

|  |
| --- |
| [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 |
|  | Product­ID Product identification number. Foreign key to Product.Product­ID | int | 4 | NOT NULL |  |
|  | 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\_­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]](#mO6TpkyW7q4W/OMRHISuYfWNCME=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Product­List­Price­History]  (  [Product­ID] [int] NOT NULL,  [Start­Date] [datetime] NOT NULL,  [End­Date] [datetime] NULL,  [List­Price] [money] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Product­List­Price­History\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­List­Price­History] ADD CONSTRAINT [CK\_­Product­List­Price­History\_­End­Date] CHECK (([End­Date]>=[Start­Date] OR [End­Date] IS NULL))  GO  ALTER TABLE [Production].[Product­List­Price­History] ADD CONSTRAINT [CK\_­Product­List­Price­History\_­List­Price] CHECK (([List­Price]>(0.00)))  GO  ALTER TABLE [Production].[Product­List­Price­History] ADD CONSTRAINT [PK\_­Product­List­Price­History\_­Product­ID\_­Start­Date] PRIMARY KEY CLUSTERED ([Product­ID], [Start­Date]) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­List­Price­History] ADD CONSTRAINT [FK\_­Product­List­Price­History\_­Product\_­Product­ID] FOREIGN KEY ([Product­ID]) REFERENCES [Production].[Product] ([Product­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Changes in the list price of a product over time.', 'SCHEMA', N'Production', 'TABLE', N'Product­List­Price­History', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'List price end date', 'SCHEMA', N'Production', 'TABLE', N'Product­List­Price­History', 'COLUMN', N'End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product list price.', 'SCHEMA', N'Production', 'TABLE', N'Product­List­Price­History', 'COLUMN', N'List­Price'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Product­List­Price­History', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product identification number. Foreign key to Product.Product­ID', 'SCHEMA', N'Production', 'TABLE', N'Product­List­Price­History', 'COLUMN', N'Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'List price start date.', 'SCHEMA', N'Production', 'TABLE', N'Product­List­Price­History', 'COLUMN', N'Start­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [End­Date] >= [Start­Date] OR [End­Date] IS NULL', 'SCHEMA', N'Production', 'TABLE', N'Product­List­Price­History', 'CONSTRAINT', N'CK\_­Product­List­Price­History\_­End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [List­Price] > (0.00)', 'SCHEMA', N'Production', 'TABLE', N'Product­List­Price­History', 'CONSTRAINT', N'CK\_­Product­List­Price­History\_­List­Price'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Product­List­Price­History', 'CONSTRAINT', N'DF\_­Product­List­Price­History\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Product.Product­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product­List­Price­History', 'CONSTRAINT', N'FK\_­Product­List­Price­History\_­Product\_­Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Product­List­Price­History', 'CONSTRAINT', N'PK\_­Product­List­Price­History\_­Product­ID\_­Start­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Product­List­Price­History', 'INDEX', N'PK\_­Product­List­Price­History\_­Product­ID\_­Start­Date'  GO |

Uses

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

Used By

[[dbo].[ufn­Get­Product­Dealer­Price]](#qCHqC6Ssa6gRKHNZ1A5hYnf4CXs=)

[[dbo].[ufn­Get­Product­List­Price]](#8mn5CnDFl2xPNBEKQCwZ9YJdOMk=)

|  |
| --- |
| [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) | Nullability | Identity | Default |
|  | Product­Model­ID Primary key for Product­Model records. | int | 4 | NOT NULL | 1 - 1 |  |
|  | Name Product model description. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NOT NULL |  |  |
|  | Catalog­Description Detailed product catalog information in xml format. | [xml([Production].[Product­Description­Schema­Collection])](#ZaR2DNJAzm2r+KkNxKx06/AFx84=) | max | NULL allowed |  |  |
|  | Instructions Manufacturing instructions in xml format. | [xml([Production].[Manu­Instructions­Schema­Collection])](#8FXP/Rirpa1dKaRW9DMCZBD9740=) | 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 |  | (getdate()) |

Indexes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Key Columns | Type | Unique | XML Type |
|  | PK\_­Product­Model\_­Product­Model­ID Primary key (clustered) constraint | Product­Model­ID |  | 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. | Catalog­Description | xml |  | Primary |
|  | PXML\_­Product­Model\_­Instructions Primary XML index. | Instructions | xml |  | Primary |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Product­Model]  (  [Product­Model­ID] [int] NOT NULL IDENTITY(1, 1),  [Name] [dbo].[Name] NOT NULL,  [Catalog­Description] [xml] (CONTENT [Production].[Product­Description­Schema­Collection]) NULL,  [Instructions] [xml] (CONTENT [Production].[Manu­Instructions­Schema­Collection]) NULL,  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Product­Model\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Product­Model\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­Model] ADD CONSTRAINT [PK\_­Product­Model\_­Product­Model­ID] PRIMARY KEY CLUSTERED ([Product­Model­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Product­Model\_­Name] ON [Production].[Product­Model] ([Name]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Product­Model\_rowguid] ON [Production].[Product­Model] ([rowguid]) ON [PRIMARY]  GO  CREATE PRIMARY XML INDEX [PXML\_­Product­Model\_­Catalog­Description]  ON [Production].[Product­Model] ([Catalog­Description])  GO  CREATE PRIMARY XML INDEX [PXML\_­Product­Model\_­Instructions]  ON [Production].[Product­Model] ([Instructions])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product model classification.', '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'Catalog­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Manufacturing instructions in xml format.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model', 'COLUMN', N'Instructions'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product model description.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model', 'COLUMN', N'Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Product­Model records.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model', 'COLUMN', N'Product­Model­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Product­Model', 'CONSTRAINT', N'DF\_­Product­Model\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Production', 'TABLE', N'Product­Model', 'CONSTRAINT', N'DF\_­Product­Model\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Product­Model', 'CONSTRAINT', N'PK\_­Product­Model\_­Product­Model­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model', 'INDEX', N'AK\_­Product­Model\_­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model', 'INDEX', N'AK\_­Product­Model\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model', 'INDEX', N'PK\_­Product­Model\_­Product­Model­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary XML index.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model', 'INDEX', N'PXML\_­Product­Model\_­Catalog­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary XML index.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model', 'INDEX', N'PXML\_­Product­Model\_­Instructions'  GO |

Uses

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

[[Production].[Manu­Instructions­Schema­Collection]](#8FXP/Rirpa1dKaRW9DMCZBD9740=)

[[Production].[Product­Description­Schema­Collection]](#ZaR2DNJAzm2r+KkNxKx06/AFx84=)

Used By

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[[Production].[Product­Model­Illustration]](#ha0ZY5wgktKInuoqdKFBVRbeJaI=)

[[Production].[Product­Model­Product­Description­Culture]](#Kkf63jUYCrQKelqjjcG51JBQDNI=)

[[Production].[v­Product­And­Description]](#2wtLw+LERdCNthBlev7/7Ew1+JQ=)

[[Production].[v­Product­Model­Catalog­Description]](#j53/ggQeAT7efkm5uXeLRIbSppI=)

[[Production].[v­Product­Model­Instructions]](#GCJbY/fUa640Nyz3prLsQye2L+M=)

|  |
| --- |
| [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 |
|  | Product­Model­ID Primary key. Foreign key to Product­Model.Product­Model­ID. | int | 4 | NOT NULL |  |
|  | 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\_­Product­Model­Illustration\_­Product­Model­ID\_­Illustration­ID Primary key (clustered) constraint | Product­Model­ID, Illustration­ID | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Product­Model­Illustration\_­Illustration\_­Illustration­ID Foreign key constraint referencing Illustration.Illustration­ID. | Illustration­ID->[[Production].[Illustration].[Illustration­ID]](#q4hmTMYciVWtuNDgg2tVsn/4PAw=) |
| FK\_­Product­Model­Illustration\_­Product­Model\_­Product­Model­ID Foreign key constraint referencing Product­Model.Product­Model­ID. | Product­Model­ID->[[Production].[Product­Model].[Product­Model­ID]](#n9VdHBX3Ji5OWfmMzrHHD+J8HSM=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Product­Model­Illustration]  (  [Product­Model­ID] [int] NOT NULL,  [Illustration­ID] [int] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Product­Model­Illustration\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­Model­Illustration] ADD CONSTRAINT [PK\_­Product­Model­Illustration\_­Product­Model­ID\_­Illustration­ID] PRIMARY KEY CLUSTERED ([Product­Model­ID], [Illustration­ID]) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­Model­Illustration] ADD CONSTRAINT [FK\_­Product­Model­Illustration\_­Illustration\_­Illustration­ID] FOREIGN KEY ([Illustration­ID]) REFERENCES [Production].[Illustration] ([Illustration­ID])  GO  ALTER TABLE [Production].[Product­Model­Illustration] ADD CONSTRAINT [FK\_­Product­Model­Illustration\_­Product­Model\_­Product­Model­ID] FOREIGN KEY ([Product­Model­ID]) REFERENCES [Production].[Product­Model] ([Product­Model­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Cross-reference table mapping product models and illustrations.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model­Illustration', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Foreign key to Illustration.Illustration­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model­Illustration', 'COLUMN', N'Illustration­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model­Illustration', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Foreign key to Product­Model.Product­Model­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model­Illustration', 'COLUMN', N'Product­Model­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Product­Model­Illustration', 'CONSTRAINT', N'DF\_­Product­Model­Illustration\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Illustration.Illustration­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model­Illustration', 'CONSTRAINT', N'FK\_­Product­Model­Illustration\_­Illustration\_­Illustration­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Product­Model.Product­Model­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model­Illustration', 'CONSTRAINT', N'FK\_­Product­Model­Illustration\_­Product­Model\_­Product­Model­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Product­Model­Illustration', 'CONSTRAINT', N'PK\_­Product­Model­Illustration\_­Product­Model­ID\_­Illustration­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model­Illustration', 'INDEX', N'PK\_­Product­Model­Illustration\_­Product­Model­ID\_­Illustration­ID'  GO |

Uses

[[Production].[Illustration]](#q4hmTMYciVWtuNDgg2tVsn/4PAw=)

[[Production].[Product­Model]](#n9VdHBX3Ji5OWfmMzrHHD+J8HSM=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

|  |
| --- |
| [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 |
|  | Product­Model­ID Primary key. Foreign key to Product­Model.Product­Model­ID. | int | 4 | NOT NULL |  |
|  | Product­Description­ID Primary key. Foreign key to Product­Description.Product­Description­ID. | int | 4 | NOT NULL |  |
|  | 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 |
|  | PK\_­Product­Model­Product­Description­Culture\_­Product­Model­ID\_­Product­Description­ID\_­Culture­ID Primary key (clustered) constraint | Product­Model­ID, Product­Description­ID, Culture­ID | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Product­Model­Product­Description­Culture\_­Culture\_­Culture­ID Foreign key constraint referencing Culture.Culture­ID. | Culture­ID->[[Production].[Culture].[Culture­ID]](#D4g0rs1owPY1TNtNA3F8BcetcxU=) |
| FK\_­Product­Model­Product­Description­Culture\_­Product­Description\_­Product­Description­ID Foreign key constraint referencing Product­Description.Product­Description­ID. | Product­Description­ID->[[Production].[Product­Description].[Product­Description­ID]](#4iEXAFuDYt/4TBUTP9DP8dTGazY=) |
| FK\_­Product­Model­Product­Description­Culture\_­Product­Model\_­Product­Model­ID Foreign key constraint referencing Product­Model.Product­Model­ID. | Product­Model­ID->[[Production].[Product­Model].[Product­Model­ID]](#n9VdHBX3Ji5OWfmMzrHHD+J8HSM=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Product­Model­Product­Description­Culture]  (  [Product­Model­ID] [int] NOT NULL,  [Product­Description­ID] [int] NOT NULL,  [Culture­ID] [nchar] (6) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Product­Model­Product­Description­Culture\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­Model­Product­Description­Culture] ADD CONSTRAINT [PK\_­Product­Model­Product­Description­Culture\_­Product­Model­ID\_­Product­Description­ID\_­Culture­ID] PRIMARY KEY CLUSTERED ([Product­Model­ID], [Product­Description­ID], [Culture­ID]) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­Model­Product­Description­Culture] ADD CONSTRAINT [FK\_­Product­Model­Product­Description­Culture\_­Culture\_­Culture­ID] FOREIGN KEY ([Culture­ID]) REFERENCES [Production].[Culture] ([Culture­ID])  GO  ALTER TABLE [Production].[Product­Model­Product­Description­Culture] ADD CONSTRAINT [FK\_­Product­Model­Product­Description­Culture\_­Product­Description\_­Product­Description­ID] FOREIGN KEY ([Product­Description­ID]) REFERENCES [Production].[Product­Description] ([Product­Description­ID])  GO  ALTER TABLE [Production].[Product­Model­Product­Description­Culture] ADD CONSTRAINT [FK\_­Product­Model­Product­Description­Culture\_­Product­Model\_­Product­Model­ID] FOREIGN KEY ([Product­Model­ID]) REFERENCES [Production].[Product­Model] ([Product­Model­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Cross-reference table mapping product descriptions and the language the description is written in.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model­Product­Description­Culture', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Culture identification number. Foreign key to Culture.Culture­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model­Product­Description­Culture', 'COLUMN', N'Culture­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model­Product­Description­Culture', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Foreign key to Product­Description.Product­Description­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model­Product­Description­Culture', 'COLUMN', N'Product­Description­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Foreign key to Product­Model.Product­Model­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model­Product­Description­Culture', 'COLUMN', N'Product­Model­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Product­Model­Product­Description­Culture', 'CONSTRAINT', N'DF\_­Product­Model­Product­Description­Culture\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Culture.Culture­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model­Product­Description­Culture', 'CONSTRAINT', N'FK\_­Product­Model­Product­Description­Culture\_­Culture\_­Culture­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Product­Description.Product­Description­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model­Product­Description­Culture', 'CONSTRAINT', N'FK\_­Product­Model­Product­Description­Culture\_­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\_­Product­Model\_­Product­Model­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Product­Model­Product­Description­Culture', 'CONSTRAINT', N'PK\_­Product­Model­Product­Description­Culture\_­Product­Model­ID\_­Product­Description­ID\_­Culture­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Product­Model­Product­Description­Culture', 'INDEX', N'PK\_­Product­Model­Product­Description­Culture\_­Product­Model­ID\_­Product­Description­ID\_­Culture­ID'  GO |

Uses

[[Production].[Culture]](#D4g0rs1owPY1TNtNA3F8BcetcxU=)

[[Production].[Product­Description]](#4iEXAFuDYt/4TBUTP9DP8dTGazY=)

[[Production].[Product­Model]](#n9VdHBX3Ji5OWfmMzrHHD+J8HSM=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

Used By

[[Production].[v­Product­And­Description]](#2wtLw+LERdCNthBlev7/7Ew1+JQ=)

|  |
| --- |
| [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 |
|  | 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 |
|  | PK\_­Product­Photo\_­Product­Photo­ID Primary key (clustered) constraint | Product­Photo­ID | True |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Product­Photo]  (  [Product­Photo­ID] [int] NOT NULL IDENTITY(1, 1),  [Thumb­Nail­Photo] [varbinary] (max) NULL,  [Thumbnail­Photo­File­Name] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Large­Photo] [varbinary] (max) NULL,  [Large­Photo­File­Name] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Product­Photo\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­Photo] ADD CONSTRAINT [PK\_­Product­Photo\_­Product­Photo­ID] PRIMARY KEY CLUSTERED ([Product­Photo­ID]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product images.', 'SCHEMA', N'Production', 'TABLE', N'Product­Photo', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Large image of the product.', 'SCHEMA', N'Production', 'TABLE', N'Product­Photo', 'COLUMN', N'Large­Photo'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Large image file name.', 'SCHEMA', N'Production', 'TABLE', N'Product­Photo', 'COLUMN', N'Large­Photo­File­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Product­Photo', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Product­Photo records.', 'SCHEMA', N'Production', 'TABLE', N'Product­Photo', 'COLUMN', N'Product­Photo­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Small image of the product.', 'SCHEMA', N'Production', 'TABLE', N'Product­Photo', 'COLUMN', N'Thumb­Nail­Photo'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Small image file name.', 'SCHEMA', N'Production', 'TABLE', N'Product­Photo', 'COLUMN', N'Thumbnail­Photo­File­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Product­Photo', 'CONSTRAINT', N'DF\_­Product­Photo\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Product­Photo', 'CONSTRAINT', N'PK\_­Product­Photo\_­Product­Photo­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Product­Photo', 'INDEX', N'PK\_­Product­Photo\_­Product­Photo­ID'  GO |

Uses

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

Used By

[[Production].[Product­Product­Photo]](#40voS36ihwbm/xtmbxsgEQceBOQ=)

|  |
| --- |
| [Production].[Product­Product­Photo] |

MS\_­Description

Cross-reference table mapping products and product photos.

Properties

|  |  |
| --- | --- |
| Property | Value |
| Heap | 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 |
|  | Product­ID Product identification number. Foreign key to Product.Product­ID. | int | 4 | NOT NULL |  |
|  | 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]](#j88jjDtwCbK8khz/m8jxryH/SFk=) | 1 | NOT NULL | ((0)) |
|  | Modified­Date Date and time the record was last updated. | datetime | 8 | NOT NULL | (getdate()) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Product­Product­Photo\_­Product­ID\_­Product­Photo­ID Primary key (clustered) constraint | Product­ID, Product­Photo­ID | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Product­Product­Photo\_­Product\_­Product­ID Foreign key constraint referencing Product.Product­ID. | Product­ID->[[Production].[Product].[Product­ID]](#mO6TpkyW7q4W/OMRHISuYfWNCME=) |
| FK\_­Product­Product­Photo\_­Product­Photo\_­Product­Photo­ID Foreign key constraint referencing Product­Photo.Product­Photo­ID. | Product­Photo­ID->[[Production].[Product­Photo].[Product­Photo­ID]](#wYFkOtQB0E8+2MDoAl84kiyzc9A=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Product­Product­Photo]  (  [Product­ID] [int] NOT NULL,  [Product­Photo­ID] [int] NOT NULL,  [Primary] [dbo].[Flag] NOT NULL CONSTRAINT [DF\_­Product­Product­Photo\_­Primary] DEFAULT ((0)),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Product­Product­Photo\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­Product­Photo] ADD CONSTRAINT [PK\_­Product­Product­Photo\_­Product­ID\_­Product­Photo­ID] PRIMARY KEY NONCLUSTERED ([Product­ID], [Product­Photo­ID]) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­Product­Photo] ADD CONSTRAINT [FK\_­Product­Product­Photo\_­Product\_­Product­ID] FOREIGN KEY ([Product­ID]) REFERENCES [Production].[Product] ([Product­ID])  GO  ALTER TABLE [Production].[Product­Product­Photo] ADD CONSTRAINT [FK\_­Product­Product­Photo\_­Product­Photo\_­Product­Photo­ID] FOREIGN KEY ([Product­Photo­ID]) REFERENCES [Production].[Product­Photo] ([Product­Photo­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Cross-reference table mapping products and product photos.', 'SCHEMA', N'Production', 'TABLE', N'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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = Photo is not the principal image. 1 = Photo is the principal image.', 'SCHEMA', N'Production', 'TABLE', N'Product­Product­Photo', 'COLUMN', N'Primary'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product identification number. Foreign key to Product.Product­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product­Product­Photo', 'COLUMN', N'Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product photo identification number. Foreign key to Product­Photo.Product­Photo­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product­Product­Photo', 'COLUMN', N'Product­Photo­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Product­Product­Photo', 'CONSTRAINT', N'DF\_­Product­Product­Photo\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0 (FALSE)', 'SCHEMA', N'Production', 'TABLE', N'Product­Product­Photo', 'CONSTRAINT', N'DF\_­Product­Product­Photo\_­Primary'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Product.Product­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product­Product­Photo', 'CONSTRAINT', N'FK\_­Product­Product­Photo\_­Product\_­Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Product­Photo.Product­Photo­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product­Product­Photo', 'CONSTRAINT', N'FK\_­Product­Product­Photo\_­Product­Photo\_­Product­Photo­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Product­Product­Photo', 'CONSTRAINT', N'PK\_­Product­Product­Photo\_­Product­ID\_­Product­Photo­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Product­Product­Photo', 'INDEX', N'PK\_­Product­Product­Photo\_­Product­ID\_­Product­Photo­ID'  GO |

Uses

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[[Production].[Product­Photo]](#wYFkOtQB0E8+2MDoAl84kiyzc9A=)

[[dbo].[Flag]](#j88jjDtwCbK8khz/m8jxryH/SFk=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

|  |
| --- |
| [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](#KQM1/2huuuPy+pMNx4F37D5+p0Y=) |
| 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) | Nullability | Full Text Indexed | Language | Identity | Default |
|  | Product­Review­ID Primary key for Product­Review records. | int | 4 | NOT NULL |  |  | 1 - 1 |  |
|  | Product­ID Product identification number. Foreign key to Product.Product­ID. | int | 4 | NOT NULL |  |  |  |  |
|  | Reviewer­Name Name of the reviewer. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NOT NULL |  |  |  |  |
|  | Review­Date Date review was submitted. | datetime | 8 | NOT NULL |  |  |  | (getdate()) |
|  | 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 |  |  |  |  |
|  | Comments Reviewer's comments | nvarchar(3850) | 7700 | NULL allowed | True | 1033 |  |  |
|  | Modified­Date Date and time the record was last updated. | datetime | 8 | NOT NULL |  |  |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Key Columns | Included Columns | Unique |
|  | 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]](#mO6TpkyW7q4W/OMRHISuYfWNCME=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Product­Review]  (  [Product­Review­ID] [int] NOT NULL IDENTITY(1, 1),  [Product­ID] [int] NOT NULL,  [Reviewer­Name] [dbo].[Name] NOT NULL,  [Review­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Product­Review\_­Review­Date] DEFAULT (getdate()),  [Email­Address] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Rating] [int] NOT NULL,  [Comments] [nvarchar] (3850) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Product­Review\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­Review] ADD CONSTRAINT [CK\_­Product­Review\_­Rating] CHECK (([Rating]>=(1) AND [Rating]<=(5)))  GO  ALTER TABLE [Production].[Product­Review] ADD CONSTRAINT [PK\_­Product­Review\_­Product­Review­ID] PRIMARY KEY CLUSTERED ([Product­Review­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Product­Review\_­Product­ID\_­Name] ON [Production].[Product­Review] ([Product­ID], [Reviewer­Name]) INCLUDE ([Comments]) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­Review] ADD CONSTRAINT [FK\_­Product­Review\_­Product\_­Product­ID] FOREIGN KEY ([Product­ID]) REFERENCES [Production].[Product] ([Product­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Customer reviews of products they have purchased.', 'SCHEMA', N'Production', 'TABLE', N'Product­Review', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Reviewer''s comments', 'SCHEMA', N'Production', 'TABLE', N'Product­Review', 'COLUMN', N'Comments'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Reviewer''s e-mail address.', 'SCHEMA', N'Production', 'TABLE', N'Product­Review', 'COLUMN', N'Email­Address'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Product­Review', 'COLUMN', N'Modified­Date'  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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Product­Review records.', 'SCHEMA', N'Production', 'TABLE', N'Product­Review', 'COLUMN', N'Product­Review­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product rating given by the reviewer. Scale is 1 to 5 with 5 as the highest rating.', 'SCHEMA', N'Production', 'TABLE', N'Product­Review', 'COLUMN', N'Rating'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date review was submitted.', 'SCHEMA', N'Production', 'TABLE', N'Product­Review', 'COLUMN', N'Review­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Name of the reviewer.', 'SCHEMA', N'Production', 'TABLE', N'Product­Review', 'COLUMN', N'Reviewer­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Rating] BETWEEN (1) AND (5)', 'SCHEMA', N'Production', 'TABLE', N'Product­Review', 'CONSTRAINT', N'CK\_­Product­Review\_­Rating'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Product­Review', 'CONSTRAINT', N'DF\_­Product­Review\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Product­Review', 'CONSTRAINT', N'DF\_­Product­Review\_­Review­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Product.Product­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product­Review', 'CONSTRAINT', N'FK\_­Product­Review\_­Product\_­Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Product­Review', 'CONSTRAINT', N'PK\_­Product­Review\_­Product­Review­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Product­Review', 'INDEX', N'IX\_­Product­Review\_­Product­ID\_­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Product­Review', 'INDEX', N'PK\_­Product­Review\_­Product­Review­ID'  GO  CREATE FULLTEXT INDEX ON [Production].[Product­Review] KEY INDEX [PK\_­Product­Review\_­Product­Review­ID] ON [AW2016Full­Text­Catalog]  GO  ALTER FULLTEXT INDEX ON [Production].[Product­Review] ADD ([Comments] LANGUAGE 1033)  GO |

Uses

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

Used By

[AW2016Full­Text­Catalog](#KQM1/2huuuPy+pMNx4F37D5+p0Y=)

|  |
| --- |
| [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 |
|  | Product­Subcategory­ID Primary key for Product­Subcategory records. | int | 4 | NOT NULL | 1 - 1 |  |
|  | Product­Category­ID Product category identification number. Foreign key to Product­Category.Product­Category­ID. | int | 4 | NOT NULL |  |  |
|  | Name Subcategory description. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 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\_­Product­Subcategory\_­Product­Subcategory­ID Primary key (clustered) constraint | Product­Subcategory­ID | 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\_­Product­Category\_­Product­Category­ID Foreign key constraint referencing Product­Category.Product­Category­ID. | Product­Category­ID->[[Production].[Product­Category].[Product­Category­ID]](#upfoEFG3EP8VWKmQ3NJKJ7V9JTs=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Product­Subcategory]  (  [Product­Subcategory­ID] [int] NOT NULL IDENTITY(1, 1),  [Product­Category­ID] [int] NOT NULL,  [Name] [dbo].[Name] NOT NULL,  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Product­Subcategory\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Product­Subcategory\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­Subcategory] ADD CONSTRAINT [PK\_­Product­Subcategory\_­Product­Subcategory­ID] PRIMARY KEY CLUSTERED ([Product­Subcategory­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Product­Subcategory\_­Name] ON [Production].[Product­Subcategory] ([Name]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Product­Subcategory\_rowguid] ON [Production].[Product­Subcategory] ([rowguid]) ON [PRIMARY]  GO  ALTER TABLE [Production].[Product­Subcategory] ADD CONSTRAINT [FK\_­Product­Subcategory\_­Product­Category\_­Product­Category­ID] FOREIGN KEY ([Product­Category­ID]) REFERENCES [Production].[Product­Category] ([Product­Category­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product subcategories. See Product­Category table.', 'SCHEMA', N'Production', 'TABLE', N'Product­Subcategory', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Product­Subcategory', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Subcategory description.', 'SCHEMA', N'Production', 'TABLE', N'Product­Subcategory', 'COLUMN', N'Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product category identification number. Foreign key to Product­Category.Product­Category­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product­Subcategory', 'COLUMN', N'Product­Category­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Product­Subcategory records.', 'SCHEMA', N'Production', 'TABLE', N'Product­Subcategory', 'COLUMN', N'Product­Subcategory­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Production', 'TABLE', N'Product­Subcategory', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Product­Subcategory', 'CONSTRAINT', N'DF\_­Product­Subcategory\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Production', 'TABLE', N'Product­Subcategory', 'CONSTRAINT', N'DF\_­Product­Subcategory\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Product­Category.Product­Category­ID.', 'SCHEMA', N'Production', 'TABLE', N'Product­Subcategory', 'CONSTRAINT', N'FK\_­Product­Subcategory\_­Product­Category\_­Product­Category­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Product­Subcategory', 'CONSTRAINT', N'PK\_­Product­Subcategory\_­Product­Subcategory­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Product­Subcategory', 'INDEX', N'AK\_­Product­Subcategory\_­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Production', 'TABLE', N'Product­Subcategory', 'INDEX', N'AK\_­Product­Subcategory\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Product­Subcategory', 'INDEX', N'PK\_­Product­Subcategory\_­Product­Subcategory­ID'  GO |

Uses

[[Production].[Product­Category]](#upfoEFG3EP8VWKmQ3NJKJ7V9JTs=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

Used By

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

|  |
| --- |
| [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 |
|  | Scrap­Reason­ID Primary key for Scrap­Reason records. | smallint | 2 | NOT NULL | 1 - 1 |  |
|  | Name Failure description. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NOT NULL |  |  |
|  | Modified­Date Date and time the record was last updated. | datetime | 8 | NOT NULL |  | (getdate()) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Scrap­Reason\_­Scrap­Reason­ID Primary key (clustered) constraint | Scrap­Reason­ID | True |
|  | AK\_­Scrap­Reason\_­Name Unique nonclustered index. | Name | True |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Scrap­Reason]  (  [Scrap­Reason­ID] [smallint] NOT NULL IDENTITY(1, 1),  [Name] [dbo].[Name] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Scrap­Reason\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Scrap­Reason] ADD CONSTRAINT [PK\_­Scrap­Reason\_­Scrap­Reason­ID] PRIMARY KEY CLUSTERED ([Scrap­Reason­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Scrap­Reason\_­Name] ON [Production].[Scrap­Reason] ([Name]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Manufacturing failure reasons lookup table.', 'SCHEMA', N'Production', 'TABLE', N'Scrap­Reason', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Scrap­Reason', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Failure description.', 'SCHEMA', N'Production', 'TABLE', N'Scrap­Reason', 'COLUMN', N'Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Scrap­Reason records.', 'SCHEMA', N'Production', 'TABLE', N'Scrap­Reason', 'COLUMN', N'Scrap­Reason­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Scrap­Reason', 'CONSTRAINT', N'DF\_­Scrap­Reason\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Scrap­Reason', 'CONSTRAINT', N'PK\_­Scrap­Reason\_­Scrap­Reason­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Scrap­Reason', 'INDEX', N'AK\_­Scrap­Reason\_­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Scrap­Reason', 'INDEX', N'PK\_­Scrap­Reason\_­Scrap­Reason­ID'  GO |

Uses

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

Used By

[[Production].[Work­Order]](#BSRICvfPko6shp7o8HSu003oCno=)

|  |
| --- |
| [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 |
|  | Transaction­ID Primary key for Transaction­History records. | int | 4 | NOT NULL | 100000 - 1 |  |
|  | 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 |  |  |
|  | 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\_­Transaction­History\_­Transaction­ID Primary key (clustered) constraint | Transaction­ID | True |
|  | IX\_­Transaction­History\_­Product­ID Nonclustered index. | Product­ID |  |
|  | IX\_­Transaction­History\_­Reference­Order­ID\_­Reference­Order­Line­ID Nonclustered index. | Reference­Order­ID, Reference­Order­Line­ID |  |

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([Transaction­Type])='P' OR upper([Transaction­Type])='S' OR upper([Transaction­Type])='W') |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Transaction­History\_­Product\_­Product­ID Foreign key constraint referencing Product.Product­ID. | Product­ID->[[Production].[Product].[Product­ID]](#mO6TpkyW7q4W/OMRHISuYfWNCME=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Transaction­History]  (  [Transaction­ID] [int] NOT NULL IDENTITY(100000, 1),  [Product­ID] [int] NOT NULL,  [Reference­Order­ID] [int] NOT NULL,  [Reference­Order­Line­ID] [int] NOT NULL CONSTRAINT [DF\_­Transaction­History\_­Reference­Order­Line­ID] DEFAULT ((0)),  [Transaction­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Transaction­History\_­Transaction­Date] DEFAULT (getdate()),  [Transaction­Type] [nchar] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Quantity] [int] NOT NULL,  [Actual­Cost] [money] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Transaction­History\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Transaction­History] ADD CONSTRAINT [CK\_­Transaction­History\_­Transaction­Type] CHECK ((upper([Transaction­Type])='P' OR upper([Transaction­Type])='S' OR upper([Transaction­Type])='W'))  GO  ALTER TABLE [Production].[Transaction­History] ADD CONSTRAINT [PK\_­Transaction­History\_­Transaction­ID] PRIMARY KEY CLUSTERED ([Transaction­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Transaction­History\_­Product­ID] ON [Production].[Transaction­History] ([Product­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Transaction­History\_­Reference­Order­ID\_­Reference­Order­Line­ID] ON [Production].[Transaction­History] ([Reference­Order­ID], [Reference­Order­Line­ID]) ON [PRIMARY]  GO  ALTER TABLE [Production].[Transaction­History] ADD CONSTRAINT [FK\_­Transaction­History\_­Product\_­Product­ID] FOREIGN KEY ([Product­ID]) REFERENCES [Production].[Product] ([Product­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Record of each purchase order, sales order, or work order transaction year to date.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product cost.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History', 'COLUMN', N'Actual­Cost'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product identification number. Foreign key to Product.Product­ID.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History', 'COLUMN', N'Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product quantity.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History', 'COLUMN', N'Quantity'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Purchase order, sales order, or work order identification number.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History', 'COLUMN', N'Reference­Order­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Line number associated with the purchase order, sales order, or work order.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History', 'COLUMN', N'Reference­Order­Line­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time of the transaction.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History', 'COLUMN', N'Transaction­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Transaction­History records.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History', 'COLUMN', N'Transaction­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'W = Work­Order, S = Sales­Order, P = Purchase­Order', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History', 'COLUMN', N'Transaction­Type'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Transaction­Type]=''p'' OR [Transaction­Type]=''s'' OR [Transaction­Type]=''w'' OR [Transaction­Type]=''P'' OR [Transaction­Type]=''S'' OR [Transaction­Type]=''W'')', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History', 'CONSTRAINT', N'CK\_­Transaction­History\_­Transaction­Type'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History', 'CONSTRAINT', N'DF\_­Transaction­History\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History', 'CONSTRAINT', N'DF\_­Transaction­History\_­Reference­Order­Line­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History', 'CONSTRAINT', N'DF\_­Transaction­History\_­Transaction­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Product.Product­ID.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History', 'CONSTRAINT', N'FK\_­Transaction­History\_­Product\_­Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History', 'CONSTRAINT', N'PK\_­Transaction­History\_­Transaction­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History', 'INDEX', N'IX\_­Transaction­History\_­Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History', 'INDEX', N'IX\_­Transaction­History\_­Reference­Order­ID\_­Reference­Order­Line­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History', 'INDEX', N'PK\_­Transaction­History\_­Transaction­ID'  GO |

Uses

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

|  |
| --- |
| [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 |
|  | 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 |  |
|  | 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

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Transaction­History­Archive\_­Transaction­ID Primary key (clustered) constraint | Transaction­ID | True |
|  | IX\_­Transaction­History­Archive\_­Product­ID Nonclustered index. | Product­ID |  |
|  | IX\_­Transaction­History­Archive\_­Reference­Order­ID\_­Reference­Order­Line­ID Nonclustered index. | Reference­Order­ID, Reference­Order­Line­ID |  |

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([Transaction­Type])='P' OR upper([Transaction­Type])='S' OR upper([Transaction­Type])='W') |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Transaction­History­Archive]  (  [Transaction­ID] [int] NOT NULL,  [Product­ID] [int] NOT NULL,  [Reference­Order­ID] [int] NOT NULL,  [Reference­Order­Line­ID] [int] NOT NULL CONSTRAINT [DF\_­Transaction­History­Archive\_­Reference­Order­Line­ID] DEFAULT ((0)),  [Transaction­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Transaction­History­Archive\_­Transaction­Date] DEFAULT (getdate()),  [Transaction­Type] [nchar] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Quantity] [int] NOT NULL,  [Actual­Cost] [money] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Transaction­History­Archive\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Transaction­History­Archive] ADD CONSTRAINT [CK\_­Transaction­History­Archive\_­Transaction­Type] CHECK ((upper([Transaction­Type])='P' OR upper([Transaction­Type])='S' OR upper([Transaction­Type])='W'))  GO  ALTER TABLE [Production].[Transaction­History­Archive] ADD CONSTRAINT [PK\_­Transaction­History­Archive\_­Transaction­ID] PRIMARY KEY CLUSTERED ([Transaction­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Transaction­History­Archive\_­Product­ID] ON [Production].[Transaction­History­Archive] ([Product­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Transaction­History­Archive\_­Reference­Order­ID\_­Reference­Order­Line­ID] ON [Production].[Transaction­History­Archive] ([Reference­Order­ID], [Reference­Order­Line­ID]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Transactions for previous years.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History­Archive', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product cost.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History­Archive', 'COLUMN', N'Actual­Cost'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History­Archive', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product identification number. Foreign key to Product.Product­ID.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History­Archive', 'COLUMN', N'Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product quantity.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History­Archive', 'COLUMN', N'Quantity'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Purchase order, sales order, or work order identification number.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History­Archive', 'COLUMN', N'Reference­Order­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Line number associated with the purchase order, sales order, or work order.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History­Archive', 'COLUMN', N'Reference­Order­Line­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time of the transaction.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History­Archive', 'COLUMN', N'Transaction­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Transaction­History­Archive records.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History­Archive', 'COLUMN', N'Transaction­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'W = Work Order, S = Sales Order, P = Purchase Order', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History­Archive', 'COLUMN', N'Transaction­Type'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Transaction­Type]=''p'' OR [Transaction­Type]=''s'' OR [Transaction­Type]=''w'' OR [Transaction­Type]=''P'' OR [Transaction­Type]=''S'' OR [Transaction­Type]=''W''', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History­Archive', 'CONSTRAINT', N'CK\_­Transaction­History­Archive\_­Transaction­Type'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History­Archive', 'CONSTRAINT', N'DF\_­Transaction­History­Archive\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History­Archive', 'CONSTRAINT', N'DF\_­Transaction­History­Archive\_­Reference­Order­Line­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History­Archive', 'CONSTRAINT', N'DF\_­Transaction­History­Archive\_­Transaction­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History­Archive', 'CONSTRAINT', N'PK\_­Transaction­History­Archive\_­Transaction­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History­Archive', 'INDEX', N'IX\_­Transaction­History­Archive\_­Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Transaction­History­Archive', 'INDEX', N'IX\_­Transaction­History­Archive\_­Reference­Order­ID\_­Reference­Order­Line­ID'  GO  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'  GO |

Uses

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

|  |
| --- |
| [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 |
|  | Unit­Measure­Code Primary key. | nchar(3) | 6 | NOT NULL |  |
|  | Name Unit of measure description. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NOT NULL |  |
|  | Modified­Date Date and time the record was last updated. | datetime | 8 | NOT NULL | (getdate()) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Unit­Measure\_­Unit­Measure­Code Primary key (clustered) constraint | Unit­Measure­Code | True |
|  | AK\_­Unit­Measure\_­Name Unique nonclustered index. | Name | True |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Unit­Measure]  (  [Unit­Measure­Code] [nchar] (3) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Name] [dbo].[Name] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Unit­Measure\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Unit­Measure] ADD CONSTRAINT [PK\_­Unit­Measure\_­Unit­Measure­Code] PRIMARY KEY CLUSTERED ([Unit­Measure­Code]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Unit­Measure\_­Name] ON [Production].[Unit­Measure] ([Name]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unit of measure lookup table.', 'SCHEMA', N'Production', 'TABLE', N'Unit­Measure', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Unit­Measure', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unit of measure description.', 'SCHEMA', N'Production', 'TABLE', N'Unit­Measure', 'COLUMN', N'Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key.', 'SCHEMA', N'Production', 'TABLE', N'Unit­Measure', 'COLUMN', N'Unit­Measure­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Unit­Measure', 'CONSTRAINT', N'DF\_­Unit­Measure\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Unit­Measure', 'CONSTRAINT', N'PK\_­Unit­Measure\_­Unit­Measure­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Unit­Measure', 'INDEX', N'AK\_­Unit­Measure\_­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Unit­Measure', 'INDEX', N'PK\_­Unit­Measure\_­Unit­Measure­Code'  GO |

Uses

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

Used By

[[Production].[Bill­Of­Materials]](#m9P+2olmZFXOowkUY18+Ri7YHx4=)

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[[Purchasing].[Product­Vendor]](#igWcZ2fEqwA+AlsF0xQn66oTGAU=)

|  |
| --- |
| [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 | Computed | Max Length (Bytes) | Nullability | Identity | Default |
|  | Work­Order­ID Primary key for Work­Order records. | int |  | 4 | NOT NULL | 1 - 1 |  |
|  | 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 |  |  |
|  | 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\_­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 Transaction­History table. | True | True | After Insert |
| u­Work­Order AFTER UPDATE trigger that inserts a row in the Transaction­History 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]>=[Start­Date] 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]](#mO6TpkyW7q4W/OMRHISuYfWNCME=) |
| FK\_­Work­Order\_­Scrap­Reason\_­Scrap­Reason­ID Foreign key constraint referencing Scrap­Reason.Scrap­Reason­ID. | Scrap­Reason­ID->[[Production].[Scrap­Reason].[Scrap­Reason­ID]](#0dK3gJTjqlelVRIns1VtvoCHDlU=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Work­Order]  (  [Work­Order­ID] [int] NOT NULL IDENTITY(1, 1),  [Product­ID] [int] NOT NULL,  [Order­Qty] [int] NOT NULL,  [Stocked­Qty] AS (isnull([Order­Qty]-[Scrapped­Qty],(0))),  [Scrapped­Qty] [smallint] NOT NULL,  [Start­Date] [datetime] NOT NULL,  [End­Date] [datetime] NULL,  [Due­Date] [datetime] NOT NULL,  [Scrap­Reason­ID] [smallint] NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Work­Order\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  CREATE TRIGGER [Production].[i­Work­Order] ON [Production].[Work­Order]  AFTER INSERT AS  BEGIN  DECLARE @Count int;  SET @Count = @@ROWCOUNT;  IF @Count = 0  RETURN;  SET NOCOUNT ON;  BEGIN TRY  INSERT INTO [Production].[Transaction­History](  [Product­ID]  ,[Reference­Order­ID]  ,[Transaction­Type]  ,[Transaction­Date]  ,[Quantity]  ,[Actual­Cost])  SELECT  inserted.[Product­ID]  ,inserted.[Work­Order­ID]  ,'W'  ,GETDATE()  ,inserted.[Order­Qty]  ,0  FROM inserted;  END TRY  BEGIN CATCH  EXECUTE [dbo].[usp­Print­Error];  -- Rollback any active or uncommittable transactions before  -- inserting information in the Error­Log  IF @@TRANCOUNT > 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE [dbo].[usp­Log­Error];  END CATCH;  END;  GO  CREATE TRIGGER [Production].[u­Work­Order] ON [Production].[Work­Order]  AFTER UPDATE AS  BEGIN  DECLARE @Count int;  SET @Count = @@ROWCOUNT;  IF @Count = 0  RETURN;  SET NOCOUNT ON;  BEGIN TRY  IF UPDATE([Product­ID]) OR UPDATE([Order­Qty])  BEGIN  INSERT INTO [Production].[Transaction­History](  [Product­ID]  ,[Reference­Order­ID]  ,[Transaction­Type]  ,[Transaction­Date]  ,[Quantity])  SELECT  inserted.[Product­ID]  ,inserted.[Work­Order­ID]  ,'W'  ,GETDATE()  ,inserted.[Order­Qty]  FROM inserted;  END;  END TRY  BEGIN CATCH  EXECUTE [dbo].[usp­Print­Error];  -- Rollback any active or uncommittable transactions before  -- inserting information in the Error­Log  IF @@TRANCOUNT > 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE [dbo].[usp­Log­Error];  END CATCH;  END;  GO  ALTER TABLE [Production].[Work­Order] ADD CONSTRAINT [CK\_­Work­Order\_­End­Date] CHECK (([End­Date]>=[Start­Date] OR [End­Date] IS NULL))  GO  ALTER TABLE [Production].[Work­Order] ADD CONSTRAINT [CK\_­Work­Order\_­Order­Qty] CHECK (([Order­Qty]>(0)))  GO  ALTER TABLE [Production].[Work­Order] ADD CONSTRAINT [CK\_­Work­Order\_­Scrapped­Qty] CHECK (([Scrapped­Qty]>=(0)))  GO  ALTER TABLE [Production].[Work­Order] ADD CONSTRAINT [PK\_­Work­Order\_­Work­Order­ID] PRIMARY KEY CLUSTERED ([Work­Order­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Work­Order\_­Product­ID] ON [Production].[Work­Order] ([Product­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Work­Order\_­Scrap­Reason­ID] ON [Production].[Work­Order] ([Scrap­Reason­ID]) ON [PRIMARY]  GO  ALTER TABLE [Production].[Work­Order] ADD CONSTRAINT [FK\_­Work­Order\_­Product\_­Product­ID] FOREIGN KEY ([Product­ID]) REFERENCES [Production].[Product] ([Product­ID])  GO  ALTER TABLE [Production].[Work­Order] ADD CONSTRAINT [FK\_­Work­Order\_­Scrap­Reason\_­Scrap­Reason­ID] FOREIGN KEY ([Scrap­Reason­ID]) REFERENCES [Production].[Scrap­Reason] ([Scrap­Reason­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Manufacturing work orders.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Work order due date.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'COLUMN', N'Due­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Work order end date.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'COLUMN', N'End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product quantity to build.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'COLUMN', N'Order­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product identification number. Foreign key to Product.Product­ID.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'COLUMN', N'Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Quantity that failed inspection.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'COLUMN', N'Scrapped­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Reason for inspection failure.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'COLUMN', N'Scrap­Reason­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Work order start date.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'COLUMN', N'Start­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Quantity built and put in inventory.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'COLUMN', N'Stocked­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Work­Order records.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'COLUMN', N'Work­Order­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [End­Date] >= [Start­Date] OR [End­Date] IS NULL', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'CONSTRAINT', N'CK\_­Work­Order\_­End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Order­Qty] > (0)', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'CONSTRAINT', N'CK\_­Work­Order\_­Order­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Scrapped­Qty] >= (0)', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'CONSTRAINT', N'CK\_­Work­Order\_­Scrapped­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'CONSTRAINT', N'DF\_­Work­Order\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Product.Product­ID.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'CONSTRAINT', N'FK\_­Work­Order\_­Product\_­Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Scrap­Reason.Scrap­Reason­ID.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'CONSTRAINT', N'FK\_­Work­Order\_­Scrap­Reason\_­Scrap­Reason­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'CONSTRAINT', N'PK\_­Work­Order\_­Work­Order­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'INDEX', N'IX\_­Work­Order\_­Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'INDEX', N'IX\_­Work­Order\_­Scrap­Reason­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'INDEX', N'PK\_­Work­Order\_­Work­Order­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'AFTER INSERT trigger that inserts a row in the Transaction­History table.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'TRIGGER', N'i­Work­Order'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'AFTER UPDATE trigger that inserts a row in the Transaction­History table, updates Modified­Date in the Work­Order table.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order', 'TRIGGER', N'u­Work­Order'  GO |

Uses

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[[Production].[Scrap­Reason]](#0dK3gJTjqlelVRIns1VtvoCHDlU=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

Used By

[[Production].[Work­Order­Routing]](#wxYgGkEd/o/k/roV2tY7ZCLH+qE=)

|  |
| --- |
| [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 |
|  | Work­Order­ID Primary key. Foreign key to Work­Order.Work­Order­ID. | int | 4 | NOT NULL |  |
|  | Product­ID Primary key. Foreign key to Product.Product­ID. | int | 4 | NOT NULL |  |
|  | Operation­Sequence Primary key. Indicates the manufacturing process sequence. | smallint | 2 | NOT NULL |  |
|  | 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 |  |
|  | Actual­Resource­Hrs Number of manufacturing hours used. | decimal(9,4) | 5 | NULL allowed |  |
|  | Planned­Cost Estimated manufacturing cost. | money | 8 | NOT NULL |  |
|  | 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 |
|  | PK\_­Work­Order­Routing\_­Work­Order­ID\_­Product­ID\_­Operation­Sequence Primary key (clustered) constraint | Work­Order­ID, Product­ID, Operation­Sequence | True |
|  | IX\_­Work­Order­Routing\_­Product­ID Nonclustered index. | Product­ID |  |

Check Constraints

|  |  |  |
| --- | --- | --- |
| Name | On Column | Constraint |
| CK\_­Work­Order­Routing\_­Actual­Cost Check constraint [Actual­Cost] > (0.00) | Actual­Cost | ([Actual­Cost]>(0.00)) |
| CK\_­Work­Order­Routing\_­Actual­End­Date Check constraint [Actual­End­Date] >= [Actual­Start­Date] OR [Actual­End­Date] IS NULL OR [Actual­Start­Date] IS NULL |  | ([Actual­End­Date]>=[Actual­Start­Date] OR [Actual­End­Date] IS NULL OR [Actual­Start­Date] IS NULL) |
| CK\_­Work­Order­Routing\_­Actual­Resource­Hrs Check constraint [Actual­Resource­Hrs] >= (0.0000) | Actual­Resource­Hrs | ([Actual­Resource­Hrs]>=(0.0000)) |
| CK\_­Work­Order­Routing\_­Planned­Cost Check constraint [Planned­Cost] > (0.00) | Planned­Cost | ([Planned­Cost]>(0.00)) |
| CK\_­Work­Order­Routing\_­Scheduled­End­Date Check constraint [Scheduled­End­Date] >= [Scheduled­Start­Date] |  | ([Scheduled­End­Date]>=[Scheduled­Start­Date]) |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Work­Order­Routing\_­Location\_­Location­ID Foreign key constraint referencing Location.Location­ID. | Location­ID->[[Production].[Location].[Location­ID]](#ciHD61a29Ub1OJEWhZ1zarNhvcs=) |
| FK\_­Work­Order­Routing\_­Work­Order\_­Work­Order­ID Foreign key constraint referencing Work­Order.Work­Order­ID. | Work­Order­ID->[[Production].[Work­Order].[Work­Order­ID]](#BSRICvfPko6shp7o8HSu003oCno=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Production].[Work­Order­Routing]  (  [Work­Order­ID] [int] NOT NULL,  [Product­ID] [int] NOT NULL,  [Operation­Sequence] [smallint] NOT NULL,  [Location­ID] [smallint] NOT NULL,  [Scheduled­Start­Date] [datetime] NOT NULL,  [Scheduled­End­Date] [datetime] NOT NULL,  [Actual­Start­Date] [datetime] NULL,  [Actual­End­Date] [datetime] NULL,  [Actual­Resource­Hrs] [decimal] (9, 4) NULL,  [Planned­Cost] [money] NOT NULL,  [Actual­Cost] [money] NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Work­Order­Routing\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Production].[Work­Order­Routing] ADD CONSTRAINT [CK\_­Work­Order­Routing\_­Actual­Cost] CHECK (([Actual­Cost]>(0.00)))  GO  ALTER TABLE [Production].[Work­Order­Routing] ADD CONSTRAINT [CK\_­Work­Order­Routing\_­Actual­End­Date] CHECK (([Actual­End­Date]>=[Actual­Start­Date] OR [Actual­End­Date] IS NULL OR [Actual­Start­Date] IS NULL))  GO  ALTER TABLE [Production].[Work­Order­Routing] ADD CONSTRAINT [CK\_­Work­Order­Routing\_­Actual­Resource­Hrs] CHECK (([Actual­Resource­Hrs]>=(0.0000)))  GO  ALTER TABLE [Production].[Work­Order­Routing] ADD CONSTRAINT [CK\_­Work­Order­Routing\_­Planned­Cost] CHECK (([Planned­Cost]>(0.00)))  GO  ALTER TABLE [Production].[Work­Order­Routing] ADD CONSTRAINT [CK\_­Work­Order­Routing\_­Scheduled­End­Date] CHECK (([Scheduled­End­Date]>=[Scheduled­Start­Date]))  GO  ALTER TABLE [Production].[Work­Order­Routing] ADD CONSTRAINT [PK\_­Work­Order­Routing\_­Work­Order­ID\_­Product­ID\_­Operation­Sequence] PRIMARY KEY CLUSTERED ([Work­Order­ID], [Product­ID], [Operation­Sequence]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Work­Order­Routing\_­Product­ID] ON [Production].[Work­Order­Routing] ([Product­ID]) ON [PRIMARY]  GO  ALTER TABLE [Production].[Work­Order­Routing] ADD CONSTRAINT [FK\_­Work­Order­Routing\_­Location\_­Location­ID] FOREIGN KEY ([Location­ID]) REFERENCES [Production].[Location] ([Location­ID])  GO  ALTER TABLE [Production].[Work­Order­Routing] ADD CONSTRAINT [FK\_­Work­Order­Routing\_­Work­Order\_­Work­Order­ID] FOREIGN KEY ([Work­Order­ID]) REFERENCES [Production].[Work­Order] ([Work­Order­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Work order details.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order­Routing', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Actual manufacturing cost.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order­Routing', 'COLUMN', N'Actual­Cost'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Actual end date.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order­Routing', 'COLUMN', N'Actual­End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Number of manufacturing hours used.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order­Routing', 'COLUMN', N'Actual­Resource­Hrs'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Actual start date.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order­Routing', 'COLUMN', N'Actual­Start­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Manufacturing location where the part is processed. Foreign key to Location.Location­ID.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order­Routing', 'COLUMN', N'Location­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order­Routing', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Indicates the manufacturing process sequence.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order­Routing', 'COLUMN', N'Operation­Sequence'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Estimated manufacturing cost.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order­Routing', 'COLUMN', N'Planned­Cost'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Foreign key to Product.Product­ID.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order­Routing', 'COLUMN', N'Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Planned manufacturing end date.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order­Routing', 'COLUMN', N'Scheduled­End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Planned manufacturing start date.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order­Routing', 'COLUMN', N'Scheduled­Start­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Foreign key to Work­Order.Work­Order­ID.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order­Routing', 'COLUMN', N'Work­Order­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Actual­Cost] > (0.00)', 'SCHEMA', N'Production', 'TABLE', N'Work­Order­Routing', 'CONSTRAINT', N'CK\_­Work­Order­Routing\_­Actual­Cost'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Actual­End­Date] >= [Actual­Start­Date] OR [Actual­End­Date] IS NULL OR [Actual­Start­Date] IS NULL', 'SCHEMA', N'Production', 'TABLE', N'Work­Order­Routing', 'CONSTRAINT', N'CK\_­Work­Order­Routing\_­Actual­End­Date'  GO  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\_­Work­Order­Routing\_­Planned­Cost'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Scheduled­End­Date] >= [Scheduled­Start­Date]', 'SCHEMA', N'Production', 'TABLE', N'Work­Order­Routing', 'CONSTRAINT', N'CK\_­Work­Order­Routing\_­Scheduled­End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Production', 'TABLE', N'Work­Order­Routing', 'CONSTRAINT', N'DF\_­Work­Order­Routing\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Location.Location­ID.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order­Routing', 'CONSTRAINT', N'FK\_­Work­Order­Routing\_­Location\_­Location­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Work­Order.Work­Order­ID.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order­Routing', 'CONSTRAINT', N'FK\_­Work­Order­Routing\_­Work­Order\_­Work­Order­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Production', 'TABLE', N'Work­Order­Routing', 'CONSTRAINT', N'PK\_­Work­Order­Routing\_­Work­Order­ID\_­Product­ID\_­Operation­Sequence'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Production', 'TABLE', N'Work­Order­Routing', 'INDEX', N'IX\_­Work­Order­Routing\_­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 |

Uses

[[Production].[Location]](#ciHD61a29Ub1OJEWhZ1zarNhvcs=)

[[Production].[Work­Order]](#BSRICvfPko6shp7o8HSu003oCno=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

|  |
| --- |
| [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 |
|  | Product­ID Primary key. Foreign key to Product.Product­ID. | int | 4 | NOT NULL |  |
|  | Business­Entity­ID Primary key. Foreign key to Vendor.Business­Entity­ID. | int | 4 | NOT NULL |  |
|  | 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 |  |
|  | Min­Order­Qty The maximum quantity that should be ordered. | int | 4 | NOT NULL |  |
|  | Max­Order­Qty The minimum quantity that should be ordered. | int | 4 | NOT NULL |  |
|  | On­Order­Qty The quantity currently on order. | int | 4 | NULL allowed |  |
|  | 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\_­Product­Vendor\_­Product­ID\_­Business­Entity­ID Primary key (clustered) constraint | Product­ID, Business­Entity­ID | 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]](#mO6TpkyW7q4W/OMRHISuYfWNCME=) |
| FK\_­Product­Vendor\_­Unit­Measure\_­Unit­Measure­Code Foreign key constraint referencing Unit­Measure.Unit­Measure­Code. | Unit­Measure­Code->[[Production].[Unit­Measure].[Unit­Measure­Code]](#11scPpeMkfJepgAP6eC9jOIFK+Y=) |
| FK\_­Product­Vendor\_­Vendor\_­Business­Entity­ID Foreign key constraint referencing Vendor.Business­Entity­ID. | Business­Entity­ID->[[Purchasing].[Vendor].[Business­Entity­ID]](#yfc9dLywcQtxtbztLjV6J3bb0tM=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Purchasing].[Product­Vendor]  (  [Product­ID] [int] NOT NULL,  [Business­Entity­ID] [int] NOT NULL,  [Average­Lead­Time] [int] NOT NULL,  [Standard­Price] [money] NOT NULL,  [Last­Receipt­Cost] [money] NULL,  [Last­Receipt­Date] [datetime] NULL,  [Min­Order­Qty] [int] NOT NULL,  [Max­Order­Qty] [int] NOT NULL,  [On­Order­Qty] [int] NULL,  [Unit­Measure­Code] [nchar] (3) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Product­Vendor\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Purchasing].[Product­Vendor] ADD CONSTRAINT [CK\_­Product­Vendor\_­Average­Lead­Time] CHECK (([Average­Lead­Time]>=(1)))  GO  ALTER TABLE [Purchasing].[Product­Vendor] ADD CONSTRAINT [CK\_­Product­Vendor\_­Last­Receipt­Cost] CHECK (([Last­Receipt­Cost]>(0.00)))  GO  ALTER TABLE [Purchasing].[Product­Vendor] ADD CONSTRAINT [CK\_­Product­Vendor\_­Max­Order­Qty] CHECK (([Max­Order­Qty]>=(1)))  GO  ALTER TABLE [Purchasing].[Product­Vendor] ADD CONSTRAINT [CK\_­Product­Vendor\_­Min­Order­Qty] CHECK (([Min­Order­Qty]>=(1)))  GO  ALTER TABLE [Purchasing].[Product­Vendor] ADD CONSTRAINT [CK\_­Product­Vendor\_­On­Order­Qty] CHECK (([On­Order­Qty]>=(0)))  GO  ALTER TABLE [Purchasing].[Product­Vendor] ADD CONSTRAINT [CK\_­Product­Vendor\_­Standard­Price] CHECK (([Standard­Price]>(0.00)))  GO  ALTER TABLE [Purchasing].[Product­Vendor] ADD CONSTRAINT [PK\_­Product­Vendor\_­Product­ID\_­Business­Entity­ID] PRIMARY KEY CLUSTERED ([Product­ID], [Business­Entity­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Product­Vendor\_­Business­Entity­ID] ON [Purchasing].[Product­Vendor] ([Business­Entity­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Product­Vendor\_­Unit­Measure­Code] ON [Purchasing].[Product­Vendor] ([Unit­Measure­Code]) ON [PRIMARY]  GO  ALTER TABLE [Purchasing].[Product­Vendor] ADD CONSTRAINT [FK\_­Product­Vendor\_­Product\_­Product­ID] FOREIGN KEY ([Product­ID]) REFERENCES [Production].[Product] ([Product­ID])  GO  ALTER TABLE [Purchasing].[Product­Vendor] ADD CONSTRAINT [FK\_­Product­Vendor\_­Unit­Measure\_­Unit­Measure­Code] FOREIGN KEY ([Unit­Measure­Code]) REFERENCES [Production].[Unit­Measure] ([Unit­Measure­Code])  GO  ALTER TABLE [Purchasing].[Product­Vendor] ADD CONSTRAINT [FK\_­Product­Vendor\_­Vendor\_­Business­Entity­ID] FOREIGN KEY ([Business­Entity­ID]) REFERENCES [Purchasing].[Vendor] ([Business­Entity­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Cross-reference table mapping vendors with the products they supply.', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The average span of time (in days) between placing an order with the vendor and receiving the purchased product.', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'COLUMN', N'Average­Lead­Time'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Foreign key to Vendor.Business­Entity­ID.', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'COLUMN', N'Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The selling price when last purchased.', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'COLUMN', N'Last­Receipt­Cost'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date the product was last received by the vendor.', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'COLUMN', N'Last­Receipt­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The minimum quantity that should be ordered.', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'COLUMN', N'Max­Order­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The maximum quantity that should be ordered.', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'COLUMN', N'Min­Order­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The quantity currently on order.', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'COLUMN', N'On­Order­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Foreign key to Product.Product­ID.', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'COLUMN', N'Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The vendor''s usual selling price.', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'COLUMN', N'Standard­Price'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The product''s unit of measure.', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'COLUMN', N'Unit­Measure­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Average­Lead­Time] >= (1)', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'CONSTRAINT', N'CK\_­Product­Vendor\_­Average­Lead­Time'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Last­Receipt­Cost] > (0.00)', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'CONSTRAINT', N'CK\_­Product­Vendor\_­Last­Receipt­Cost'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Max­Order­Qty] >= (1)', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'CONSTRAINT', N'CK\_­Product­Vendor\_­Max­Order­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Min­Order­Qty] >= (1)', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'CONSTRAINT', N'CK\_­Product­Vendor\_­Min­Order­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [On­Order­Qty] >= (0)', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'CONSTRAINT', N'CK\_­Product­Vendor\_­On­Order­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Standard­Price] > (0.00)', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'CONSTRAINT', N'CK\_­Product­Vendor\_­Standard­Price'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'CONSTRAINT', N'DF\_­Product­Vendor\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Product.Product­ID.', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'CONSTRAINT', N'FK\_­Product­Vendor\_­Product\_­Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Unit­Measure.Unit­Measure­Code.', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'CONSTRAINT', N'FK\_­Product­Vendor\_­Unit­Measure\_­Unit­Measure­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Vendor.Business­Entity­ID.', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'CONSTRAINT', N'FK\_­Product­Vendor\_­Vendor\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'CONSTRAINT', N'PK\_­Product­Vendor\_­Product­ID\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'INDEX', N'IX\_­Product­Vendor\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'INDEX', N'IX\_­Product­Vendor\_­Unit­Measure­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Purchasing', 'TABLE', N'Product­Vendor', 'INDEX', N'PK\_­Product­Vendor\_­Product­ID\_­Business­Entity­ID'  GO |

Uses

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[[Production].[Unit­Measure]](#11scPpeMkfJepgAP6eC9jOIFK+Y=)

[[Purchasing].[Vendor]](#yfc9dLywcQtxtbztLjV6J3bb0tM=)

[Purchasing](#r48uLAuwsJTIAvAuIPBTbVrjT9s=)

|  |
| --- |
| [Purchasing].[Purchase­Order­Detail] |

MS\_­Description

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

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 | Computed | Max Length (Bytes) | Nullability | Identity | Default |
|  | Purchase­Order­ID Primary key. Foreign key to Purchase­Order­Header.Purchase­Order­ID. | int |  | 4 | NOT NULL |  |  |
|  | 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 |  |  |
|  | 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 Received­Qty - Rejected­Qty. | decimal(9,2) | True | 5 | NOT NULL |  |  |
|  | 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 |
|  | PK\_­Purchase­Order­Detail\_­Purchase­Order­ID\_­Purchase­Order­Detail­ID Primary key (clustered) constraint | Purchase­Order­ID, Purchase­Order­Detail­ID | 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 Transaction­History table and updates the Purchase­Order­Header.Sub­Total column. | True | True | After Insert |
| u­Purchase­Order­Detail AFTER UPDATE trigger that inserts a row in the Transaction­History 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]](#mO6TpkyW7q4W/OMRHISuYfWNCME=) |
| FK\_­Purchase­Order­Detail\_­Purchase­Order­Header\_­Purchase­Order­ID Foreign key constraint referencing Purchase­Order­Header.Purchase­Order­ID. | Purchase­Order­ID->[[Purchasing].[Purchase­Order­Header].[Purchase­Order­ID]](#wVItRHlooKKFqOhuItRUJuA60Fs=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Purchasing].[Purchase­Order­Detail]  (  [Purchase­Order­ID] [int] NOT NULL,  [Purchase­Order­Detail­ID] [int] NOT NULL IDENTITY(1, 1),  [Due­Date] [datetime] NOT NULL,  [Order­Qty] [smallint] NOT NULL,  [Product­ID] [int] NOT NULL,  [Unit­Price] [money] NOT NULL,  [Line­Total] AS (isnull([Order­Qty]\*[Unit­Price],(0.00))),  [Received­Qty] [decimal] (8, 2) NOT NULL,  [Rejected­Qty] [decimal] (8, 2) NOT NULL,  [Stocked­Qty] AS (isnull([Received­Qty]-[Rejected­Qty],(0.00))),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Purchase­Order­Detail\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  CREATE TRIGGER [Purchasing].[i­Purchase­Order­Detail] ON [Purchasing].[Purchase­Order­Detail]  AFTER INSERT AS  BEGIN  DECLARE @Count int;  SET @Count = @@ROWCOUNT;  IF @Count = 0  RETURN;  SET NOCOUNT ON;  BEGIN TRY  INSERT INTO [Production].[Transaction­History]  ([Product­ID]  ,[Reference­Order­ID]  ,[Reference­Order­Line­ID]  ,[Transaction­Type]  ,[Transaction­Date]  ,[Quantity]  ,[Actual­Cost])  SELECT  inserted.[Product­ID]  ,inserted.[Purchase­Order­ID]  ,inserted.[Purchase­Order­Detail­ID]  ,'P'  ,GETDATE()  ,inserted.[Order­Qty]  ,inserted.[Unit­Price]  FROM inserted  INNER JOIN [Purchasing].[Purchase­Order­Header]  ON inserted.[Purchase­Order­ID] = [Purchasing].[Purchase­Order­Header].[Purchase­Order­ID];  -- Update Sub­Total in Purchase­Order­Header record. Note that this causes the  -- Purchase­Order­Header trigger to fire which will update the Revision­Number.  UPDATE [Purchasing].[Purchase­Order­Header]  SET [Purchasing].[Purchase­Order­Header].[Sub­Total] =  (SELECT SUM([Purchasing].[Purchase­Order­Detail].[Line­Total])  FROM [Purchasing].[Purchase­Order­Detail]  WHERE [Purchasing].[Purchase­Order­Header].[Purchase­Order­ID] = [Purchasing].[Purchase­Order­Detail].[Purchase­Order­ID])  WHERE [Purchasing].[Purchase­Order­Header].[Purchase­Order­ID] IN (SELECT inserted.[Purchase­Order­ID] FROM inserted);  END TRY  BEGIN CATCH  EXECUTE [dbo].[usp­Print­Error];  -- Rollback any active or uncommittable transactions before  -- inserting information in the Error­Log  IF @@TRANCOUNT > 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE [dbo].[usp­Log­Error];  END CATCH;  END;  GO  CREATE TRIGGER [Purchasing].[u­Purchase­Order­Detail] ON [Purchasing].[Purchase­Order­Detail]  AFTER UPDATE AS  BEGIN  DECLARE @Count int;  SET @Count = @@ROWCOUNT;  IF @Count = 0  RETURN;  SET NOCOUNT ON;  BEGIN TRY  IF UPDATE([Product­ID]) OR UPDATE([Order­Qty]) OR UPDATE([Unit­Price])  -- Insert record into Transaction­History  BEGIN  INSERT INTO [Production].[Transaction­History]  ([Product­ID]  ,[Reference­Order­ID]  ,[Reference­Order­Line­ID]  ,[Transaction­Type]  ,[Transaction­Date]  ,[Quantity]  ,[Actual­Cost])  SELECT  inserted.[Product­ID]  ,inserted.[Purchase­Order­ID]  ,inserted.[Purchase­Order­Detail­ID]  ,'P'  ,GETDATE()  ,inserted.[Order­Qty]  ,inserted.[Unit­Price]  FROM inserted  INNER JOIN [Purchasing].[Purchase­Order­Detail]  ON inserted.[Purchase­Order­ID] = [Purchasing].[Purchase­Order­Detail].[Purchase­Order­ID];  -- Update Sub­Total in Purchase­Order­Header record. Note that this causes the  -- Purchase­Order­Header trigger to fire which will update the Revision­Number.  UPDATE [Purchasing].[Purchase­Order­Header]  SET [Purchasing].[Purchase­Order­Header].[Sub­Total] =  (SELECT SUM([Purchasing].[Purchase­Order­Detail].[Line­Total])  FROM [Purchasing].[Purchase­Order­Detail]  WHERE [Purchasing].[Purchase­Order­Header].[Purchase­Order­ID]  = [Purchasing].[Purchase­Order­Detail].[Purchase­Order­ID])  WHERE [Purchasing].[Purchase­Order­Header].[Purchase­Order­ID]  IN (SELECT inserted.[Purchase­Order­ID] FROM inserted);  UPDATE [Purchasing].[Purchase­Order­Detail]  SET [Purchasing].[Purchase­Order­Detail].[Modified­Date] = GETDATE()  FROM inserted  WHERE inserted.[Purchase­Order­ID] = [Purchasing].[Purchase­Order­Detail].[Purchase­Order­ID]  AND inserted.[Purchase­Order­Detail­ID] = [Purchasing].[Purchase­Order­Detail].[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)))  GO  ALTER TABLE [Purchasing].[Purchase­Order­Detail] ADD CONSTRAINT [CK\_­Purchase­Order­Detail\_­Received­Qty] CHECK (([Received­Qty]>=(0.00)))  GO  ALTER TABLE [Purchasing].[Purchase­Order­Detail] ADD CONSTRAINT [CK\_­Purchase­Order­Detail\_­Rejected­Qty] CHECK (([Rejected­Qty]>=(0.00)))  GO  ALTER TABLE [Purchasing].[Purchase­Order­Detail] ADD CONSTRAINT [CK\_­Purchase­Order­Detail\_­Unit­Price] CHECK (([Unit­Price]>=(0.00)))  GO  ALTER TABLE [Purchasing].[Purchase­Order­Detail] ADD CONSTRAINT [PK\_­Purchase­Order­Detail\_­Purchase­Order­ID\_­Purchase­Order­Detail­ID] PRIMARY KEY CLUSTERED ([Purchase­Order­ID], [Purchase­Order­Detail­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Purchase­Order­Detail\_­Product­ID] ON [Purchasing].[Purchase­Order­Detail] ([Product­ID]) ON [PRIMARY]  GO  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])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Individual products associated with a specific purchase order. See Purchase­Order­Header.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date the product is expected to be received.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', 'COLUMN', N'Due­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Per product subtotal. Computed as Order­Qty \* Unit­Price.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', 'COLUMN', N'Line­Total'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Quantity ordered.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', 'COLUMN', N'Order­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product identification number. Foreign key to Product.Product­ID.', 'SCHEMA', N'Purchasing', 'TABLE', N'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 Purchase­Order­Header.Purchase­Order­ID.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', 'COLUMN', N'Purchase­Order­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Quantity actually received from the vendor.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', 'COLUMN', N'Received­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Quantity rejected during inspection.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', 'COLUMN', N'Rejected­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Quantity accepted into inventory. Computed as Received­Qty - Rejected­Qty.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', 'COLUMN', N'Stocked­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Vendor''s selling price of a single product.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', 'COLUMN', N'Unit­Price'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Order­Qty] > (0)', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', 'CONSTRAINT', N'CK\_­Purchase­Order­Detail\_­Order­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Received­Qty] >= (0.00)', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', 'CONSTRAINT', N'CK\_­Purchase­Order­Detail\_­Received­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Rejected­Qty] >= (0.00)', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', 'CONSTRAINT', N'CK\_­Purchase­Order­Detail\_­Rejected­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Unit­Price] >= (0.00)', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', 'CONSTRAINT', N'CK\_­Purchase­Order­Detail\_­Unit­Price'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', 'CONSTRAINT', N'DF\_­Purchase­Order­Detail\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Product.Product­ID.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', 'CONSTRAINT', N'FK\_­Purchase­Order­Detail\_­Product\_­Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Purchase­Order­Header.Purchase­Order­ID.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', 'CONSTRAINT', N'FK\_­Purchase­Order­Detail\_­Purchase­Order­Header\_­Purchase­Order­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', 'CONSTRAINT', N'PK\_­Purchase­Order­Detail\_­Purchase­Order­ID\_­Purchase­Order­Detail­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', 'INDEX', N'IX\_­Purchase­Order­Detail\_­Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', 'INDEX', N'PK\_­Purchase­Order­Detail\_­Purchase­Order­ID\_­Purchase­Order­Detail­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'AFTER INSERT trigger that inserts a row in the Transaction­History table and updates the Purchase­Order­Header.Sub­Total column.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', 'TRIGGER', N'i­Purchase­Order­Detail'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'AFTER UPDATE trigger that inserts a row in the Transaction­History table, updates Modified­Date in Purchase­Order­Detail and updates the Purchase­Order­Header.Sub­Total column.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Detail', 'TRIGGER', N'u­Purchase­Order­Detail'  GO |

Uses

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[[Purchasing].[Purchase­Order­Header]](#wVItRHlooKKFqOhuItRUJuA60Fs=)

[Purchasing](#r48uLAuwsJTIAvAuIPBTbVrjT9s=)

|  |
| --- |
| [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 | Persisted | Computed | Max Length (Bytes) | Nullability | Identity | Default |
|  | 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)) |
|  | Employee­ID Employee who created the purchase order. Foreign key to Employee.Business­Entity­ID. | int |  |  | 4 | NOT NULL |  |  |
|  | Vendor­ID Vendor with whom the purchase order is placed. Foreign key to Vendor.Business­Entity­ID. | int |  |  | 4 | NOT NULL |  |  |
|  | Ship­Method­ID Shipping method. Foreign key to Ship­Method.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(Purchase­Order­Detail.Line­Total)for the appropriate Purchase­Order­ID. | 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\_­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]>=[Order­Date] 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.Employee­ID. | Employee­ID->[[Human­Resources].[Employee].[Business­Entity­ID]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=) |
| FK\_­Purchase­Order­Header\_­Ship­Method\_­Ship­Method­ID Foreign key constraint referencing Ship­Method.Ship­Method­ID. | Ship­Method­ID->[[Purchasing].[Ship­Method].[Ship­Method­ID]](#Xt5ka3ip5ljwxYo+n5TjCRZypoY=) |
| FK\_­Purchase­Order­Header\_­Vendor\_­Vendor­ID Foreign key constraint referencing Vendor.Vendor­ID. | Vendor­ID->[[Purchasing].[Vendor].[Business­Entity­ID]](#yfc9dLywcQtxtbztLjV6J3bb0tM=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Purchasing].[Purchase­Order­Header]  (  [Purchase­Order­ID] [int] NOT NULL IDENTITY(1, 1),  [Revision­Number] [tinyint] NOT NULL CONSTRAINT [DF\_­Purchase­Order­Header\_­Revision­Number] DEFAULT ((0)),  [Status] [tinyint] NOT NULL CONSTRAINT [DF\_­Purchase­Order­Header\_­Status] DEFAULT ((1)),  [Employee­ID] [int] NOT NULL,  [Vendor­ID] [int] NOT NULL,  [Ship­Method­ID] [int] NOT NULL,  [Order­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Purchase­Order­Header\_­Order­Date] DEFAULT (getdate()),  [Ship­Date] [datetime] NULL,  [Sub­Total] [money] NOT NULL CONSTRAINT [DF\_­Purchase­Order­Header\_­Sub­Total] DEFAULT ((0.00)),  [Tax­Amt] [money] NOT NULL CONSTRAINT [DF\_­Purchase­Order­Header\_­Tax­Amt] DEFAULT ((0.00)),  [Freight] [money] NOT NULL CONSTRAINT [DF\_­Purchase­Order­Header\_­Freight] DEFAULT ((0.00)),  [Total­Due] AS (isnull(([Sub­Total]+[Tax­Amt])+[Freight],(0))) PERSISTED NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Purchase­Order­Header\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  CREATE TRIGGER [Purchasing].[u­Purchase­Order­Header] ON [Purchasing].[Purchase­Order­Header]  AFTER UPDATE AS  BEGIN  DECLARE @Count int;  SET @Count = @@ROWCOUNT;  IF @Count = 0  RETURN;  SET NOCOUNT ON;  BEGIN TRY  -- Update Revision­Number for modification of any field EXCEPT the Status.  IF NOT UPDATE([Status])  BEGIN  UPDATE [Purchasing].[Purchase­Order­Header]  SET [Purchasing].[Purchase­Order­Header].[Revision­Number] =  [Purchasing].[Purchase­Order­Header].[Revision­Number] + 1  WHERE [Purchasing].[Purchase­Order­Header].[Purchase­Order­ID] IN  (SELECT inserted.[Purchase­Order­ID] FROM inserted);  END;  END TRY  BEGIN CATCH  EXECUTE [dbo].[usp­Print­Error];  -- Rollback any active or uncommittable transactions before  -- inserting information in the Error­Log  IF @@TRANCOUNT > 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE [dbo].[usp­Log­Error];  END CATCH;  END;  GO  ALTER TABLE [Purchasing].[Purchase­Order­Header] ADD CONSTRAINT [CK\_­Purchase­Order­Header\_­Freight] CHECK (([Freight]>=(0.00)))  GO  ALTER TABLE [Purchasing].[Purchase­Order­Header] ADD CONSTRAINT [CK\_­Purchase­Order­Header\_­Ship­Date] CHECK (([Ship­Date]>=[Order­Date] OR [Ship­Date] IS NULL))  GO  ALTER TABLE [Purchasing].[Purchase­Order­Header] ADD CONSTRAINT [CK\_­Purchase­Order­Header\_­Status] CHECK (([Status]>=(1) AND [Status]<=(4)))  GO  ALTER TABLE [Purchasing].[Purchase­Order­Header] ADD CONSTRAINT [CK\_­Purchase­Order­Header\_­Sub­Total] CHECK (([Sub­Total]>=(0.00)))  GO  ALTER TABLE [Purchasing].[Purchase­Order­Header] ADD CONSTRAINT [CK\_­Purchase­Order­Header\_­Tax­Amt] CHECK (([Tax­Amt]>=(0.00)))  GO  ALTER TABLE [Purchasing].[Purchase­Order­Header] ADD CONSTRAINT [PK\_­Purchase­Order­Header\_­Purchase­Order­ID] PRIMARY KEY CLUSTERED ([Purchase­Order­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Purchase­Order­Header\_­Employee­ID] ON [Purchasing].[Purchase­Order­Header] ([Employee­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Purchase­Order­Header\_­Vendor­ID] ON [Purchasing].[Purchase­Order­Header] ([Vendor­ID]) ON [PRIMARY]  GO  ALTER TABLE [Purchasing].[Purchase­Order­Header] ADD CONSTRAINT [FK\_­Purchase­Order­Header\_­Employee\_­Employee­ID] FOREIGN KEY ([Employee­ID]) REFERENCES [Human­Resources].[Employee] ([Business­Entity­ID])  GO  ALTER TABLE [Purchasing].[Purchase­Order­Header] ADD CONSTRAINT [FK\_­Purchase­Order­Header\_­Ship­Method\_­Ship­Method­ID] FOREIGN KEY ([Ship­Method­ID]) REFERENCES [Purchasing].[Ship­Method] ([Ship­Method­ID])  GO  ALTER TABLE [Purchasing].[Purchase­Order­Header] ADD CONSTRAINT [FK\_­Purchase­Order­Header\_­Vendor\_­Vendor­ID] FOREIGN KEY ([Vendor­ID]) REFERENCES [Purchasing].[Vendor] ([Business­Entity­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'General purchase order information. See Purchase­Order­Detail.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Employee who created the purchase order. Foreign key to Employee.Business­Entity­ID.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'COLUMN', N'Employee­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Shipping cost.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'COLUMN', N'Freight'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Purchase order creation date.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'COLUMN', N'Order­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'COLUMN', N'Purchase­Order­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Incremental number to track changes to the purchase order over time.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'COLUMN', N'Revision­Number'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Estimated shipment date from the vendor.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'COLUMN', N'Ship­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Shipping method. Foreign key to Ship­Method.Ship­Method­ID.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'COLUMN', N'Ship­Method­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Order current status. 1 = Pending; 2 = Approved; 3 = Rejected; 4 = Complete', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'COLUMN', N'Status'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Purchase order subtotal. Computed as SUM(Purchase­Order­Detail.Line­Total)for the appropriate Purchase­Order­ID.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'COLUMN', N'Sub­Total'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Tax amount.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'COLUMN', N'Tax­Amt'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Total due to vendor. Computed as Subtotal + Tax­Amt + Freight.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'COLUMN', N'Total­Due'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Vendor with whom the purchase order is placed. Foreign key to Vendor.Business­Entity­ID.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'COLUMN', N'Vendor­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Freight] >= (0.00)', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'CONSTRAINT', N'CK\_­Purchase­Order­Header\_­Freight'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Ship­Date] >= [Order­Date] OR [Ship­Date] IS NULL', 'SCHEMA', N'Purchasing', 'TABLE', N'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\_­Purchase­Order­Header\_­Tax­Amt'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0.0', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'CONSTRAINT', N'DF\_­Purchase­Order­Header\_­Freight'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'CONSTRAINT', N'DF\_­Purchase­Order­Header\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'CONSTRAINT', N'DF\_­Purchase­Order­Header\_­Order­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'CONSTRAINT', N'DF\_­Purchase­Order­Header\_­Revision­Number'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 1', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'CONSTRAINT', N'DF\_­Purchase­Order­Header\_­Status'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0.0', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'CONSTRAINT', N'DF\_­Purchase­Order­Header\_­Sub­Total'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0.0', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'CONSTRAINT', N'DF\_­Purchase­Order­Header\_­Tax­Amt'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Employee.Employee­ID.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'CONSTRAINT', N'FK\_­Purchase­Order­Header\_­Employee\_­Employee­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Ship­Method.Ship­Method­ID.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'CONSTRAINT', N'FK\_­Purchase­Order­Header\_­Ship­Method\_­Ship­Method­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Vendor.Vendor­ID.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'CONSTRAINT', N'FK\_­Purchase­Order­Header\_­Vendor\_­Vendor­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'CONSTRAINT', N'PK\_­Purchase­Order­Header\_­Purchase­Order­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'INDEX', N'IX\_­Purchase­Order­Header\_­Employee­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'INDEX', N'IX\_­Purchase­Order­Header\_­Vendor­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'INDEX', N'PK\_­Purchase­Order­Header\_­Purchase­Order­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'AFTER UPDATE trigger that updates the Revision­Number and Modified­Date columns in the Purchase­Order­Header table.', 'SCHEMA', N'Purchasing', 'TABLE', N'Purchase­Order­Header', 'TRIGGER', N'u­Purchase­Order­Header'  GO |

Uses

[[Human­Resources].[Employee]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=)

[[Purchasing].[Ship­Method]](#Xt5ka3ip5ljwxYo+n5TjCRZypoY=)

[[Purchasing].[Vendor]](#yfc9dLywcQtxtbztLjV6J3bb0tM=)

[Purchasing](#r48uLAuwsJTIAvAuIPBTbVrjT9s=)

Used By

[[Purchasing].[Purchase­Order­Detail]](#dsxg+I54iAv4FByATvQGW4l6Cco=)

|  |
| --- |
| [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 |
|  | Ship­Method­ID Primary key for Ship­Method records. | int | 4 | NOT NULL | 1 - 1 |  |
|  | Name Shipping company name. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NOT NULL |  |  |
|  | Ship­Base Minimum shipping charge. | money | 8 | NOT NULL |  | ((0.00)) |
|  | Ship­Rate Shipping charge per pound. | 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\_­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)) |

SQL Script

|  |
| --- |
| CREATE TABLE [Purchasing].[Ship­Method]  (  [Ship­Method­ID] [int] NOT NULL IDENTITY(1, 1),  [Name] [dbo].[Name] NOT NULL,  [Ship­Base] [money] NOT NULL CONSTRAINT [DF\_­Ship­Method\_­Ship­Base] DEFAULT ((0.00)),  [Ship­Rate] [money] NOT NULL CONSTRAINT [DF\_­Ship­Method\_­Ship­Rate] DEFAULT ((0.00)),  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Ship­Method\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Ship­Method\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Purchasing].[Ship­Method] ADD CONSTRAINT [CK\_­Ship­Method\_­Ship­Base] CHECK (([Ship­Base]>(0.00)))  GO  ALTER TABLE [Purchasing].[Ship­Method] ADD CONSTRAINT [CK\_­Ship­Method\_­Ship­Rate] CHECK (([Ship­Rate]>(0.00)))  GO  ALTER TABLE [Purchasing].[Ship­Method] ADD CONSTRAINT [PK\_­Ship­Method\_­Ship­Method­ID] PRIMARY KEY CLUSTERED ([Ship­Method­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Ship­Method\_­Name] ON [Purchasing].[Ship­Method] ([Name]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Ship­Method\_rowguid] ON [Purchasing].[Ship­Method] ([rowguid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Shipping company lookup table.', 'SCHEMA', N'Purchasing', 'TABLE', N'Ship­Method', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Purchasing', 'TABLE', N'Ship­Method', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Shipping company name.', 'SCHEMA', N'Purchasing', 'TABLE', N'Ship­Method', 'COLUMN', N'Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Purchasing', 'TABLE', N'Ship­Method', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Minimum shipping charge.', 'SCHEMA', N'Purchasing', 'TABLE', N'Ship­Method', 'COLUMN', N'Ship­Base'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Ship­Method records.', 'SCHEMA', N'Purchasing', 'TABLE', N'Ship­Method', 'COLUMN', N'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\_­Ship­Base'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Ship­Rate] > (0.00)', 'SCHEMA', N'Purchasing', 'TABLE', N'Ship­Method', 'CONSTRAINT', N'CK\_­Ship­Method\_­Ship­Rate'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Purchasing', 'TABLE', N'Ship­Method', 'CONSTRAINT', N'DF\_­Ship­Method\_­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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0.0', 'SCHEMA', N'Purchasing', 'TABLE', N'Ship­Method', 'CONSTRAINT', N'DF\_­Ship­Method\_­Ship­Base'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0.0', 'SCHEMA', N'Purchasing', 'TABLE', N'Ship­Method', 'CONSTRAINT', N'DF\_­Ship­Method\_­Ship­Rate'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Purchasing', 'TABLE', N'Ship­Method', 'CONSTRAINT', N'PK\_­Ship­Method\_­Ship­Method­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Purchasing', 'TABLE', N'Ship­Method', 'INDEX', N'AK\_­Ship­Method\_­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Purchasing', 'TABLE', N'Ship­Method', 'INDEX', N'AK\_­Ship­Method\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Purchasing', 'TABLE', N'Ship­Method', 'INDEX', N'PK\_­Ship­Method\_­Ship­Method­ID'  GO |

Uses

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Purchasing](#r48uLAuwsJTIAvAuIPBTbVrjT9s=)

Used By

[[Purchasing].[Purchase­Order­Header]](#wVItRHlooKKFqOhuItRUJuA60Fs=)

[[Sales].[Sales­Order­Header]](#J0oqEfXKE3WFfJ7PoPfs1QUbO+c=)

|  |
| --- |
| [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) | Nullability | Default |
|  | Business­Entity­ID Primary key for Vendor records. Foreign key to Business­Entity.Business­Entity­ID | int | 4 | NOT NULL |  |
|  | Account­Number Vendor account (identification) number. | [[dbo].[Account­Number]](#wfUPMi1TrfOhNsrHkC5PC5H/4RE=) | 30 | NOT NULL |  |
|  | Name Company name. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 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]](#j88jjDtwCbK8khz/m8jxryH/SFk=) | 1 | NOT NULL | ((1)) |
|  | Active­Flag 0 = Vendor no longer used. 1 = Vendor is actively used. | [[dbo].[Flag]](#j88jjDtwCbK8khz/m8jxryH/SFk=) | 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\_­Vendor\_­Business­Entity­ID Primary key (clustered) constraint | Business­Entity­ID | True |
|  | AK\_­Vendor\_­Account­Number Unique nonclustered index. | Account­Number | True |

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 [Credit­Rating]<=(5)) |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Vendor\_­Business­Entity\_­Business­Entity­ID Foreign key constraint referencing Business­Entity.Business­Entity­ID | Business­Entity­ID->[[Person].[Business­Entity].[Business­Entity­ID]](#I7t6VoTbrv7x5ZQ/kE/EZs1npdg=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Purchasing].[Vendor]  (  [Business­Entity­ID] [int] NOT NULL,  [Account­Number] [dbo].[Account­Number] NOT NULL,  [Name] [dbo].[Name] NOT NULL,  [Credit­Rating] [tinyint] NOT NULL,  [Preferred­Vendor­Status] [dbo].[Flag] NOT NULL CONSTRAINT [DF\_­Vendor\_­Preferred­Vendor­Status] DEFAULT ((1)),  [Active­Flag] [dbo].[Flag] NOT NULL CONSTRAINT [DF\_­Vendor\_­Active­Flag] DEFAULT ((1)),  [Purchasing­Web­Service­URL] [nvarchar] (1024) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Vendor\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  CREATE TRIGGER [Purchasing].[d­Vendor] ON [Purchasing].[Vendor]  INSTEAD OF DELETE NOT FOR REPLICATION AS  BEGIN  DECLARE @Count int;  SET @Count = @@ROWCOUNT;  IF @Count = 0  RETURN;  SET NOCOUNT ON;  BEGIN TRY  DECLARE @Delete­Count int;  SELECT @Delete­Count = COUNT(\*) FROM deleted;  IF @Delete­Count > 0  BEGIN  RAISERROR  (N'Vendors cannot be deleted. They can only be marked as not active.', -- Message  10, -- Severity.  1); -- State.  -- Rollback any active or uncommittable transactions  IF @@TRANCOUNT > 0  BEGIN  ROLLBACK TRANSACTION;  END  END;  END TRY  BEGIN CATCH  EXECUTE [dbo].[usp­Print­Error];  -- Rollback any active or uncommittable transactions before  -- inserting information in the Error­Log  IF @@TRANCOUNT > 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE [dbo].[usp­Log­Error];  END CATCH;  END;  GO  ALTER TABLE [Purchasing].[Vendor] ADD CONSTRAINT [CK\_­Vendor\_­Credit­Rating] CHECK (([Credit­Rating]>=(1) AND [Credit­Rating]<=(5)))  GO  ALTER TABLE [Purchasing].[Vendor] ADD CONSTRAINT [PK\_­Vendor\_­Business­Entity­ID] PRIMARY KEY CLUSTERED ([Business­Entity­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Vendor\_­Account­Number] ON [Purchasing].[Vendor] ([Account­Number]) ON [PRIMARY]  GO  ALTER TABLE [Purchasing].[Vendor] ADD CONSTRAINT [FK\_­Vendor\_­Business­Entity\_­Business­Entity­ID] FOREIGN KEY ([Business­Entity­ID]) REFERENCES [Person].[Business­Entity] ([Business­Entity­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Companies from whom Adventure Works Cycles purchases parts or other goods.', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Vendor account (identification) number.', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'COLUMN', N'Account­Number'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = Vendor no longer used. 1 = Vendor is actively used.', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'COLUMN', N'Active­Flag'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Vendor records. Foreign key to Business­Entity.Business­Entity­ID', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'COLUMN', N'Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'1 = Superior, 2 = Excellent, 3 = Above average, 4 = Average, 5 = Below average', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'COLUMN', N'Credit­Rating'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Company name.', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'COLUMN', N'Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = Do not use if another vendor is available. 1 = Preferred over other vendors supplying the same product.', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'COLUMN', N'Preferred­Vendor­Status'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Vendor URL.', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'COLUMN', N'Purchasing­Web­Service­URL'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Credit­Rating] BETWEEN (1) AND (5)', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'CONSTRAINT', N'CK\_­Vendor\_­Credit­Rating'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 1 (TRUE)', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'CONSTRAINT', N'DF\_­Vendor\_­Active­Flag'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'CONSTRAINT', N'DF\_­Vendor\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 1 (TRUE)', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'CONSTRAINT', N'DF\_­Vendor\_­Preferred­Vendor­Status'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Business­Entity.Business­Entity­ID', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'CONSTRAINT', N'FK\_­Vendor\_­Business­Entity\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'CONSTRAINT', N'PK\_­Vendor\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'INDEX', N'AK\_­Vendor\_­Account­Number'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'INDEX', N'PK\_­Vendor\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'INSTEAD OF DELETE trigger which keeps Vendors from being deleted.', 'SCHEMA', N'Purchasing', 'TABLE', N'Vendor', 'TRIGGER', N'd­Vendor'  GO |

Uses

[[Person].[Business­Entity]](#I7t6VoTbrv7x5ZQ/kE/EZs1npdg=)

[[dbo].[Account­Number]](#wfUPMi1TrfOhNsrHkC5PC5H/4RE=)

[[dbo].[Flag]](#j88jjDtwCbK8khz/m8jxryH/SFk=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Purchasing](#r48uLAuwsJTIAvAuIPBTbVrjT9s=)

Used By

[[Purchasing].[Product­Vendor]](#igWcZ2fEqwA+AlsF0xQn66oTGAU=)

[[Purchasing].[Purchase­Order­Header]](#wVItRHlooKKFqOhuItRUJuA60Fs=)

[[Purchasing].[v­Vendor­With­Addresses]](#NdlQ62Q6DZw3GS30J9H9dm/SOqg=)

[[Purchasing].[v­Vendor­With­Contacts]](#kSLkxLBh/Shb6NobziJenpOvdFo=)

[[dbo].[ufn­Get­Contact­Information]](#MAbUSn/1caUuJ7XN6dczcbTRRyA=)

|  |
| --- |
| [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 |
|  | Country­Region­Code ISO code for countries and regions. Foreign key to Country­Region.Country­Region­Code. | nvarchar(3) | 6 | NOT NULL |  |
|  | 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 |
|  | PK\_­Country­Region­Currency\_­Country­Region­Code\_­Currency­Code Primary key (clustered) constraint | Country­Region­Code, Currency­Code | True |
|  | IX\_­Country­Region­Currency\_­Currency­Code Nonclustered index. | Currency­Code |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Country­Region­Currency\_­Country­Region\_­Country­Region­Code Foreign key constraint referencing Country­Region.Country­Region­Code. | Country­Region­Code->[[Person].[Country­Region].[Country­Region­Code]](#efOGmVlBeofSPEY2jc0F91joV9Q=) |
| FK\_­Country­Region­Currency\_­Currency\_­Currency­Code Foreign key constraint referencing Currency.Currency­Code. | Currency­Code->[[Sales].[Currency].[Currency­Code]](#FEyILkTLRQRZ4zkAPe+AjFF8vg8=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Sales].[Country­Region­Currency]  (  [Country­Region­Code] [nvarchar] (3) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Currency­Code] [nchar] (3) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Country­Region­Currency\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Country­Region­Currency] ADD CONSTRAINT [PK\_­Country­Region­Currency\_­Country­Region­Code\_­Currency­Code] PRIMARY KEY CLUSTERED ([Country­Region­Code], [Currency­Code]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Country­Region­Currency\_­Currency­Code] ON [Sales].[Country­Region­Currency] ([Currency­Code]) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Country­Region­Currency] ADD CONSTRAINT [FK\_­Country­Region­Currency\_­Country­Region\_­Country­Region­Code] FOREIGN KEY ([Country­Region­Code]) REFERENCES [Person].[Country­Region] ([Country­Region­Code])  GO  ALTER TABLE [Sales].[Country­Region­Currency] ADD CONSTRAINT [FK\_­Country­Region­Currency\_­Currency\_­Currency­Code] FOREIGN KEY ([Currency­Code]) REFERENCES [Sales].[Currency] ([Currency­Code])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Cross-reference table mapping ISO currency codes to a country or region.', 'SCHEMA', N'Sales', 'TABLE', N'Country­Region­Currency', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ISO code for countries and regions. Foreign key to Country­Region.Country­Region­Code.', 'SCHEMA', N'Sales', 'TABLE', N'Country­Region­Currency', 'COLUMN', N'Country­Region­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ISO standard currency code. Foreign key to Currency.Currency­Code.', 'SCHEMA', N'Sales', 'TABLE', N'Country­Region­Currency', 'COLUMN', N'Currency­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Sales', 'TABLE', N'Country­Region­Currency', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Country­Region­Currency', 'CONSTRAINT', N'DF\_­Country­Region­Currency\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Country­Region.Country­Region­Code.', 'SCHEMA', N'Sales', 'TABLE', N'Country­Region­Currency', 'CONSTRAINT', N'FK\_­Country­Region­Currency\_­Country­Region\_­Country­Region­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Currency.Currency­Code.', 'SCHEMA', N'Sales', 'TABLE', N'Country­Region­Currency', 'CONSTRAINT', N'FK\_­Country­Region­Currency\_­Currency\_­Currency­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Sales', 'TABLE', N'Country­Region­Currency', 'CONSTRAINT', N'PK\_­Country­Region­Currency\_­Country­Region­Code\_­Currency­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Sales', 'TABLE', N'Country­Region­Currency', 'INDEX', N'IX\_­Country­Region­Currency\_­Currency­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'Country­Region­Currency', 'INDEX', N'PK\_­Country­Region­Currency\_­Country­Region­Code\_­Currency­Code'  GO |

Uses

[[Person].[Country­Region]](#efOGmVlBeofSPEY2jc0F91joV9Q=)

[[Sales].[Currency]](#FEyILkTLRQRZ4zkAPe+AjFF8vg8=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

|  |
| --- |
| [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 |
|  | 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

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Credit­Card\_­Credit­Card­ID Primary key (clustered) constraint | Credit­Card­ID | True |
|  | AK\_­Credit­Card\_­Card­Number Unique nonclustered index. | Card­Number | True |

SQL Script

|  |
| --- |
| CREATE TABLE [Sales].[Credit­Card]  (  [Credit­Card­ID] [int] NOT NULL IDENTITY(1, 1),  [Card­Type] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Card­Number] [nvarchar] (25) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Exp­Month] [tinyint] NOT NULL,  [Exp­Year] [smallint] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Credit­Card\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Credit­Card] ADD CONSTRAINT [PK\_­Credit­Card\_­Credit­Card­ID] PRIMARY KEY CLUSTERED ([Credit­Card­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Credit­Card\_­Card­Number] ON [Sales].[Credit­Card] ([Card­Number]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Customer credit card information.', 'SCHEMA', N'Sales', 'TABLE', N'Credit­Card', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Credit card number.', 'SCHEMA', N'Sales', 'TABLE', N'Credit­Card', 'COLUMN', N'Card­Number'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Credit card name.', 'SCHEMA', N'Sales', 'TABLE', N'Credit­Card', 'COLUMN', N'Card­Type'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Credit­Card records.', 'SCHEMA', N'Sales', 'TABLE', N'Credit­Card', 'COLUMN', N'Credit­Card­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Credit card expiration month.', 'SCHEMA', N'Sales', 'TABLE', N'Credit­Card', 'COLUMN', N'Exp­Month'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Credit card expiration year.', 'SCHEMA', N'Sales', 'TABLE', N'Credit­Card', 'COLUMN', N'Exp­Year'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Sales', 'TABLE', N'Credit­Card', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Credit­Card', 'CONSTRAINT', N'DF\_­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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Sales', 'TABLE', N'Credit­Card', 'INDEX', N'AK\_­Credit­Card\_­Card­Number'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'Credit­Card', 'INDEX', N'PK\_­Credit­Card\_­Credit­Card­ID'  GO |

Uses

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

Used By

[[Sales].[Person­Credit­Card]](#HOEWmD+eQ6y/+8bq8hkiLEiBBjU=)

[[Sales].[Sales­Order­Header]](#J0oqEfXKE3WFfJ7PoPfs1QUbO+c=)

|  |
| --- |
| [Sales].[Currency] |

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 |
|  | Currency­Code The ISO code for the Currency. | nchar(3) | 6 | NOT NULL |  |
|  | Name Currency name. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NOT NULL |  |
|  | Modified­Date Date and time the record was last updated. | datetime | 8 | NOT NULL | (getdate()) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Currency\_­Currency­Code Primary key (clustered) constraint | Currency­Code | True |
|  | AK\_­Currency\_­Name Unique nonclustered index. | Name | True |

SQL Script

|  |
| --- |
| CREATE TABLE [Sales].[Currency]  (  [Currency­Code] [nchar] (3) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Name] [dbo].[Name] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Currency\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Currency] ADD CONSTRAINT [PK\_­Currency\_­Currency­Code] PRIMARY KEY CLUSTERED ([Currency­Code]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Currency\_­Name] ON [Sales].[Currency] ([Name]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Lookup table containing standard ISO currencies.', 'SCHEMA', N'Sales', 'TABLE', N'Currency', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The ISO code for the Currency.', 'SCHEMA', N'Sales', 'TABLE', N'Currency', 'COLUMN', N'Currency­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Sales', 'TABLE', N'Currency', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Currency name.', 'SCHEMA', N'Sales', 'TABLE', N'Currency', 'COLUMN', N'Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Currency', 'CONSTRAINT', N'DF\_­Currency\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Sales', 'TABLE', N'Currency', 'CONSTRAINT', N'PK\_­Currency\_­Currency­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Sales', 'TABLE', N'Currency', 'INDEX', N'AK\_­Currency\_­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'Currency', 'INDEX', N'PK\_­Currency\_­Currency­Code'  GO |

Uses

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

Used By

[[Sales].[Country­Region­Currency]](#ZTDvJcpGJmjH9tZPEMXFxv/sN0o=)

[[Sales].[Currency­Rate]](#DbnMf0tD82NB+BGbYOX9jWqScts=)

|  |
| --- |
| [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 |
|  | 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 |  |  |
|  | From­Currency­Code Exchange rate was converted from this currency code. | nchar(3) | 6 | NOT NULL |  |  |
|  | 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\_­Currency­Rate\_­Currency­Rate­ID Primary key (clustered) constraint | Currency­Rate­ID | True |
|  | AK\_­Currency­Rate\_­Currency­Rate­Date\_­From­Currency­Code\_­To­Currency­Code Unique nonclustered index. | Currency­Rate­Date, From­Currency­Code, To­Currency­Code | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Currency­Rate\_­Currency\_­From­Currency­Code Foreign key constraint referencing Currency.From­Currency­Code. | From­Currency­Code->[[Sales].[Currency].[Currency­Code]](#FEyILkTLRQRZ4zkAPe+AjFF8vg8=) |
| FK\_­Currency­Rate\_­Currency\_­To­Currency­Code Foreign key constraint referencing Currency.To­Currency­Code. | To­Currency­Code->[[Sales].[Currency].[Currency­Code]](#FEyILkTLRQRZ4zkAPe+AjFF8vg8=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Sales].[Currency­Rate]  (  [Currency­Rate­ID] [int] NOT NULL IDENTITY(1, 1),  [Currency­Rate­Date] [datetime] NOT NULL,  [From­Currency­Code] [nchar] (3) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [To­Currency­Code] [nchar] (3) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Average­Rate] [money] NOT NULL,  [End­Of­Day­Rate] [money] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Currency­Rate\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Currency­Rate] ADD CONSTRAINT [PK\_­Currency­Rate\_­Currency­Rate­ID] PRIMARY KEY CLUSTERED ([Currency­Rate­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Currency­Rate\_­Currency­Rate­Date\_­From­Currency­Code\_­To­Currency­Code] ON [Sales].[Currency­Rate] ([Currency­Rate­Date], [From­Currency­Code], [To­Currency­Code]) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Currency­Rate] ADD CONSTRAINT [FK\_­Currency­Rate\_­Currency\_­From­Currency­Code] FOREIGN KEY ([From­Currency­Code]) REFERENCES [Sales].[Currency] ([Currency­Code])  GO  ALTER TABLE [Sales].[Currency­Rate] ADD CONSTRAINT [FK\_­Currency­Rate\_­Currency\_­To­Currency­Code] FOREIGN KEY ([To­Currency­Code]) REFERENCES [Sales].[Currency] ([Currency­Code])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Currency exchange rates.', 'SCHEMA', N'Sales', 'TABLE', N'Currency­Rate', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Average exchange rate for the day.', 'SCHEMA', N'Sales', 'TABLE', N'Currency­Rate', 'COLUMN', N'Average­Rate'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the exchange rate was obtained.', 'SCHEMA', N'Sales', 'TABLE', N'Currency­Rate', 'COLUMN', N'Currency­Rate­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Currency­Rate records.', 'SCHEMA', N'Sales', 'TABLE', N'Currency­Rate', 'COLUMN', N'Currency­Rate­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Final exchange rate for the day.', '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'From­Currency­Code'  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'To­Currency­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Currency­Rate', 'CONSTRAINT', N'DF\_­Currency­Rate\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Currency.From­Currency­Code.', 'SCHEMA', N'Sales', 'TABLE', N'Currency­Rate', 'CONSTRAINT', N'FK\_­Currency­Rate\_­Currency\_­From­Currency­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Currency.To­Currency­Code.', 'SCHEMA', N'Sales', 'TABLE', N'Currency­Rate', 'CONSTRAINT', N'FK\_­Currency­Rate\_­Currency\_­To­Currency­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Sales', 'TABLE', N'Currency­Rate', 'CONSTRAINT', N'PK\_­Currency­Rate\_­Currency­Rate­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Sales', 'TABLE', N'Currency­Rate', 'INDEX', N'AK\_­Currency­Rate\_­Currency­Rate­Date\_­From­Currency­Code\_­To­Currency­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'Currency­Rate', 'INDEX', N'PK\_­Currency­Rate\_­Currency­Rate­ID'  GO |

Uses

[[Sales].[Currency]](#FEyILkTLRQRZ4zkAPe+AjFF8vg8=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

Used By

[[Sales].[Sales­Order­Header]](#J0oqEfXKE3WFfJ7PoPfs1QUbO+c=)

|  |
| --- |
| [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 | Computed | Max Length (Bytes) | Nullability | Identity | Identity Replication | Default |
|  | Customer­ID Primary key. | int |  | 4 | NOT NULL | 1 - 1 | False |  |
|  | Person­ID Foreign key to Person.Business­Entity­ID | int |  | 4 | NULL allowed |  |  |  |
|  | Store­ID Foreign key to Store.Business­Entity­ID | int |  | 4 | NULL allowed |  |  |  |
|  | Territory­ID ID of the territory in which the customer is located. Foreign key to Sales­Territory.Sales­Territory­ID. | int |  | 4 | NULL allowed |  |  |  |
|  | 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 |  |  | (getdate()) |

Computed columns

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

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | 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].[Person].[Business­Entity­ID]](#TZe4JItyjmK+s/tUWju0b9W3HXU=) |
| FK\_­Customer\_­Sales­Territory\_­Territory­ID Foreign key constraint referencing Sales­Territory.Territory­ID. | Territory­ID->[[Sales].[Sales­Territory].[Territory­ID]](#hvNbyL+2AhjcTkzEdMCOV3mg8XQ=) |
| FK\_­Customer\_­Store\_­Store­ID Foreign key constraint referencing Store.Business­Entity­ID. | Store­ID->[[Sales].[Store].[Business­Entity­ID]](#o4yE5vuw4wTVa38E7sxNcXx3KhA=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Sales].[Customer]  (  [Customer­ID] [int] NOT NULL IDENTITY(1, 1) NOT FOR REPLICATION,  [Person­ID] [int] NULL,  [Store­ID] [int] NULL,  [Territory­ID] [int] NULL,  [Account­Number] AS (isnull('AW'+[dbo].[ufn­Leading­Zeros]([Customer­ID]),'')),  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Customer\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Customer\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Customer] ADD CONSTRAINT [PK\_­Customer\_­Customer­ID] PRIMARY KEY CLUSTERED ([Customer­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Customer\_­Account­Number] ON [Sales].[Customer] ([Account­Number]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Customer\_rowguid] ON [Sales].[Customer] ([rowguid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Customer\_­Territory­ID] ON [Sales].[Customer] ([Territory­ID]) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Customer] ADD CONSTRAINT [FK\_­Customer\_­Person\_­Person­ID] FOREIGN KEY ([Person­ID]) REFERENCES [Person].[Person] ([Business­Entity­ID])  GO  ALTER TABLE [Sales].[Customer] ADD CONSTRAINT [FK\_­Customer\_­Sales­Territory\_­Territory­ID] FOREIGN KEY ([Territory­ID]) REFERENCES [Sales].[Sales­Territory] ([Territory­ID])  GO  ALTER TABLE [Sales].[Customer] ADD CONSTRAINT [FK\_­Customer\_­Store\_­Store­ID] FOREIGN KEY ([Store­ID]) REFERENCES [Sales].[Store] ([Business­Entity­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Current customer information. Also see the Person and Store tables.', 'SCHEMA', N'Sales', 'TABLE', N'Customer', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique number identifying the customer assigned by the accounting system.', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'COLUMN', N'Account­Number'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key.', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'COLUMN', N'Customer­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key to Person.Business­Entity­ID', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'COLUMN', N'Person­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key to Store.Business­Entity­ID', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'COLUMN', N'Store­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ID of the territory in which the customer is located. Foreign key to Sales­Territory.Sales­Territory­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'COLUMN', N'Territory­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'CONSTRAINT', N'DF\_­Customer\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'CONSTRAINT', N'DF\_­Customer\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Person.Business­Entity­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'CONSTRAINT', N'FK\_­Customer\_­Person\_­Person­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Sales­Territory.Territory­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'CONSTRAINT', N'FK\_­Customer\_­Sales­Territory\_­Territory­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Store.Business­Entity­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'CONSTRAINT', N'FK\_­Customer\_­Store\_­Store­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'CONSTRAINT', N'PK\_­Customer\_­Customer­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'INDEX', N'AK\_­Customer\_­Account­Number'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'INDEX', N'AK\_­Customer\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'INDEX', N'IX\_­Customer\_­Territory­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'Customer', 'INDEX', N'PK\_­Customer\_­Customer­ID'  GO |

Uses

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[[Sales].[Sales­Territory]](#hvNbyL+2AhjcTkzEdMCOV3mg8XQ=)

[[Sales].[Store]](#o4yE5vuw4wTVa38E7sxNcXx3KhA=)

[[dbo].[ufn­Leading­Zeros]](#50oMRfjOJe+9VZTmAIx0V5kh0ds=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

Used By

[[Sales].[Sales­Order­Header]](#J0oqEfXKE3WFfJ7PoPfs1QUbO+c=)

[[Sales].[v­Individual­Customer]](#WcBS54CxROpNMbufsyU5V6t9Ykw=)

[[dbo].[ufn­Get­Contact­Information]](#MAbUSn/1caUuJ7XN6dczcbTRRyA=)

|  |
| --- |
| [Sales].[Person­Credit­Card] |

MS\_­Description

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

Properties

|  |  |
| --- | --- |
| Property | Value |
| Row Count (~) | 19118 |
| Created | 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 |
|  | Business­Entity­ID Business entity identification number. Foreign key to Person.Business­Entity­ID. | int | 4 | NOT NULL |  |
|  | 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\_­Person­Credit­Card\_­Business­Entity­ID\_­Credit­Card­ID Primary key (clustered) constraint | Business­Entity­ID, Credit­Card­ID | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Person­Credit­Card\_­Credit­Card\_­Credit­Card­ID Foreign key constraint referencing Credit­Card.Credit­Card­ID. | Credit­Card­ID->[[Sales].[Credit­Card].[Credit­Card­ID]](#Juoic8jJ3a0e1KCG+cj8LixVRbQ=) |
| FK\_­Person­Credit­Card\_­Person\_­Business­Entity­ID Foreign key constraint referencing Person.Business­Entity­ID. | Business­Entity­ID->[[Person].[Person].[Business­Entity­ID]](#TZe4JItyjmK+s/tUWju0b9W3HXU=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Sales].[Person­Credit­Card]  (  [Business­Entity­ID] [int] NOT NULL,  [Credit­Card­ID] [int] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Person­Credit­Card\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Person­Credit­Card] ADD CONSTRAINT [PK\_­Person­Credit­Card\_­Business­Entity­ID\_­Credit­Card­ID] PRIMARY KEY CLUSTERED ([Business­Entity­ID], [Credit­Card­ID]) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Person­Credit­Card] ADD CONSTRAINT [FK\_­Person­Credit­Card\_­Credit­Card\_­Credit­Card­ID] FOREIGN KEY ([Credit­Card­ID]) REFERENCES [Sales].[Credit­Card] ([Credit­Card­ID])  GO  ALTER TABLE [Sales].[Person­Credit­Card] ADD CONSTRAINT [FK\_­Person­Credit­Card\_­Person\_­Business­Entity­ID] FOREIGN KEY ([Business­Entity­ID]) REFERENCES [Person].[Person] ([Business­Entity­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Cross-reference table mapping people to their credit card information in the Credit­Card table. ', 'SCHEMA', N'Sales', 'TABLE', N'Person­Credit­Card', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Business entity identification number. Foreign key to Person.Business­Entity­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Person­Credit­Card', 'COLUMN', N'Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Credit card identification number. Foreign key to Credit­Card.Credit­Card­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Person­Credit­Card', 'COLUMN', N'Credit­Card­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Sales', 'TABLE', N'Person­Credit­Card', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Person­Credit­Card', 'CONSTRAINT', N'DF\_­Person­Credit­Card\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Credit­Card.Credit­Card­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Person­Credit­Card', 'CONSTRAINT', N'FK\_­Person­Credit­Card\_­Credit­Card\_­Credit­Card­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Person.Business­Entity­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Person­Credit­Card', 'CONSTRAINT', N'FK\_­Person­Credit­Card\_­Person\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Sales', 'TABLE', N'Person­Credit­Card', 'CONSTRAINT', N'PK\_­Person­Credit­Card\_­Business­Entity­ID\_­Credit­Card­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'Person­Credit­Card', 'INDEX', N'PK\_­Person­Credit­Card\_­Business­Entity­ID\_­Credit­Card­ID'  GO |

Uses

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[[Sales].[Credit­Card]](#Juoic8jJ3a0e1KCG+cj8LixVRbQ=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

|  |
| --- |
| [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 | Computed | Max Length (Bytes) | Nullability | Identity | Default |
|  | Sales­Order­ID Primary key. Foreign key to Sales­Order­Header.Sales­Order­ID. | int |  | 4 | NOT NULL |  |  |
|  | 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 |  |  |
|  | Order­Qty Quantity ordered per product. | smallint |  | 2 | NOT NULL |  |  |
|  | Product­ID Product sold to customer. Foreign key to Product.Product­ID. | int |  | 4 | NOT NULL |  |  |
|  | Special­Offer­ID Promotional code. Foreign key to Special­Offer.Special­Offer­ID. | int |  | 4 | NOT NULL |  |  |
|  | 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) \* Order­Qty. | numeric(38,6) | True | 17 | 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()) |

Computed columns

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

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Sales­Order­Detail\_­Sales­Order­ID\_­Sales­Order­Detail­ID Primary key (clustered) constraint | Sales­Order­ID, Sales­Order­Detail­ID | 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].[Sales­Order­Header].[Sales­Order­ID]](#J0oqEfXKE3WFfJ7PoPfs1QUbO+c=) |
| FK\_­Sales­Order­Detail\_­Special­Offer­Product\_­Special­Offer­IDProduct­ID Foreign key constraint referencing Special­Offer­Product.Special­Offer­IDProduct­ID. |  | Special­Offer­ID->[[Sales].[Special­Offer­Product].[Special­Offer­ID]](#8e4RMrihzXJLXw2gPdnNJ+2ijQ0=)  Product­ID->[[Sales].[Special­Offer­Product].[Product­ID]](#8e4RMrihzXJLXw2gPdnNJ+2ijQ0=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Sales].[Sales­Order­Detail]  (  [Sales­Order­ID] [int] NOT NULL,  [Sales­Order­Detail­ID] [int] NOT NULL IDENTITY(1, 1),  [Carrier­Tracking­Number] [nvarchar] (25) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Order­Qty] [smallint] NOT NULL,  [Product­ID] [int] NOT NULL,  [Special­Offer­ID] [int] NOT NULL,  [Unit­Price] [money] NOT NULL,  [Unit­Price­Discount] [money] NOT NULL CONSTRAINT [DF\_­Sales­Order­Detail\_­Unit­Price­Discount] DEFAULT ((0.0)),  [Line­Total] AS (isnull(([Unit­Price]\*((1.0)-[Unit­Price­Discount]))\*[Order­Qty],(0.0))),  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Sales­Order­Detail\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Sales­Order­Detail\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  CREATE TRIGGER [Sales].[idu­Sales­Order­Detail] ON [Sales].[Sales­Order­Detail]  AFTER INSERT, DELETE, UPDATE AS  BEGIN  DECLARE @Count int;  SET @Count = @@ROWCOUNT;  IF @Count = 0  RETURN;  SET NOCOUNT ON;  BEGIN TRY  -- If inserting or updating these columns  IF UPDATE([Product­ID]) OR UPDATE([Order­Qty]) OR UPDATE([Unit­Price]) OR UPDATE([Unit­Price­Discount])  -- Insert record into Transaction­History  BEGIN  INSERT INTO [Production].[Transaction­History]  ([Product­ID]  ,[Reference­Order­ID]  ,[Reference­Order­Line­ID]  ,[Transaction­Type]  ,[Transaction­Date]  ,[Quantity]  ,[Actual­Cost])  SELECT  inserted.[Product­ID]  ,inserted.[Sales­Order­ID]  ,inserted.[Sales­Order­Detail­ID]  ,'S'  ,GETDATE()  ,inserted.[Order­Qty]  ,inserted.[Unit­Price]  FROM inserted  INNER JOIN [Sales].[Sales­Order­Header]  ON inserted.[Sales­Order­ID] = [Sales].[Sales­Order­Header].[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].[Sales­Order­Detail].[Sales­Order­ID])  WHERE [Sales].[Sales­Order­Header].[Sales­Order­ID] IN (SELECT inserted.[Sales­Order­ID] FROM inserted);  UPDATE [Person].[Person]  SET [Demographics].modify('declare default element namespace  "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Individual­Survey";  replace value of (/Individual­Survey/Total­Purchase­YTD)[1]  with data(/Individual­Survey/Total­Purchase­YTD)[1] - sql:column("deleted.Line­Total")')  FROM deleted  INNER JOIN [Sales].[Sales­Order­Header]  ON deleted.[Sales­Order­ID] = [Sales].[Sales­Order­Header].[Sales­Order­ID]  INNER JOIN [Sales].[Customer]  ON [Sales].[Customer].[Customer­ID] = [Sales].[Sales­Order­Header].[Customer­ID]  WHERE [Sales].[Customer].[Person­ID] = [Person].[Person].[Business­Entity­ID];  END TRY  BEGIN CATCH  EXECUTE [dbo].[usp­Print­Error];  -- Rollback any active or uncommittable transactions before  -- inserting information in the Error­Log  IF @@TRANCOUNT > 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE [dbo].[usp­Log­Error];  END CATCH;  END;  GO  ALTER TABLE [Sales].[Sales­Order­Detail] ADD CONSTRAINT [CK\_­Sales­Order­Detail\_­Order­Qty] CHECK (([Order­Qty]>(0)))  GO  ALTER TABLE [Sales].[Sales­Order­Detail] ADD CONSTRAINT [CK\_­Sales­Order­Detail\_­Unit­Price] CHECK (([Unit­Price]>=(0.00)))  GO  ALTER TABLE [Sales].[Sales­Order­Detail] ADD CONSTRAINT [CK\_­Sales­Order­Detail\_­Unit­Price­Discount] CHECK (([Unit­Price­Discount]>=(0.00)))  GO  ALTER TABLE [Sales].[Sales­Order­Detail] ADD CONSTRAINT [PK\_­Sales­Order­Detail\_­Sales­Order­ID\_­Sales­Order­Detail­ID] PRIMARY KEY CLUSTERED ([Sales­Order­ID], [Sales­Order­Detail­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Sales­Order­Detail\_­Product­ID] ON [Sales].[Sales­Order­Detail] ([Product­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Sales­Order­Detail\_rowguid] ON [Sales].[Sales­Order­Detail] ([rowguid]) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Sales­Order­Detail] ADD CONSTRAINT [FK\_­Sales­Order­Detail\_­Sales­Order­Header\_­Sales­Order­ID] FOREIGN KEY ([Sales­Order­ID]) REFERENCES [Sales].[Sales­Order­Header] ([Sales­Order­ID]) ON DELETE CASCADE  GO  ALTER TABLE [Sales].[Sales­Order­Detail] ADD CONSTRAINT [FK\_­Sales­Order­Detail\_­Special­Offer­Product\_­Special­Offer­IDProduct­ID] FOREIGN KEY ([Special­Offer­ID], [Product­ID]) REFERENCES [Sales].[Special­Offer­Product] ([Special­Offer­ID], [Product­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Individual products associated with a specific sales order. See Sales­Order­Header.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Shipment tracking number supplied by the shipper.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'COLUMN', N'Carrier­Tracking­Number'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Per product subtotal. Computed as Unit­Price \* (1 - Unit­Price­Discount) \* Order­Qty.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'COLUMN', N'Line­Total'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Quantity ordered per product.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'COLUMN', N'Order­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product sold to customer. Foreign key to Product.Product­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'COLUMN', N'Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. One incremental unique number per product sold.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'COLUMN', N'Sales­Order­Detail­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Foreign key to Sales­Order­Header.Sales­Order­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'COLUMN', N'Sales­Order­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Promotional code. Foreign key to Special­Offer.Special­Offer­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'COLUMN', N'Special­Offer­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Selling price of a single product.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'COLUMN', N'Unit­Price'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Discount amount.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'COLUMN', N'Unit­Price­Discount'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Order­Qty] > (0)', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'CONSTRAINT', N'CK\_­Sales­Order­Detail\_­Order­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Unit­Price] >= (0.00)', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'CONSTRAINT', N'CK\_­Sales­Order­Detail\_­Unit­Price'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Unit­Price­Discount] >= (0.00)', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'CONSTRAINT', N'CK\_­Sales­Order­Detail\_­Unit­Price­Discount'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'CONSTRAINT', N'DF\_­Sales­Order­Detail\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'CONSTRAINT', N'DF\_­Sales­Order­Detail\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0.0', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'CONSTRAINT', N'DF\_­Sales­Order­Detail\_­Unit­Price­Discount'  GO  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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Special­Offer­Product.Special­Offer­IDProduct­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'CONSTRAINT', N'FK\_­Sales­Order­Detail\_­Special­Offer­Product\_­Special­Offer­IDProduct­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'CONSTRAINT', N'PK\_­Sales­Order­Detail\_­Sales­Order­ID\_­Sales­Order­Detail­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'INDEX', N'AK\_­Sales­Order­Detail\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'INDEX', N'IX\_­Sales­Order­Detail\_­Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'INDEX', N'PK\_­Sales­Order­Detail\_­Sales­Order­ID\_­Sales­Order­Detail­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'AFTER INSERT, DELETE, UPDATE trigger that inserts a row in the Transaction­History table, updates Modified­Date in Sales­Order­Detail and updates the Sales­Order­Header.Sub­Total column.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Detail', 'TRIGGER', N'idu­Sales­Order­Detail'  GO |

Uses

[[Sales].[Sales­Order­Header]](#J0oqEfXKE3WFfJ7PoPfs1QUbO+c=)

[[Sales].[Special­Offer­Product]](#8e4RMrihzXJLXw2gPdnNJ+2ijQ0=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

|  |
| --- |
| [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 | Computed | Max Length (Bytes) | Nullability | Identity | Identity Replication | Default |
|  | 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 |  |  | (getdate()) |
|  | 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 allowed |  |  |  |
|  | 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]](#j88jjDtwCbK8khz/m8jxryH/SFk=) |  | 1 | NOT NULL |  |  | ((1)) |
|  | Sales­Order­Number Unique sales order identification number. | nvarchar(25) | True | 50 | NOT NULL |  |  |  |
|  | Purchase­Order­Number Customer purchase order number reference. | [[dbo].[Order­Number]](#SZqX2/vCAFjEqiLsacel6rQr3G4=) |  | 50 | NULL allowed |  |  |  |
|  | Account­Number Financial accounting number reference. | [[dbo].[Account­Number]](#wfUPMi1TrfOhNsrHkC5PC5H/4RE=) |  | 30 | NULL allowed |  |  |  |
|  | Customer­ID Customer identification number. Foreign key to Customer.Business­Entity­ID. | int |  | 4 | NOT NULL |  |  |  |
|  | Sales­Person­ID Sales person who created the sales order. Foreign key to Sales­Person.Business­Entity­ID. | int |  | 4 | NULL allowed |  |  |  |
|  | Territory­ID Territory in which the sale was made. Foreign key to Sales­Territory.Sales­Territory­ID. | int |  | 4 | NULL allowed |  |  |  |
|  | Bill­To­Address­ID Customer billing address. Foreign key to Address.Address­ID. | int |  | 4 | NOT NULL |  |  |  |
|  | Ship­To­Address­ID Customer shipping address. Foreign key to Address.Address­ID. | int |  | 4 | NOT NULL |  |  |  |
|  | Ship­Method­ID Shipping method. Foreign key to Ship­Method.Ship­Method­ID. | int |  | 4 | NOT NULL |  |  |  |
|  | Credit­Card­ID Credit card identification number. Foreign key to Credit­Card.Credit­Card­ID. | int |  | 4 | NULL allowed |  |  |  |
|  | Credit­Card­Approval­Code Approval code provided by the credit card company. | varchar(15) |  | 15 | NULL allowed |  |  |  |
|  | Currency­Rate­ID Currency exchange rate used. Foreign key to Currency­Rate.Currency­Rate­ID. | int |  | 4 | NULL allowed |  |  |  |
|  | Sub­Total Sales subtotal. Computed as SUM(Sales­Order­Detail.Line­Total)for the appropriate Sales­Order­ID. | 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 from customer. Computed as Subtotal + Tax­Amt + Freight. | money | True | 8 | NOT NULL |  |  |  |
|  | Comment Sales representative comments. | nvarchar(128) |  | 256 | 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()) |

Computed columns

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

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Sales­Order­Header\_­Sales­Order­ID Primary key (clustered) constraint | Sales­Order­ID | True |
|  | AK\_­Sales­Order­Header\_­Sales­Order­Number Unique nonclustered index. | Sales­Order­Number | 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]>=[Order­Date]) |
| CK\_­Sales­Order­Header\_­Freight Check constraint [Freight] >= (0.00) | Freight | ([Freight]>=(0.00)) |
| CK\_­Sales­Order­Header\_­Ship­Date Check constraint [Ship­Date] >= [Order­Date] OR [Ship­Date] IS NULL |  | ([Ship­Date]>=[Order­Date] 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]](#YHZkNb8Ts04ByOupF/OAfoeI2qk=) |
| FK\_­Sales­Order­Header\_­Address\_­Ship­To­Address­ID Foreign key constraint referencing Address.Address­ID. | Ship­To­Address­ID->[[Person].[Address].[Address­ID]](#YHZkNb8Ts04ByOupF/OAfoeI2qk=) |
| FK\_­Sales­Order­Header\_­Credit­Card\_­Credit­Card­ID Foreign key constraint referencing Credit­Card.Credit­Card­ID. | Credit­Card­ID->[[Sales].[Credit­Card].[Credit­Card­ID]](#Juoic8jJ3a0e1KCG+cj8LixVRbQ=) |
| FK\_­Sales­Order­Header\_­Currency­Rate\_­Currency­Rate­ID Foreign key constraint referencing Currency­Rate.Currency­Rate­ID. | Currency­Rate­ID->[[Sales].[Currency­Rate].[Currency­Rate­ID]](#DbnMf0tD82NB+BGbYOX9jWqScts=) |
| FK\_­Sales­Order­Header\_­Customer\_­Customer­ID Foreign key constraint referencing Customer.Customer­ID. | Customer­ID->[[Sales].[Customer].[Customer­ID]](#UK53WWAjbLvEqOKY0KByf769x9M=) |
| FK\_­Sales­Order­Header\_­Sales­Person\_­Sales­Person­ID Foreign key constraint referencing Sales­Person.Sales­Person­ID. | Sales­Person­ID->[[Sales].[Sales­Person].[Business­Entity­ID]](#hKN58zHn4ffskJv6ldcUqjf87do=) |
| FK\_­Sales­Order­Header\_­Sales­Territory\_­Territory­ID Foreign key constraint referencing Sales­Territory.Territory­ID. | Territory­ID->[[Sales].[Sales­Territory].[Territory­ID]](#hvNbyL+2AhjcTkzEdMCOV3mg8XQ=) |
| FK\_­Sales­Order­Header\_­Ship­Method\_­Ship­Method­ID Foreign key constraint referencing Ship­Method.Ship­Method­ID. | Ship­Method­ID->[[Purchasing].[Ship­Method].[Ship­Method­ID]](#Xt5ka3ip5ljwxYo+n5TjCRZypoY=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Sales].[Sales­Order­Header]  (  [Sales­Order­ID] [int] NOT NULL IDENTITY(1, 1) NOT FOR REPLICATION,  [Revision­Number] [tinyint] NOT NULL CONSTRAINT [DF\_­Sales­Order­Header\_­Revision­Number] DEFAULT ((0)),  [Order­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Sales­Order­Header\_­Order­Date] DEFAULT (getdate()),  [Due­Date] [datetime] NOT NULL,  [Ship­Date] [datetime] NULL,  [Status] [tinyint] NOT NULL CONSTRAINT [DF\_­Sales­Order­Header\_­Status] DEFAULT ((1)),  [Online­Order­Flag] [dbo].[Flag] NOT NULL CONSTRAINT [DF\_­Sales­Order­Header\_­Online­Order­Flag] DEFAULT ((1)),  [Sales­Order­Number] AS (isnull(N'SO'+CONVERT([nvarchar](23),[Sales­Order­ID],(0)),N'\*\*\* ERROR \*\*\*')),  [Purchase­Order­Number] [dbo].[Order­Number] NULL,  [Account­Number] [dbo].[Account­Number] NULL,  [Customer­ID] [int] NOT NULL,  [Sales­Person­ID] [int] NULL,  [Territory­ID] [int] NULL,  [Bill­To­Address­ID] [int] NOT NULL,  [Ship­To­Address­ID] [int] NOT NULL,  [Ship­Method­ID] [int] NOT NULL,  [Credit­Card­ID] [int] NULL,  [Credit­Card­Approval­Code] [varchar] (15) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Currency­Rate­ID] [int] NULL,  [Sub­Total] [money] NOT NULL CONSTRAINT [DF\_­Sales­Order­Header\_­Sub­Total] DEFAULT ((0.00)),  [Tax­Amt] [money] NOT NULL CONSTRAINT [DF\_­Sales­Order­Header\_­Tax­Amt] DEFAULT ((0.00)),  [Freight] [money] NOT NULL CONSTRAINT [DF\_­Sales­Order­Header\_­Freight] DEFAULT ((0.00)),  [Total­Due] AS (isnull(([Sub­Total]+[Tax­Amt])+[Freight],(0))),  [Comment] [nvarchar] (128) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Sales­Order­Header\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Sales­Order­Header\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  CREATE TRIGGER [Sales].[u­Sales­Order­Header] ON [Sales].[Sales­Order­Header]  AFTER UPDATE NOT FOR REPLICATION AS  BEGIN  DECLARE @Count int;  SET @Count = @@ROWCOUNT;  IF @Count = 0  RETURN;  SET NOCOUNT ON;  BEGIN TRY  -- Update Revision­Number for modification of any field EXCEPT the Status.  IF NOT UPDATE([Status])  BEGIN  UPDATE [Sales].[Sales­Order­Header]  SET [Sales].[Sales­Order­Header].[Revision­Number] =  [Sales].[Sales­Order­Header].[Revision­Number] + 1  WHERE [Sales].[Sales­Order­Header].[Sales­Order­ID] IN  (SELECT inserted.[Sales­Order­ID] FROM inserted);  END;  -- Update the Sales­Person Sales­YTD when Sub­Total is updated  IF UPDATE([Sub­Total])  BEGIN  DECLARE @Start­Date datetime,  @End­Date datetime  SET @Start­Date = [dbo].[ufn­Get­Accounting­Start­Date]();  SET @End­Date = [dbo].[ufn­Get­Accounting­End­Date]();  UPDATE [Sales].[Sales­Person]  SET [Sales].[Sales­Person].[Sales­YTD] =  (SELECT SUM([Sales].[Sales­Order­Header].[Sub­Total])  FROM [Sales].[Sales­Order­Header]  WHERE [Sales].[Sales­Person].[Business­Entity­ID] = [Sales].[Sales­Order­Header].[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].[Sales­Order­Header].[Territory­ID]  AND ([Sales].[Sales­Order­Header].[Status] = 5) -- Shipped  AND [Sales].[Sales­Order­Header].[Order­Date] BETWEEN @Start­Date AND @End­Date)  WHERE [Sales].[Sales­Territory].[Territory­ID]  IN (SELECT DISTINCT inserted.[Territory­ID] FROM inserted  WHERE inserted.[Order­Date] BETWEEN @Start­Date AND @End­Date);  END;  END TRY  BEGIN CATCH  EXECUTE [dbo].[usp­Print­Error];  -- Rollback any active or uncommittable transactions before  -- inserting information in the Error­Log  IF @@TRANCOUNT > 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE [dbo].[usp­Log­Error];  END CATCH;  END;  GO  ALTER TABLE [Sales].[Sales­Order­Header] ADD CONSTRAINT [CK\_­Sales­Order­Header\_­Due­Date] CHECK (([Due­Date]>=[Order­Date]))  GO  ALTER TABLE [Sales].[Sales­Order­Header] ADD CONSTRAINT [CK\_­Sales­Order­Header\_­Freight] CHECK (([Freight]>=(0.00)))  GO  ALTER TABLE [Sales].[Sales­Order­Header] ADD CONSTRAINT [CK\_­Sales­Order­Header\_­Ship­Date] CHECK (([Ship­Date]>=[Order­Date] OR [Ship­Date] IS NULL))  GO  ALTER TABLE [Sales].[Sales­Order­Header] ADD CONSTRAINT [CK\_­Sales­Order­Header\_­Status] CHECK (([Status]>=(0) AND [Status]<=(8)))  GO  ALTER TABLE [Sales].[Sales­Order­Header] ADD CONSTRAINT [CK\_­Sales­Order­Header\_­Sub­Total] CHECK (([Sub­Total]>=(0.00)))  GO  ALTER TABLE [Sales].[Sales­Order­Header] ADD CONSTRAINT [CK\_­Sales­Order­Header\_­Tax­Amt] CHECK (([Tax­Amt]>=(0.00)))  GO  ALTER TABLE [Sales].[Sales­Order­Header] ADD CONSTRAINT [PK\_­Sales­Order­Header\_­Sales­Order­ID] PRIMARY KEY CLUSTERED ([Sales­Order­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Sales­Order­Header\_­Customer­ID] ON [Sales].[Sales­Order­Header] ([Customer­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Sales­Order­Header\_rowguid] ON [Sales].[Sales­Order­Header] ([rowguid]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Sales­Order­Header\_­Sales­Order­Number] ON [Sales].[Sales­Order­Header] ([Sales­Order­Number]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Sales­Order­Header\_­Sales­Person­ID] ON [Sales].[Sales­Order­Header] ([Sales­Person­ID]) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Sales­Order­Header] ADD CONSTRAINT [FK\_­Sales­Order­Header\_­Address\_­Bill­To­Address­ID] FOREIGN KEY ([Bill­To­Address­ID]) REFERENCES [Person].[Address] ([Address­ID])  GO  ALTER TABLE [Sales].[Sales­Order­Header] ADD CONSTRAINT [FK\_­Sales­Order­Header\_­Address\_­Ship­To­Address­ID] FOREIGN KEY ([Ship­To­Address­ID]) REFERENCES [Person].[Address] ([Address­ID])  GO  ALTER TABLE [Sales].[Sales­Order­Header] ADD CONSTRAINT [FK\_­Sales­Order­Header\_­Credit­Card\_­Credit­Card­ID] FOREIGN KEY ([Credit­Card­ID]) REFERENCES [Sales].[Credit­Card] ([Credit­Card­ID])  GO  ALTER TABLE [Sales].[Sales­Order­Header] ADD CONSTRAINT [FK\_­Sales­Order­Header\_­Currency­Rate\_­Currency­Rate­ID] FOREIGN KEY ([Currency­Rate­ID]) REFERENCES [Sales].[Currency­Rate] ([Currency­Rate­ID])  GO  ALTER TABLE [Sales].[Sales­Order­Header] ADD CONSTRAINT [FK\_­Sales­Order­Header\_­Customer\_­Customer­ID] FOREIGN KEY ([Customer­ID]) REFERENCES [Sales].[Customer] ([Customer­ID])  GO  ALTER TABLE [Sales].[Sales­Order­Header] ADD CONSTRAINT [FK\_­Sales­Order­Header\_­Sales­Person\_­Sales­Person­ID] FOREIGN KEY ([Sales­Person­ID]) REFERENCES [Sales].[Sales­Person] ([Business­Entity­ID])  GO  ALTER TABLE [Sales].[Sales­Order­Header] ADD CONSTRAINT [FK\_­Sales­Order­Header\_­Sales­Territory\_­Territory­ID] FOREIGN KEY ([Territory­ID]) REFERENCES [Sales].[Sales­Territory] ([Territory­ID])  GO  ALTER TABLE [Sales].[Sales­Order­Header] ADD CONSTRAINT [FK\_­Sales­Order­Header\_­Ship­Method\_­Ship­Method­ID] FOREIGN KEY ([Ship­Method­ID]) REFERENCES [Purchasing].[Ship­Method] ([Ship­Method­ID])  GO  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'Account­Number'  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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Approval code provided by the credit card company.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Credit­Card­Approval­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Credit card identification number. Foreign key to Credit­Card.Credit­Card­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Credit­Card­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Currency exchange rate used. Foreign key to Currency­Rate.Currency­Rate­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Currency­Rate­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Customer identification number. Foreign key to Customer.Business­Entity­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Customer­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date the order is due to the customer.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Due­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Shipping cost.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Freight'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = Order placed by sales person. 1 = Order placed online by customer.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Online­Order­Flag'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Dates the sales order was created.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Order­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Customer purchase order number reference. ', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Purchase­Order­Number'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Incremental number to track changes to the sales order over time.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Revision­Number'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Sales­Order­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique sales order identification number.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Sales­Order­Number'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Sales person who created the sales order. Foreign key to Sales­Person.Business­Entity­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Sales­Person­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date the order was shipped to the customer.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Ship­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Shipping method. Foreign key to Ship­Method.Ship­Method­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Ship­Method­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Customer shipping address. Foreign key to Address.Address­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Ship­To­Address­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Order current status. 1 = In process; 2 = Approved; 3 = Backordered; 4 = Rejected; 5 = Shipped; 6 = Cancelled', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Status'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Sales subtotal. Computed as SUM(Sales­Order­Detail.Line­Total)for the appropriate Sales­Order­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Sub­Total'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Tax amount.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Tax­Amt'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Territory in which the sale was made. Foreign key to Sales­Territory.Sales­Territory­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Territory­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Total due from customer. Computed as Subtotal + Tax­Amt + Freight.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'COLUMN', N'Total­Due'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Due­Date] >= [Order­Date]', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'CK\_­Sales­Order­Header\_­Due­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Freight] >= (0.00)', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'CK\_­Sales­Order­Header\_­Freight'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Ship­Date] >= [Order­Date] OR [Ship­Date] IS NULL', 'SCHEMA', N'Sales', 'TABLE', N'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\_­Sales­Order­Header\_­Status'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Sub­Total] >= (0.00)', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'CK\_­Sales­Order­Header\_­Sub­Total'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Tax­Amt] >= (0.00)', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'CK\_­Sales­Order­Header\_­Tax­Amt'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0.0', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'DF\_­Sales­Order­Header\_­Freight'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'DF\_­Sales­Order­Header\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 1 (TRUE)', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'DF\_­Sales­Order­Header\_­Online­Order­Flag'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'DF\_­Sales­Order­Header\_­Order­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'DF\_­Sales­Order­Header\_­Revision­Number'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'DF\_­Sales­Order­Header\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 1', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'DF\_­Sales­Order­Header\_­Status'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0.0', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'DF\_­Sales­Order­Header\_­Sub­Total'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0.0', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'DF\_­Sales­Order­Header\_­Tax­Amt'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Address.Address­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'FK\_­Sales­Order­Header\_­Address\_­Bill­To­Address­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Address.Address­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'FK\_­Sales­Order­Header\_­Address\_­Ship­To­Address­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Credit­Card.Credit­Card­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'FK\_­Sales­Order­Header\_­Credit­Card\_­Credit­Card­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Currency­Rate.Currency­Rate­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'FK\_­Sales­Order­Header\_­Currency­Rate\_­Currency­Rate­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Customer.Customer­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'FK\_­Sales­Order­Header\_­Customer\_­Customer­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Sales­Person.Sales­Person­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'FK\_­Sales­Order­Header\_­Sales­Person\_­Sales­Person­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Sales­Territory.Territory­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'FK\_­Sales­Order­Header\_­Sales­Territory\_­Territory­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Ship­Method.Ship­Method­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'FK\_­Sales­Order­Header\_­Ship­Method\_­Ship­Method­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'CONSTRAINT', N'PK\_­Sales­Order­Header\_­Sales­Order­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'INDEX', N'AK\_­Sales­Order­Header\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'INDEX', N'AK\_­Sales­Order­Header\_­Sales­Order­Number'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'INDEX', N'IX\_­Sales­Order­Header\_­Customer­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'INDEX', N'IX\_­Sales­Order­Header\_­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\_­Sales­Order­Header\_­Sales­Order­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'AFTER UPDATE trigger that updates the Revision­Number and Modified­Date columns in the Sales­Order­Header table.Updates the Sales­YTD column in the Sales­Person and Sales­Territory tables.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header', 'TRIGGER', N'u­Sales­Order­Header'  GO |

Uses

[[Person].[Address]](#YHZkNb8Ts04ByOupF/OAfoeI2qk=)

[[Purchasing].[Ship­Method]](#Xt5ka3ip5ljwxYo+n5TjCRZypoY=)

[[Sales].[Credit­Card]](#Juoic8jJ3a0e1KCG+cj8LixVRbQ=)

[[Sales].[Currency­Rate]](#DbnMf0tD82NB+BGbYOX9jWqScts=)

[[Sales].[Customer]](#UK53WWAjbLvEqOKY0KByf769x9M=)

[[Sales].[Sales­Person]](#hKN58zHn4ffskJv6ldcUqjf87do=)

[[Sales].[Sales­Territory]](#hvNbyL+2AhjcTkzEdMCOV3mg8XQ=)

[[dbo].[Account­Number]](#wfUPMi1TrfOhNsrHkC5PC5H/4RE=)

[[dbo].[Flag]](#j88jjDtwCbK8khz/m8jxryH/SFk=)

[[dbo].[Order­Number]](#SZqX2/vCAFjEqiLsacel6rQr3G4=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

Used By

[[Sales].[Sales­Order­Detail]](#OkvyE7uLBgi7y2P/yCWzRf5c0wE=)

[[Sales].[Sales­Order­Header­Sales­Reason]](#8Csb5Ws8nFAO1qWhqs6WCgPvmp4=)

[[Sales].[v­Sales­Person­Sales­By­Fiscal­Years]](#GfF5ZKdsT1+xL6X5ks+ghnXIfng=)

|  |
| --- |
| [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 |
|  | Sales­Order­ID Primary key. Foreign key to Sales­Order­Header.Sales­Order­ID. | int | 4 | NOT NULL |  |
|  | Sales­Reason­ID Primary key. Foreign key to Sales­Reason.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\_­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].[Sales­Order­Header].[Sales­Order­ID]](#J0oqEfXKE3WFfJ7PoPfs1QUbO+c=) |
| FK\_­Sales­Order­Header­Sales­Reason\_­Sales­Reason\_­Sales­Reason­ID Foreign key constraint referencing Sales­Reason.Sales­Reason­ID. |  | Sales­Reason­ID->[[Sales].[Sales­Reason].[Sales­Reason­ID]](#a3VrLxK/Cx1h+4jZ069Ywnkt+DM=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Sales].[Sales­Order­Header­Sales­Reason]  (  [Sales­Order­ID] [int] NOT NULL,  [Sales­Reason­ID] [int] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Sales­Order­Header­Sales­Reason\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Sales­Order­Header­Sales­Reason] ADD CONSTRAINT [PK\_­Sales­Order­Header­Sales­Reason\_­Sales­Order­ID\_­Sales­Reason­ID] PRIMARY KEY CLUSTERED ([Sales­Order­ID], [Sales­Reason­ID]) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Sales­Order­Header­Sales­Reason] ADD CONSTRAINT [FK\_­Sales­Order­Header­Sales­Reason\_­Sales­Order­Header\_­Sales­Order­ID] FOREIGN KEY ([Sales­Order­ID]) REFERENCES [Sales].[Sales­Order­Header] ([Sales­Order­ID]) ON DELETE CASCADE  GO  ALTER TABLE [Sales].[Sales­Order­Header­Sales­Reason] ADD CONSTRAINT [FK\_­Sales­Order­Header­Sales­Reason\_­Sales­Reason\_­Sales­Reason­ID] FOREIGN KEY ([Sales­Reason­ID]) REFERENCES [Sales].[Sales­Reason] ([Sales­Reason­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Cross-reference table mapping sales orders to sales reason codes.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header­Sales­Reason', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header­Sales­Reason', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Foreign key to Sales­Order­Header.Sales­Order­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header­Sales­Reason', 'COLUMN', N'Sales­Order­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Foreign key to Sales­Reason.Sales­Reason­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header­Sales­Reason', 'COLUMN', N'Sales­Reason­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header­Sales­Reason', 'CONSTRAINT', N'DF\_­Sales­Order­Header­Sales­Reason\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Sales­Order­Header.Sales­Order­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header­Sales­Reason', 'CONSTRAINT', N'FK\_­Sales­Order­Header­Sales­Reason\_­Sales­Order­Header\_­Sales­Order­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Sales­Reason.Sales­Reason­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header­Sales­Reason', 'CONSTRAINT', N'FK\_­Sales­Order­Header­Sales­Reason\_­Sales­Reason\_­Sales­Reason­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header­Sales­Reason', 'CONSTRAINT', N'PK\_­Sales­Order­Header­Sales­Reason\_­Sales­Order­ID\_­Sales­Reason­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Order­Header­Sales­Reason', 'INDEX', N'PK\_­Sales­Order­Header­Sales­Reason\_­Sales­Order­ID\_­Sales­Reason­ID'  GO |

Uses

[[Sales].[Sales­Order­Header]](#J0oqEfXKE3WFfJ7PoPfs1QUbO+c=)

[[Sales].[Sales­Reason]](#a3VrLxK/Cx1h+4jZ069Ywnkt+DM=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

|  |
| --- |
| [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 |
|  | Business­Entity­ID Primary key for Sales­Person records. Foreign key to Employee.Business­Entity­ID | int | 4 | NOT NULL |  |
|  | 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)) |
|  | 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\_­Sales­Person\_­Business­Entity­ID Primary key (clustered) constraint | Business­Entity­ID | True |
|  | AK\_­Sales­Person\_rowguid Unique nonclustered index. Used to support replication samples. | rowguid | True |

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.Employee­ID. | Business­Entity­ID->[[Human­Resources].[Employee].[Business­Entity­ID]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=) |
| FK\_­Sales­Person\_­Sales­Territory\_­Territory­ID Foreign key constraint referencing Sales­Territory.Territory­ID. | Territory­ID->[[Sales].[Sales­Territory].[Territory­ID]](#hvNbyL+2AhjcTkzEdMCOV3mg8XQ=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Sales].[Sales­Person]  (  [Business­Entity­ID] [int] NOT NULL,  [Territory­ID] [int] NULL,  [Sales­Quota] [money] NULL,  [Bonus] [money] NOT NULL CONSTRAINT [DF\_­Sales­Person\_­Bonus] DEFAULT ((0.00)),  [Commission­Pct] [smallmoney] NOT NULL CONSTRAINT [DF\_­Sales­Person\_­Commission­Pct] DEFAULT ((0.00)),  [Sales­YTD] [money] NOT NULL CONSTRAINT [DF\_­Sales­Person\_­Sales­YTD] DEFAULT ((0.00)),  [Sales­Last­Year] [money] NOT NULL CONSTRAINT [DF\_­Sales­Person\_­Sales­Last­Year] DEFAULT ((0.00)),  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Sales­Person\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Sales­Person\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Sales­Person] ADD CONSTRAINT [CK\_­Sales­Person\_­Bonus] CHECK (([Bonus]>=(0.00)))  GO  ALTER TABLE [Sales].[Sales­Person] ADD CONSTRAINT [CK\_­Sales­Person\_­Commission­Pct] CHECK (([Commission­Pct]>=(0.00)))  GO  ALTER TABLE [Sales].[Sales­Person] ADD CONSTRAINT [CK\_­Sales­Person\_­Sales­Last­Year] CHECK (([Sales­Last­Year]>=(0.00)))  GO  ALTER TABLE [Sales].[Sales­Person] ADD CONSTRAINT [CK\_­Sales­Person\_­Sales­Quota] CHECK (([Sales­Quota]>(0.00)))  GO  ALTER TABLE [Sales].[Sales­Person] ADD CONSTRAINT [CK\_­Sales­Person\_­Sales­YTD] CHECK (([Sales­YTD]>=(0.00)))  GO  ALTER TABLE [Sales].[Sales­Person] ADD CONSTRAINT [PK\_­Sales­Person\_­Business­Entity­ID] PRIMARY KEY CLUSTERED ([Business­Entity­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Sales­Person\_rowguid] ON [Sales].[Sales­Person] ([rowguid]) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Sales­Person] ADD CONSTRAINT [FK\_­Sales­Person\_­Employee\_­Business­Entity­ID] FOREIGN KEY ([Business­Entity­ID]) REFERENCES [Human­Resources].[Employee] ([Business­Entity­ID])  GO  ALTER TABLE [Sales].[Sales­Person] ADD CONSTRAINT [FK\_­Sales­Person\_­Sales­Territory\_­Territory­ID] FOREIGN KEY ([Territory­ID]) REFERENCES [Sales].[Sales­Territory] ([Territory­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Sales representative current information.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Bonus due if quota is met.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'COLUMN', N'Bonus'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Sales­Person records. Foreign key to Employee.Business­Entity­ID', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'COLUMN', N'Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Commision percent received per sale.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'COLUMN', N'Commission­Pct'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Sales total of previous year.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'COLUMN', N'Sales­Last­Year'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Projected yearly sales.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'COLUMN', N'Sales­Quota'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Sales total year to date.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'COLUMN', N'Sales­YTD'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Territory currently assigned to. Foreign key to Sales­Territory.Sales­Territory­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'COLUMN', N'Territory­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Bonus] >= (0.00)', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'CONSTRAINT', N'CK\_­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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Sales­Last­Year] >= (0.00)', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'CONSTRAINT', N'CK\_­Sales­Person\_­Sales­Last­Year'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Sales­Quota] > (0.00)', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'CONSTRAINT', N'CK\_­Sales­Person\_­Sales­Quota'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Sales­YTD] >= (0.00)', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'CONSTRAINT', N'CK\_­Sales­Person\_­Sales­YTD'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0.0', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'CONSTRAINT', N'DF\_­Sales­Person\_­Bonus'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0.0', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'CONSTRAINT', N'DF\_­Sales­Person\_­Commission­Pct'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'CONSTRAINT', N'DF\_­Sales­Person\_­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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0.0', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'CONSTRAINT', N'DF\_­Sales­Person\_­Sales­Last­Year'  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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Employee.Employee­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'CONSTRAINT', N'FK\_­Sales­Person\_­Employee\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Sales­Territory.Territory­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'CONSTRAINT', N'FK\_­Sales­Person\_­Sales­Territory\_­Territory­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'CONSTRAINT', N'PK\_­Sales­Person\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'INDEX', N'AK\_­Sales­Person\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person', 'INDEX', N'PK\_­Sales­Person\_­Business­Entity­ID'  GO |

Uses

[[Human­Resources].[Employee]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=)

[[Sales].[Sales­Territory]](#hvNbyL+2AhjcTkzEdMCOV3mg8XQ=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

Used By

[[Sales].[Sales­Order­Header]](#J0oqEfXKE3WFfJ7PoPfs1QUbO+c=)

[[Sales].[Sales­Person­Quota­History]](#ZEp3m1cn9szNTztS7qyCt1Ng+u8=)

[[Sales].[Sales­Territory­History]](#STLuHwwNIM3t3qtp2pV+eqOT5iE=)

[[Sales].[Store]](#o4yE5vuw4wTVa38E7sxNcXx3KhA=)

[[Sales].[v­Sales­Person]](#pOjFS54tNwjD3Yvi8IMnn7dtRuA=)

[[Sales].[v­Sales­Person­Sales­By­Fiscal­Years]](#GfF5ZKdsT1+xL6X5ks+ghnXIfng=)

|  |
| --- |
| [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 |
|  | Business­Entity­ID Sales person identification number. Foreign key to Sales­Person.Business­Entity­ID. | int | 4 | NOT NULL |  |
|  | Quota­Date Sales quota date. | datetime | 8 | NOT NULL |  |
|  | Sales­Quota Sales quota amount. | money | 8 | 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\_­Sales­Person­Quota­History\_­Business­Entity­ID\_­Quota­Date Primary key (clustered) constraint | Business­Entity­ID, 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\_­Business­Entity­ID Foreign key constraint referencing Sales­Person.Sales­Person­ID. | Business­Entity­ID->[[Sales].[Sales­Person].[Business­Entity­ID]](#hKN58zHn4ffskJv6ldcUqjf87do=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Sales].[Sales­Person­Quota­History]  (  [Business­Entity­ID] [int] NOT NULL,  [Quota­Date] [datetime] NOT NULL,  [Sales­Quota] [money] NOT NULL,  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Sales­Person­Quota­History\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Sales­Person­Quota­History\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Sales­Person­Quota­History] ADD CONSTRAINT [CK\_­Sales­Person­Quota­History\_­Sales­Quota] CHECK (([Sales­Quota]>(0.00)))  GO  ALTER TABLE [Sales].[Sales­Person­Quota­History] ADD CONSTRAINT [PK\_­Sales­Person­Quota­History\_­Business­Entity­ID\_­Quota­Date] PRIMARY KEY CLUSTERED ([Business­Entity­ID], [Quota­Date]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Sales­Person­Quota­History\_rowguid] ON [Sales].[Sales­Person­Quota­History] ([rowguid]) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Sales­Person­Quota­History] ADD CONSTRAINT [FK\_­Sales­Person­Quota­History\_­Sales­Person\_­Business­Entity­ID] FOREIGN KEY ([Business­Entity­ID]) REFERENCES [Sales].[Sales­Person] ([Business­Entity­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Sales performance tracking.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person­Quota­History', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Sales person identification number. Foreign key to Sales­Person.Business­Entity­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person­Quota­History', 'COLUMN', N'Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person­Quota­History', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Sales quota date.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person­Quota­History', 'COLUMN', N'Quota­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person­Quota­History', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Sales quota amount.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person­Quota­History', 'COLUMN', N'Sales­Quota'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Sales­Quota] > (0.00)', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person­Quota­History', 'CONSTRAINT', N'CK\_­Sales­Person­Quota­History\_­Sales­Quota'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person­Quota­History', 'CONSTRAINT', N'DF\_­Sales­Person­Quota­History\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Person­Quota­History', 'CONSTRAINT', N'DF\_­Sales­Person­Quota­History\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Sales­Person.Sales­Person­ID.', 'SCHEMA', N'Sales', 'TABLE', N'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\_­Sales­Person­Quota­History\_­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 |

Uses

[[Sales].[Sales­Person]](#hKN58zHn4ffskJv6ldcUqjf87do=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

|  |
| --- |
| [Sales].[Sales­Reason] |

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 |
|  | Sales­Reason­ID Primary key for Sales­Reason records. | int | 4 | NOT NULL | 1 - 1 |  |
|  | Name Sales reason description. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NOT NULL |  |  |
|  | Reason­Type Category the sales reason belongs to. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NOT NULL |  |  |
|  | Modified­Date Date and time the record was last updated. | datetime | 8 | NOT NULL |  | (getdate()) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Sales­Reason\_­Sales­Reason­ID Primary key (clustered) constraint | Sales­Reason­ID | True |

SQL Script

|  |
| --- |
| CREATE TABLE [Sales].[Sales­Reason]  (  [Sales­Reason­ID] [int] NOT NULL IDENTITY(1, 1),  [Name] [dbo].[Name] NOT NULL,  [Reason­Type] [dbo].[Name] NOT NULL,  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Sales­Reason\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Sales­Reason] ADD CONSTRAINT [PK\_­Sales­Reason\_­Sales­Reason­ID] PRIMARY KEY CLUSTERED ([Sales­Reason­ID]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Lookup table of customer purchase reasons.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Reason', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Reason', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Sales reason description.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Reason', 'COLUMN', N'Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Category the sales reason belongs to.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Reason', 'COLUMN', N'Reason­Type'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Sales­Reason records.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Reason', 'COLUMN', N'Sales­Reason­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Reason', 'CONSTRAINT', N'DF\_­Sales­Reason\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Reason', 'CONSTRAINT', N'PK\_­Sales­Reason\_­Sales­Reason­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Reason', 'INDEX', N'PK\_­Sales­Reason\_­Sales­Reason­ID'  GO |

Uses

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

Used By

[[Sales].[Sales­Order­Header­Sales­Reason]](#8Csb5Ws8nFAO1qWhqs6WCgPvmp4=)

|  |
| --- |
| [Sales].[Sales­Tax­Rate] |

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) | Nullability | Identity | Default |
|  | Sales­Tax­Rate­ID Primary key for Sales­Tax­Rate records. | int | 4 | NOT NULL | 1 - 1 |  |
|  | 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]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 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\_­Sales­Tax­Rate\_­Sales­Tax­Rate­ID Primary key (clustered) constraint | Sales­Tax­Rate­ID | True |
|  | AK\_­Sales­Tax­Rate\_­State­Province­ID\_­Tax­Type Unique nonclustered index. | State­Province­ID, 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) | Tax­Type | ([Tax­Type]>=(1) AND [Tax­Type]<=(3)) |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Sales­Tax­Rate\_­State­Province\_­State­Province­ID Foreign key constraint referencing State­Province.State­Province­ID. | State­Province­ID->[[Person].[State­Province].[State­Province­ID]](#fYuToAkdHMSws7pXek8EIcFTfvY=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Sales].[Sales­Tax­Rate]  (  [Sales­Tax­Rate­ID] [int] NOT NULL IDENTITY(1, 1),  [State­Province­ID] [int] NOT NULL,  [Tax­Type] [tinyint] NOT NULL,  [Tax­Rate] [smallmoney] NOT NULL CONSTRAINT [DF\_­Sales­Tax­Rate\_­Tax­Rate] DEFAULT ((0.00)),  [Name] [dbo].[Name] NOT NULL,  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Sales­Tax­Rate\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Sales­Tax­Rate\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Sales­Tax­Rate] ADD CONSTRAINT [CK\_­Sales­Tax­Rate\_­Tax­Type] CHECK (([Tax­Type]>=(1) AND [Tax­Type]<=(3)))  GO  ALTER TABLE [Sales].[Sales­Tax­Rate] ADD CONSTRAINT [PK\_­Sales­Tax­Rate\_­Sales­Tax­Rate­ID] PRIMARY KEY CLUSTERED ([Sales­Tax­Rate­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Sales­Tax­Rate\_rowguid] ON [Sales].[Sales­Tax­Rate] ([rowguid]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Sales­Tax­Rate\_­State­Province­ID\_­Tax­Type] ON [Sales].[Sales­Tax­Rate] ([State­Province­ID], [Tax­Type]) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Sales­Tax­Rate] ADD CONSTRAINT [FK\_­Sales­Tax­Rate\_­State­Province\_­State­Province­ID] FOREIGN KEY ([State­Province­ID]) REFERENCES [Person].[State­Province] ([State­Province­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Tax rate lookup table.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Tax­Rate', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Tax­Rate', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Tax rate description.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Tax­Rate', 'COLUMN', N'Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Tax­Rate', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Sales­Tax­Rate records.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Tax­Rate', 'COLUMN', N'Sales­Tax­Rate­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'State, province, or country/region the sales tax applies to.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Tax­Rate', 'COLUMN', N'State­Province­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Tax rate amount.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Tax­Rate', 'COLUMN', N'Tax­Rate'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'1 = Tax applied to retail transactions, 2 = Tax applied to wholesale transactions, 3 = Tax applied to all sales (retail and wholesale) transactions.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Tax­Rate', 'COLUMN', N'Tax­Type'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Tax­Type] BETWEEN (1) AND (3)', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Tax­Rate', 'CONSTRAINT', N'CK\_­Sales­Tax­Rate\_­Tax­Type'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Tax­Rate', 'CONSTRAINT', N'DF\_­Sales­Tax­Rate\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Tax­Rate', 'CONSTRAINT', N'DF\_­Sales­Tax­Rate\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0.0', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Tax­Rate', 'CONSTRAINT', N'DF\_­Sales­Tax­Rate\_­Tax­Rate'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing State­Province.State­Province­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Tax­Rate', 'CONSTRAINT', N'FK\_­Sales­Tax­Rate\_­State­Province\_­State­Province­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Tax­Rate', 'CONSTRAINT', N'PK\_­Sales­Tax­Rate\_­Sales­Tax­Rate­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Tax­Rate', 'INDEX', N'AK\_­Sales­Tax­Rate\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Tax­Rate', 'INDEX', N'AK\_­Sales­Tax­Rate\_­State­Province­ID\_­Tax­Type'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Tax­Rate', 'INDEX', N'PK\_­Sales­Tax­Rate\_­Sales­Tax­Rate­ID'  GO |

Uses

[[Person].[State­Province]](#fYuToAkdHMSws7pXek8EIcFTfvY=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

|  |
| --- |
| [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 |
|  | Territory­ID Primary key for Sales­Territory records. | int | 4 | NOT NULL | 1 - 1 |  |
|  | Name Sales territory description | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NOT NULL |  |  |
|  | Country­Region­Code ISO standard country or region code. Foreign key to Country­Region.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\_­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 Country­Region.Country­Region­Code. | Country­Region­Code->[[Person].[Country­Region].[Country­Region­Code]](#efOGmVlBeofSPEY2jc0F91joV9Q=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Sales].[Sales­Territory]  (  [Territory­ID] [int] NOT NULL IDENTITY(1, 1),  [Name] [dbo].[Name] NOT NULL,  [Country­Region­Code] [nvarchar] (3) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Group] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Sales­YTD] [money] NOT NULL CONSTRAINT [DF\_­Sales­Territory\_­Sales­YTD] DEFAULT ((0.00)),  [Sales­Last­Year] [money] NOT NULL CONSTRAINT [DF\_­Sales­Territory\_­Sales­Last­Year] DEFAULT ((0.00)),  [Cost­YTD] [money] NOT NULL CONSTRAINT [DF\_­Sales­Territory\_­Cost­YTD] DEFAULT ((0.00)),  [Cost­Last­Year] [money] NOT NULL CONSTRAINT [DF\_­Sales­Territory\_­Cost­Last­Year] DEFAULT ((0.00)),  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­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)))  GO  ALTER TABLE [Sales].[Sales­Territory] ADD CONSTRAINT [CK\_­Sales­Territory\_­Cost­YTD] CHECK (([Cost­YTD]>=(0.00)))  GO  ALTER TABLE [Sales].[Sales­Territory] ADD CONSTRAINT [CK\_­Sales­Territory\_­Sales­Last­Year] CHECK (([Sales­Last­Year]>=(0.00)))  GO  ALTER TABLE [Sales].[Sales­Territory] ADD CONSTRAINT [CK\_­Sales­Territory\_­Sales­YTD] CHECK (([Sales­YTD]>=(0.00)))  GO  ALTER TABLE [Sales].[Sales­Territory] ADD CONSTRAINT [PK\_­Sales­Territory\_­Territory­ID] PRIMARY KEY CLUSTERED ([Territory­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Sales­Territory\_­Name] ON [Sales].[Sales­Territory] ([Name]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Sales­Territory\_rowguid] ON [Sales].[Sales­Territory] ([rowguid]) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Sales­Territory] ADD CONSTRAINT [FK\_­Sales­Territory\_­Country­Region\_­Country­Region­Code] FOREIGN KEY ([Country­Region­Code]) REFERENCES [Person].[Country­Region] ([Country­Region­Code])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Sales territory lookup table.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Business costs in the territory the previous year.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'COLUMN', N'Cost­Last­Year'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Business costs in the territory year to date.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'COLUMN', N'Cost­YTD'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ISO standard country or region code. Foreign key to Country­Region.Country­Region­Code. ', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'COLUMN', N'Country­Region­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Geographic area to which the sales territory belong.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'COLUMN', N'Group'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Sales territory description', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'COLUMN', N'Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Sales in the territory the previous year.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'COLUMN', N'Sales­Last­Year'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Sales in the territory year to date.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'COLUMN', N'Sales­YTD'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Sales­Territory records.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'COLUMN', N'Territory­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Cost­Last­Year] >= (0.00)', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'CONSTRAINT', N'CK\_­Sales­Territory\_­Cost­Last­Year'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Cost­YTD] >= (0.00)', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'CONSTRAINT', N'CK\_­Sales­Territory\_­Cost­YTD'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Sales­Last­Year] >= (0.00)', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'CONSTRAINT', N'CK\_­Sales­Territory\_­Sales­Last­Year'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Sales­YTD] >= (0.00)', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'CONSTRAINT', N'CK\_­Sales­Territory\_­Sales­YTD'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0.0', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'CONSTRAINT', N'DF\_­Sales­Territory\_­Cost­Last­Year'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0.0', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'CONSTRAINT', N'DF\_­Sales­Territory\_­Cost­YTD'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'CONSTRAINT', N'DF\_­Sales­Territory\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'CONSTRAINT', N'DF\_­Sales­Territory\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0.0', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'CONSTRAINT', N'DF\_­Sales­Territory\_­Sales­Last­Year'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 0.0', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'CONSTRAINT', N'DF\_­Sales­Territory\_­Sales­YTD'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Country­Region.Country­Region­Code.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'CONSTRAINT', N'FK\_­Sales­Territory\_­Country­Region\_­Country­Region­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'CONSTRAINT', N'PK\_­Sales­Territory\_­Territory­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'INDEX', N'AK\_­Sales­Territory\_­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'INDEX', N'AK\_­Sales­Territory\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory', 'INDEX', N'PK\_­Sales­Territory\_­Territory­ID'  GO |

Uses

[[Person].[Country­Region]](#efOGmVlBeofSPEY2jc0F91joV9Q=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

Used By

[[Person].[State­Province]](#fYuToAkdHMSws7pXek8EIcFTfvY=)

[[Sales].[Customer]](#UK53WWAjbLvEqOKY0KByf769x9M=)

[[Sales].[Sales­Order­Header]](#J0oqEfXKE3WFfJ7PoPfs1QUbO+c=)

[[Sales].[Sales­Person]](#hKN58zHn4ffskJv6ldcUqjf87do=)

[[Sales].[Sales­Territory­History]](#STLuHwwNIM3t3qtp2pV+eqOT5iE=)

[[Sales].[v­Sales­Person]](#pOjFS54tNwjD3Yvi8IMnn7dtRuA=)

[[Sales].[v­Sales­Person­Sales­By­Fiscal­Years]](#GfF5ZKdsT1+xL6X5ks+ghnXIfng=)

|  |
| --- |
| [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 |
|  | Business­Entity­ID Primary key. The sales rep. Foreign key to Sales­Person.Business­Entity­ID. | int | 4 | NOT NULL |  |
|  | Territory­ID Primary key. Territory identification number. Foreign key to Sales­Territory.Sales­Territory­ID. | int | 4 | NOT NULL |  |
|  | 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 |
|  | PK\_­Sales­Territory­History\_­Business­Entity­ID\_­Start­Date\_­Territory­ID Primary key (clustered) constraint | Business­Entity­ID, 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.Sales­Person­ID. | Business­Entity­ID->[[Sales].[Sales­Person].[Business­Entity­ID]](#hKN58zHn4ffskJv6ldcUqjf87do=) |
| FK\_­Sales­Territory­History\_­Sales­Territory\_­Territory­ID Foreign key constraint referencing Sales­Territory.Territory­ID. | Territory­ID->[[Sales].[Sales­Territory].[Territory­ID]](#hvNbyL+2AhjcTkzEdMCOV3mg8XQ=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Sales].[Sales­Territory­History]  (  [Business­Entity­ID] [int] NOT NULL,  [Territory­ID] [int] NOT NULL,  [Start­Date] [datetime] NOT NULL,  [End­Date] [datetime] NULL,  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Sales­Territory­History\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Sales­Territory­History\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Sales­Territory­History] ADD CONSTRAINT [CK\_­Sales­Territory­History\_­End­Date] CHECK (([End­Date]>=[Start­Date] OR [End­Date] IS NULL))  GO  ALTER TABLE [Sales].[Sales­Territory­History] ADD CONSTRAINT [PK\_­Sales­Territory­History\_­Business­Entity­ID\_­Start­Date\_­Territory­ID] PRIMARY KEY CLUSTERED ([Business­Entity­ID], [Start­Date], [Territory­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Sales­Territory­History\_rowguid] ON [Sales].[Sales­Territory­History] ([rowguid]) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Sales­Territory­History] ADD CONSTRAINT [FK\_­Sales­Territory­History\_­Sales­Person\_­Business­Entity­ID] FOREIGN KEY ([Business­Entity­ID]) REFERENCES [Sales].[Sales­Person] ([Business­Entity­ID])  GO  ALTER TABLE [Sales].[Sales­Territory­History] ADD CONSTRAINT [FK\_­Sales­Territory­History\_­Sales­Territory\_­Territory­ID] FOREIGN KEY ([Territory­ID]) REFERENCES [Sales].[Sales­Territory] ([Territory­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Sales representative transfers to other sales territories.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory­History', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. The sales rep. Foreign key to Sales­Person.Business­Entity­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory­History', 'COLUMN', N'Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date the sales representative left work in the territory.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory­History', 'COLUMN', N'End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory­History', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory­History', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Date the sales representive started work in the territory.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory­History', 'COLUMN', N'Start­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Territory identification number. Foreign key to Sales­Territory.Sales­Territory­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory­History', 'COLUMN', N'Territory­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [End­Date] >= [Start­Date] OR [End­Date] IS NULL', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory­History', 'CONSTRAINT', N'CK\_­Sales­Territory­History\_­End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory­History', 'CONSTRAINT', N'DF\_­Sales­Territory­History\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory­History', 'CONSTRAINT', N'DF\_­Sales­Territory­History\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Sales­Person.Sales­Person­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory­History', 'CONSTRAINT', N'FK\_­Sales­Territory­History\_­Sales­Person\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Sales­Territory.Territory­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory­History', 'CONSTRAINT', N'FK\_­Sales­Territory­History\_­Sales­Territory\_­Territory­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory­History', 'CONSTRAINT', N'PK\_­Sales­Territory­History\_­Business­Entity­ID\_­Start­Date\_­Territory­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory­History', 'INDEX', N'AK\_­Sales­Territory­History\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'Sales­Territory­History', 'INDEX', N'PK\_­Sales­Territory­History\_­Business­Entity­ID\_­Start­Date\_­Territory­ID'  GO |

Uses

[[Sales].[Sales­Person]](#hKN58zHn4ffskJv6ldcUqjf87do=)

[[Sales].[Sales­Territory]](#hvNbyL+2AhjcTkzEdMCOV3mg8XQ=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

|  |
| --- |
| [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 |
|  | 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)) |
|  | 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\_­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]](#mO6TpkyW7q4W/OMRHISuYfWNCME=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Sales].[Shopping­Cart­Item]  (  [Shopping­Cart­Item­ID] [int] NOT NULL IDENTITY(1, 1),  [Shopping­Cart­ID] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Quantity] [int] NOT NULL CONSTRAINT [DF\_­Shopping­Cart­Item\_­Quantity] DEFAULT ((1)),  [Product­ID] [int] NOT NULL,  [Date­Created] [datetime] NOT NULL CONSTRAINT [DF\_­Shopping­Cart­Item\_­Date­Created] DEFAULT (getdate()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Shopping­Cart­Item\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Shopping­Cart­Item] ADD CONSTRAINT [CK\_­Shopping­Cart­Item\_­Quantity] CHECK (([Quantity]>=(1)))  GO  ALTER TABLE [Sales].[Shopping­Cart­Item] ADD CONSTRAINT [PK\_­Shopping­Cart­Item\_­Shopping­Cart­Item­ID] PRIMARY KEY CLUSTERED ([Shopping­Cart­Item­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Shopping­Cart­Item\_­Shopping­Cart­ID\_­Product­ID] ON [Sales].[Shopping­Cart­Item] ([Shopping­Cart­ID], [Product­ID]) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Shopping­Cart­Item] ADD CONSTRAINT [FK\_­Shopping­Cart­Item\_­Product\_­Product­ID] FOREIGN KEY ([Product­ID]) REFERENCES [Production].[Product] ([Product­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Contains online customer orders until the order is submitted or cancelled.', 'SCHEMA', N'Sales', 'TABLE', N'Shopping­Cart­Item', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date the time the record was created.', 'SCHEMA', N'Sales', 'TABLE', N'Shopping­Cart­Item', 'COLUMN', N'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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product ordered. Foreign key to Product.Product­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Shopping­Cart­Item', 'COLUMN', N'Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Product quantity ordered.', 'SCHEMA', N'Sales', 'TABLE', N'Shopping­Cart­Item', 'COLUMN', N'Quantity'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Shopping cart identification number.', 'SCHEMA', N'Sales', 'TABLE', N'Shopping­Cart­Item', 'COLUMN', N'Shopping­Cart­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Shopping­Cart­Item records.', 'SCHEMA', N'Sales', 'TABLE', N'Shopping­Cart­Item', 'COLUMN', N'Shopping­Cart­Item­ID'  GO  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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Shopping­Cart­Item', 'CONSTRAINT', N'DF\_­Shopping­Cart­Item\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of 1', 'SCHEMA', N'Sales', 'TABLE', N'Shopping­Cart­Item', 'CONSTRAINT', N'DF\_­Shopping­Cart­Item\_­Quantity'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Product.Product­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Shopping­Cart­Item', 'CONSTRAINT', N'FK\_­Shopping­Cart­Item\_­Product\_­Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Sales', 'TABLE', N'Shopping­Cart­Item', 'CONSTRAINT', N'PK\_­Shopping­Cart­Item\_­Shopping­Cart­Item­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Sales', 'TABLE', N'Shopping­Cart­Item', 'INDEX', N'IX\_­Shopping­Cart­Item\_­Shopping­Cart­ID\_­Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'Shopping­Cart­Item', 'INDEX', N'PK\_­Shopping­Cart­Item\_­Shopping­Cart­Item­ID'  GO |

Uses

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

|  |
| --- |
| [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 |
|  | 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 |  |  |
|  | 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\_­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)) |

SQL Script

|  |
| --- |
| CREATE TABLE [Sales].[Special­Offer]  (  [Special­Offer­ID] [int] NOT NULL IDENTITY(1, 1),  [Description] [nvarchar] (255) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Discount­Pct] [smallmoney] NOT NULL CONSTRAINT [DF\_­Special­Offer\_­Discount­Pct] DEFAULT ((0.00)),  [Type] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Category] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Start­Date] [datetime] NOT NULL,  [End­Date] [datetime] NOT NULL,  [Min­Qty] [int] NOT NULL CONSTRAINT [DF\_­Special­Offer\_­Min­Qty] DEFAULT ((0)),  [Max­Qty] [int] NULL,  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Special­Offer\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Special­Offer\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Special­Offer] ADD CONSTRAINT [CK\_­Special­Offer\_­Discount­Pct] CHECK (([Discount­Pct]>=(0.00)))  GO  ALTER TABLE [Sales].[Special­Offer] ADD CONSTRAINT [CK\_­Special­Offer\_­End­Date] CHECK (([End­Date]>=[Start­Date]))  GO  ALTER TABLE [Sales].[Special­Offer] ADD CONSTRAINT [CK\_­Special­Offer\_­Max­Qty] CHECK (([Max­Qty]>=(0)))  GO  ALTER TABLE [Sales].[Special­Offer] ADD CONSTRAINT [CK\_­Special­Offer\_­Min­Qty] CHECK (([Min­Qty]>=(0)))  GO  ALTER TABLE [Sales].[Special­Offer] ADD CONSTRAINT [PK\_­Special­Offer\_­Special­Offer­ID] PRIMARY KEY CLUSTERED ([Special­Offer­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Special­Offer\_rowguid] ON [Sales].[Special­Offer] ([rowguid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Sale discounts lookup table.', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Group the discount applies to such as Reseller or Customer.', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer', 'COLUMN', N'Category'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Discount description.', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer', 'COLUMN', N'Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Discount precentage.', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer', 'COLUMN', N'Discount­Pct'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Discount end date.', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer', 'COLUMN', N'End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Maximum discount percent allowed.', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer', 'COLUMN', N'Max­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Minimum discount percent allowed.', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer', 'COLUMN', N'Min­Qty'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key for Special­Offer records.', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer', 'COLUMN', N'Special­Offer­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Discount start date.', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer', 'COLUMN', N'Start­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Discount type category.', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer', 'COLUMN', N'Type'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Discount­Pct] >= (0.00)', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer', 'CONSTRAINT', N'CK\_­Special­Offer\_­Discount­Pct'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [End­Date] >= [Start­Date]', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer', 'CONSTRAINT', N'CK\_­Special­Offer\_­End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Check constraint [Max­Qty] >= (0)', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer', 'CONSTRAINT', N'CK\_­Special­Offer\_­Max­Qty'  GO  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\_­Discount­Pct'  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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer', 'CONSTRAINT', N'DF\_­Special­Offer\_­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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer', 'CONSTRAINT', N'PK\_­Special­Offer\_­Special­Offer­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer', 'INDEX', N'AK\_­Special­Offer\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer', 'INDEX', N'PK\_­Special­Offer\_­Special­Offer­ID'  GO |

Uses

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

Used By

[[Sales].[Special­Offer­Product]](#8e4RMrihzXJLXw2gPdnNJ+2ijQ0=)

|  |
| --- |
| [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 |
|  | Special­Offer­ID Primary key for Special­Offer­Product records. | int | 4 | NOT NULL |  |
|  | 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\_­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 Foreign key constraint referencing Product.Product­ID. | Product­ID->[[Production].[Product].[Product­ID]](#mO6TpkyW7q4W/OMRHISuYfWNCME=) |
| FK\_­Special­Offer­Product\_­Special­Offer\_­Special­Offer­ID Foreign key constraint referencing Special­Offer.Special­Offer­ID. | Special­Offer­ID->[[Sales].[Special­Offer].[Special­Offer­ID]](#H28uErAJ2leiaB1YxFBqojYITRE=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Sales].[Special­Offer­Product]  (  [Special­Offer­ID] [int] NOT NULL,  [Product­ID] [int] NOT NULL,  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Special­Offer­Product\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Special­Offer­Product\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Special­Offer­Product] ADD CONSTRAINT [PK\_­Special­Offer­Product\_­Special­Offer­ID\_­Product­ID] PRIMARY KEY CLUSTERED ([Special­Offer­ID], [Product­ID]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Special­Offer­Product\_­Product­ID] ON [Sales].[Special­Offer­Product] ([Product­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Special­Offer­Product\_rowguid] ON [Sales].[Special­Offer­Product] ([rowguid]) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Special­Offer­Product] ADD CONSTRAINT [FK\_­Special­Offer­Product\_­Product\_­Product­ID] FOREIGN KEY ([Product­ID]) REFERENCES [Production].[Product] ([Product­ID])  GO  ALTER TABLE [Sales].[Special­Offer­Product] ADD CONSTRAINT [FK\_­Special­Offer­Product\_­Special­Offer\_­Special­Offer­ID] FOREIGN KEY ([Special­Offer­ID]) REFERENCES [Sales].[Special­Offer] ([Special­Offer­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Cross-reference table mapping products to special offer discounts.', 'SCHEMA', N'Sales', 'TABLE', N'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'Modified­Date'  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'Special­Offer­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer­Product', 'CONSTRAINT', N'DF\_­Special­Offer­Product\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer­Product', 'CONSTRAINT', N'DF\_­Special­Offer­Product\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Product.Product­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer­Product', 'CONSTRAINT', N'FK\_­Special­Offer­Product\_­Product\_­Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Special­Offer.Special­Offer­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer­Product', 'CONSTRAINT', N'FK\_­Special­Offer­Product\_­Special­Offer\_­Special­Offer­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer­Product', 'CONSTRAINT', N'PK\_­Special­Offer­Product\_­Special­Offer­ID\_­Product­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer­Product', 'INDEX', N'AK\_­Special­Offer­Product\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Sales', 'TABLE', N'Special­Offer­Product', 'INDEX', N'IX\_­Special­Offer­Product\_­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 |

Uses

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[[Sales].[Special­Offer]](#H28uErAJ2leiaB1YxFBqojYITRE=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

Used By

[[Sales].[Sales­Order­Detail]](#OkvyE7uLBgi7y2P/yCWzRf5c0wE=)

|  |
| --- |
| [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) | Nullability | Default |
|  | Business­Entity­ID Primary key. Foreign key to Customer.Business­Entity­ID. | int | 4 | NOT NULL |  |
|  | Name Name of the store. | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 | NOT NULL |  |
|  | Sales­Person­ID ID of the sales person assigned to the customer. Foreign key to Sales­Person.Business­Entity­ID. | int | 4 | NULL allowed |  |
|  | Demographics Demographic informationg about the store such as the number of employees, annual sales and store type. | [xml([Sales].[Store­Survey­Schema­Collection])](#Y25RTEvxjWoBOBvJpbWnAGVIQnU=) | 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 | (getdate()) |

Indexes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Key Columns | Type | Unique | XML Type |
|  | PK\_­Store\_­Business­Entity­ID Primary key (clustered) constraint | Business­Entity­ID |  | True |  |
|  | AK\_­Store\_rowguid Unique nonclustered index. Used to support replication samples. | rowguid |  | True |  |
|  | IX\_­Store\_­Sales­Person­ID Nonclustered index. | Sales­Person­ID |  |  |  |
|  | PXML\_­Store\_­Demographics Primary XML index. | Demographics | xml |  | Primary |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Store\_­Business­Entity\_­Business­Entity­ID Foreign key constraint referencing Business­Entity.Business­Entity­ID | Business­Entity­ID->[[Person].[Business­Entity].[Business­Entity­ID]](#I7t6VoTbrv7x5ZQ/kE/EZs1npdg=) |
| FK\_­Store\_­Sales­Person\_­Sales­Person­ID Foreign key constraint referencing Sales­Person.Sales­Person­ID | Sales­Person­ID->[[Sales].[Sales­Person].[Business­Entity­ID]](#hKN58zHn4ffskJv6ldcUqjf87do=) |

SQL Script

|  |
| --- |
| CREATE TABLE [Sales].[Store]  (  [Business­Entity­ID] [int] NOT NULL,  [Name] [dbo].[Name] NOT NULL,  [Sales­Person­ID] [int] NULL,  [Demographics] [xml] (CONTENT [Sales].[Store­Survey­Schema­Collection]) NULL,  [rowguid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_­Store\_rowguid] DEFAULT (newid()),  [Modified­Date] [datetime] NOT NULL CONSTRAINT [DF\_­Store\_­Modified­Date] DEFAULT (getdate())  ) ON [PRIMARY]  GO  ALTER TABLE [Sales].[Store] ADD CONSTRAINT [PK\_­Store\_­Business­Entity­ID] PRIMARY KEY CLUSTERED ([Business­Entity­ID]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [AK\_­Store\_rowguid] ON [Sales].[Store] ([rowguid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­Store\_­Sales­Person­ID] ON [Sales].[Store] ([Sales­Person­ID]) ON [PRIMARY]  GO  CREATE PRIMARY XML INDEX [PXML\_­Store\_­Demographics]  ON [Sales].[Store] ([Demographics])  GO  ALTER TABLE [Sales].[Store] ADD CONSTRAINT [FK\_­Store\_­Business­Entity\_­Business­Entity­ID] FOREIGN KEY ([Business­Entity­ID]) REFERENCES [Person].[Business­Entity] ([Business­Entity­ID])  GO  ALTER TABLE [Sales].[Store] ADD CONSTRAINT [FK\_­Store\_­Sales­Person\_­Sales­Person­ID] FOREIGN KEY ([Sales­Person­ID]) REFERENCES [Sales].[Sales­Person] ([Business­Entity­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Customers (resellers) of Adventure Works products.', 'SCHEMA', N'Sales', 'TABLE', N'Store', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key. Foreign key to Customer.Business­Entity­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'COLUMN', N'Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Demographic informationg about the store such as the number of employees, annual sales and store type.', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'COLUMN', N'Demographics'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date and time the record was last updated.', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Name of the store.', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'COLUMN', N'Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'COLUMN', N'rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'ID of the sales person assigned to the customer. Foreign key to Sales­Person.Business­Entity­ID.', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'COLUMN', N'Sales­Person­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of GETDATE()', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'CONSTRAINT', N'DF\_­Store\_­Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Default constraint value of NEWID()', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'CONSTRAINT', N'DF\_­Store\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Business­Entity.Business­Entity­ID', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'CONSTRAINT', N'FK\_­Store\_­Business­Entity\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Foreign key constraint referencing Sales­Person.Sales­Person­ID', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'CONSTRAINT', N'FK\_­Store\_­Sales­Person\_­Sales­Person­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary key (clustered) constraint', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'CONSTRAINT', N'PK\_­Store\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Unique nonclustered index. Used to support replication samples.', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'INDEX', N'AK\_­Store\_rowguid'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Nonclustered index.', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'INDEX', N'IX\_­Store\_­Sales­Person­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index created by a primary key constraint.', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'INDEX', N'PK\_­Store\_­Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary XML index.', 'SCHEMA', N'Sales', 'TABLE', N'Store', 'INDEX', N'PXML\_­Store\_­Demographics'  GO |

Uses

[[Person].[Business­Entity]](#I7t6VoTbrv7x5ZQ/kE/EZs1npdg=)

[[Sales].[Sales­Person]](#hKN58zHn4ffskJv6ldcUqjf87do=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

[[Sales].[Store­Survey­Schema­Collection]](#Y25RTEvxjWoBOBvJpbWnAGVIQnU=)

Used By

[[Sales].[Customer]](#UK53WWAjbLvEqOKY0KByf769x9M=)

[[Sales].[v­Store­With­Addresses]](#lyHdOUZeccdbPaj16bKzv2RjJKw=)

[[Sales].[v­Store­With­Contacts]](#HCZuGdamTtZV4FxWl+4pACIsexM=)

[[Sales].[v­Store­With­Demographics]](#Y5BMY9GzFlSsci5vrMo/PYvMfsM=)

[[dbo].[ufn­Get­Contact­Information]](#MAbUSn/1caUuJ7XN6dczcbTRRyA=)

|  |
| --- |
| Views |

Objects

|  |
| --- |
| Name |
| [Human­Resources.v­Employee](#TVXJJHSO4vPDsqDtNRd5tGQdsO0=) Employee names and addresses. |
| [Human­Resources.v­Employee­Department](#JhbW+DYcE93jITPkfvVrtIWEHzE=) Returns employee name, title, and current department. |
| [Human­Resources.v­Employee­Department­History](#Ppb9efXEcRDxKIGzLbHZkOB+Nto=) Returns employee name and current and previous departments. |
| [Human­Resources.v­Job­Candidate](#OJ+5x6Pf56todcFiq0tOA8QJbeA=) Job candidate names and resumes. |
| [Human­Resources.v­Job­Candidate­Education](#L3qRlBiBKB45IvcTjpNzlK4Zu0c=) 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. |
| [Human­Resources.v­Job­Candidate­Employment](#xDdPFCC7cBQwDROim4Heve+/Okw=) Displays the content from each employement history related element in the xml column Resume in the Human­Resources.Job­Candidate table. The content has been localized into French, Simplified Chinese and Thai. Some data may not display correctly unless supplemental language support is installed. |
| [Person.v­Additional­Contact­Info](#VDCcGn5Gs1OPJIG7Ywjk/m2DaqY=) Displays the contact name and content from each element in the xml column Additional­Contact­Info for that person. |
| [Person.v­State­Province­Country­Region](#9RXeFj8QLY2V2kjrQ9Nq0mUcx70=) Joins State­Province table with Country­Region table. |
| [Production.v­Product­And­Description](#2wtLw+LERdCNthBlev7/7Ew1+JQ=) Product names and descriptions. Product descriptions are provided in multiple languages. |
| [Production.v­Product­Model­Catalog­Description](#j53/ggQeAT7efkm5uXeLRIbSppI=) 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](#GCJbY/fUa640Nyz3prLsQye2L+M=) Displays the content from each element in the xml column Instructions for each product in the Production.Product­Model table that has manufacturing instructions. |
| [Purchasing.v­Vendor­With­Addresses](#NdlQ62Q6DZw3GS30J9H9dm/SOqg=) Vendor (company) names and addresses . |
| [Purchasing.v­Vendor­With­Contacts](#kSLkxLBh/Shb6NobziJenpOvdFo=) Vendor (company) names and the names of vendor employees to contact. |
| [Sales.v­Individual­Customer](#WcBS54CxROpNMbufsyU5V6t9Ykw=) Individual customers (names and addresses) that purchase Adventure Works Cycles products online. |
| [Sales.v­Person­Demographics](#rOUy+E6WBf1AY40+Q9SudpuP9xU=) Displays the content from each element in the xml column Demographics for each customer in the Person.Person table. |
| [Sales.v­Sales­Person](#pOjFS54tNwjD3Yvi8IMnn7dtRuA=) Sales representiatives (names and addresses) and their sales-related information. |
| [Sales.v­Sales­Person­Sales­By­Fiscal­Years](#GfF5ZKdsT1+xL6X5ks+ghnXIfng=) Uses PIVOT to return aggregated sales information for each sales representative. |
| [Sales.v­Store­With­Addresses](#lyHdOUZeccdbPaj16bKzv2RjJKw=) Stores (including store addresses) that sell Adventure Works Cycles products to consumers. |
| [Sales.v­Store­With­Contacts](#HCZuGdamTtZV4FxWl+4pACIsexM=) Stores (including store contacts) that sell Adventure Works Cycles products to consumers. |
| [Sales.v­Store­With­Demographics](#Y5BMY9GzFlSsci5vrMo/PYvMfsM=) 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]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Middle­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Last­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Suffix | nvarchar(10) | 20 |
| Job­Title | nvarchar(50) | 100 |
| Phone­Number | [[dbo].[Phone]](#Dz5dpqul0rw8eU0/JuKLh0EO1ek=) | 50 |
| Phone­Number­Type | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Email­Address | nvarchar(50) | 100 |
| Email­Promotion | int | 4 |
| Address­Line1 | nvarchar(60) | 120 |
| Address­Line2 | nvarchar(60) | 120 |
| City | nvarchar(30) | 60 |
| State­Province­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Postal­Code | nvarchar(15) | 30 |
| Country­Region­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Additional­Contact­Info | xml | max |

SQL Script

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

Uses

[[Human­Resources].[Employee]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=)

[[Person].[Address]](#YHZkNb8Ts04ByOupF/OAfoeI2qk=)

[[Person].[Business­Entity­Address]](#HK6CehnfZ4YYh7ckP8cx74El9pY=)

[[Person].[Country­Region]](#efOGmVlBeofSPEY2jc0F91joV9Q=)

[[Person].[Email­Address]](#3EL9ymWvglf8ADJZjk1nQ8j/hLY=)

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[[Person].[Person­Phone]](#NyzXUZBpBs0rFeuVHGsZlpmRP0Y=)

[[Person].[Phone­Number­Type]](#05nCels02cq7i+PWSDjMFGEmWFU=)

[[Person].[State­Province]](#fYuToAkdHMSws7pXek8EIcFTfvY=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[[dbo].[Phone]](#Dz5dpqul0rw8eU0/JuKLh0EO1ek=)

[Human­Resources](#wRNBi9QOObVjEO+Wxjr4GvvwBBs=)

|  |
| --- |
| [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]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Middle­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Last­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Suffix | nvarchar(10) | 20 |
| Job­Title | nvarchar(50) | 100 |
| Department | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Group­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Start­Date | date | 3 |

SQL Script

|  |
| --- |
| CREATE VIEW [Human­Resources].[v­Employee­Department]  AS  SELECT  e.[Business­Entity­ID]  ,p.[Title]  ,p.[First­Name]  ,p.[Middle­Name]  ,p.[Last­Name]  ,p.[Suffix]  ,e.[Job­Title]  ,d.[Name] AS [Department]  ,d.[Group­Name]  ,edh.[Start­Date]  FROM [Human­Resources].[Employee] e  INNER JOIN [Person].[Person] p  ON p.[Business­Entity­ID] = e.[Business­Entity­ID]  INNER JOIN [Human­Resources].[Employee­Department­History] edh  ON e.[Business­Entity­ID] = edh.[Business­Entity­ID]  INNER JOIN [Human­Resources].[Department] d  ON edh.[Department­ID] = d.[Department­ID]  WHERE edh.End­Date IS NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Returns employee name, title, and current department.', 'SCHEMA', N'Human­Resources', 'VIEW', N'v­Employee­Department', NULL, NULL  GO |

Uses

[[Human­Resources].[Department]](#TH5R06keykSJYajBUWfR3phTY8s=)

[[Human­Resources].[Employee]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=)

[[Human­Resources].[Employee­Department­History]](#tDjny2Q9sWvuqg97wWzd145nXkA=)

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Human­Resources](#wRNBi9QOObVjEO+Wxjr4GvvwBBs=)

|  |
| --- |
| [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]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Middle­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Last­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Suffix | nvarchar(10) | 20 |
| Shift | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Department | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Group­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Start­Date | date | 3 |
| End­Date | date | 3 |

SQL Script

|  |
| --- |
| CREATE VIEW [Human­Resources].[v­Employee­Department­History]  AS  SELECT  e.[Business­Entity­ID]  ,p.[Title]  ,p.[First­Name]  ,p.[Middle­Name]  ,p.[Last­Name]  ,p.[Suffix]  ,s.[Name] AS [Shift]  ,d.[Name] AS [Department]  ,d.[Group­Name]  ,edh.[Start­Date]  ,edh.[End­Date]  FROM [Human­Resources].[Employee] e  INNER JOIN [Person].[Person] p  ON p.[Business­Entity­ID] = e.[Business­Entity­ID]  INNER JOIN [Human­Resources].[Employee­Department­History] edh  ON e.[Business­Entity­ID] = edh.[Business­Entity­ID]  INNER JOIN [Human­Resources].[Department] d  ON edh.[Department­ID] = d.[Department­ID]  INNER JOIN [Human­Resources].[Shift] s  ON s.[Shift­ID] = edh.[Shift­ID];  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Returns employee name and current and previous departments.', 'SCHEMA', N'Human­Resources', 'VIEW', N'v­Employee­Department­History', NULL, NULL  GO |

Uses

[[Human­Resources].[Department]](#TH5R06keykSJYajBUWfR3phTY8s=)

[[Human­Resources].[Employee]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=)

[[Human­Resources].[Employee­Department­History]](#tDjny2Q9sWvuqg97wWzd145nXkA=)

[[Human­Resources].[Shift]](#oOQXice18uca5/e1dhJkYUmlNt0=)

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Human­Resources](#wRNBi9QOObVjEO+Wxjr4GvvwBBs=)

|  |
| --- |
| [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 |  |

SQL Script

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

Uses

[[Human­Resources].[Job­Candidate]](#WVRXAhzXWcgHQbdlg570h3Fv6QI=)

[Human­Resources](#wRNBi9QOObVjEO+Wxjr4GvvwBBs=)

|  |
| --- |
| [Human­Resources].[v­Job­Candidate­Education] |

MS\_­Description

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

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| ANSI Nulls On | True |
| Quoted Identifier On | True |
| Created | 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 |  |

SQL Script

|  |
| --- |
| CREATE VIEW [Human­Resources].[v­Job­Candidate­Education]  AS  SELECT  jc.[Job­Candidate­ID]  ,[Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Edu.Level)[1]', 'nvarchar(max)') AS [Edu.Level]  ,CONVERT(datetime, REPLACE([Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Edu.Start­Date)[1]', 'nvarchar(20)') ,'Z', ''), 101) AS [Edu.Start­Date]  ,CONVERT(datetime, REPLACE([Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Edu.End­Date)[1]', 'nvarchar(20)') ,'Z', ''), 101) AS [Edu.End­Date]  ,[Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Edu.Degree)[1]', 'nvarchar(50)') AS [Edu.Degree]  ,[Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Edu.Major)[1]', 'nvarchar(50)') AS [Edu.Major]  ,[Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Edu.Minor)[1]', 'nvarchar(50)') AS [Edu.Minor]  ,[Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Edu.GPA)[1]', 'nvarchar(5)') AS [Edu.GPA]  ,[Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Edu.GPAScale)[1]', 'nvarchar(5)') AS [Edu.GPAScale]  ,[Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Edu.School)[1]', 'nvarchar(100)') AS [Edu.School]  ,[Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Edu.Location/Location/Loc.Country­Region)[1]', 'nvarchar(100)') AS [Edu.Loc.Country­Region]  ,[Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Edu.Location/Location/Loc.State)[1]', 'nvarchar(100)') AS [Edu.Loc.State]  ,[Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Edu.Location/Location/Loc.City)[1]', 'nvarchar(100)') AS [Edu.Loc.City]  FROM [Human­Resources].[Job­Candidate] jc  CROSS APPLY jc.[Resume].nodes(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  /Resume/Education') AS [Education](ref);  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Displays the content from each education related element in the xml column Resume in the Human­Resources.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 |

Uses

[[Human­Resources].[Job­Candidate]](#WVRXAhzXWcgHQbdlg570h3Fv6QI=)

[Human­Resources](#wRNBi9QOObVjEO+Wxjr4GvvwBBs=)

|  |
| --- |
| [Human­Resources].[v­Job­Candidate­Employment] |

MS\_­Description

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

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 |  |

SQL Script

|  |
| --- |
| CREATE VIEW [Human­Resources].[v­Job­Candidate­Employment]  AS  SELECT  jc.[Job­Candidate­ID]  ,CONVERT(datetime, REPLACE([Employment].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Emp.Start­Date)[1]', 'nvarchar(20)') ,'Z', ''), 101) AS [Emp.Start­Date]  ,CONVERT(datetime, REPLACE([Employment].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Emp.End­Date)[1]', 'nvarchar(20)') ,'Z', ''), 101) AS [Emp.End­Date]  ,[Employment].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Emp.Org­Name)[1]', 'nvarchar(100)') AS [Emp.Org­Name]  ,[Employment].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Emp.Job­Title)[1]', 'nvarchar(100)') AS [Emp.Job­Title]  ,[Employment].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Emp.Responsibility)[1]', 'nvarchar(max)') AS [Emp.Responsibility]  ,[Employment].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Emp.Function­Category)[1]', 'nvarchar(max)') AS [Emp.Function­Category]  ,[Employment].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Emp.Industry­Category)[1]', 'nvarchar(max)') AS [Emp.Industry­Category]  ,[Employment].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Emp.Location/Location/Loc.Country­Region)[1]', 'nvarchar(max)') AS [Emp.Loc.Country­Region]  ,[Employment].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Emp.Location/Location/Loc.State)[1]', 'nvarchar(max)') AS [Emp.Loc.State]  ,[Employment].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  (Emp.Location/Location/Loc.City)[1]', 'nvarchar(max)') AS [Emp.Loc.City]  FROM [Human­Resources].[Job­Candidate] jc  CROSS APPLY jc.[Resume].nodes(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";  /Resume/Employment') AS Employment(ref);  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Displays the content from each employement history related element in the xml column Resume in the Human­Resources.Job­Candidate table. The content has been localized into French, Simplified Chinese and Thai. Some data may not display correctly unless supplemental language support is installed.', 'SCHEMA', N'Human­Resources', 'VIEW', N'v­Job­Candidate­Employment', NULL, NULL  GO |

Uses

[[Human­Resources].[Job­Candidate]](#WVRXAhzXWcgHQbdlg570h3Fv6QI=)

[Human­Resources](#wRNBi9QOObVjEO+Wxjr4GvvwBBs=)

|  |
| --- |
| [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]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Middle­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Last­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 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 [State­Province]  ,[Contact­Info].ref.value(N'declare namespace ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact­Info";  declare namespace act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact­Types";  (act:home­Postal­Address/act:Postal­Code)[1]', 'nvarchar(50)') AS [Postal­Code]  ,[Contact­Info].ref.value(N'declare namespace ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact­Info";  declare namespace act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact­Types";  (act:home­Postal­Address/act:Country­Region)[1]', 'nvarchar(50)') AS [Country­Region]  ,[Contact­Info].ref.value(N'declare namespace ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact­Info";  declare namespace act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact­Types";  (act:home­Postal­Address/act:Special­Instructions/text())[1]', 'nvarchar(max)') AS [Home­Address­Special­Instructions]  ,[Contact­Info].ref.value(N'declare namespace ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact­Info";  declare namespace act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact­Types";  (act:e­Mail/act:e­Mail­Address)[1]', 'nvarchar(128)') AS [EMail­Address]  ,LTRIM(RTRIM([Contact­Info].ref.value(N'declare namespace ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact­Info";  declare namespace act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact­Types";  (act:e­Mail/act:Special­Instructions/text())[1]', 'nvarchar(max)'))) AS [EMail­Special­Instructions]  ,[Contact­Info].ref.value(N'declare namespace ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact­Info";  declare namespace act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact­Types";  (act:e­Mail/act:Special­Instructions/act:telephone­Number/act:number)[1]', 'nvarchar(50)') AS [EMail­Telephone­Number]  ,[rowguid]  ,[Modified­Date]  FROM [Person].[Person]  OUTER APPLY [Additional­Contact­Info].nodes(  'declare namespace ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact­Info";  /ci:Additional­Contact­Info') AS Contact­Info(ref)  WHERE [Additional­Contact­Info] IS NOT NULL;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Displays the contact name and content from each element in the xml column Additional­Contact­Info for that person.', 'SCHEMA', N'Person', 'VIEW', N'v­Additional­Contact­Info', NULL, NULL  GO |

Uses

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Person](#GSXHhzVRNjGQYtnCQHosa2uthyw=)

|  |
| --- |
| [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) |
|  | State­Province­ID | int | 4 |
|  | State­Province­Code | nchar(3) | 6 |
|  | Is­Only­State­Province­Flag | [[dbo].[Flag]](#j88jjDtwCbK8khz/m8jxryH/SFk=) | 1 |
|  | State­Province­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
|  | Territory­ID | int | 4 |
|  | Country­Region­Code | nvarchar(3) | 6 |
|  | Country­Region­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | IX\_v­State­Province­Country­Region Clustered index on the view v­State­Province­Country­Region. | State­Province­ID, Country­Region­Code | True |

SQL Script

|  |
| --- |
| CREATE VIEW [Person].[v­State­Province­Country­Region]  WITH SCHEMABINDING  AS  SELECT  sp.[State­Province­ID]  ,sp.[State­Province­Code]  ,sp.[Is­Only­State­Province­Flag]  ,sp.[Name] AS [State­Province­Name]  ,sp.[Territory­ID]  ,cr.[Country­Region­Code]  ,cr.[Name] AS [Country­Region­Name]  FROM [Person].[State­Province] sp  INNER JOIN [Person].[Country­Region] cr  ON sp.[Country­Region­Code] = cr.[Country­Region­Code];  GO  CREATE UNIQUE CLUSTERED INDEX [IX\_v­State­Province­Country­Region] ON [Person].[v­State­Province­Country­Region] ([State­Province­ID], [Country­Region­Code]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Joins State­Province table with Country­Region table.', 'SCHEMA', N'Person', 'VIEW', N'v­State­Province­Country­Region', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Clustered index on the view v­State­Province­Country­Region.', 'SCHEMA', N'Person', 'VIEW', N'v­State­Province­Country­Region', 'INDEX', N'IX\_v­State­Province­Country­Region'  GO |

Uses

[[Person].[Country­Region]](#efOGmVlBeofSPEY2jc0F91joV9Q=)

[[Person].[State­Province]](#fYuToAkdHMSws7pXek8EIcFTfvY=)

[[dbo].[Flag]](#j88jjDtwCbK8khz/m8jxryH/SFk=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Person](#GSXHhzVRNjGQYtnCQHosa2uthyw=)

|  |
| --- |
| [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) |
|  | Product­ID | int | 4 |
|  | Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
|  | Product­Model | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
|  | 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 |

SQL Script

|  |
| --- |
| CREATE VIEW [Production].[v­Product­And­Description]  WITH SCHEMABINDING  AS  -- View (indexed or standard) to display products and product descriptions by language.  SELECT  p.[Product­ID]  ,p.[Name]  ,pm.[Name] AS [Product­Model]  ,pmx.[Culture­ID]  ,pd.[Description]  FROM [Production].[Product] p  INNER JOIN [Production].[Product­Model] pm  ON p.[Product­Model­ID] = pm.[Product­Model­ID]  INNER JOIN [Production].[Product­Model­Product­Description­Culture] pmx  ON pm.[Product­Model­ID] = pmx.[Product­Model­ID]  INNER JOIN [Production].[Product­Description] pd  ON pmx.[Product­Description­ID] = pd.[Product­Description­ID];  GO  CREATE UNIQUE CLUSTERED INDEX [IX\_v­Product­And­Description] ON [Production].[v­Product­And­Description] ([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 v­Product­And­Description.', 'SCHEMA', N'Production', 'VIEW', N'v­Product­And­Description', 'INDEX', N'IX\_v­Product­And­Description'  GO |

Uses

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[[Production].[Product­Description]](#4iEXAFuDYt/4TBUTP9DP8dTGazY=)

[[Production].[Product­Model]](#n9VdHBX3Ji5OWfmMzrHHD+J8HSM=)

[[Production].[Product­Model­Product­Description­Culture]](#Kkf63jUYCrQKelqjjcG51JBQDNI=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

|  |
| --- |
| [Production].[v­Product­Model­Catalog­Description] |

MS\_­Description

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

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| ANSI Nulls On | True |
| Quoted Identifier On | True |
| Created | 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]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |  |
| Summary | nvarchar(max) | max |  |
| Manufacturer | nvarchar(max) | max |  |
| Copyright | nvarchar(30) | 60 |  |
| Product­URL | nvarchar(256) | 512 |  |
| Warranty­Period | nvarchar(256) | 512 |  |
| Warranty­Description | nvarchar(256) | 512 |  |
| No­Of­Years | nvarchar(256) | 512 |  |
| Maintenance­Description | nvarchar(256) | 512 |  |
| Wheel | nvarchar(256) | 512 |  |
| Saddle | nvarchar(256) | 512 |  |
| Pedal | nvarchar(256) | 512 |  |
| Bike­Frame | nvarchar(max) | max |  |
| Crankset | nvarchar(256) | 512 |  |
| Picture­Angle | nvarchar(256) | 512 |  |
| Picture­Size | nvarchar(256) | 512 |  |
| Product­Photo­ID | nvarchar(256) | 512 |  |
| Material | nvarchar(256) | 512 |  |
| Color | nvarchar(256) | 512 |  |
| Product­Line | nvarchar(256) | 512 |  |
| Style | nvarchar(256) | 512 |  |
| Rider­Experience | nvarchar(1024) | 2048 |  |
| rowguid | uniqueidentifier | 16 |  |
| Modified­Date | datetime | 8 |  |

SQL Script

|  |
| --- |
| CREATE VIEW [Production].[v­Product­Model­Catalog­Description]  AS  SELECT  [Product­Model­ID]  ,[Name]  ,[Catalog­Description].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Description";  declare namespace html="http://www.w3.org/1999/xhtml";  (/p1:Product­Description/p1:Summary/html:p)[1]', 'nvarchar(max)') AS [Summary]  ,[Catalog­Description].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Description";  (/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/Product­Model­Description";  (/p1:Product­Description/p1:Manufacturer/p1:Copyright)[1]', 'nvarchar(30)') AS [Copyright]  ,[Catalog­Description].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Description";  (/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/Product­Model­Description";  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 p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Description";  declare namespace wm="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Warr­And­Main";  (/p1:Product­Description/p1:Features/wm:Warranty/wm:Description)[1]', 'nvarchar(256)') AS [Warranty­Description]  ,[Catalog­Description].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Description";  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 p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Description";  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/Product­Model­Description";  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/Product­Model­Description";  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/Product­Model­Description";  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/Product­Model­Description";  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/Product­Model­Description";  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/Product­Model­Description";  (/p1:Product­Description/p1:Picture/p1:Angle)[1]', 'nvarchar(256)') AS [Picture­Angle]  ,[Catalog­Description].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Description";  (/p1:Product­Description/p1:Picture/p1:Size)[1]', 'nvarchar(256)') AS [Picture­Size]  ,[Catalog­Description].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Description";  (/p1:Product­Description/p1:Picture/p1:Product­Photo­ID)[1]', 'nvarchar(256)') AS [Product­Photo­ID]  ,[Catalog­Description].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Description";  (/p1:Product­Description/p1:Specifications/Material)[1]', 'nvarchar(256)') AS [Material]  ,[Catalog­Description].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Description";  (/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/Product­Model­Description";  (/p1:Product­Description/p1:Specifications/Product­Line)[1]', 'nvarchar(256)') AS [Product­Line]  ,[Catalog­Description].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Description";  (/p1:Product­Description/p1:Specifications/Style)[1]', 'nvarchar(256)') AS [Style]  ,[Catalog­Description].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Description";  (/p1:Product­Description/p1:Specifications/Rider­Experience)[1]', 'nvarchar(1024)') AS [Rider­Experience]  ,[rowguid]  ,[Modified­Date]  FROM [Production].[Product­Model]  WHERE [Catalog­Description] IS NOT NULL;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Displays the content from each element in the xml column Catalog­Description for each product in the Production.Product­Model table that has catalog data.', 'SCHEMA', N'Production', 'VIEW', N'v­Product­Model­Catalog­Description', NULL, NULL  GO |

Uses

[[Production].[Product­Model]](#n9VdHBX3Ji5OWfmMzrHHD+J8HSM=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

|  |
| --- |
| [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]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |  |
| Instructions | nvarchar(max) | max |  |
| Location­ID | int | 4 |  |
| Setup­Hours | decimal(9,4) | 5 |  |
| Machine­Hours | decimal(9,4) | 5 |  |
| Labor­Hours | decimal(9,4) | 5 |  |
| Lot­Size | int | 4 |  |
| Step | nvarchar(1024) | 2048 |  |
| rowguid | uniqueidentifier | 16 |  |
| Modified­Date | datetime | 8 |  |

SQL Script

|  |
| --- |
| CREATE VIEW [Production].[v­Product­Model­Instructions]  AS  SELECT  [Product­Model­ID]  ,[Name]  ,[Instructions].value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Manu­Instructions";  (/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 "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Manu­Instructions";  /root/Location') Mfg­Instructions(ref)  CROSS APPLY [Mfg­Instructions].ref.nodes('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Manu­Instructions";  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'v­Product­Model­Instructions', NULL, NULL  GO |

Uses

[[Production].[Product­Model]](#n9VdHBX3Ji5OWfmMzrHHD+J8HSM=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

|  |
| --- |
| [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]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Address­Type | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Address­Line1 | nvarchar(60) | 120 |
| Address­Line2 | nvarchar(60) | 120 |
| City | nvarchar(30) | 60 |
| State­Province­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Postal­Code | nvarchar(15) | 30 |
| Country­Region­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |

SQL Script

|  |
| --- |
| CREATE VIEW [Purchasing].[v­Vendor­With­Addresses] AS  SELECT  v.[Business­Entity­ID]  ,v.[Name]  ,at.[Name] AS [Address­Type]  ,a.[Address­Line1]  ,a.[Address­Line2]  ,a.[City]  ,sp.[Name] AS [State­Province­Name]  ,a.[Postal­Code]  ,cr.[Name] AS [Country­Region­Name]  FROM [Purchasing].[Vendor] v  INNER JOIN [Person].[Business­Entity­Address] bea  ON bea.[Business­Entity­ID] = v.[Business­Entity­ID]  INNER JOIN [Person].[Address] a  ON a.[Address­ID] = bea.[Address­ID]  INNER JOIN [Person].[State­Province] sp  ON sp.[State­Province­ID] = a.[State­Province­ID]  INNER JOIN [Person].[Country­Region] cr  ON cr.[Country­Region­Code] = sp.[Country­Region­Code]  INNER JOIN [Person].[Address­Type] at  ON at.[Address­Type­ID] = bea.[Address­Type­ID];  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Vendor (company) names and addresses .', 'SCHEMA', N'Purchasing', 'VIEW', N'v­Vendor­With­Addresses', NULL, NULL  GO |

Uses

[[Person].[Address]](#YHZkNb8Ts04ByOupF/OAfoeI2qk=)

[[Person].[Address­Type]](#sC112VucbUsBFhIBrJnCpuBv1os=)

[[Person].[Business­Entity­Address]](#HK6CehnfZ4YYh7ckP8cx74El9pY=)

[[Person].[Country­Region]](#efOGmVlBeofSPEY2jc0F91joV9Q=)

[[Person].[State­Province]](#fYuToAkdHMSws7pXek8EIcFTfvY=)

[[Purchasing].[Vendor]](#yfc9dLywcQtxtbztLjV6J3bb0tM=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Purchasing](#r48uLAuwsJTIAvAuIPBTbVrjT9s=)

|  |
| --- |
| [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]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Contact­Type | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Title | nvarchar(8) | 16 |
| First­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Middle­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Last­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Suffix | nvarchar(10) | 20 |
| Phone­Number | [[dbo].[Phone]](#Dz5dpqul0rw8eU0/JuKLh0EO1ek=) | 50 |
| Phone­Number­Type | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Email­Address | nvarchar(50) | 100 |
| Email­Promotion | int | 4 |

SQL Script

|  |
| --- |
| CREATE VIEW [Purchasing].[v­Vendor­With­Contacts] AS  SELECT  v.[Business­Entity­ID]  ,v.[Name]  ,ct.[Name] AS [Contact­Type]  ,p.[Title]  ,p.[First­Name]  ,p.[Middle­Name]  ,p.[Last­Name]  ,p.[Suffix]  ,pp.[Phone­Number]  ,pnt.[Name] AS [Phone­Number­Type]  ,ea.[Email­Address]  ,p.[Email­Promotion]  FROM [Purchasing].[Vendor] v  INNER JOIN [Person].[Business­Entity­Contact] bec  ON bec.[Business­Entity­ID] = v.[Business­Entity­ID]  INNER JOIN [Person].Contact­Type ct  ON ct.[Contact­Type­ID] = bec.[Contact­Type­ID]  INNER JOIN [Person].[Person] p  ON p.[Business­Entity­ID] = bec.[Person­ID]  LEFT OUTER JOIN [Person].[Email­Address] ea  ON ea.[Business­Entity­ID] = p.[Business­Entity­ID]  LEFT OUTER JOIN [Person].[Person­Phone] pp  ON pp.[Business­Entity­ID] = p.[Business­Entity­ID]  LEFT OUTER JOIN [Person].[Phone­Number­Type] pnt  ON pnt.[Phone­Number­Type­ID] = pp.[Phone­Number­Type­ID];  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Vendor (company) names and the names of vendor employees to contact.', 'SCHEMA', N'Purchasing', 'VIEW', N'v­Vendor­With­Contacts', NULL, NULL  GO |

Uses

[[Person].[Business­Entity­Contact]](#9Zj2hWeFVzGQn5EUQXP6lLMtAZI=)

[[Person].[Contact­Type]](#uyTva1snlxlEidD4b0wN1HQzMw4=)

[[Person].[Email­Address]](#3EL9ymWvglf8ADJZjk1nQ8j/hLY=)

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[[Person].[Person­Phone]](#NyzXUZBpBs0rFeuVHGsZlpmRP0Y=)

[[Person].[Phone­Number­Type]](#05nCels02cq7i+PWSDjMFGEmWFU=)

[[Purchasing].[Vendor]](#yfc9dLywcQtxtbztLjV6J3bb0tM=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[[dbo].[Phone]](#Dz5dpqul0rw8eU0/JuKLh0EO1ek=)

[Purchasing](#r48uLAuwsJTIAvAuIPBTbVrjT9s=)

|  |
| --- |
| [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]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Middle­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Last­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Suffix | nvarchar(10) | 20 |
| Phone­Number | [[dbo].[Phone]](#Dz5dpqul0rw8eU0/JuKLh0EO1ek=) | 50 |
| Phone­Number­Type | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Email­Address | nvarchar(50) | 100 |
| Email­Promotion | int | 4 |
| Address­Type | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Address­Line1 | nvarchar(60) | 120 |
| Address­Line2 | nvarchar(60) | 120 |
| City | nvarchar(30) | 60 |
| State­Province­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Postal­Code | nvarchar(15) | 30 |
| Country­Region­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Demographics | xml | max |

SQL Script

|  |
| --- |
| CREATE VIEW [Sales].[v­Individual­Customer]  AS  SELECT  p.[Business­Entity­ID]  ,p.[Title]  ,p.[First­Name]  ,p.[Middle­Name]  ,p.[Last­Name]  ,p.[Suffix]  ,pp.[Phone­Number]  ,pnt.[Name] AS [Phone­Number­Type]  ,ea.[Email­Address]  ,p.[Email­Promotion]  ,at.[Name] AS [Address­Type]  ,a.[Address­Line1]  ,a.[Address­Line2]  ,a.[City]  ,[State­Province­Name] = sp.[Name]  ,a.[Postal­Code]  ,[Country­Region­Name] = cr.[Name]  ,p.[Demographics]  FROM [Person].[Person] p  INNER JOIN [Person].[Business­Entity­Address] bea  ON bea.[Business­Entity­ID] = p.[Business­Entity­ID]  INNER JOIN [Person].[Address] a  ON a.[Address­ID] = bea.[Address­ID]  INNER JOIN [Person].[State­Province] sp  ON sp.[State­Province­ID] = a.[State­Province­ID]  INNER JOIN [Person].[Country­Region] cr  ON cr.[Country­Region­Code] = sp.[Country­Region­Code]  INNER JOIN [Person].[Address­Type] at  ON at.[Address­Type­ID] = bea.[Address­Type­ID]  INNER JOIN [Sales].[Customer] c  ON c.[Person­ID] = p.[Business­Entity­ID]  LEFT OUTER JOIN [Person].[Email­Address] ea  ON ea.[Business­Entity­ID] = p.[Business­Entity­ID]  LEFT OUTER JOIN [Person].[Person­Phone] pp  ON pp.[Business­Entity­ID] = p.[Business­Entity­ID]  LEFT OUTER JOIN [Person].[Phone­Number­Type] pnt  ON pnt.[Phone­Number­Type­ID] = pp.[Phone­Number­Type­ID]  WHERE c.Store­ID IS NULL;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Individual customers (names and addresses) that purchase Adventure Works Cycles products online.', 'SCHEMA', N'Sales', 'VIEW', N'v­Individual­Customer', NULL, NULL  GO |

Uses

[[Person].[Address]](#YHZkNb8Ts04ByOupF/OAfoeI2qk=)

[[Person].[Address­Type]](#sC112VucbUsBFhIBrJnCpuBv1os=)

[[Person].[Business­Entity­Address]](#HK6CehnfZ4YYh7ckP8cx74El9pY=)

[[Person].[Country­Region]](#efOGmVlBeofSPEY2jc0F91joV9Q=)

[[Person].[Email­Address]](#3EL9ymWvglf8ADJZjk1nQ8j/hLY=)

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[[Person].[Person­Phone]](#NyzXUZBpBs0rFeuVHGsZlpmRP0Y=)

[[Person].[Phone­Number­Type]](#05nCels02cq7i+PWSDjMFGEmWFU=)

[[Person].[State­Province]](#fYuToAkdHMSws7pXek8EIcFTfvY=)

[[Sales].[Customer]](#UK53WWAjbLvEqOKY0KByf769x9M=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[[dbo].[Phone]](#Dz5dpqul0rw8eU0/JuKLh0EO1ek=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

|  |
| --- |
| [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 |

SQL Script

|  |
| --- |
| CREATE VIEW [Sales].[v­Person­Demographics]  AS  SELECT  p.[Business­Entity­ID]  ,[Individual­Survey].[ref].[value](N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Individual­Survey";  Total­Purchase­YTD[1]', 'money') AS [Total­Purchase­YTD]  ,CONVERT(datetime, REPLACE([Individual­Survey].[ref].[value](N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Individual­Survey";  Date­First­Purchase[1]', 'nvarchar(20)') ,'Z', ''), 101) AS [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;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Displays the content from each element in the xml column Demographics for each customer in the Person.Person table.', 'SCHEMA', N'Sales', 'VIEW', N'v­Person­Demographics', NULL, NULL  GO |

Uses

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

|  |
| --- |
| [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]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Middle­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Last­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Suffix | nvarchar(10) | 20 |
| Job­Title | nvarchar(50) | 100 |
| Phone­Number | [[dbo].[Phone]](#Dz5dpqul0rw8eU0/JuKLh0EO1ek=) | 50 |
| Phone­Number­Type | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Email­Address | nvarchar(50) | 100 |
| Email­Promotion | int | 4 |
| Address­Line1 | nvarchar(60) | 120 |
| Address­Line2 | nvarchar(60) | 120 |
| City | nvarchar(30) | 60 |
| State­Province­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Postal­Code | nvarchar(15) | 30 |
| Country­Region­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Territory­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Territory­Group | nvarchar(50) | 100 |
| Sales­Quota | money | 8 |
| Sales­YTD | money | 8 |
| Sales­Last­Year | money | 8 |

SQL Script

|  |
| --- |
| CREATE VIEW [Sales].[v­Sales­Person]  AS  SELECT  s.[Business­Entity­ID]  ,p.[Title]  ,p.[First­Name]  ,p.[Middle­Name]  ,p.[Last­Name]  ,p.[Suffix]  ,e.[Job­Title]  ,pp.[Phone­Number]  ,pnt.[Name] AS [Phone­Number­Type]  ,ea.[Email­Address]  ,p.[Email­Promotion]  ,a.[Address­Line1]  ,a.[Address­Line2]  ,a.[City]  ,[State­Province­Name] = sp.[Name]  ,a.[Postal­Code]  ,[Country­Region­Name] = cr.[Name]  ,[Territory­Name] = st.[Name]  ,[Territory­Group] = st.[Group]  ,s.[Sales­Quota]  ,s.[Sales­YTD]  ,s.[Sales­Last­Year]  FROM [Sales].[Sales­Person] s  INNER JOIN [Human­Resources].[Employee] e  ON e.[Business­Entity­ID] = s.[Business­Entity­ID]  INNER JOIN [Person].[Person] p  ON p.[Business­Entity­ID] = s.[Business­Entity­ID]  INNER JOIN [Person].[Business­Entity­Address] bea  ON bea.[Business­Entity­ID] = s.[Business­Entity­ID]  INNER JOIN [Person].[Address] a  ON a.[Address­ID] = bea.[Address­ID]  INNER JOIN [Person].[State­Province] sp  ON sp.[State­Province­ID] = a.[State­Province­ID]  INNER JOIN [Person].[Country­Region] cr  ON cr.[Country­Region­Code] = sp.[Country­Region­Code]  LEFT OUTER JOIN [Sales].[Sales­Territory] st  ON st.[Territory­ID] = s.[Territory­ID]  LEFT OUTER JOIN [Person].[Email­Address] ea  ON ea.[Business­Entity­ID] = p.[Business­Entity­ID]  LEFT OUTER JOIN [Person].[Person­Phone] pp  ON pp.[Business­Entity­ID] = p.[Business­Entity­ID]  LEFT OUTER JOIN [Person].[Phone­Number­Type] pnt  ON pnt.[Phone­Number­Type­ID] = pp.[Phone­Number­Type­ID];  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Sales representiatives (names and addresses) and their sales-related information.', 'SCHEMA', N'Sales', 'VIEW', N'v­Sales­Person', NULL, NULL  GO |

Uses

[[Human­Resources].[Employee]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=)

[[Person].[Address]](#YHZkNb8Ts04ByOupF/OAfoeI2qk=)

[[Person].[Business­Entity­Address]](#HK6CehnfZ4YYh7ckP8cx74El9pY=)

[[Person].[Country­Region]](#efOGmVlBeofSPEY2jc0F91joV9Q=)

[[Person].[Email­Address]](#3EL9ymWvglf8ADJZjk1nQ8j/hLY=)

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[[Person].[Person­Phone]](#NyzXUZBpBs0rFeuVHGsZlpmRP0Y=)

[[Person].[Phone­Number­Type]](#05nCels02cq7i+PWSDjMFGEmWFU=)

[[Person].[State­Province]](#fYuToAkdHMSws7pXek8EIcFTfvY=)

[[Sales].[Sales­Person]](#hKN58zHn4ffskJv6ldcUqjf87do=)

[[Sales].[Sales­Territory]](#hvNbyL+2AhjcTkzEdMCOV3mg8XQ=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[[dbo].[Phone]](#Dz5dpqul0rw8eU0/JuKLh0EO1ek=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

|  |
| --- |
| [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]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| 2002 | money | 8 |
| 2003 | money | 8 |
| 2004 | money | 8 |

SQL Script

|  |
| --- |
| CREATE VIEW [Sales].[v­Sales­Person­Sales­By­Fiscal­Years]  AS  SELECT  pvt.[Sales­Person­ID]  ,pvt.[Full­Name]  ,pvt.[Job­Title]  ,pvt.[Sales­Territory]  ,pvt.[2002]  ,pvt.[2003]  ,pvt.[2004]  FROM (SELECT  soh.[Sales­Person­ID]  ,p.[First­Name] + ' ' + COALESCE(p.[Middle­Name], '') + ' ' + p.[Last­Name] AS [Full­Name]  ,e.[Job­Title]  ,st.[Name] AS [Sales­Territory]  ,soh.[Sub­Total]  ,YEAR(DATEADD(m, 6, soh.[Order­Date])) AS [Fiscal­Year]  FROM [Sales].[Sales­Person] sp  INNER JOIN [Sales].[Sales­Order­Header] soh  ON sp.[Business­Entity­ID] = soh.[Sales­Person­ID]  INNER JOIN [Sales].[Sales­Territory] st  ON sp.[Territory­ID] = st.[Territory­ID]  INNER JOIN [Human­Resources].[Employee] e  ON soh.[Sales­Person­ID] = e.[Business­Entity­ID]  INNER JOIN [Person].[Person] p  ON p.[Business­Entity­ID] = sp.[Business­Entity­ID]  ) AS soh  PIVOT  (  SUM([Sub­Total])  FOR [Fiscal­Year]  IN ([2002], [2003], [2004])  ) AS pvt;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Uses PIVOT to return aggregated sales information for each sales representative.', 'SCHEMA', N'Sales', 'VIEW', N'v­Sales­Person­Sales­By­Fiscal­Years', NULL, NULL  GO |

Uses

[[Human­Resources].[Employee]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=)

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[[Sales].[Sales­Order­Header]](#J0oqEfXKE3WFfJ7PoPfs1QUbO+c=)

[[Sales].[Sales­Person]](#hKN58zHn4ffskJv6ldcUqjf87do=)

[[Sales].[Sales­Territory]](#hvNbyL+2AhjcTkzEdMCOV3mg8XQ=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

|  |
| --- |
| [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]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Address­Type | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Address­Line1 | nvarchar(60) | 120 |
| Address­Line2 | nvarchar(60) | 120 |
| City | nvarchar(30) | 60 |
| State­Province­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Postal­Code | nvarchar(15) | 30 |
| Country­Region­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |

SQL Script

|  |
| --- |
| CREATE VIEW [Sales].[v­Store­With­Addresses] AS  SELECT  s.[Business­Entity­ID]  ,s.[Name]  ,at.[Name] AS [Address­Type]  ,a.[Address­Line1]  ,a.[Address­Line2]  ,a.[City]  ,sp.[Name] AS [State­Province­Name]  ,a.[Postal­Code]  ,cr.[Name] AS [Country­Region­Name]  FROM [Sales].[Store] s  INNER JOIN [Person].[Business­Entity­Address] bea  ON bea.[Business­Entity­ID] = s.[Business­Entity­ID]  INNER JOIN [Person].[Address] a  ON a.[Address­ID] = bea.[Address­ID]  INNER JOIN [Person].[State­Province] sp  ON sp.[State­Province­ID] = a.[State­Province­ID]  INNER JOIN [Person].[Country­Region] cr  ON cr.[Country­Region­Code] = sp.[Country­Region­Code]  INNER JOIN [Person].[Address­Type] at  ON at.[Address­Type­ID] = bea.[Address­Type­ID];  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Stores (including store addresses) that sell Adventure Works Cycles products to consumers.', 'SCHEMA', N'Sales', 'VIEW', N'v­Store­With­Addresses', NULL, NULL  GO |

Uses

[[Person].[Address]](#YHZkNb8Ts04ByOupF/OAfoeI2qk=)

[[Person].[Address­Type]](#sC112VucbUsBFhIBrJnCpuBv1os=)

[[Person].[Business­Entity­Address]](#HK6CehnfZ4YYh7ckP8cx74El9pY=)

[[Person].[Country­Region]](#efOGmVlBeofSPEY2jc0F91joV9Q=)

[[Person].[State­Province]](#fYuToAkdHMSws7pXek8EIcFTfvY=)

[[Sales].[Store]](#o4yE5vuw4wTVa38E7sxNcXx3KhA=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

|  |
| --- |
| [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]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Contact­Type | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Title | nvarchar(8) | 16 |
| First­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Middle­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Last­Name | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Suffix | nvarchar(10) | 20 |
| Phone­Number | [[dbo].[Phone]](#Dz5dpqul0rw8eU0/JuKLh0EO1ek=) | 50 |
| Phone­Number­Type | [[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Email­Address | nvarchar(50) | 100 |
| Email­Promotion | int | 4 |

SQL Script

|  |
| --- |
| CREATE VIEW [Sales].[v­Store­With­Contacts] AS  SELECT  s.[Business­Entity­ID]  ,s.[Name]  ,ct.[Name] AS [Contact­Type]  ,p.[Title]  ,p.[First­Name]  ,p.[Middle­Name]  ,p.[Last­Name]  ,p.[Suffix]  ,pp.[Phone­Number]  ,pnt.[Name] AS [Phone­Number­Type]  ,ea.[Email­Address]  ,p.[Email­Promotion]  FROM [Sales].[Store] s  INNER JOIN [Person].[Business­Entity­Contact] bec  ON bec.[Business­Entity­ID] = s.[Business­Entity­ID]  INNER JOIN [Person].[Contact­Type] ct  ON ct.[Contact­Type­ID] = bec.[Contact­Type­ID]  INNER JOIN [Person].[Person] p  ON p.[Business­Entity­ID] = bec.[Person­ID]  LEFT OUTER JOIN [Person].[Email­Address] ea  ON ea.[Business­Entity­ID] = p.[Business­Entity­ID]  LEFT OUTER JOIN [Person].[Person­Phone] pp  ON pp.[Business­Entity­ID] = p.[Business­Entity­ID]  LEFT OUTER JOIN [Person].[Phone­Number­Type] pnt  ON pnt.[Phone­Number­Type­ID] = pp.[Phone­Number­Type­ID];  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Stores (including store contacts) that sell Adventure Works Cycles products to consumers.', 'SCHEMA', N'Sales', 'VIEW', N'v­Store­With­Contacts', NULL, NULL  GO |

Uses

[[Person].[Business­Entity­Contact]](#9Zj2hWeFVzGQn5EUQXP6lLMtAZI=)

[[Person].[Contact­Type]](#uyTva1snlxlEidD4b0wN1HQzMw4=)

[[Person].[Email­Address]](#3EL9ymWvglf8ADJZjk1nQ8j/hLY=)

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[[Person].[Person­Phone]](#NyzXUZBpBs0rFeuVHGsZlpmRP0Y=)

[[Person].[Phone­Number­Type]](#05nCels02cq7i+PWSDjMFGEmWFU=)

[[Sales].[Store]](#o4yE5vuw4wTVa38E7sxNcXx3KhA=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[[dbo].[Phone]](#Dz5dpqul0rw8eU0/JuKLh0EO1ek=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

|  |
| --- |
| [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]](#j6R3OCcicAWC4on+/yNYdrBS0c0=) | 100 |
| Annual­Sales | money | 8 |
| Annual­Revenue | money | 8 |
| Bank­Name | nvarchar(50) | 100 |
| Business­Type | nvarchar(5) | 10 |
| Year­Opened | int | 4 |
| Specialty | nvarchar(50) | 100 |
| Square­Feet | int | 4 |
| Brands | nvarchar(30) | 60 |
| Internet | nvarchar(30) | 60 |
| Number­Employees | int | 4 |

SQL Script

|  |
| --- |
| CREATE VIEW [Sales].[v­Store­With­Demographics] AS  SELECT  s.[Business­Entity­ID]  ,s.[Name]  ,s.[Demographics].value('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Store­Survey";  (/Store­Survey/Annual­Sales)[1]', 'money') AS [Annual­Sales]  ,s.[Demographics].value('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Store­Survey";  (/Store­Survey/Annual­Revenue)[1]', 'money') AS [Annual­Revenue]  ,s.[Demographics].value('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Store­Survey";  (/Store­Survey/Bank­Name)[1]', 'nvarchar(50)') AS [Bank­Name]  ,s.[Demographics].value('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Store­Survey";  (/Store­Survey/Business­Type)[1]', 'nvarchar(5)') AS [Business­Type]  ,s.[Demographics].value('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Store­Survey";  (/Store­Survey/Year­Opened)[1]', 'integer') AS [Year­Opened]  ,s.[Demographics].value('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Store­Survey";  (/Store­Survey/Specialty)[1]', 'nvarchar(50)') AS [Specialty]  ,s.[Demographics].value('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Store­Survey";  (/Store­Survey/Square­Feet)[1]', 'integer') AS [Square­Feet]  ,s.[Demographics].value('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Store­Survey";  (/Store­Survey/Brands)[1]', 'nvarchar(30)') AS [Brands]  ,s.[Demographics].value('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Store­Survey";  (/Store­Survey/Internet)[1]', 'nvarchar(30)') AS [Internet]  ,s.[Demographics].value('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Store­Survey";  (/Store­Survey/Number­Employees)[1]', 'integer') AS [Number­Employees]  FROM [Sales].[Store] s;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Stores (including demographics) that sell Adventure Works Cycles products to consumers.', 'SCHEMA', N'Sales', 'VIEW', N'v­Store­With­Demographics', NULL, NULL  GO |

Uses

[[Sales].[Store]](#o4yE5vuw4wTVa38E7sxNcXx3KhA=)

[[dbo].[Name]](#j6R3OCcicAWC4on+/yNYdrBS0c0=)

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

|  |
| --- |
| Stored Procedures |

Objects

|  |
| --- |
| Name |
| [dbo.usp­Get­Bill­Of­Materials](#C41ile3bJ5ydReR7L8IkkBAf7Ac=) Stored procedure using a recursive query to return a multi-level bill of material for the specified Product­ID. |
| [dbo.usp­Get­Employee­Managers](#yaIVF+o80FMpJygtFV827IXwYrg=) Stored procedure using a recursive query to return the direct and indirect managers of the specified employee. |
| [dbo.usp­Get­Manager­Employees](#j0llh4oBM6uN5VRP3n9vBDBGwbc=) Stored procedure using a recursive query to return the direct and indirect employees of the specified manager. |
| [dbo.usp­Get­Where­Used­Product­ID](#Nj+nzkRbS2L5x7zZW3glT02uJTA=) Stored procedure using a recursive query to return all components or assemblies that directly or indirectly use the specified Product­ID. |
| [dbo.usp­Log­Error](#5QeOhszr4MkHJhy/bzXlVS2S9sg=) 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](#pp3lctRP6teZGHKstzwOkN1TOzs=) 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](#/DT71/7etHi4sXQ3xAYmBMvn9JA=) |
| [Human­Resources.usp­Update­Employee­Hire­Info](#6ydkl6OBzxZTt8aQ90AP6hr4cFU=) Updates the Employee table and inserts a new row in the Employee­Pay­History table with the values specified in the input parameters. |
| [Human­Resources.usp­Update­Employee­Login](#C5oNhWtNH5mLMaKD+WajVUPlL6E=) Updates the Employee table with the values specified in the input parameters for the given Business­Entity­ID. |
| [Human­Resources.usp­Update­Employee­Personal­Info](#vCx7xMQcjRyuiOsUmNSNg9QeITY=) 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 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[usp­Get­Bill­Of­Materials]  @Start­Product­ID [int],  @Check­Date [datetime]  AS  BEGIN  SET NOCOUNT ON;  -- Use recursive query to generate a multi-level Bill of Material (i.e. all level 1  -- components of a level 0 assembly, all level 2 components of a level 1 assembly)  -- The Check­Date eliminates any components that are no longer used in the product on this date.  WITH [BOM\_cte]([Product­Assembly­ID], [Component­ID], [Component­Desc], [Per­Assembly­Qty], [Standard­Cost], [List­Price], [BOMLevel], [Recursion­Level]) -- CTE name and columns  AS (  SELECT b.[Product­Assembly­ID], b.[Component­ID], p.[Name], b.[Per­Assembly­Qty], p.[Standard­Cost], p.[List­Price], b.[BOMLevel], 0 -- Get the initial list of components for the bike assembly  FROM [Production].[Bill­Of­Materials] b  INNER JOIN [Production].[Product] p  ON b.[Component­ID] = p.[Product­ID]  WHERE b.[Product­Assembly­ID] = @Start­Product­ID  AND @Check­Date >= b.[Start­Date]  AND @Check­Date <= ISNULL(b.[End­Date], @Check­Date)  UNION ALL  SELECT b.[Product­Assembly­ID], b.[Component­ID], p.[Name], b.[Per­Assembly­Qty], p.[Standard­Cost], p.[List­Price], b.[BOMLevel], [Recursion­Level] + 1 -- Join recursive member to anchor  FROM [BOM\_cte] cte  INNER JOIN [Production].[Bill­Of­Materials] b  ON b.[Product­Assembly­ID] = cte.[Component­ID]  INNER JOIN [Production].[Product] p  ON b.[Component­ID] = p.[Product­ID]  WHERE @Check­Date >= b.[Start­Date]  AND @Check­Date <= ISNULL(b.[End­Date], @Check­Date)  )  -- Outer select from the CTE  SELECT b.[Product­Assembly­ID], b.[Component­ID], b.[Component­Desc], SUM(b.[Per­Assembly­Qty]) AS [Total­Quantity] , b.[Standard­Cost], b.[List­Price], b.[BOMLevel], b.[Recursion­Level]  FROM [BOM\_cte] b  GROUP BY b.[Component­ID], b.[Component­Desc], b.[Product­Assembly­ID], b.[BOMLevel], b.[Recursion­Level], b.[Standard­Cost], b.[List­Price]  ORDER BY b.[BOMLevel], b.[Product­Assembly­ID], b.[Component­ID]  OPTION (MAXRECURSION 25)  END;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Stored procedure using a recursive query to return a multi-level bill of material for the specified Product­ID.', '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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the stored procedure usp­Get­Bill­Of­Materials. Enter a valid Product­ID from the Production.Product table.', 'SCHEMA', N'dbo', 'PROCEDURE', N'usp­Get­Bill­Of­Materials', 'PARAMETER', N'@Start­Product­ID'  GO |

Uses

[[Production].[Bill­Of­Materials]](#m9P+2olmZFXOowkUY18+Ri7YHx4=)

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

|  |
| --- |
| [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 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[usp­Get­Employee­Managers]  @Business­Entity­ID [int]  AS  BEGIN  SET NOCOUNT ON;  -- Use recursive query to list out all Employees required for a particular Manager  WITH [EMP\_cte]([Business­Entity­ID], [Organization­Node], [First­Name], [Last­Name], [Job­Title], [Recursion­Level]) -- CTE name and columns  AS (  SELECT e.[Business­Entity­ID], e.[Organization­Node], p.[First­Name], p.[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].[First­Name], [EMP\_cte].[Last­Name],  [EMP\_cte].[Organization­Node].To­String() AS [Organization­Node], p.[First­Name] AS 'Manager­First­Name', p.[Last­Name] AS 'Manager­Last­Name' -- Outer select from the CTE  FROM [EMP\_cte]  INNER JOIN [Human­Resources].[Employee] e  ON [EMP\_cte].[Organization­Node].Get­Ancestor(1) = e.[Organization­Node]  INNER JOIN [Person].[Person] p  ON p.[Business­Entity­ID] = e.[Business­Entity­ID]  ORDER BY [Recursion­Level], [EMP\_cte].[Organization­Node].To­String()  OPTION (MAXRECURSION 25)  END;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Stored procedure using a recursive query to return the direct and indirect managers of the specified employee.', 'SCHEMA', N'dbo', 'PROCEDURE', N'usp­Get­Employee­Managers', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the stored procedure usp­Get­Employee­Managers. Enter a valid Business­Entity­ID from the Human­Resources.Employee table.', 'SCHEMA', N'dbo', 'PROCEDURE', N'usp­Get­Employee­Managers', 'PARAMETER', N'@Business­Entity­ID'  GO |

Uses

[[Human­Resources].[Employee]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=)

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

|  |
| --- |
| [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 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[usp­Get­Manager­Employees]  @Business­Entity­ID [int]  AS  BEGIN  SET NOCOUNT ON;  -- Use recursive query to list out all Employees required for a particular Manager  WITH [EMP\_cte]([Business­Entity­ID], [Organization­Node], [First­Name], [Last­Name], [Recursion­Level]) -- CTE name and columns  AS (  SELECT e.[Business­Entity­ID], e.[Organization­Node], p.[First­Name], p.[Last­Name], 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 'Manager­Last­Name',  [EMP\_cte].[Business­Entity­ID], [EMP\_cte].[First­Name], [EMP\_cte].[Last­Name] -- Outer select from the CTE  FROM [EMP\_cte]  INNER JOIN [Human­Resources].[Employee] e  ON [EMP\_cte].[Organization­Node].Get­Ancestor(1) = e.[Organization­Node]  INNER JOIN [Person].[Person] p  ON p.[Business­Entity­ID] = e.[Business­Entity­ID]  ORDER BY [Recursion­Level], [EMP\_cte].[Organization­Node].To­String()  OPTION (MAXRECURSION 25)  END;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Stored procedure using a recursive query to return the direct and indirect employees of the specified manager.', 'SCHEMA', N'dbo', 'PROCEDURE', N'usp­Get­Manager­Employees', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the stored procedure usp­Get­Manager­Employees. Enter a valid Business­Entity­ID of the manager from the Human­Resources.Employee table.', 'SCHEMA', N'dbo', 'PROCEDURE', N'usp­Get­Manager­Employees', 'PARAMETER', N'@Business­Entity­ID'  GO |

Uses

[[Human­Resources].[Employee]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=)

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

|  |
| --- |
| [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 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[usp­Get­Where­Used­Product­ID]  @Start­Product­ID [int],  @Check­Date [datetime]  AS  BEGIN  SET NOCOUNT ON;  --Use recursive query to generate a multi-level Bill of Material (i.e. all level 1 components of a level 0 assembly, all level 2 components of a level 1 assembly)  WITH [BOM\_cte]([Product­Assembly­ID], [Component­ID], [Component­Desc], [Per­Assembly­Qty], [Standard­Cost], [List­Price], [BOMLevel], [Recursion­Level]) -- CTE name and columns  AS (  SELECT b.[Product­Assembly­ID], b.[Component­ID], p.[Name], b.[Per­Assembly­Qty], p.[Standard­Cost], p.[List­Price], b.[BOMLevel], 0 -- Get the initial list of components for the bike assembly  FROM [Production].[Bill­Of­Materials] b  INNER JOIN [Production].[Product] p  ON b.[Product­Assembly­ID] = p.[Product­ID]  WHERE b.[Component­ID] = @Start­Product­ID  AND @Check­Date >= b.[Start­Date]  AND @Check­Date <= ISNULL(b.[End­Date], @Check­Date)  UNION ALL  SELECT b.[Product­Assembly­ID], b.[Component­ID], p.[Name], b.[Per­Assembly­Qty], p.[Standard­Cost], p.[List­Price], b.[BOMLevel], [Recursion­Level] + 1 -- Join recursive member to anchor  FROM [BOM\_cte] cte  INNER JOIN [Production].[Bill­Of­Materials] b  ON cte.[Product­Assembly­ID] = b.[Component­ID]  INNER JOIN [Production].[Product] p  ON b.[Product­Assembly­ID] = p.[Product­ID]  WHERE @Check­Date >= b.[Start­Date]  AND @Check­Date <= ISNULL(b.[End­Date], @Check­Date)  )  -- Outer select from the CTE  SELECT b.[Product­Assembly­ID], b.[Component­ID], b.[Component­Desc], SUM(b.[Per­Assembly­Qty]) 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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the stored procedure usp­Get­Where­Used­Product­ID. Enter a valid Product­ID from the Production.Product table.', 'SCHEMA', N'dbo', 'PROCEDURE', N'usp­Get­Where­Used­Product­ID', 'PARAMETER', N'@Start­Product­ID'  GO |

Uses

[[Production].[Bill­Of­Materials]](#m9P+2olmZFXOowkUY18+Ri7YHx4=)

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

|  |
| --- |
| [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 |

SQL Script

|  |
| --- |
| -- usp­Log­Error logs error information in the Error­Log table about the  -- error that caused execution to jump to the CATCH block of a  -- TRY...CATCH construct. This should be executed from within the scope  -- of a CATCH block otherwise it will return without inserting error  -- information.  CREATE PROCEDURE [dbo].[usp­Log­Error]  @Error­Log­ID [int] = 0 OUTPUT -- contains the Error­Log­ID of the row inserted  AS -- by usp­Log­Error in the Error­Log table  BEGIN  SET NOCOUNT ON;  -- Output parameter value of 0 indicates that error  -- information was not logged  SET @Error­Log­ID = 0;  BEGIN TRY  -- Return if there is no error information to log  IF ERROR\_­NUMBER() IS NULL  RETURN;  -- Return if inside an uncommittable transaction.  -- Data insertion/modification is not allowed when  -- a transaction is in an uncommittable state.  IF XACT\_­STATE() = -1  BEGIN  PRINT 'Cannot log error since the current transaction is in an uncommittable state. '  + 'Rollback the transaction before executing usp­Log­Error in order to successfully log error information.';  RETURN;  END  INSERT [dbo].[Error­Log]  (  [User­Name],  [Error­Number],  [Error­Severity],  [Error­State],  [Error­Procedure],  [Error­Line],  [Error­Message]  )  VALUES  (  CONVERT(sysname, CURRENT\_­USER),  ERROR\_­NUMBER(),  ERROR\_­SEVERITY(),  ERROR\_­STATE(),  ERROR\_­PROCEDURE(),  ERROR\_­LINE(),  ERROR\_­MESSAGE()  );  -- Pass back the Error­Log­ID of the row inserted  SET @Error­Log­ID = @@IDENTITY;  END TRY  BEGIN CATCH  PRINT 'An error occurred in stored procedure usp­Log­Error: ';  EXECUTE [dbo].[usp­Print­Error];  RETURN -1;  END CATCH  END;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Logs error information in the 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'usp­Log­Error', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Output parameter for the stored procedure usp­Log­Error. Contains the Error­Log­ID value corresponding to the row inserted by usp­Log­Error in the Error­Log table.', 'SCHEMA', N'dbo', 'PROCEDURE', N'usp­Log­Error', 'PARAMETER', N'@Error­Log­ID'  GO |

Uses

[[dbo].[Error­Log]](#XCrMUsfgKMjh8aFAM1/ck+QFKjQ=)

[[dbo].[usp­Print­Error]](#pp3lctRP6teZGHKstzwOkN1TOzs=)

Used By

[[Human­Resources].[usp­Update­Employee­Hire­Info]](#6ydkl6OBzxZTt8aQ90AP6hr4cFU=)

[[Human­Resources].[usp­Update­Employee­Login]](#C5oNhWtNH5mLMaKD+WajVUPlL6E=)

[[Human­Resources].[usp­Update­Employee­Personal­Info]](#vCx7xMQcjRyuiOsUmNSNg9QeITY=)

|  |
| --- |
| [dbo].[usp­Print­Error] |

MS\_­Description

Prints error information about the error that caused execution to jump to the CATCH block of a TRY...CATCH construct. Should be executed from within the scope of a CATCH block otherwise it will return without printing any error information.

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

SQL Script

|  |
| --- |
| -- usp­Print­Error prints error information about the error that caused  -- execution to jump to the CATCH block of a TRY...CATCH construct.  -- Should be executed from within the scope of a CATCH block otherwise  -- it will return without printing any error information.  CREATE PROCEDURE [dbo].[usp­Print­Error]  AS  BEGIN  SET NOCOUNT ON;  -- Print error information.  PRINT 'Error ' + CONVERT(varchar(50), ERROR\_­NUMBER()) +  ', Severity ' + CONVERT(varchar(5), ERROR\_­SEVERITY()) +  ', State ' + CONVERT(varchar(5), ERROR\_­STATE()) +  ', Procedure ' + ISNULL(ERROR\_­PROCEDURE(), '-') +  ', Line ' + CONVERT(varchar(5), ERROR\_­LINE());  PRINT ERROR\_­MESSAGE();  END;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Prints error information about the error that caused execution to jump to the CATCH block of a TRY...CATCH construct. Should be executed from within the scope of a CATCH block otherwise it will return without printing any error information.', 'SCHEMA', N'dbo', 'PROCEDURE', N'usp­Print­Error', NULL, NULL  GO |

Used By

[[dbo].[usp­Log­Error]](#5QeOhszr4MkHJhy/bzXlVS2S9sg=)

|  |
| --- |
| [dbo].[usp­Search­Candidate­Resumes] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @search­String | nvarchar(1000) | 2000 |
| @use­Inflectional | bit | 1 |
| @use­Thesaurus | bit | 1 |
| @language | int | 4 |

SQL Script

|  |
| --- |
| --A stored procedure which demonstrates integrated full text search  CREATE PROCEDURE [dbo].[usp­Search­Candidate­Resumes]  @search­String [nvarchar](1000),  @use­Inflectional [bit]=0,  @use­Thesaurus [bit]=0,  @language[int]=0  WITH EXECUTE AS CALLER  AS  BEGIN  SET NOCOUNT ON;  DECLARE @string nvarchar(1050)  --setting the lcid to the default instance LCID if needed  IF @language = NULL OR @language = 0  BEGIN  SELECT @language =CONVERT(int, serverproperty('lcid'))  END    --FREETEXTTABLE case as inflectional and Thesaurus were required  IF @use­Thesaurus = 1 AND @use­Inflectional = 1  BEGIN  SELECT FT\_­TBL.[Job­Candidate­ID], KEY\_­TBL.[RANK] FROM [Human­Resources].[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 [Human­Resources].[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 [Human­Resources].[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 [Human­Resources].[Job­Candidate] AS FT\_­TBL  INNER JOIN CONTAINSTABLE([Human­Resources].[Job­Candidate],\*,@string,LANGUAGE @language) AS KEY\_­TBL  ON FT\_­TBL.[Job­Candidate­ID] =KEY\_­TBL.[KEY]  END  END;  GO |

Uses

[[Human­Resources].[Job­Candidate]](#WVRXAhzXWcgHQbdlg570h3Fv6QI=)

|  |
| --- |
| [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 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [Human­Resources].[usp­Update­Employee­Hire­Info]  @Business­Entity­ID [int],  @Job­Title [nvarchar](50),  @Hire­Date [datetime],  @Rate­Change­Date [datetime],  @Rate [money],  @Pay­Frequency [tinyint],  @Current­Flag [dbo].[Flag]  WITH EXECUTE AS CALLER  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  BEGIN TRANSACTION;  UPDATE [Human­Resources].[Employee]  SET [Job­Title] = @Job­Title  ,[Hire­Date] = @Hire­Date  ,[Current­Flag] = @Current­Flag  WHERE [Business­Entity­ID] = @Business­Entity­ID;  INSERT INTO [Human­Resources].[Employee­Pay­History]  ([Business­Entity­ID]  ,[Rate­Change­Date]  ,[Rate]  ,[Pay­Frequency])  VALUES (@Business­Entity­ID, @Rate­Change­Date, @Rate, @Pay­Frequency);  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  -- Rollback any active or uncommittable transactions before  -- inserting information in the Error­Log  IF @@TRANCOUNT > 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE [dbo].[usp­Log­Error];  END CATCH;  END;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Updates the Employee table and inserts a new row in the Employee­Pay­History table with the values specified in the input parameters.', 'SCHEMA', N'Human­Resources', 'PROCEDURE', N'usp­Update­Employee­Hire­Info', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the stored procedure usp­Update­Employee­Hire­Info. Enter a valid Business­Entity­ID from the Employee table.', 'SCHEMA', N'Human­Resources', 'PROCEDURE', N'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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the stored procedure usp­Update­Employee­Hire­Info. Enter a hire date for the employee.', 'SCHEMA', N'Human­Resources', 'PROCEDURE', N'usp­Update­Employee­Hire­Info', 'PARAMETER', N'@Hire­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the stored procedure usp­Update­Employee­Hire­Info. Enter a title for the employee.', 'SCHEMA', N'Human­Resources', 'PROCEDURE', N'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'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the stored procedure usp­Update­Employee­Hire­Info. Enter the new rate for the employee.', 'SCHEMA', N'Human­Resources', 'PROCEDURE', N'usp­Update­Employee­Hire­Info', 'PARAMETER', N'@Rate'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the stored procedure usp­Update­Employee­Hire­Info. Enter the date the rate changed for the employee.', 'SCHEMA', N'Human­Resources', 'PROCEDURE', N'usp­Update­Employee­Hire­Info', 'PARAMETER', N'@Rate­Change­Date'  GO |

Uses

[[Human­Resources].[Employee]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=)

[[Human­Resources].[Employee­Pay­History]](#WzREUJ9VRXbYWerHGwiGJinYM0A=)

[[dbo].[usp­Log­Error]](#5QeOhszr4MkHJhy/bzXlVS2S9sg=)

[[dbo].[Flag]](#j88jjDtwCbK8khz/m8jxryH/SFk=)

[Human­Resources](#wRNBi9QOObVjEO+Wxjr4GvvwBBs=)

|  |
| --- |
| [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 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [Human­Resources].[usp­Update­Employee­Login]  @Business­Entity­ID [int],  @Organization­Node [hierarchyid],  @Login­ID [nvarchar](256),  @Job­Title [nvarchar](50),  @Hire­Date [datetime],  @Current­Flag [dbo].[Flag]  WITH EXECUTE AS CALLER  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  UPDATE [Human­Resources].[Employee]  SET [Organization­Node] = @Organization­Node  ,[Login­ID] = @Login­ID  ,[Job­Title] = @Job­Title  ,[Hire­Date] = @Hire­Date  ,[Current­Flag] = @Current­Flag  WHERE [Business­Entity­ID] = @Business­Entity­ID;  END TRY  BEGIN CATCH  EXECUTE [dbo].[usp­Log­Error];  END CATCH;  END;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Updates the Employee table with the values specified in the input parameters for the given Business­Entity­ID.', 'SCHEMA', N'Human­Resources', 'PROCEDURE', N'usp­Update­Employee­Login', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the stored procedure usp­Update­Employee­Login. Enter a valid Employee­ID from the Employee table.', 'SCHEMA', N'Human­Resources', 'PROCEDURE', N'usp­Update­Employee­Login', 'PARAMETER', N'@Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the stored procedure usp­Update­Employee­Hire­Info. Enter the current flag for the employee.', 'SCHEMA', N'Human­Resources', 'PROCEDURE', N'usp­Update­Employee­Login', 'PARAMETER', N'@Current­Flag'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the stored procedure usp­Update­Employee­Hire­Info. Enter a hire date for the employee.', 'SCHEMA', N'Human­Resources', 'PROCEDURE', N'usp­Update­Employee­Login', 'PARAMETER', N'@Hire­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the stored procedure usp­Update­Employee­Hire­Info. Enter a title for the employee.', 'SCHEMA', N'Human­Resources', 'PROCEDURE', N'usp­Update­Employee­Login', 'PARAMETER', N'@Job­Title'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the stored procedure usp­Update­Employee­Hire­Info. Enter a valid login for the employee.', 'SCHEMA', N'Human­Resources', 'PROCEDURE', N'usp­Update­Employee­Login', 'PARAMETER', N'@Login­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the stored procedure usp­Update­Employee­Hire­Info. Enter a valid Manager­ID for the employee.', 'SCHEMA', N'Human­Resources', 'PROCEDURE', N'usp­Update­Employee­Login', 'PARAMETER', N'@Organization­Node'  GO |

Uses

[[Human­Resources].[Employee]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=)

[[dbo].[usp­Log­Error]](#5QeOhszr4MkHJhy/bzXlVS2S9sg=)

[[dbo].[Flag]](#j88jjDtwCbK8khz/m8jxryH/SFk=)

[Human­Resources](#wRNBi9QOObVjEO+Wxjr4GvvwBBs=)

|  |
| --- |
| [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 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [Human­Resources].[usp­Update­Employee­Personal­Info]  @Business­Entity­ID [int],  @National­IDNumber [nvarchar](15),  @Birth­Date [datetime],  @Marital­Status [nchar](1),  @Gender [nchar](1)  WITH EXECUTE AS CALLER  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  UPDATE [Human­Resources].[Employee]  SET [National­IDNumber] = @National­IDNumber  ,[Birth­Date] = @Birth­Date  ,[Marital­Status] = @Marital­Status  ,[Gender] = @Gender  WHERE [Business­Entity­ID] = @Business­Entity­ID;  END TRY  BEGIN CATCH  EXECUTE [dbo].[usp­Log­Error];  END CATCH;  END;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Updates the Employee table with the values specified in the input parameters for the given Employee­ID.', 'SCHEMA', N'Human­Resources', 'PROCEDURE', N'usp­Update­Employee­Personal­Info', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the stored procedure usp­Update­Employee­Hire­Info. Enter a birth date for the employee.', 'SCHEMA', N'Human­Resources', 'PROCEDURE', N'usp­Update­Employee­Personal­Info', 'PARAMETER', N'@Birth­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the stored procedure usp­Update­Employee­Personal­Info. Enter a valid Business­Entity­ID from the Human­Resources.Employee table.', 'SCHEMA', N'Human­Resources', 'PROCEDURE', N'usp­Update­Employee­Personal­Info', 'PARAMETER', N'@Business­Entity­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the stored procedure usp­Update­Employee­Hire­Info. Enter a gender for the employee.', 'SCHEMA', N'Human­Resources', 'PROCEDURE', N'usp­Update­Employee­Personal­Info', 'PARAMETER', N'@Gender'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the stored procedure usp­Update­Employee­Hire­Info. Enter a marital status for the employee.', 'SCHEMA', N'Human­Resources', 'PROCEDURE', N'usp­Update­Employee­Personal­Info', 'PARAMETER', N'@Marital­Status'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the stored procedure usp­Update­Employee­Hire­Info. Enter a national ID for the employee.', 'SCHEMA', N'Human­Resources', 'PROCEDURE', N'usp­Update­Employee­Personal­Info', 'PARAMETER', N'@National­IDNumber'  GO |

Uses

[[Human­Resources].[Employee]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=)

[[dbo].[usp­Log­Error]](#5QeOhszr4MkHJhy/bzXlVS2S9sg=)

[Human­Resources](#wRNBi9QOObVjEO+Wxjr4GvvwBBs=)

|  |
| --- |
| Table-valued Functions |

Objects

|  |
| --- |
| Name |
| [dbo.ufn­Get­Contact­Information](#MAbUSn/1caUuJ7XN6dczcbTRRyA=) 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 |

SQL Script

|  |
| --- |
| CREATE FUNCTION [dbo].[ufn­Get­Contact­Information](@Person­ID int)  RETURNS @ret­Contact­Information TABLE  (  -- Columns returned by the function  [Person­ID] int NOT NULL,  [First­Name] [nvarchar](50) NULL,  [Last­Name] [nvarchar](50) NULL,  [Job­Title] [nvarchar](50) NULL,  [Business­Entity­Type] [nvarchar](50) NULL  )  AS  -- Returns the first name, last name, job title and business entity type for the specified contact.  -- Since a contact can serve multiple roles, more than one row may be returned.  BEGIN  IF @Person­ID IS NOT NULL  BEGIN  IF EXISTS(SELECT \* FROM [Human­Resources].[Employee] e  WHERE e.[Business­Entity­ID] = @Person­ID)  INSERT INTO @ret­Contact­Information  SELECT @Person­ID, p.First­Name, p.Last­Name, e.[Job­Title], 'Employee'  FROM [Human­Resources].[Employee] AS e  INNER JOIN [Person].[Person] p  ON p.[Business­Entity­ID] = e.[Business­Entity­ID]  WHERE e.[Business­Entity­ID] = @Person­ID;  IF EXISTS(SELECT \* FROM [Purchasing].[Vendor] AS v  INNER JOIN [Person].[Business­Entity­Contact] bec  ON bec.[Business­Entity­ID] = v.[Business­Entity­ID]  WHERE bec.[Person­ID] = @Person­ID)  INSERT INTO @ret­Contact­Information  SELECT @Person­ID, p.First­Name, p.Last­Name, ct.[Name], 'Vendor Contact'  FROM [Purchasing].[Vendor] AS v  INNER JOIN [Person].[Business­Entity­Contact] bec  ON bec.[Business­Entity­ID] = v.[Business­Entity­ID]  INNER JOIN [Person].Contact­Type ct  ON ct.[Contact­Type­ID] = bec.[Contact­Type­ID]  INNER JOIN [Person].[Person] p  ON p.[Business­Entity­ID] = bec.[Person­ID]  WHERE bec.[Person­ID] = @Person­ID;    IF EXISTS(SELECT \* FROM [Sales].[Store] AS s  INNER JOIN [Person].[Business­Entity­Contact] bec  ON bec.[Business­Entity­ID] = s.[Business­Entity­ID]  WHERE bec.[Person­ID] = @Person­ID)  INSERT INTO @ret­Contact­Information  SELECT @Person­ID, p.First­Name, p.Last­Name, ct.[Name], 'Store Contact'  FROM [Sales].[Store] AS s  INNER JOIN [Person].[Business­Entity­Contact] bec  ON bec.[Business­Entity­ID] = s.[Business­Entity­ID]  INNER JOIN [Person].Contact­Type ct  ON ct.[Contact­Type­ID] = bec.[Contact­Type­ID]  INNER JOIN [Person].[Person] p  ON p.[Business­Entity­ID] = bec.[Person­ID]  WHERE bec.[Person­ID] = @Person­ID;  IF EXISTS(SELECT \* FROM [Person].[Person] AS p  INNER JOIN [Sales].[Customer] AS c  ON c.[Person­ID] = p.[Business­Entity­ID]  WHERE p.[Business­Entity­ID] = @Person­ID AND c.[Store­ID] IS NULL)  INSERT INTO @ret­Contact­Information  SELECT @Person­ID, p.First­Name, p.Last­Name, NULL, 'Consumer'  FROM [Person].[Person] AS p  INNER JOIN [Sales].[Customer] AS c  ON c.[Person­ID] = p.[Business­Entity­ID]  WHERE p.[Business­Entity­ID] = @Person­ID AND c.[Store­ID] IS NULL;  END  RETURN;  END;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Table value function returning the first name, last name, job title and contact type for a given contact.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Get­Contact­Information', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the table value function ufn­Get­Contact­Information. Enter a valid Person­ID from the Person.Contact table.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Get­Contact­Information', 'PARAMETER', N'@Person­ID'  GO |

Uses

[[Human­Resources].[Employee]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=)

[[Person].[Business­Entity­Contact]](#9Zj2hWeFVzGQn5EUQXP6lLMtAZI=)

[[Person].[Contact­Type]](#uyTva1snlxlEidD4b0wN1HQzMw4=)

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[[Purchasing].[Vendor]](#yfc9dLywcQtxtbztLjV6J3bb0tM=)

[[Sales].[Customer]](#UK53WWAjbLvEqOKY0KByf769x9M=)

[[Sales].[Store]](#o4yE5vuw4wTVa38E7sxNcXx3KhA=)

|  |
| --- |
| Scalar-valued Functions |

Objects

|  |
| --- |
| Name |
| [dbo.ufn­Get­Accounting­End­Date](#l+9oERkOGqy7NCMVk5qhMxSvPUE=) Scalar function used in the u­Sales­Order­Header trigger to set the starting account date. |
| [dbo.ufn­Get­Accounting­Start­Date](#qPaspZL/9Jqp3AaHMQ9Srtyi+gk=) Scalar function used in the u­Sales­Order­Header trigger to set the ending account date. |
| [dbo.ufn­Get­Document­Status­Text](#w+9GVsJ4QWsZw9zvddOOm+6VE9U=) Scalar function returning the text representation of the Status column in the Document table. |
| [dbo.ufn­Get­Product­Dealer­Price](#qCHqC6Ssa6gRKHNZ1A5hYnf4CXs=) Scalar function returning the dealer price for a given product on a particular order date. |
| [dbo.ufn­Get­Product­List­Price](#8mn5CnDFl2xPNBEKQCwZ9YJdOMk=) Scalar function returning the list price for a given product on a particular order date. |
| [dbo.ufn­Get­Product­Standard­Cost](#iQHxQM8zJLh+qEeVmjb6xgEGWO4=) Scalar function returning the standard cost for a given product on a particular order date. |
| [dbo.ufn­Get­Purchase­Order­Status­Text](#PMU1tQL4fyT1furkFgb8xilbu7M=) Scalar function returning the text representation of the Status column in the Purchase­Order­Header table. |
| [dbo.ufn­Get­Sales­Order­Status­Text](#pQ5iAe9Ece080z4eJt7JNpIM0pc=) Scalar function returning the text representation of the Status column in the Sales­Order­Header table. |
| [dbo.ufn­Get­Stock](#uhj1mdMC7NK57d7HxekWpOKGYxI=) Scalar function returning the quantity of inventory in Location­ID 6 (Miscellaneous Storage)for a specified Product­ID. |
| [dbo.ufn­Leading­Zeros](#50oMRfjOJe+9VZTmAIx0V5kh0ds=) 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 |

SQL Script

|  |
| --- |
| CREATE FUNCTION [dbo].[ufn­Get­Accounting­End­Date]()  RETURNS [datetime]  AS  BEGIN  RETURN DATEADD(millisecond, -2, CONVERT(datetime, '20040701', 112));  END;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Scalar function used in the u­Sales­Order­Header trigger to set the starting account date.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Get­Accounting­End­Date', NULL, NULL  GO |

|  |
| --- |
| [dbo].[ufn­Get­Accounting­Start­Date] |

MS\_­Description

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

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

SQL Script

|  |
| --- |
| CREATE FUNCTION [dbo].[ufn­Get­Accounting­Start­Date]()  RETURNS [datetime]  AS  BEGIN  RETURN CONVERT(datetime, '20030701', 112);  END;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Scalar function used in the u­Sales­Order­Header 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 |

SQL Script

|  |
| --- |
| CREATE FUNCTION [dbo].[ufn­Get­Document­Status­Text](@Status [tinyint])  RETURNS [nvarchar](16)  AS  -- Returns the sales order status text representation for the status value.  BEGIN  DECLARE @ret [nvarchar](16);  SET @ret =  CASE @Status  WHEN 1 THEN N'Pending approval'  WHEN 2 THEN N'Approved'  WHEN 3 THEN N'Obsolete'  ELSE N'\*\* Invalid \*\*'  END;    RETURN @ret  END;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Scalar function returning the text representation of the Status column in the Document table.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Get­Document­Status­Text', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the scalar function ufn­Get­Document­Status­Text. Enter a valid integer.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Get­Document­Status­Text', 'PARAMETER', N'@Status'  GO |

|  |
| --- |
| [dbo].[ufn­Get­Product­Dealer­Price] |

MS\_­Description

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

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Product­ID 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 |

SQL Script

|  |
| --- |
| CREATE FUNCTION [dbo].[ufn­Get­Product­Dealer­Price](@Product­ID [int], @Order­Date [datetime])  RETURNS [money]  AS  -- Returns the dealer price for the product on a specific date.  BEGIN  DECLARE @Dealer­Price money;  DECLARE @Dealer­Discount money;  SET @Dealer­Discount = 0.60 -- 60% of list price  SELECT @Dealer­Price = plph.[List­Price] \* @Dealer­Discount  FROM [Production].[Product] p  INNER JOIN [Production].[Product­List­Price­History] plph  ON p.[Product­ID] = plph.[Product­ID]  AND p.[Product­ID] = @Product­ID  AND @Order­Date BETWEEN plph.[Start­Date] AND COALESCE(plph.[End­Date], CONVERT(datetime, '99991231', 112)); -- Make sure we get all the prices!  RETURN @Dealer­Price;  END;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Scalar function returning the dealer price for a given product on a particular order date.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Get­Product­Dealer­Price', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the scalar function ufn­Get­Product­Dealer­Price. Enter a valid order date.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Get­Product­Dealer­Price', 'PARAMETER', N'@Order­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the scalar function ufn­Get­Product­Dealer­Price. Enter a valid Product­ID from the Production.Product table.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Get­Product­Dealer­Price', 'PARAMETER', N'@Product­ID'  GO |

Uses

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[[Production].[Product­List­Price­History]](#lSXw7btC5MVIn+g9cscCbIFpkU0=)

|  |
| --- |
| [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 |

SQL Script

|  |
| --- |
| CREATE FUNCTION [dbo].[ufn­Get­Product­List­Price](@Product­ID [int], @Order­Date [datetime])  RETURNS [money]  AS  BEGIN  DECLARE @List­Price money;  SELECT @List­Price = plph.[List­Price]  FROM [Production].[Product] p  INNER JOIN [Production].[Product­List­Price­History] plph  ON p.[Product­ID] = plph.[Product­ID]  AND p.[Product­ID] = @Product­ID  AND @Order­Date BETWEEN plph.[Start­Date] AND COALESCE(plph.[End­Date], CONVERT(datetime, '99991231', 112)); -- Make sure we get all the prices!  RETURN @List­Price;  END;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Scalar function returning the list price for a given product on a particular order date.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Get­Product­List­Price', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the scalar function ufn­Get­Product­List­Price. Enter a valid order date.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Get­Product­List­Price', 'PARAMETER', N'@Order­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the scalar function ufn­Get­Product­List­Price. Enter a valid Product­ID from the Production.Product table.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Get­Product­List­Price', 'PARAMETER', N'@Product­ID'  GO |

Uses

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[[Production].[Product­List­Price­History]](#lSXw7btC5MVIn+g9cscCbIFpkU0=)

|  |
| --- |
| [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 |

SQL Script

|  |
| --- |
| CREATE FUNCTION [dbo].[ufn­Get­Product­Standard­Cost](@Product­ID [int], @Order­Date [datetime])  RETURNS [money]  AS  -- Returns the standard cost for the product on a specific date.  BEGIN  DECLARE @Standard­Cost money;  SELECT @Standard­Cost = pch.[Standard­Cost]  FROM [Production].[Product] p  INNER JOIN [Production].[Product­Cost­History] pch  ON p.[Product­ID] = pch.[Product­ID]  AND p.[Product­ID] = @Product­ID  AND @Order­Date BETWEEN pch.[Start­Date] AND COALESCE(pch.[End­Date], CONVERT(datetime, '99991231', 112)); -- Make sure we get all the prices!  RETURN @Standard­Cost;  END;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Scalar function returning the standard cost for a given product on a particular order date.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Get­Product­Standard­Cost', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the scalar function ufn­Get­Product­Standard­Cost. Enter a valid order date.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Get­Product­Standard­Cost', 'PARAMETER', N'@Order­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the scalar function ufn­Get­Product­Standard­Cost. Enter a valid Product­ID from the Production.Product table.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Get­Product­Standard­Cost', 'PARAMETER', N'@Product­ID'  GO |

Uses

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[[Production].[Product­Cost­History]](#V4hNECfDVL8rNsab6KfgsgmqeHE=)

|  |
| --- |
| [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 |

SQL Script

|  |
| --- |
| CREATE FUNCTION [dbo].[ufn­Get­Purchase­Order­Status­Text](@Status [tinyint])  RETURNS [nvarchar](15)  AS  -- Returns the sales order status text representation for the status value.  BEGIN  DECLARE @ret [nvarchar](15);  SET @ret =  CASE @Status  WHEN 1 THEN 'Pending'  WHEN 2 THEN 'Approved'  WHEN 3 THEN 'Rejected'  WHEN 4 THEN 'Complete'  ELSE '\*\* Invalid \*\*'  END;    RETURN @ret  END;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Scalar function returning the text representation of the Status column in the Purchase­Order­Header table.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Get­Purchase­Order­Status­Text', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the scalar function ufn­Get­Purchase­Ordert­Status­Text. Enter a valid integer.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Get­Purchase­Order­Status­Text', 'PARAMETER', N'@Status'  GO |

|  |
| --- |
| [dbo].[ufn­Get­Sales­Order­Status­Text] |

MS\_­Description

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

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Status Input parameter for the scalar function ufn­Get­Sales­Order­Status­Text. Enter a valid integer. | tinyint | 1 |

SQL Script

|  |
| --- |
| CREATE FUNCTION [dbo].[ufn­Get­Sales­Order­Status­Text](@Status [tinyint])  RETURNS [nvarchar](15)  AS  -- Returns the sales order status text representation for the status value.  BEGIN  DECLARE @ret [nvarchar](15);  SET @ret =  CASE @Status  WHEN 1 THEN 'In process'  WHEN 2 THEN 'Approved'  WHEN 3 THEN 'Backordered'  WHEN 4 THEN 'Rejected'  WHEN 5 THEN 'Shipped'  WHEN 6 THEN 'Cancelled'  ELSE '\*\* Invalid \*\*'  END;    RETURN @ret  END;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Scalar function returning the text representation of the Status column in the Sales­Order­Header table.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Get­Sales­Order­Status­Text', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the scalar function ufn­Get­Sales­Order­Status­Text. Enter a valid integer.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Get­Sales­Order­Status­Text', 'PARAMETER', N'@Status'  GO |

|  |
| --- |
| [dbo].[ufn­Get­Stock] |

MS\_­Description

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

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Product­ID Input parameter for the scalar function ufn­Get­Stock. Enter a valid Product­ID from the Production.Product­Inventory table. | int | 4 |

SQL Script

|  |
| --- |
| CREATE FUNCTION [dbo].[ufn­Get­Stock](@Product­ID [int])  RETURNS [int]  AS  -- Returns the stock level for the product. This function is used internally only  BEGIN  DECLARE @ret int;    SELECT @ret = SUM(p.[Quantity])  FROM [Production].[Product­Inventory] p  WHERE p.[Product­ID] = @Product­ID  AND p.[Location­ID] = '6'; -- Only look at inventory in the misc storage    IF (@ret IS NULL)  SET @ret = 0    RETURN @ret  END;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Scalar function returning the quantity of inventory in Location­ID 6 (Miscellaneous Storage)for a specified Product­ID.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Get­Stock', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the scalar function ufn­Get­Stock. Enter a valid Product­ID from the Production.Product­Inventory table.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Get­Stock', 'PARAMETER', N'@Product­ID'  GO |

Uses

[[Production].[Product­Inventory]](#JnGLivmNHPEgSuzGYyO0WefiqDw=)

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

SQL Script

|  |
| --- |
| CREATE FUNCTION [dbo].[ufn­Leading­Zeros](  @Value int  )  RETURNS varchar(8)  WITH SCHEMABINDING  AS  BEGIN  DECLARE @Return­Value varchar(8);  SET @Return­Value = CONVERT(varchar(8), @Value);  SET @Return­Value = REPLICATE('0', 8 - DATALENGTH(@Return­Value)) + @Return­Value;  RETURN (@Return­Value);  END;  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Scalar function used by the Sales.Customer table to help set the account number.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Leading­Zeros', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Input parameter for the scalar function ufn­Leading­Zeros. Enter a valid integer.', 'SCHEMA', N'dbo', 'FUNCTION', N'ufn­Leading­Zeros', 'PARAMETER', N'@Value'  GO |

Used By

[[Sales].[Customer]](#UK53WWAjbLvEqOKY0KByf769x9M=)

|  |
| --- |
| Database Triggers |

Objects

|  |
| --- |
| Name |
| [ddl­Database­Trigger­Log](#p1Mmvdtgs2cfWxslvgENDSSqRBg=) 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 |

SQL Script

|  |
| --- |
| CREATE TRIGGER [ddl­Database­Trigger­Log] ON DATABASE  FOR DDL\_­DATABASE\_­LEVEL\_­EVENTS AS  BEGIN  SET NOCOUNT ON;  DECLARE @data XML;  DECLARE @schema sysname;  DECLARE @object sysname;  DECLARE @event­Type sysname;  SET @data = EVENTDATA();  SET @event­Type = @data.value('(/EVENT\_­INSTANCE/Event­Type)[1]', 'sysname');  SET @schema = @data.value('(/EVENT\_­INSTANCE/Schema­Name)[1]', 'sysname');  SET @object = @data.value('(/EVENT\_­INSTANCE/Object­Name)[1]', 'sysname')  IF @object IS NOT NULL  PRINT ' ' + @event­Type + ' - ' + @schema + '.' + @object;  ELSE  PRINT ' ' + @event­Type + ' - ' + @schema;  IF @event­Type IS NULL  PRINT CONVERT(nvarchar(max), @data);  INSERT [dbo].[Database­Log]  (  [Post­Time],  [Database­User],  [Event],  [Schema],  [Object],  [TSQL],  [Xml­Event]  )  VALUES  (  GETDATE(),  CONVERT(sysname, CURRENT\_­USER),  @event­Type,  CONVERT(sysname, @schema),  CONVERT(sysname, @object),  @data.value('(/EVENT\_­INSTANCE/TSQLCommand)[1]', 'nvarchar(max)'),  @data  );  END;  GO  DISABLE TRIGGER ddl­Database­Trigger­Log ON DATABASE  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Database trigger to audit all of the DDL changes made to the Adventure­Works 2016 database.', 'TRIGGER', N'ddl­Database­Trigger­Log', NULL, NULL, NULL, NULL  GO |

|  |
| --- |
| User-Defined Data Types |

Objects

|  |
| --- |
| Name |
| [dbo.Account­Number](#wfUPMi1TrfOhNsrHkC5PC5H/4RE=) |
| [dbo.Flag](#j88jjDtwCbK8khz/m8jxryH/SFk=) |
| [dbo.Name](#j6R3OCcicAWC4on+/yNYdrBS0c0=) |
| [dbo.Name­Style](#WOmoxZnY6bzyL6GPFaVQqJCk+Sw=) |
| [dbo.Order­Number](#SZqX2/vCAFjEqiLsacel6rQr3G4=) |
| [dbo.Phone](#Dz5dpqul0rw8eU0/JuKLh0EO1ek=) |

|  |
| --- |
| [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]](#yfc9dLywcQtxtbztLjV6J3bb0tM=)

[[Sales].[Sales­Order­Header]](#J0oqEfXKE3WFfJ7PoPfs1QUbO+c=)

|  |
| --- |
| [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]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=)

[[Person].[State­Province]](#fYuToAkdHMSws7pXek8EIcFTfvY=)

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[[Production].[Product­Product­Photo]](#40voS36ihwbm/xtmbxsgEQceBOQ=)

[[Purchasing].[Vendor]](#yfc9dLywcQtxtbztLjV6J3bb0tM=)

[[Sales].[Sales­Order­Header]](#J0oqEfXKE3WFfJ7PoPfs1QUbO+c=)

[[Person].[v­State­Province­Country­Region]](#9RXeFj8QLY2V2kjrQ9Nq0mUcx70=)

[[Human­Resources].[usp­Update­Employee­Hire­Info]](#6ydkl6OBzxZTt8aQ90AP6hr4cFU=)

[[Human­Resources].[usp­Update­Employee­Login]](#C5oNhWtNH5mLMaKD+WajVUPlL6E=)

|  |
| --- |
| [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]](#TH5R06keykSJYajBUWfR3phTY8s=)

[[Human­Resources].[Shift]](#oOQXice18uca5/e1dhJkYUmlNt0=)

[[Person].[Address­Type]](#sC112VucbUsBFhIBrJnCpuBv1os=)

[[Person].[Contact­Type]](#uyTva1snlxlEidD4b0wN1HQzMw4=)

[[Person].[Country­Region]](#efOGmVlBeofSPEY2jc0F91joV9Q=)

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[[Person].[Phone­Number­Type]](#05nCels02cq7i+PWSDjMFGEmWFU=)

[[Person].[State­Province]](#fYuToAkdHMSws7pXek8EIcFTfvY=)

[[Production].[Culture]](#D4g0rs1owPY1TNtNA3F8BcetcxU=)

[[Production].[Location]](#ciHD61a29Ub1OJEWhZ1zarNhvcs=)

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[[Production].[Product­Category]](#upfoEFG3EP8VWKmQ3NJKJ7V9JTs=)

[[Production].[Product­Model]](#n9VdHBX3Ji5OWfmMzrHHD+J8HSM=)

[[Production].[Product­Review]](#Z1+PLnel1CKvek/LdN60/nVl5cI=)

[[Production].[Product­Subcategory]](#WthFJ9hmewdhBB85kY7NAjtek7Y=)

[[Production].[Scrap­Reason]](#0dK3gJTjqlelVRIns1VtvoCHDlU=)

[[Production].[Unit­Measure]](#11scPpeMkfJepgAP6eC9jOIFK+Y=)

[[Purchasing].[Ship­Method]](#Xt5ka3ip5ljwxYo+n5TjCRZypoY=)

[[Purchasing].[Vendor]](#yfc9dLywcQtxtbztLjV6J3bb0tM=)

[[Sales].[Currency]](#FEyILkTLRQRZ4zkAPe+AjFF8vg8=)

[[Sales].[Sales­Reason]](#a3VrLxK/Cx1h+4jZ069Ywnkt+DM=)

[[Sales].[Sales­Tax­Rate]](#qRl1u9t1FSaWq4TfyLPf6H9xl4k=)

[[Sales].[Sales­Territory]](#hvNbyL+2AhjcTkzEdMCOV3mg8XQ=)

[[Sales].[Store]](#o4yE5vuw4wTVa38E7sxNcXx3KhA=)

[[Human­Resources].[v­Employee]](#TVXJJHSO4vPDsqDtNRd5tGQdsO0=)

[[Human­Resources].[v­Employee­Department]](#JhbW+DYcE93jITPkfvVrtIWEHzE=)

[[Human­Resources].[v­Employee­Department­History]](#Ppb9efXEcRDxKIGzLbHZkOB+Nto=)

[[Person].[v­Additional­Contact­Info]](#VDCcGn5Gs1OPJIG7Ywjk/m2DaqY=)

[[Person].[v­State­Province­Country­Region]](#9RXeFj8QLY2V2kjrQ9Nq0mUcx70=)

[[Production].[v­Product­And­Description]](#2wtLw+LERdCNthBlev7/7Ew1+JQ=)

[[Production].[v­Product­Model­Catalog­Description]](#j53/ggQeAT7efkm5uXeLRIbSppI=)

[[Production].[v­Product­Model­Instructions]](#GCJbY/fUa640Nyz3prLsQye2L+M=)

[[Purchasing].[v­Vendor­With­Addresses]](#NdlQ62Q6DZw3GS30J9H9dm/SOqg=)

[[Purchasing].[v­Vendor­With­Contacts]](#kSLkxLBh/Shb6NobziJenpOvdFo=)

[[Sales].[v­Individual­Customer]](#WcBS54CxROpNMbufsyU5V6t9Ykw=)

[[Sales].[v­Sales­Person]](#pOjFS54tNwjD3Yvi8IMnn7dtRuA=)

[[Sales].[v­Sales­Person­Sales­By­Fiscal­Years]](#GfF5ZKdsT1+xL6X5ks+ghnXIfng=)

[[Sales].[v­Store­With­Addresses]](#lyHdOUZeccdbPaj16bKzv2RjJKw=)

[[Sales].[v­Store­With­Contacts]](#HCZuGdamTtZV4FxWl+4pACIsexM=)

[[Sales].[v­Store­With­Demographics]](#Y5BMY9GzFlSsci5vrMo/PYvMfsM=)

|  |
| --- |
| [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]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

|  |
| --- |
| [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]](#J0oqEfXKE3WFfJ7PoPfs1QUbO+c=)

|  |
| --- |
| [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]](#NyzXUZBpBs0rFeuVHGsZlpmRP0Y=)

[[Human­Resources].[v­Employee]](#TVXJJHSO4vPDsqDtNRd5tGQdsO0=)

[[Purchasing].[v­Vendor­With­Contacts]](#kSLkxLBh/Shb6NobziJenpOvdFo=)

[[Sales].[v­Individual­Customer]](#WcBS54CxROpNMbufsyU5V6t9Ykw=)

[[Sales].[v­Sales­Person]](#pOjFS54tNwjD3Yvi8IMnn7dtRuA=)

[[Sales].[v­Store­With­Contacts]](#HCZuGdamTtZV4FxWl+4pACIsexM=)

|  |
| --- |
| XML Schema Collections |

Objects

|  |
| --- |
| Name |
| [Human­Resources.HRResume­Schema­Collection](#l5jXOsaY8noZjfgneQd+33ebs58=) Collection of XML schemas for the Resume column in the Human­Resources.Job­Candidate table. |
| [Person.Additional­Contact­Info­Schema­Collection](#KOFxa4KV3R/J1jPMka6b7eLe2ZM=) Collection of XML schemas for the Additional­Contact­Info column in the Person.Contact table. |
| [Person.Individual­Survey­Schema­Collection](#KUt3GRqxMZv6wiyAwm6mIpRjcgc=) Collection of XML schemas for the Demographics column in the Person.Person table. |
| [Production.Manu­Instructions­Schema­Collection](#8FXP/Rirpa1dKaRW9DMCZBD9740=) Collection of XML schemas for the Instructions column in the Production.Product­Model table. |
| [Production.Product­Description­Schema­Collection](#ZaR2DNJAzm2r+KkNxKx06/AFx84=) Collection of XML schemas for the Catalog­Description column in the Production.Product­Model table. |
| [Sales.Store­Survey­Schema­Collection](#Y25RTEvxjWoBOBvJpbWnAGVIQnU=) 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]](#WVRXAhzXWcgHQbdlg570h3Fv6QI=)

SQL Script

|  |
| --- |
| 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" target­Namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume" element­Form­Default="qualified">  <xsd:element name="Address" type="t:Address­Type" />  <xsd:element name="Education" type="t:Education­Type" />  <xsd:element name="Employment" type="t:Employment­Type" />  <xsd:element name="Location" type="t:Location­Type" />  <xsd:element name="Name" type="t:Name­Type" />  <xsd:element name="Resume" type="t:Resume­Type" />  <xsd:element name="Telephone" type="t:Telephone­Type" />  <xsd:complex­Type name="Address­Type">  <xsd:complex­Content>  <xsd:restriction base="xsd:any­Type">  <xsd:sequence>  <xsd:element name="Addr.Type" type="xsd:string" />  <xsd:element name="Addr.Org­Name" type="xsd:string" min­Occurs="0" />  <xsd:element name="Addr.Street" type="xsd:string" max­Occurs="unbounded" />  <xsd:element name="Addr.Location">  <xsd:complex­Type>  <xsd:complex­Content>  <xsd:restriction base="xsd:any­Type">  <xsd:sequence>  <xsd:element ref="t:Location" />  </xsd:sequence>  </xsd:restriction>  </xsd:complex­Content>  </xsd:complex­Type>  </xsd:element>  <xsd:element name="Addr.Postal­Code" type="xsd:string" />  <xsd:element name="Addr.Telephone" min­Occurs="0">  <xsd:complex­Type>  <xsd:complex­Content>  <xsd:restriction base="xsd:any­Type">  <xsd:sequence>  <xsd:element ref="t:Telephone" max­Occurs="unbounded" />  </xsd:sequence>  </xsd:restriction>  </xsd:complex­Content>  </xsd:complex­Type>  </xsd:element>  </xsd:sequence>  </xsd:restriction>  </xsd:complex­Content>  </xsd:complex­Type>  <xsd: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">  <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](#wRNBi9QOObVjEO+Wxjr4GvvwBBs=)

Used By

[[Human­Resources].[Job­Candidate]](#WVRXAhzXWcgHQbdlg570h3Fv6QI=)

|  |
| --- |
| [Person].[Additional­Contact­Info­Schema­Collection] |

MS\_­Description

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

Dependent Columns

* [[Person].[Person].[Additional­Contact­Info]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)
* [[Person].[Person].[Additional­Contact­Info]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

SQL Script

|  |
| --- |
| CREATE XML SCHEMA COLLECTION [Person].[Additional­Contact­Info­Schema­Collection]  AS N'<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:t="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact­Info" target­Namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact­Info">  <xsd:element name="Additional­Contact­Info">  <xsd:complex­Type mixed="true">  <xsd:complex­Content mixed="true">  <xsd:restriction base="xsd:any­Type">  <xsd:sequence>  <xsd:any namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact­Record http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact­Types" min­Occurs="0" max­Occurs="unbounded" />  </xsd:sequence>  </xsd:restriction>  </xsd:complex­Content>  </xsd:complex­Type>  </xsd:element>  </xsd:schema>  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:t="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact­Record" target­Namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact­Record">  <xsd:element name="Contact­Record">  <xsd:complex­Type mixed="true">  <xsd:complex­Content mixed="true">  <xsd:restriction base="xsd:any­Type">  <xsd:choice min­Occurs="0" max­Occurs="unbounded">  <xsd:any namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/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" xmlns:t="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact­Types" target­Namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Contact­Types" element­Form­Default="qualified">  <xsd:element name="e­Mail" type="t:e­Mail­Type" />  <xsd:element name="facsimile­Telephone­Number" type="t:phone­Number­Type" />  <xsd: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" min­Occurs="0" />  </xsd:sequence>  </xsd:restriction>  </xsd:complex­Content>  </xsd:complex­Type>  <xsd:complex­Type name="e­Mail­Type">  <xsd:complex­Content>  <xsd:restriction base="xsd:any­Type">  <xsd:sequence>  <xsd:element name="e­Mail­Address" type="xsd:string" />  <xsd:element name="Special­Instructions" type="t:special­Instructions­Type" min­Occurs="0" />  </xsd:sequence>  </xsd:restriction>  </xsd:complex­Content>  </xsd:complex­Type>  <xsd:complex­Type name="phone­Number­Type">  <xsd:complex­Content>  <xsd:restriction base="xsd:any­Type">  <xsd:sequence>  <xsd:element name="number">  <xsd:simple­Type>  <xsd:restriction base="xsd:string">  <xsd:pattern value="[0-9\(\)\-]\*" />  </xsd:restriction>  </xsd:simple­Type>  </xsd:element>  <xsd:element name="Special­Instructions" type="t:special­Instructions­Type" min­Occurs="0" />  </xsd:sequence>  </xsd:restriction>  </xsd:complex­Content>  </xsd:complex­Type>  <xsd:complex­Type name="special­Instructions­Type" mixed="true">  <xsd:complex­Content mixed="true">  <xsd:restriction base="xsd:any­Type">  <xsd:sequence>  <xsd:any namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/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](#GSXHhzVRNjGQYtnCQHosa2uthyw=)

Used By

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

|  |
| --- |
| [Person].[Individual­Survey­Schema­Collection] |

MS\_­Description

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

Dependent Columns

* [[Person].[Person].[Demographics]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)
* [[Person].[Person].[Demographics]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

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/Individual­Survey" target­Namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Individual­Survey" element­Form­Default="qualified">  <xsd:element name="Individual­Survey">  <xsd:complex­Type>  <xsd:complex­Content>  <xsd:restriction base="xsd:any­Type">  <xsd:sequence>  <xsd:element name="Total­Purchase­YTD" type="xsd:decimal" min­Occurs="0" />  <xsd:element name="Date­First­Purchase" type="xsd:date" min­Occurs="0" />  <xsd:element name="Birth­Date" type="xsd:date" min­Occurs="0" />  <xsd:element name="Marital­Status" type="xsd:string" min­Occurs="0" />  <xsd:element name="Yearly­Income" type="t:Salary­Type" min­Occurs="0" />  <xsd:element name="Gender" type="xsd:string" min­Occurs="0" />  <xsd:element name="Total­Children" type="xsd:int" min­Occurs="0" />  <xsd:element name="Number­Children­At­Home" type="xsd:int" min­Occurs="0" />  <xsd:element name="Education" type="xsd:string" min­Occurs="0" />  <xsd:element name="Occupation" type="xsd:string" min­Occurs="0" />  <xsd:element name="Home­Owner­Flag" type="xsd:string" min­Occurs="0" />  <xsd:element name="Number­Cars­Owned" type="xsd:int" min­Occurs="0" />  <xsd:element name="Hobby" type="xsd:string" min­Occurs="0" max­Occurs="unbounded" />  <xsd:element name="Commute­Distance" type="t:Mile­Range­Type" min­Occurs="0" />  <xsd:element name="Comments" type="xsd:string" min­Occurs="0" />  </xsd:sequence>  </xsd:restriction>  </xsd:complex­Content>  </xsd:complex­Type>  </xsd:element>  <xsd:simple­Type name="Mile­Range­Type">  <xsd:restriction base="xsd:string">  <xsd:enumeration value="0-1 Miles" />  <xsd:enumeration value="1-2 Miles" />  <xsd:enumeration value="2-5 Miles" />  <xsd:enumeration value="5-10 Miles" />  <xsd:enumeration value="10+ Miles" />  </xsd:restriction>  </xsd:simple­Type>  <xsd:simple­Type name="Salary­Type">  <xsd:restriction base="xsd:string">  <xsd:enumeration value="0-25000" />  <xsd:enumeration value="25001-50000" />  <xsd:enumeration value="50001-75000" />  <xsd:enumeration value="75001-100000" />  <xsd:enumeration value="greater than 100000" />  </xsd:restriction>  </xsd:simple­Type>  </xsd:schema>'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Collection of XML schemas for the Demographics column in the Person.Person table.', 'SCHEMA', N'Person', 'XML SCHEMA COLLECTION', N'Individual­Survey­Schema­Collection', NULL, NULL  GO |

Uses

[Person](#GSXHhzVRNjGQYtnCQHosa2uthyw=)

Used By

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

|  |
| --- |
| [Production].[Manu­Instructions­Schema­Collection] |

MS\_­Description

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

Dependent Columns

* [[Production].[Product­Model].[Instructions]](#n9VdHBX3Ji5OWfmMzrHHD+J8HSM=)
* [[Production].[Product­Model].[Instructions]](#n9VdHBX3Ji5OWfmMzrHHD+J8HSM=)

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/Product­Model­Manu­Instructions" target­Namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Manu­Instructions" 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" />  </xsd:sequence>  <xsd:attribute name="Location­ID" type="xsd:integer" use="required" />  <xsd:attribute name="Setup­Hours" type="xsd:decimal" />  <xsd:attribute name="Machine­Hours" type="xsd:decimal" />  <xsd:attribute name="Labor­Hours" type="xsd:decimal" />  <xsd:attribute name="Lot­Size" type="xsd:decimal" />  </xsd:restriction>  </xsd:complex­Content>  </xsd:complex­Type>  </xsd:element>  </xsd:sequence>  </xsd:restriction>  </xsd:complex­Content>  </xsd:complex­Type>  </xsd:element>  <xsd:complex­Type name="Step­Type" mixed="true">  <xsd:complex­Content mixed="true">  <xsd:restriction base="xsd:any­Type">  <xsd:choice min­Occurs="0" max­Occurs="unbounded">  <xsd:element name="tool" type="xsd:string" />  <xsd:element name="material" type="xsd:string" />  <xsd:element name="blueprint" type="xsd:string" />  <xsd:element name="specs" type="xsd:string" />  <xsd:element name="diag" type="xsd:string" />  </xsd:choice>  </xsd:restriction>  </xsd:complex­Content>  </xsd:complex­Type>  </xsd:schema>'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Collection of XML schemas for the Instructions column in the Production.Product­Model table.', 'SCHEMA', N'Production', 'XML SCHEMA COLLECTION', N'Manu­Instructions­Schema­Collection', NULL, NULL  GO |

Uses

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

Used By

[[Production].[Product­Model]](#n9VdHBX3Ji5OWfmMzrHHD+J8HSM=)

|  |
| --- |
| [Production].[Product­Description­Schema­Collection] |

MS\_­Description

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

Dependent Columns

* [[Production].[Product­Model].[Catalog­Description]](#n9VdHBX3Ji5OWfmMzrHHD+J8HSM=)
* [[Production].[Product­Model].[Catalog­Description]](#n9VdHBX3Ji5OWfmMzrHHD+J8HSM=)

SQL Script

|  |
| --- |
| CREATE XML SCHEMA COLLECTION [Production].[Product­Description­Schema­Collection]  AS N'<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:t="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Warr­And­Main" 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" xmlns:ns1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Warr­And­Main" xmlns:t="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Description" target­Namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Description" element­Form­Default="qualified">  <xsd:import namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Product­Model­Warr­And­Main" />  <xsd:element name="Code" type="xsd:string" />  <xsd:element name="Description" type="xsd:string" />  <xsd:element name="Product­Description" type="t:Product­Description" />  <xsd:element name="Taxonomy" type="xsd:string" />  <xsd:complex­Type name="Category">  <xsd:complex­Content>  <xsd:restriction base="xsd:any­Type">  <xsd:sequence>  <xsd:element ref="t:Taxonomy" />  <xsd:element ref="t:Code" />  <xsd:element ref="t:Description" min­Occurs="0" />  </xsd:sequence>  </xsd:restriction>  </xsd:complex­Content>  </xsd:complex­Type>  <xsd:complex­Type name="Features" mixed="true">  <xsd:complex­Content mixed="true">  <xsd:restriction base="xsd:any­Type">  <xsd:sequence>  <xsd:element ref="ns1:Warranty" />  <xsd:element ref="ns1:Maintenance" />  <xsd:any namespace="##other" process­Contents="skip" min­Occurs="0" max­Occurs="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" max­Occurs="unbounded" />  <xsd:element name="Picture" type="t:Picture" min­Occurs="0" max­Occurs="unbounded" />  <xsd:element name="Category" type="t:Category" min­Occurs="0" max­Occurs="unbounded" />  <xsd:element name="Specifications" type="t:Specifications" min­Occurs="0" max­Occurs="unbounded" />  </xsd:sequence>  <xsd:attribute name="Product­Model­ID" type="xsd:string" />  <xsd:attribute name="Product­Model­Name" type="xsd:string" />  </xsd:restriction>  </xsd:complex­Content>  </xsd:complex­Type>  <xsd:complex­Type name="Specifications" mixed="true">  <xsd:complex­Content mixed="true">  <xsd:restriction base="xsd:any­Type">  <xsd:sequence>  <xsd:any process­Contents="skip" min­Occurs="0" max­Occurs="unbounded" />  </xsd:sequence>  </xsd:restriction>  </xsd:complex­Content>  </xsd:complex­Type>  <xsd:complex­Type name="Summary" mixed="true">  <xsd:complex­Content mixed="true">  <xsd:restriction base="xsd:any­Type">  <xsd:sequence>  <xsd:any namespace="http://www.w3.org/1999/xhtml" process­Contents="skip" min­Occurs="0" max­Occurs="unbounded" />  </xsd:sequence>  </xsd:restriction>  </xsd:complex­Content>  </xsd:complex­Type>  </xsd:schema>'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Collection of XML schemas for the Catalog­Description column in the Production.Product­Model table.', 'SCHEMA', N'Production', 'XML SCHEMA COLLECTION', N'Product­Description­Schema­Collection', NULL, NULL  GO |

Uses

[Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=)

Used By

[[Production].[Product­Model]](#n9VdHBX3Ji5OWfmMzrHHD+J8HSM=)

|  |
| --- |
| [Sales].[Store­Survey­Schema­Collection] |

MS\_­Description

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

Dependent Columns

* [[Sales].[Store].[Demographics]](#o4yE5vuw4wTVa38E7sxNcXx3KhA=)
* [[Sales].[Store].[Demographics]](#o4yE5vuw4wTVa38E7sxNcXx3KhA=)

SQL Script

|  |
| --- |
| CREATE XML SCHEMA COLLECTION [Sales].[Store­Survey­Schema­Collection]  AS N'<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:t="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Store­Survey" target­Namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Store­Survey" element­Form­Default="qualified">  <xsd:element name="Store­Survey">  <xsd:complex­Type>  <xsd:complex­Content>  <xsd:restriction base="xsd:any­Type">  <xsd:sequence>  <xsd:element name="Contact­Name" type="xsd:string" min­Occurs="0" />  <xsd:element name="Job­Title" type="xsd:string" min­Occurs="0" />  <xsd:element name="Annual­Sales" type="xsd:decimal" min­Occurs="0" />  <xsd:element name="Annual­Revenue" type="xsd:decimal" min­Occurs="0" />  <xsd:element name="Bank­Name" type="xsd:string" min­Occurs="0" />  <xsd:element name="Business­Type" type="t:Business­Type" min­Occurs="0" />  <xsd:element name="Year­Opened" type="xsd:g­Year" min­Occurs="0" />  <xsd:element name="Specialty" type="t:Specialty­Type" min­Occurs="0" />  <xsd:element name="Square­Feet" type="xsd:float" min­Occurs="0" />  <xsd:element name="Brands" type="t:Brand­Type" min­Occurs="0" />  <xsd:element name="Internet" type="t:Internet­Type" min­Occurs="0" />  <xsd:element name="Number­Employees" type="xsd:int" min­Occurs="0" />  <xsd:element name="Comments" type="xsd:string" min­Occurs="0" />  </xsd:sequence>  </xsd:restriction>  </xsd:complex­Content>  </xsd:complex­Type>  </xsd:element>  <xsd:simple­Type name="Brand­Type">  <xsd:restriction base="xsd:string">  <xsd:enumeration value="AW" />  <xsd:enumeration value="2" />  <xsd:enumeration value="3" />  <xsd:enumeration value="4+" />  </xsd:restriction>  </xsd:simple­Type>  <xsd:simple­Type name="Business­Type">  <xsd:restriction base="xsd:string">  <xsd:enumeration value="BM" />  <xsd:enumeration value="BS" />  <xsd:enumeration value="D" />  <xsd:enumeration value="OS" />  <xsd:enumeration value="SGS" />  </xsd:restriction>  </xsd:simple­Type>  <xsd:simple­Type name="Internet­Type">  <xsd:restriction base="xsd:string">  <xsd:enumeration value="56kb" />  <xsd:enumeration value="ISDN" />  <xsd:enumeration value="DSL" />  <xsd:enumeration value="T1" />  <xsd:enumeration value="T2" />  <xsd:enumeration value="T3" />  </xsd:restriction>  </xsd:simple­Type>  <xsd:simple­Type name="Specialty­Type">  <xsd:restriction base="xsd:string">  <xsd:enumeration value="Family" />  <xsd:enumeration value="Kids" />  <xsd:enumeration value="BMX" />  <xsd:enumeration value="Touring" />  <xsd:enumeration value="Road" />  <xsd:enumeration value="Mountain" />  <xsd:enumeration value="All" />  </xsd:restriction>  </xsd:simple­Type>  </xsd:schema>'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Collection of XML schemas for the Demographics column in the Sales.Store table.', 'SCHEMA', N'Sales', 'XML SCHEMA COLLECTION', N'Store­Survey­Schema­Collection', NULL, NULL  GO |

Uses

[Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=)

Used By

[[Sales].[Store]](#o4yE5vuw4wTVa38E7sxNcXx3KhA=)

|  |
| --- |
| Full Text Catalogs |

Objects

|  |
| --- |
| Name |
| [dbo.AW2016Full­Text­Catalog](#KQM1/2huuuPy+pMNx4F37D5+p0Y=) |

|  |
| --- |
| AW2016Full­Text­Catalog |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | [dbo](#YEiKkgju4q/EYI9jbGpwHRIoEIM=) |
| Default | True |
| Accent Sensitive | True |

Tables

* [Job­Candidate](#WVRXAhzXWcgHQbdlg570h3Fv6QI=)
* [Document](#v5/qynDVNwkOWZdAe2xATpOnRWk=)
* [Product­Review](#Z1+PLnel1CKvek/LdN60/nVl5cI=)

SQL Script

|  |
| --- |
| CREATE FULLTEXT CATALOG [AW2016Full­Text­Catalog]  WITH ACCENT\_­SENSITIVITY = ON  AS DEFAULT  AUTHORIZATION [dbo]  GO  CREATE FULLTEXT INDEX ON [Human­Resources].[Job­Candidate] KEY INDEX [PK\_­Job­Candidate\_­Job­Candidate­ID] ON [AW2016Full­Text­Catalog]  GO  CREATE FULLTEXT INDEX ON [Production].[Document] KEY INDEX [PK\_­Document\_­Document­Node] ON [AW2016Full­Text­Catalog]  GO  CREATE FULLTEXT INDEX ON [Production].[Product­Review] KEY INDEX [PK\_­Product­Review\_­Product­Review­ID] ON [AW2016Full­Text­Catalog]  GO |

Uses

[[Human­Resources].[Job­Candidate]](#WVRXAhzXWcgHQbdlg570h3Fv6QI=)

[[Production].[Document]](#v5/qynDVNwkOWZdAe2xATpOnRWk=)

[[Production].[Product­Review]](#Z1+PLnel1CKvek/LdN60/nVl5cI=)

|  |
| --- |
| Users |

Objects

|  |
| --- |
| Name |
| [dbo](#YEiKkgju4q/EYI9jbGpwHRIoEIM=) |
| [guest](#dFKsdgk9Z+ECSEH+YiXwoZ7EzSY=) |

|  |
| --- |
| dbo |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Type | Windows­User |
| Default Schema | dbo |

Database Level Permissions

|  |  |
| --- | --- |
| Type | Action |
| CONNECT | Grant |

SQL Script

|  |
| --- |
| GO |

|  |
| --- |
| guest |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Type | Sql­User |
| Default Schema | guest |

SQL Script

|  |
| --- |
| GO |

|  |
| --- |
| Database Roles |

Objects

|  |
| --- |
| Name |
| [db\_accessadmin](#rud42PEk4VGGUxPbnFs4+Fkd5Gw=) |
| [db\_backupoperator](#6ahADrtcCV+qeOTA3bSCTQ2+P80=) |
| [db\_datareader](#ty7gspobmhGfTXvvJA9NpdmNvt4=) |
| [db\_datawriter](#xQClRoyrdQNlnsvhRTczOJfAhuk=) |
| [db\_ddladmin](#3EGJ/FSCwhLb3b1D8oYZL68hRYw=) |
| [db\_denydatareader](#r8bRd2m7tYvE9CPUMXamyNC7f04=) |
| [db\_denydatawriter](#x7aQYUrzeoiyxqH8zccH3RUKkE8=) |
| [db\_owner](#tVkhPzyCpQhctZPmZ2cz+c9Bys8=) |
| [db\_securityadmin](#kETAC31WG0S1z5GLVbwR00aVQ7I=) |
| [public](#EWSAEBEyjmunmnQiwYqE6dLPJvE=) |

|  |
| --- |
| db\_accessadmin |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | [dbo](#YEiKkgju4q/EYI9jbGpwHRIoEIM=) |

|  |
| --- |
| db\_backupoperator |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | [dbo](#YEiKkgju4q/EYI9jbGpwHRIoEIM=) |

|  |
| --- |
| db\_datareader |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | [dbo](#YEiKkgju4q/EYI9jbGpwHRIoEIM=) |

|  |
| --- |
| db\_datawriter |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | [dbo](#YEiKkgju4q/EYI9jbGpwHRIoEIM=) |

|  |
| --- |
| db\_ddladmin |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | [dbo](#YEiKkgju4q/EYI9jbGpwHRIoEIM=) |

|  |
| --- |
| db\_denydatareader |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | [dbo](#YEiKkgju4q/EYI9jbGpwHRIoEIM=) |

|  |
| --- |
| db\_denydatawriter |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | [dbo](#YEiKkgju4q/EYI9jbGpwHRIoEIM=) |

|  |
| --- |
| db\_owner |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | [dbo](#YEiKkgju4q/EYI9jbGpwHRIoEIM=) |

|  |
| --- |
| db\_securityadmin |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | [dbo](#YEiKkgju4q/EYI9jbGpwHRIoEIM=) |

|  |
| --- |
| public |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | [dbo](#YEiKkgju4q/EYI9jbGpwHRIoEIM=) |

|  |
| --- |
| Schemas |

Objects

|  |
| --- |
| Name |
| [Human­Resources](#wRNBi9QOObVjEO+Wxjr4GvvwBBs=) Contains objects related to employees and departments. |
| [Person](#GSXHhzVRNjGQYtnCQHosa2uthyw=) Contains objects related to names and addresses of customers, vendors, and employees |
| [Production](#hroRDOFJt+sxSV6kbLuFXyt9c+I=) Contains objects related to products, inventory, and manufacturing. |
| [Purchasing](#r48uLAuwsJTIAvAuIPBTbVrjT9s=) Contains objects related to vendors and purchase orders. |
| [Sales](#LW/LQ9VeVLmciegNxjapKG86l8M=) Contains objects related to customers, sales orders, and sales territories. |

|  |
| --- |
| Human­Resources |

MS\_­Description

Contains objects related to employees and departments.

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | [dbo](#YEiKkgju4q/EYI9jbGpwHRIoEIM=) |

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]](#TH5R06keykSJYajBUWfR3phTY8s=)

[[Human­Resources].[Employee]](#Kjz9lDvrP3nS1nwr2lPE3hJmy04=)

[[Human­Resources].[Employee­Department­History]](#tDjny2Q9sWvuqg97wWzd145nXkA=)

[[Human­Resources].[Employee­Pay­History]](#WzREUJ9VRXbYWerHGwiGJinYM0A=)

[[Human­Resources].[Job­Candidate]](#WVRXAhzXWcgHQbdlg570h3Fv6QI=)

[[Human­Resources].[Shift]](#oOQXice18uca5/e1dhJkYUmlNt0=)

[[Human­Resources].[v­Employee]](#TVXJJHSO4vPDsqDtNRd5tGQdsO0=)

[[Human­Resources].[v­Employee­Department]](#JhbW+DYcE93jITPkfvVrtIWEHzE=)

[[Human­Resources].[v­Employee­Department­History]](#Ppb9efXEcRDxKIGzLbHZkOB+Nto=)

[[Human­Resources].[v­Job­Candidate]](#OJ+5x6Pf56todcFiq0tOA8QJbeA=)

[[Human­Resources].[v­Job­Candidate­Education]](#L3qRlBiBKB45IvcTjpNzlK4Zu0c=)

[[Human­Resources].[v­Job­Candidate­Employment]](#xDdPFCC7cBQwDROim4Heve+/Okw=)

[[Human­Resources].[usp­Update­Employee­Hire­Info]](#6ydkl6OBzxZTt8aQ90AP6hr4cFU=)

[[Human­Resources].[usp­Update­Employee­Login]](#C5oNhWtNH5mLMaKD+WajVUPlL6E=)

[[Human­Resources].[usp­Update­Employee­Personal­Info]](#vCx7xMQcjRyuiOsUmNSNg9QeITY=)

[[Human­Resources].[HRResume­Schema­Collection]](#l5jXOsaY8noZjfgneQd+33ebs58=)

|  |
| --- |
| Person |

MS\_­Description

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

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | [dbo](#YEiKkgju4q/EYI9jbGpwHRIoEIM=) |

SQL Script

|  |
| --- |
| CREATE SCHEMA [Person]  AUTHORIZATION [dbo]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Contains objects related to names and addresses of customers, vendors, and employees', 'SCHEMA', N'Person', NULL, NULL, NULL, NULL  GO |

Used By

[[Person].[Address]](#YHZkNb8Ts04ByOupF/OAfoeI2qk=)

[[Person].[Address­Type]](#sC112VucbUsBFhIBrJnCpuBv1os=)

[[Person].[Business­Entity]](#I7t6VoTbrv7x5ZQ/kE/EZs1npdg=)

[[Person].[Business­Entity­Address]](#HK6CehnfZ4YYh7ckP8cx74El9pY=)

[[Person].[Business­Entity­Contact]](#9Zj2hWeFVzGQn5EUQXP6lLMtAZI=)

[[Person].[Contact­Type]](#uyTva1snlxlEidD4b0wN1HQzMw4=)

[[Person].[Country­Region]](#efOGmVlBeofSPEY2jc0F91joV9Q=)

[[Person].[Email­Address]](#3EL9ymWvglf8ADJZjk1nQ8j/hLY=)

[[Person].[Password]](#z1Ylcf/ZAQ6li+sOkPO88E0ccrw=)

[[Person].[Person]](#TZe4JItyjmK+s/tUWju0b9W3HXU=)

[[Person].[Person­Phone]](#NyzXUZBpBs0rFeuVHGsZlpmRP0Y=)

[[Person].[Pet]](#etnWX6SZkHYAYL8t5N4pzt7zXKE=)

[[Person].[Phone­Number­Type]](#05nCels02cq7i+PWSDjMFGEmWFU=)

[[Person].[State­Province]](#fYuToAkdHMSws7pXek8EIcFTfvY=)

[[Person].[v­Additional­Contact­Info]](#VDCcGn5Gs1OPJIG7Ywjk/m2DaqY=)

[[Person].[v­State­Province­Country­Region]](#9RXeFj8QLY2V2kjrQ9Nq0mUcx70=)

[[Person].[Additional­Contact­Info­Schema­Collection]](#KOFxa4KV3R/J1jPMka6b7eLe2ZM=)

[[Person].[Individual­Survey­Schema­Collection]](#KUt3GRqxMZv6wiyAwm6mIpRjcgc=)

|  |
| --- |
| Production |

MS\_­Description

Contains objects related to products, inventory, and manufacturing.

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | [dbo](#YEiKkgju4q/EYI9jbGpwHRIoEIM=) |

SQL Script

|  |
| --- |
| CREATE SCHEMA [Production]  AUTHORIZATION [dbo]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Contains objects related to products, inventory, and manufacturing.', 'SCHEMA', N'Production', NULL, NULL, NULL, NULL  GO |

Used By

[[Production].[Bill­Of­Materials]](#m9P+2olmZFXOowkUY18+Ri7YHx4=)

[[Production].[Culture]](#D4g0rs1owPY1TNtNA3F8BcetcxU=)

[[Production].[Document]](#v5/qynDVNwkOWZdAe2xATpOnRWk=)

[[Production].[Illustration]](#q4hmTMYciVWtuNDgg2tVsn/4PAw=)

[[Production].[Location]](#ciHD61a29Ub1OJEWhZ1zarNhvcs=)

[[Production].[Product]](#mO6TpkyW7q4W/OMRHISuYfWNCME=)

[[Production].[Product­Category]](#upfoEFG3EP8VWKmQ3NJKJ7V9JTs=)

[[Production].[Product­Cost­History]](#V4hNECfDVL8rNsab6KfgsgmqeHE=)

[[Production].[Product­Description]](#4iEXAFuDYt/4TBUTP9DP8dTGazY=)

[[Production].[Product­Document]](#hFy0+8CYqgHPJaiGh0/YlUFf+eI=)

[[Production].[Product­Inventory]](#JnGLivmNHPEgSuzGYyO0WefiqDw=)

[[Production].[Product­List­Price­History]](#lSXw7btC5MVIn+g9cscCbIFpkU0=)

[[Production].[Product­Model]](#n9VdHBX3Ji5OWfmMzrHHD+J8HSM=)

[[Production].[Product­Model­Illustration]](#ha0ZY5wgktKInuoqdKFBVRbeJaI=)

[[Production].[Product­Model­Product­Description­Culture]](#Kkf63jUYCrQKelqjjcG51JBQDNI=)

[[Production].[Product­Photo]](#wYFkOtQB0E8+2MDoAl84kiyzc9A=)

[[Production].[Product­Product­Photo]](#40voS36ihwbm/xtmbxsgEQceBOQ=)

[[Production].[Product­Review]](#Z1+PLnel1CKvek/LdN60/nVl5cI=)

[[Production].[Product­Subcategory]](#WthFJ9hmewdhBB85kY7NAjtek7Y=)

[[Production].[Scrap­Reason]](#0dK3gJTjqlelVRIns1VtvoCHDlU=)

[[Production].[Transaction­History]](#qEI6rAj76QhivliqVRIQ5mR7wrA=)

[[Production].[Transaction­History­Archive]](#mWvwPJkOXR0q1NT8gm4kJjfJ/aA=)

[[Production].[Unit­Measure]](#11scPpeMkfJepgAP6eC9jOIFK+Y=)

[[Production].[Work­Order]](#BSRICvfPko6shp7o8HSu003oCno=)

[[Production].[Work­Order­Routing]](#wxYgGkEd/o/k/roV2tY7ZCLH+qE=)

[[Production].[v­Product­And­Description]](#2wtLw+LERdCNthBlev7/7Ew1+JQ=)

[[Production].[v­Product­Model­Catalog­Description]](#j53/ggQeAT7efkm5uXeLRIbSppI=)

[[Production].[v­Product­Model­Instructions]](#GCJbY/fUa640Nyz3prLsQye2L+M=)

[[Production].[Manu­Instructions­Schema­Collection]](#8FXP/Rirpa1dKaRW9DMCZBD9740=)

[[Production].[Product­Description­Schema­Collection]](#ZaR2DNJAzm2r+KkNxKx06/AFx84=)

|  |
| --- |
| Purchasing |

MS\_­Description

Contains objects related to vendors and purchase orders.

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | [dbo](#YEiKkgju4q/EYI9jbGpwHRIoEIM=) |

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]](#igWcZ2fEqwA+AlsF0xQn66oTGAU=)

[[Purchasing].[Purchase­Order­Detail]](#dsxg+I54iAv4FByATvQGW4l6Cco=)

[[Purchasing].[Purchase­Order­Header]](#wVItRHlooKKFqOhuItRUJuA60Fs=)

[[Purchasing].[Ship­Method]](#Xt5ka3ip5ljwxYo+n5TjCRZypoY=)

[[Purchasing].[Vendor]](#yfc9dLywcQtxtbztLjV6J3bb0tM=)

[[Purchasing].[v­Vendor­With­Addresses]](#NdlQ62Q6DZw3GS30J9H9dm/SOqg=)

[[Purchasing].[v­Vendor­With­Contacts]](#kSLkxLBh/Shb6NobziJenpOvdFo=)

|  |
| --- |
| Sales |

MS\_­Description

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

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | [dbo](#YEiKkgju4q/EYI9jbGpwHRIoEIM=) |

SQL Script

|  |
| --- |
| CREATE SCHEMA [Sales]  AUTHORIZATION [dbo]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Contains objects related to customers, sales orders, and sales territories.', 'SCHEMA', N'Sales', NULL, NULL, NULL, NULL  GO |

Used By

[[Sales].[Country­Region­Currency]](#ZTDvJcpGJmjH9tZPEMXFxv/sN0o=)

[[Sales].[Credit­Card]](#Juoic8jJ3a0e1KCG+cj8LixVRbQ=)

[[Sales].[Currency]](#FEyILkTLRQRZ4zkAPe+AjFF8vg8=)

[[Sales].[Currency­Rate]](#DbnMf0tD82NB+BGbYOX9jWqScts=)

[[Sales].[Customer]](#UK53WWAjbLvEqOKY0KByf769x9M=)

[[Sales].[Person­Credit­Card]](#HOEWmD+eQ6y/+8bq8hkiLEiBBjU=)

[[Sales].[Sales­Order­Detail]](#OkvyE7uLBgi7y2P/yCWzRf5c0wE=)

[[Sales].[Sales­Order­Header]](#J0oqEfXKE3WFfJ7PoPfs1QUbO+c=)

[[Sales].[Sales­Order­Header­Sales­Reason]](#8Csb5Ws8nFAO1qWhqs6WCgPvmp4=)

[[Sales].[Sales­Person]](#hKN58zHn4ffskJv6ldcUqjf87do=)

[[Sales].[Sales­Person­Quota­History]](#ZEp3m1cn9szNTztS7qyCt1Ng+u8=)

[[Sales].[Sales­Reason]](#a3VrLxK/Cx1h+4jZ069Ywnkt+DM=)

[[Sales].[Sales­Tax­Rate]](#qRl1u9t1FSaWq4TfyLPf6H9xl4k=)

[[Sales].[Sales­Territory]](#hvNbyL+2AhjcTkzEdMCOV3mg8XQ=)

[[Sales].[Sales­Territory­History]](#STLuHwwNIM3t3qtp2pV+eqOT5iE=)

[[Sales].[Shopping­Cart­Item]](#usOp9/vzTEpC3eW1PfV8jtNCfNQ=)

[[Sales].[Special­Offer]](#H28uErAJ2leiaB1YxFBqojYITRE=)

[[Sales].[Special­Offer­Product]](#8e4RMrihzXJLXw2gPdnNJ+2ijQ0=)

[[Sales].[Store]](#o4yE5vuw4wTVa38E7sxNcXx3KhA=)

[[Sales].[v­Individual­Customer]](#WcBS54CxROpNMbufsyU5V6t9Ykw=)

[[Sales].[v­Person­Demographics]](#rOUy+E6WBf1AY40+Q9SudpuP9xU=)

[[Sales].[v­Sales­Person]](#pOjFS54tNwjD3Yvi8IMnn7dtRuA=)

[[Sales].[v­Sales­Person­Sales­By­Fiscal­Years]](#GfF5ZKdsT1+xL6X5ks+ghnXIfng=)

[[Sales].[v­Store­With­Addresses]](#lyHdOUZeccdbPaj16bKzv2RjJKw=)

[[Sales].[v­Store­With­Contacts]](#HCZuGdamTtZV4FxWl+4pACIsexM=)

[[Sales].[v­Store­With­Demographics]](#Y5BMY9GzFlSsci5vrMo/PYvMfsM=)

[[Sales].[Store­Survey­Schema­Collection]](#Y25RTEvxjWoBOBvJpbWnAGVIQnU=)