




Database documentation

Server name:	
Database name:	AdventureWorks
Documentation date:	30/03/2009
Database size:	178.75 MB
Database description:	AdventureWorks Sample OLTP Database

Table summary

	Name	Data size	Index size	Row Count
	dbo.AWBuildVersion	8 KB	8 KB	1
	dbo.DatabaseLog	1408 KB	24 KB	389
	dbo.ErrorLog	0 KB	0 KB	0
	HumanResources.Department	8 KB	24 KB	16
	HumanResources.Employee	56 KB	120 KB	290
	HumanResources.EmployeeAddress	16 KB	32 KB	290
	HumanResources.EmployeeDepartmentHistory	16 KB	48 KB	296
	HumanResources.EmployeePayHistory	16 KB	16 KB	316
	HumanResources.JobCandidate	128 KB	32 KB	13
	HumanResources.Shift	8 KB	40 KB	3
	Person.Address	2224 KB	2440 KB	19614
	Person.AddressType	8 KB	40 KB	6
	Person.Contact	4536 KB	1936 KB	19972
	Person.ContactType	8 KB	24 KB	20
	Person.CountryRegion	16 KB	32 KB	238
	Person.StateProvince	16 KB	64 KB	181
	Production.BillOfMaterials	160 KB	200 KB	2679
	Production.Culture	8 KB	24 KB	8
	Production.Document	344 KB	24 KB	9
	Production.Illustration	184 KB	8 KB	5
	Production.Location	8 KB	24 KB	14
	Production.Product	104 KB	120 KB	504
	Production.ProductCategory	8 KB	40 KB	4
	Production.ProductCostHistory	24 KB	16 KB	395
	Production.ProductDescription	144 KB	56 KB	762
	Production.ProductDocument	8 KB	8 KB	32
	Production.ProductInventory	56 KB	16 KB	1069
	Production.ProductListPriceHistory	24 KB	16 KB	395
	Production.ProductModel	144 KB	216 KB	128
	Production.ProductModelIllustration	8 KB	8 KB	7
	Production.ProductModelProductDescriptionCulture	32 KB	16 KB	762
	Production.ProductPhoto	2240 KB	16 KB	101
	Production.ProductProductPhoto	16 KB	40 KB	504
	Production.ProductReview	16 KB	56 KB	4
	Production.ProductSubcategory	8 KB	40 KB	37
	Production.ScrapReason	8 KB	24 KB	16
	Production.TransactionHistory	6304 KB	2992 KB	113443
	Production.TransactionHistoryArchive	4960 KB	2352 KB	89253
	Production.UnitMeasure	8 KB	24 KB	38
	Production.WorkOrder	4184 KB	1704 KB	72591
	Production.WorkOrderRouting	5544 KB	912 KB	67131
	Purchasing.ProductVendor	40 KB	48 KB	460
	Purchasing.PurchaseOrderDetail	512 KB	168 KB	8845
	Purchasing.PurchaseOrderHeader	336 KB	144 KB	4012
	Purchasing.ShipMethod	8 KB	40 KB	5
	Purchasing.Vendor	16 KB	32 KB	104
	Purchasing.VendorAddress	8 KB	24 KB	104
	Purchasing.VendorContact	8 KB	40 KB	156
	Sales.ContactCreditCard	480 KB	16 KB	19118

	Sales.CountryRegionCurrency	8 KB	24 KB	109
	Sales.CreditCard	1496 KB	776 KB	19118
	Sales.Currency	8 KB	24 KB	105
	Sales.CurrencyRate	768 KB	400 KB	13532
	Sales.Customer	824 KB	1176 KB	19185
	Sales.CustomerAddress	864 KB	552 KB	19220
	Sales.Individual	24648 KB	49784 KB	18484
	Sales.SalesOrderDetail	9864 KB	5144 KB	121317
	Sales.SalesOrderHeader	5592 KB	2384 KB	31465
	Sales.SalesOrderHeaderSalesReason	688 KB	16 KB	27647
	Sales.SalesPerson	8 KB	24 KB	17
	Sales.SalesPersonQuotaHistory	16 KB	32 KB	163
	Sales.SalesReason	8 KB	8 KB	10
	Sales.SalesTaxRate	8 KB	40 KB	29
	Sales.SalesTerritory	8 KB	40 KB	10
	Sales.SalesTerritoryHistory	8 KB	24 KB	17
	Sales.ShoppingCartItem	8 KB	24 KB	3
	Sales.SpecialOffer	8 KB	24 KB	16
	Sales.SpecialOfferProduct	24 KB	64 KB	538
	Sales.Store	808 KB	608 KB	701
	Sales.StoreContact	40 KB	120 KB	753

70 table(s)



TABLES



dbo.AWBuildVersion

Current version number of the AdventureWorks sample database.

I	F	P	Column name	Data type	Nulls	Default	Description
			SystemInformationID	tinyint	NO		Primary key for AWBuildVersion records.
			Database Version	nvarchar(25)	NO		Version number of the database in 9.yy.mm.dd.00 format.
			VersionDate	datetime	NO		Date and time the record was last updated.
			ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.



Indexes


Index name	Column name	Sort direction	Is unique	Index type
PK_AWBuildVersion_SystemInformationID	SystemInformationID	ASC	Yes	CLUSTERED

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [dbo].[AWBuildVersion](
[SystemInformationID] [tinyint] IDENTITY(1,1) NOT NULL,
[Database Version] [nvarchar](25) COLLATE Latin1_General_CS_AS NOT NULL,
[VersionDate] [datetime] NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

dbo.DatabaseLog

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

I F P	Column name	Data type	Nulls	Default	Description
	DatabaseLogID	int	NO		Primary key for DatabaseLog records.
	PostTime	datetime	NO		The date and time the DDL change occurred.
	DatabaseUser	nvarchar(128)	NO		The user who implemented the DDL change.
	Event	nvarchar(128)	NO		The type of DDL statement that was executed.
	Schema	nvarchar(128)	YES		The schema to which the changed object belongs.
	Object	nvarchar(128)	YES		The object that was changed by the DDL statment.
	TSQL	nvarchar	NO		The exact Transact-SQL statement that was executed.
	XmlEvent	xml	NO		The raw XML data generated by database trigger.

Indexes



Index name	Column name	Sort direction	Is unique	Index type
PK_DatabaseLog_DatabaseLogID	DatabaseLogID	ASC	Yes	NONCLUSTERED

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [dbo].[DatabaseLog](
[DatabaseLogID] [int] IDENTITY(1,1) NOT NULL,
[PostTime] [datetime] NOT NULL,
[DatabaseUser] [sysname] COLLATE Latin1_General_CS_AS NOT NULL,
[Event] [sysname] COLLATE Latin1_General_CS_AS NOT NULL,
[Schema] [sysname] COLLATE Latin1_General_CS_AS NULL,
[Object] [sysname] COLLATE Latin1_General_CS_AS NULL,
[TSQL] [nvarchar](max) COLLATE Latin1_General_CS_AS NOT NULL,
[XmlEvent] [xml] NOT NULL
) ON [PRIMARY]
```

dbo.ErrorLog

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

I F P	Column name	Data type	Nulls	Default	Description
	ErrorLogID	int	NO		Primary key for ErrorLog records.
	ErrorTime	datetime	NO	(getdate())	The date and time at which the error occurred.
	UserName	nvarchar(128)	NO		The user who executed the batch in which the error occurred.
	ErrorNumber	int	NO		The error number of the error that occurred.
	ErrorSeverity	int	YES		The severity of the error that occurred.
	ErrorState	int	YES		The state number of the error that occurred.
	ErrorProcedure	nvarchar(126)	YES		The name of the stored procedure or trigger where the error occurred.
	ErrorLine	int	YES		The line number at which the error occurred.
	ErrorMessage	nvarchar(4000)	NO		The message text of the error that occurred.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_ErrorLog_ErrorLogID	ErrorLogID	ASC	Yes	CLUSTERED

Objects that depend on dbo.ErrorLog



[dbo.uspLogError](#) (Stored procedure)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [dbo].[ErrorLog](
[ErrorLogID] [int] IDENTITY(1,1) NOT NULL,
[ErrorTime] [datetime] NOT NULL,
[UserName] [sysname] COLLATE Latin1_General_CS_AS NOT NULL,
[ErrorNumber] [int] NOT NULL,
[ErrorSeverity] [int] NULL,
[ErrorState] [int] NULL,
[ErrorProcedure] [nvarchar](126) COLLATE Latin1_General_CS_AS NULL,
[ErrorLine] [int] NULL,
[ErrorMessage] [nvarchar](4000) COLLATE Latin1_General_CS_AS NOT NULL
) ON [PRIMARY]
```

HumanResources.Department

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

I F P	Column name	Data type	Nulls	Default	Description
	DepartmentID	smallint	NO		Primary key for Department records.
	Name	Name	NO		Name of the department.
	GroupName	Name	NO		Name of the group to which the department belongs.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_Department_Name	Name	ASC	Yes	NONCLUSTERED
PK_Department_DepartmentID	DepartmentID	ASC	Yes	CLUSTERED

Referencing tables

Table name	Foreign key	Primary key or unique constraint
HumanResources.EmployeeDepartmentHistory	FK_EmployeeDepartmentHistory_Departme nt_DepartmentID	PK_Department_DepartmentID

Objects that depend on HumanResources.Department

[HumanResources.vEmployeeDepartment \(VIEW\)](#)







[HumanResources.vEmployeeDepartmentHistory \(VIEW\)](#)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [HumanResources].[Department](
[DepartmentID] [smallint] IDENTITY(1,1) NOT NULL,
[Name] [dbo].[Name] NOT NULL,
[GroupName] [dbo].[Name] NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

HumanResources.Employee

Employee information such as salary, department, and title.

I F P	Column name	Data type	Nulls	Default	Description
	EmployeeID	int	NO		Primary key for Employee records.
	NationalIDNumber	nvarchar(15)	NO		Unique national identification number such as a social security number.
	ContactID	int	NO		Identifies the employee in the Contact table. Foreign key to Contact.ContactID.
	LoginID	nvarchar(256)	NO		Network login.
	ManagerID	int	YES		Manager to whom the employee is assigned. Foreign Key to Employee.M
	Title	nvarchar(50)	NO		Work title such as Buyer or Sales Representative.
	BirthDate	datetime	NO		Date of birth.
	MaritalStatus	nchar(1)	NO		M = Married, S = Single
	Gender	nchar(1)	NO		M = Male, F = Female
	HireDate	datetime	NO		Employee hired on this date.
	SalariedFlag	Flag	NO	((1))	Job classification. 0 = Hourly, not exempt from collective bargaining. 1 = Salaried, exempt from collective bargaining.
	VacationHours	smallint	NO	((0))	Number of available vacation hours.
	SickLeaveHours	smallint	NO	((0))	Number of available sick leave hours.
	CurrentFlag	Flag	NO	((1))	0 = Inactive, 1 = Active
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_Employee_LoginID	LoginID	ASC	Yes	NONCLUSTERED
AK_Employee_NationalIDNumber	NationalIDNumber	ASC	Yes	NONCLUSTERED
AK_Employee_rowguid	rowguid	ASC	Yes	NONCLUSTERED
IX_Employee_ManagerID	ManagerID	ASC		NONCLUSTERED
PK_Employee_EmployeeID	EmployeeID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_Employee_Employee_ManagerID	ManagerID	PK_Employee_EmployeeID (HumanResources.Employee)	Foreign key constraint referencing Employee.ManagerID.
FK_Employee_Contact_ContactID	ContactID	PK_Contact_ContactID (Person.Contact)	Foreign key constraint referencing Contact.ContactID.

Referencing tables

Table name	Foreign key	Primary key or unique constraint
HumanResources.JobCandidate	FK_JobCandidate_Employee_EmployeeID	PK_Employee_EmployeeID
Purchasing.PurchaseOrderHeader	FK_PurchaseOrderHeader_Employee_EmployeeID	PK_Employee_EmployeeID
Sales.SalesPerson	FK_SalesPerson_Employee_SalesPersonID	PK_Employee_EmployeeID
HumanResources.Employee	FK_Employee_Employee_ManagerID	PK_Employee_EmployeeID
HumanResources.EmployeeDepartmentHistory	FK_EmployeeDepartmentHistory_Employee_EmployeeID	PK_Employee_EmployeeID
HumanResources.EmployeeAddress	FK_EmployeeAddress_Employee_EmployeeID	PK_Employee_EmployeeID
HumanResources.EmployeePayHistory	FK_EmployeePayHistory_Employee_EmployeeID	PK_Employee_EmployeeID

Check constraints

Check name	Column name	Expression
CK_Employee_BirthDate	BirthDate	(([BirthDate]>= 1930-01-01 AND [BirthDate]<=dateadd(year,-18),getdate()))

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

Triggers

Trigger name: [HumanResources.dEmployee](#) Created on: 26 Apr 2006
Trigger type: [INSTEAD OF DELETE](#) Trigger active: Yes

```
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 @@TRANSCOUNT > 0
BEGIN
ROLLBACK TRANSACTION;
END
END;
END;
```

Objects that depend on [HumanResources.Employee](#)

[dbo.uspGetEmployeeManagers](#) (Stored procedure)
[dbo.uspGetManagerEmployees](#) (Stored procedure)
[HumanResources.uspUpdateEmployeeHireInfo](#) (Stored procedure)
[HumanResources.uspUpdateEmployeeLogin](#) (Stored procedure)
[HumanResources.uspUpdateEmployeePersonalInfo](#) (Stored procedure)
[HumanResources.vEmployee](#) (VIEW)
[HumanResources.vEmployeeDepartment](#) (VIEW)
[HumanResources.vEmployeeDepartmentHistory](#) (VIEW)
[Sales.vSalesPerson](#) (VIEW)
[Sales.vSalesPersonSalesByFiscalYears](#) (VIEW)






Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [HumanResources].[Employee](
[EmployeeID] [int] IDENTITY(1,1) NOT NULL,
[NationalIDNumber] [nvarchar](15) COLLATE Latin1_General_CS_AS NOT NULL,
[ContactID] [int] NOT NULL,
[LoginID] [nvarchar](256) COLLATE Latin1_General_CS_AS NOT NULL,
[ManagerID] [int] NULL,
[Title] [nvarchar](50) COLLATE Latin1_General_CS_AS NOT NULL,
[BirthDate] [datetime] NOT NULL,
[MaritalStatus] [nchar](1) COLLATE Latin1_General_CS_AS NOT NULL,
[Gender] [nchar](1) COLLATE Latin1_General_CS_AS NOT NULL,
[HireDate] [datetime] NOT NULL,
[SalariedFlag] [dbo].[Flag] NOT NULL,
[VacationHours] [smallint] NOT NULL,
```

```
[SickLeaveHours] [smallint] NOT NULL,  
[CurrentFlag] [dbo].[Flag] NOT NULL,  
[rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,  
[ModifiedDate] [datetime] NOT NULL  
) ON [PRIMARY]
```

HumanResources.EmployeeAddress

Cross-reference table mapping employees to their address(es).

I F P	Column name	Data type	Nulls	Default	Description
 	EmployeeID	int	NO		Primary key. Foreign key to Employee.EmployeeID.
 	AddressID	int	NO		Primary key. Foreign key to Address.AddressID.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_EmployeeAddress_rowguid	rowguid	ASC	Yes	NONCLUSTERED
PK_EmployeeAddress_EmployeeID_AddressID	EmployeeID	ASC	Yes	CLUSTERED
PK_EmployeeAddress_EmployeeID_AddressID	AddressID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_EmployeeAddress_Address_AddressID	AddressID	PK_Address_AddressID (Person.Address)	Foreign key constraint referencing Address.AddressID
FK_EmployeeAddress_EmployeeID_EmployeeID	EmployeeID	PK_Employee_EmployeeID (HumanResources.Employee)	Foreign key constraint referencing Employee.EmployeeID.

Objects that depend on HumanResources.EmployeeAddress

[HumanResources.vEmployee \(VIEW\)](#)








[Sales.vSalesPerson \(VIEW\)](#)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [HumanResources].[EmployeeAddress](
  [EmployeeID] [int] NOT NULL,
  [AddressID] [int] NOT NULL,
  [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
  [ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

HumanResources.EmployeeDepartmentHistory

Employee department transfers.

I F P	Column name	Data type	Nulls	Default	Description
 	EmployeeID	int	NO		Employee identification number. Foreign key to Employee.EmployeeID.
 	DepartmentID	smallint	NO		Department in which the employee worked including currently. Foreign key to Department.DepartmentID.
 	ShiftID	tinyint	NO		Identifies which 8-hour shift the employee works. Foreign key to Shift.ShiftID.
	StartDate	datetime	NO		Date the employee started work in the department.
	EndDate	datetime	YES		Date the employee left the department. NULL = Current department.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
IX_EmployeeDepartmentHistory_DepartmentID	DepartmentID	ASC		NONCLUSTERED
IX_EmployeeDepartmentHistory_ShiftID	ShiftID	ASC		NONCLUSTERED
PK_EmployeeDepartmentHistory_EmployeeID_StartDate_DepartmentID	EmployeeID	ASC	Yes	CLUSTERED
PK_EmployeeDepartmentHistory_EmployeeID_StartDate_DepartmentID	DepartmentID	ASC	Yes	CLUSTERED
PK_EmployeeDepartmentHistory_EmployeeID_StartDate_DepartmentID	ShiftID	ASC	Yes	CLUSTERED
PK_EmployeeDepartmentHistory_EmployeeID_StartDate_DepartmentID	StartDate	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_EmployeeDepartmentHistory_DepartmentID	DepartmentID	PK_Department_DepartmentID (HumanResources.Department)	Foreign key constraint referencing Department.DepartmentID.
FK_EmployeeDepartmentHistory_EmployeeID	EmployeeID	PK_Employee_EmployeeID (HumanResources.Employee)	Foreign key constraint referencing Employee.EmployeeID.
FK_EmployeeDepartmentHistory_ShiftID	ShiftID	PK_Shift_ShiftID (HumanResources.Shift)	Foreign key constraint referencing Shift.ShiftID

Check constraints

Check name	Column name	Expression
CK_EmployeeDepartmentHistory_EndDate	StartDate	([EndDate]>=[StartDate] OR [EndDate] IS NULL)
CK_EmployeeDepartmentHistory_EndDate	EndDate	([EndDate]>=[StartDate] OR [EndDate] IS NULL)

Objects that depend on HumanResources.EmployeeDepartmentHistory

[HumanResources.vEmployeeDepartment \(VIEW\)](#)





[HumanResources.vEmployeeDepartmentHistory \(VIEW\)](#)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [HumanResources].[EmployeeDepartmentHistory](
[EmployeeID] [int] NOT NULL,
[DepartmentID] [smallint] NOT NULL,
[ShiftID] [tinyint] NOT NULL,
[StartDate] [datetime] NOT NULL,
[EndDate] [datetime] NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

HumanResources.EmployeePayHistory

Employee pay history.

I F P	Column name	Data type	Nulls	Default	Description
 	EmployeeID	int	NO		Employee identification number. Foreign key to Employee.EmployeeID.
 	RateChangeDate	datetime	NO		Date the change in pay is effective
	Rate	money	NO		Salary hourly rate.
	PayFrequency	tinyint	NO		1 = Salary received monthly, 2 = Salary received biweekly
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_EmployeePayHistory_EmployeeID_ RateChangeDate	EmployeeID	ASC	Yes	CLUSTERED
PK_EmployeePayHistory_EmployeeID_ RateChangeDate	RateChangeDate	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_EmployeePayHistory_EmployeeID	EmployeeID	PK_Employee_EmployeeID (HumanResources.Employee)	Foreign key constraint referencing Employee.EmployeeID.

Check constraints

Check name	Column name	Expression
CK_EmployeePayHistory_PayFrequency	PayFrequency	((PayFrequency)=(2) OR [PayFrequency]=(1))
CK_EmployeePayHistory_Rate	Rate	((Rate)>=(6.50) AND [Rate]<=(200.00))

Objects that depend on HumanResources.EmployeePayHistory



[HumanResources.uspUpdateEmployeeHireInfo](#) (Stored procedure)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [HumanResources].[EmployeePayHistory](
[EmployeeID] [int] NOT NULL,
[RateChangeDate] [datetime] NOT NULL,
[Rate] [money] NOT NULL,
[PayFrequency] [tinyint] NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

HumanResources.JobCandidate

Résumés submitted to Human Resources by job applicants.

I F P	Column name	Data type	Nulls	Default	Description
	JobCandidateID	int	NO		Primary key for JobCandidate records.
	EmployeeID	int	YES		Employee identification number if applicant was hired. Foreign key to Employee.EmployeeID.
	Resume	xml	YES		Résumé in XML format.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
IX_JobCandidate_EmployeeID	EmployeeID	ASC		NONCLUSTERED
PK_JobCandidate_JobCandidateID	JobCandidateID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_JobCandidate_Employee_EmployeeID	EmployeeID	PK_Employee_EmployeeID (HumanResources.Employee)	Foreign key constraint referencing Employee.EmployeeID.

Objects that depend on HumanResources.JobCandidate

[HumanResources.vJobCandidate \(VIEW\)](#)

[HumanResources.vJobCandidateEducation \(VIEW\)](#)





[HumanResources.vJobCandidateEmployment \(VIEW\)](#)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [HumanResources].[JobCandidate](
[JobCandidateID] [int] IDENTITY(1,1) NOT NULL,
[EmployeeID] [int] NULL,
[Resume] [xml](CONTENT [HumanResources].[HRResumeSchemaCollection]) NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

HumanResources.Shift

Work shift lookup table.

I F P	Column name	Data type	Nulls	Default	Description
	ShiftID	tinyint	NO		Primary key for Shift records.
	Name	Name	NO		Shift description.
	StartTime	datetime	NO		Shift start time.
	EndTime	datetime	NO		Shift end time.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_Shift_Name	Name	ASC	Yes	NONCLUSTERED
AK_Shift_StartTime_EndTime	StartTime	ASC	Yes	NONCLUSTERED
AK_Shift_StartTime_EndTime	EndTime	ASC	Yes	NONCLUSTERED
PK_Shift_ShiftID	ShiftID	ASC	Yes	CLUSTERED

Referencing tables

Table name	Foreign key	Primary key or unique constraint
HumanResources.EmployeeDepartmentHistory	FK_EmployeeDepartmentHistory_Shift_Shif tID	PK_Shift_ShiftID

Objects that depend on HumanResources.Shift









[HumanResources.vEmployeeDepartmentHistory \(VIEW\)](#)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [HumanResources].[Shift](
[ShiftID] [tinyint] IDENTITY(1,1) NOT NULL,
[Name] [dbo].[Name] NOT NULL,
[StartTime] [datetime] NOT NULL,
[EndTime] [datetime] NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Person.Address

Street address information for customers, employees, and vendors.

I F P	Column name	Data type	Nulls	Default	Description
	AddressID	int	NO		Primary key for Address records.
	AddressLine1	nvarchar(60)	NO		First street address line.
	AddressLine2	nvarchar(60)	YES		Second street address line.
	City	nvarchar(30)	NO		Name of the city.
 	StateProvinceID	int	NO		Unique identification number for the state or province. Foreign key to StateProvince table.
	PostalCode	nvarchar(15)	NO		Postal code for the street address.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_Address_rowguid	rowguid	ASC	Yes	NONCLUSTERED
IX_Address_AddressLine1_AddressLine2_City_StateProvinceID_PostalCode	AddressLine1	ASC	Yes	NONCLUSTERED
IX_Address_AddressLine1_AddressLine2_City_StateProvinceID_PostalCode	AddressLine2	ASC	Yes	NONCLUSTERED
IX_Address_AddressLine1_AddressLine2_City_StateProvinceID_PostalCode	City	ASC	Yes	NONCLUSTERED
IX_Address_AddressLine1_AddressLine2_City_StateProvinceID_PostalCode	StateProvinceID	ASC	Yes	NONCLUSTERED
IX_Address_AddressLine1_AddressLine2_City_StateProvinceID_PostalCode	PostalCode	ASC	Yes	NONCLUSTERED
IX_Address_StateProvinceID	StateProvinceID	ASC		NONCLUSTERED
PK_Address_AddressID	AddressID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_Address_StateProvince_StateProvinceID	StateProvinceID	PK_StateProvince_StateProvinceID (Person.StateProvince)	Foreign key constraint referencing StateProvince.StateProvinceID.

Referencing tables

Table name	Foreign key	Primary key or unique constraint
Sales.SalesOrderHeader	FK_SalesOrderHeader_Address_BillToAddressID	PK_Address_AddressID
Purchasing.VendorAddress	FK_VendorAddress_Address_AddressID	PK_Address_AddressID
Sales.CustomerAddress	FK_CustomerAddress_Address_AddressID	PK_Address_AddressID
HumanResources.EmployeeAddress	FK_EmployeeAddress_Address_AddressID	PK_Address_AddressID
Sales.SalesOrderHeader	FK_SalesOrderHeader_Address_ShipToAddressID	PK_Address_AddressID

Objects that depend on Person.Address

[HumanResources.vEmployee \(VIEW\)](#)

[Purchasing.vVendor \(VIEW\)](#)

[Sales.vIndividualCustomer \(VIEW\)](#)

[Sales.vSalesPerson \(VIEW\)](#)

[Sales.vStoreWithDemographics \(VIEW\)](#)

Table definition




```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Person].[Address](
[AddressID] [int] IDENTITY(1,1) NOT FOR REPLICATION NOT NULL,
[AddressLine1] [nvarchar](60) COLLATE Latin1_General_CS_AS NOT NULL,
```



```
[AddressLine2] [nvarchar](60) COLLATE Latin1_General_CS_AS NULL,  
[City] [nvarchar](30) COLLATE Latin1_General_CS_AS NOT NULL,  
[StateProvinceID] [int] NOT NULL,  
[PostalCode] [nvarchar](15) COLLATE Latin1_General_CS_AS NOT NULL,  
[rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,  
[ModifiedDate] [datetime] NOT NULL  
) ON [PRIMARY]
```

Person.AddressType

Types of addresses stored in the Address table.

I F P	Column name	Data type	Nulls	Default	Description
	AddressTypeID	int	NO		Primary key for AddressType records.
	Name	Name	NO		Address type description. For example, Billing, Home, or Shipping.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_AddressType_Name	Name	ASC	Yes	NONCLUSTERED
AK_AddressType_rowguid	rowguid	ASC	Yes	NONCLUSTERED
PK_AddressType_AddressTypeID	AddressTypeID	ASC	Yes	CLUSTERED

Referencing tables

Table name	Foreign key	Primary key or unique constraint
Sales.CustomerAddress	FK_CustomerAddress_AddressType_AddressTypeID	PK_AddressType_AddressTypeID
Purchasing.VendorAddress	FK_VendorAddress_AddressType_AddressTypeID	PK_AddressType_AddressTypeID

Objects that depend on Person.AddressType

[Sales.vIndividualCustomer](#) (VIEW)






[Sales.vStoreWithDemographics](#) (VIEW)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Person].[AddressType](
  [AddressTypeID] [int] IDENTITY(1,1) NOT NULL,
  [Name] [dbo].[Name] NOT NULL,
  [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
  [ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Person.Contact

Names of each employee, customer contact, and vendor contact.

I F P	Column name	Data type	Nulls	Default	Description
	ContactID	int	NO		Primary key for Contact records.
	NameStyle	NameStyle	NO	((0))	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.
	Title	nvarchar(8)	YES		A courtesy title. For example, Mr. or Ms.
	FirstName	Name	NO		First name of the person.
	MiddleName	Name	YES		Middle name or middle initial of the person.
	LastName	Name	NO		Last name of the person.
	Suffix	nvarchar(10)	YES		Surname suffix. For example, Sr. or Jr.
	EmailAddress	nvarchar(50)	YES		E-mail address for the person.
	EmailPromotion	int	NO	((0))	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.
	Phone	Phone	YES		Phone number associated with the person.
	PasswordHash	varchar(128)	NO		Password for the e-mail account.
	PasswordSalt	varchar(10)	NO		Random value concatenated with the password string before the password is hashed.
	AdditionalContactInfo	xml	YES		Additional contact information about the person stored in xml format.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_Contact_rowguid	rowguid	ASC	Yes	NONCLUSTERED
IX_Contact_EmailAddress	EmailAddress	ASC		NONCLUSTERED
PK_Contact_ContactID	ContactID	ASC	Yes	CLUSTERED
PXML_Contact_AddContact	AdditionalContactInfo	ASC		XML

Referencing tables

Table name	Foreign key	Primary key or unique constraint
Sales.SalesOrderHeader	FK_SalesOrderHeader_Contact_ContactID	PK_Contact_ContactID
Sales.StoreContact	FK_StoreContact_Contact_ContactID	PK_Contact_ContactID
Purchasing.VendorContact	FK_VendorContact_Contact_ContactID	PK_Contact_ContactID
Sales.ContactCreditCard	FK_ContactCreditCard_Contact_ContactID	PK_Contact_ContactID
HumanResources.Employee	FK_Employee_Contact_ContactID	PK_Contact_ContactID
Sales.Individual	FK_Individual_Contact_ContactID	PK_Contact_ContactID

Check constraints

Check name	Column name	Expression
CK_Contact_EmailPromotion	EmailPromotion	([EmailPromotion]>=(0) AND [EmailPromotion]<=(2))

Objects that depend on Person.Contact

[dbo.uspGetEmployeeManagers](#) (Stored procedure)

[dbo.uspGetManagerEmployees](#) (Stored procedure)

[HumanResources.vEmployee](#) (VIEW)

[HumanResources.vEmployeeDepartment](#) (VIEW)

[HumanResources.vEmployeeDepartmentHistory](#) (VIEW)

[Person.vAdditionalContactInfo](#) (VIEW)

[Purchasing.vVendor](#) (VIEW)



Sales.vIndividualCustomer (VIEW)
Sales.vSalesPerson (VIEW)
Sales.vSalesPersonSalesByFiscalYears (VIEW)
Sales.vStoreWithDemographics (VIEW)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Person].[Contact](
[ContactID] [int] IDENTITY(1,1) NOT FOR REPLICATION NOT NULL,
[NameStyle] [dbo].[NameStyle] NOT NULL,
[Title] [nvarchar](8) COLLATE Latin1_General_CS_AS NULL,
[FirstName] [dbo].[Name] NOT NULL,
[MiddleName] [dbo].[Name] NULL,
[LastName] [dbo].[Name] NOT NULL,
[Suffix] [nvarchar](10) COLLATE Latin1_General_CS_AS NULL,
[EmailAddress] [nvarchar](50) COLLATE Latin1_General_CS_AS NULL,
[EmailPromotion] [int] NOT NULL,
[Phone] [dbo].[Phone] NULL,
[PasswordHash] [varchar](128) COLLATE Latin1_General_CS_AS NOT NULL,
[PasswordSalt] [varchar](10) COLLATE Latin1_General_CS_AS NOT NULL,
[AdditionalContactInfo] [xml](CONTENT [Person].[AdditionalContactInfoSchemaCollection]) NULL,
[rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Person.ContactType

Lookup table containing the types of contacts stored in Contact.

I F P	Column name	Data type	Nulls	Default	Description
	ContactTypeID	int	NO		Primary key for ContactType records.
	Name	Name	NO		Contact type description.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_ContactType_Name	Name	ASC	Yes	NONCLUSTERED
PK_ContactType_ContactTypeID	ContactTypeID	ASC	Yes	CLUSTERED

Referencing tables

Table name	Foreign key	Primary key or unique constraint
Purchasing.vVendorContact	FK_VendorContact_ContactType_ContactTypeID	PK_ContactType_ContactTypeID
Sales.StoreContact	FK_StoreContact_ContactType_ContactTypeID	PK_ContactType_ContactTypeID

Objects that depend on Person.ContactType

[Purchasing.vVendor](#) (VIEW)



[Sales.vStoreWithDemographics](#) (VIEW)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Person].[ContactType](
[ContactTypeID] [int] IDENTITY(1,1) NOT NULL,
[Name] [dbo].[Name] NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Person.CountryRegion

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

I F P	Column name	Data type	Nulls	Default	Description
	CountryRegionCode	nvarchar(3)	NO		ISO standard code for countries and regions.
	Name	Name	NO		Country or region name.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_CountryRegion_Name	Name	ASC	Yes	NONCLUSTERED
PK_CountryRegion_CountryRegionCode	CountryRegionCode	ASC	Yes	CLUSTERED

Referencing tables

Table name	Foreign key	Primary key or unique constraint
Person.StateProvince	FK_StateProvince_CountryRegion_CountryRegionCode	PK_CountryRegion_CountryRegionCode
Sales.CountryRegionCurrency	FK_CountryRegionCurrency_CountryRegion_CountryRegionCode	PK_CountryRegion_CountryRegionCode

Objects that depend on Person.CountryRegion

[HumanResources.vEmployee \(VIEW\)](#)

[Person.vStateProvinceCountryRegion \(VIEW\)](#)

[Purchasing.vVendor \(VIEW\)](#)

[Sales.vIndividualCustomer \(VIEW\)](#)

[Sales.vSalesPerson \(VIEW\)](#)







[Sales.vStoreWithDemographics \(VIEW\)](#)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Person].[CountryRegion](
[CountryRegionCode] [nvarchar](3) COLLATE Latin1_General_CS_AS NOT NULL,
[Name] [dbo].[Name] NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Person.StateProvince

State and province lookup table.

I F P	Column name	Data type	Nulls	Default	Description
	StateProvinceID	int	NO		Primary key for StateProvince records.
	StateProvinceCode	nchar(3)	NO		ISO standard state or province code.
	CountryRegionCode	nvarchar(3)	NO		ISO standard country or region code. Foreign key to CountryRegion.CountryRegionCode.
	IsOnlyStateProvinceFlag	Flag	NO	((1))	0 = StateProvinceCode exists. 1 = StateProvinceCode unavailable, using CountryRegionCode.
	Name	Name	NO		State or province description.
	TerritoryID	int	NO		ID of the territory in which the state or province is located. Foreign key to SalesTerritory.SalesTerritoryID.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_StateProvince_Name	Name	ASC	Yes	NONCLUSTERED
AK_StateProvince_rowguid	rowguid	ASC	Yes	NONCLUSTERED
AK_StateProvince_StateProvinceCode_CountryRegionCode	StateProvinceCode	ASC	Yes	NONCLUSTERED
AK_StateProvince_StateProvinceCode_CountryRegionCode	CountryRegionCode	ASC	Yes	NONCLUSTERED
PK_StateProvince_StateProvinceID	StateProvinceID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_StateProvince_CountryRegion_CountryRegionCode	CountryRegionCode	PK_CountryRegion_CountryRegionCode (Person.CountryRegion)	Foreign key constraint referencing CountryRegion.CountryRegionCode.
FK_StateProvince_SalesTerritory_TerritoryID	TerritoryID	PK_SalesTerritory_TerritoryID (Sales.SalesTerritory)	Foreign key constraint referencing SalesTerritory.TerritoryID.

Referencing tables

Table name	Foreign key	Primary key or unique constraint
Sales.SalesTaxRate	FK_SalesTaxRate_StateProvince_StateProvinceID	PK_StateProvince_StateProvinceID
Person.Address	FK_Address_StateProvince_StateProvinceID	PK_StateProvince_StateProvinceID

Objects that depend on Person.StateProvince

[HumanResources.vEmployee \(VIEW\)](#)

[Person.vStateProvinceCountryRegion \(VIEW\)](#)

[Purchasing.vVendor \(VIEW\)](#)

[Sales.vIndividualCustomer \(VIEW\)](#)

[Sales.vSalesPerson \(VIEW\)](#)

[Sales.vStoreWithDemographics \(VIEW\)](#)






Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Person].[StateProvince](
[StateProvinceID] [int] IDENTITY(1,1) NOT NULL,
[StateProvinceCode] [nchar](3) COLLATE Latin1_General_CS_AS NOT NULL,
[CountryRegionCode] [nvarchar](3) COLLATE Latin1_General_CS_AS NOT NULL,
[IsOnlyStateProvinceFlag] [dbo].[Flag] NOT NULL,
[Name] [dbo].[Name] NOT NULL,
[TerritoryID] [int] NOT NULL,
```

```
[rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,  
[ModifiedDate] [datetime] NOT NULL  
) ON [PRIMARY]
```


Production.BillOfMaterials

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

I F P	Column name	Data type	Nulls	Default	Description
	BillOfMaterialsID	int	NO		Primary key for BillOfMaterials records.
	ProductAssemblyID	int	YES		Parent product identification number. Foreign key to Product.ProductID.
	ComponentID	int	NO		Component identification number. Foreign key to Product.ProductID.
	StartDate	datetime	NO	(getdate())	Date the component started being used in the assembly item.
	EndDate	datetime	YES		Date the component stopped being used in the assembly item.
	UnitMeasureCode	nchar(3)	NO		Standard code identifying the unit of measure for the quantity.
	BOMLevel	smallint	NO		Indicates the depth the component is from its parent (AssemblyID).
	PerAssemblyQty	decimal(8,2)	NO	((1.00))	Quantity of the component needed to create the assembly.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_BillOfMaterials_ProductAssemblyID_ComponentID_StartDate	ProductAssemblyID	ASC	Yes	CLUSTERED
AK_BillOfMaterials_ProductAssemblyID_ComponentID_StartDate	ComponentID	ASC	Yes	CLUSTERED
AK_BillOfMaterials_ProductAssemblyID_ComponentID_StartDate	StartDate	ASC	Yes	CLUSTERED
IX_BillOfMaterials_UnitMeasureCode	UnitMeasureCode	ASC		NONCLUSTERED
PK_BillOfMaterials_BillOfMaterialsID	BillOfMaterialsID	ASC	Yes	NONCLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_BillOfMaterials_Product_ComponentID	ComponentID	PK_Product_ProductID (Production.Product)	Foreign key constraint referencing Product.ComponentID.
FK_BillOfMaterials_UnitMeasure_UnitMeasureCode	UnitMeasureCode	PK_UnitMeasure_UnitMeasureCode (Production.UnitMeasure)	Foreign key constraint referencing UnitMeasure.UnitMeasureCode.
FK_BillOfMaterials_Product_ProductAssemblyID	ProductAssemblyID	PK_Product_ProductID (Production.Product)	Foreign key constraint referencing Product.ProductAssemblyID.

Check constraints

Check name	Column name	Expression
CK_BillOfMaterials_EndDate	StartDate	([EndDate]>[StartDate] OR [EndDate] IS NULL)
CK_BillOfMaterials_EndDate	EndDate	([EndDate]>[StartDate] OR [EndDate] IS NULL)
CK_BillOfMaterials_ProductAssemblyID	ProductAssemblyID	([ProductAssemblyID]<>[ComponentID])
CK_BillOfMaterials_ProductAssemblyID	ComponentID	([ProductAssemblyID]<>[ComponentID])
CK_BillOfMaterials_BOMLevel	ProductAssemblyID	([ProductAssemblyID] IS NULL AND [BOMLevel]=0) AND [PerAssemblyQty]=(1.00) OR [ProductAssemblyID] IS NOT ([ProductAssemblyID] IS NULL AND [BOMLevel]=0) AND [PerAssemblyQty]=(1.00) OR [ProductAssemblyID] IS NOT ([ProductAssemblyID] IS NULL AND [BOMLevel]=0) AND [PerAssemblyQty]=(1.00) OR [ProductAssemblyID] IS NOT ([PerAssemblyQty]>=(1.00))
CK_BillOfMaterials_BOMLevel	BOMLevel	
CK_BillOfMaterials_BOMLevel	PerAssemblyQty	
CK_BillOfMaterials_PerAssemblyQty	PerAssemblyQty	

Objects that depend on Production.BillOfMaterials

[dbo.uspGetBillOfMaterials](#) (Stored procedure)

[dbo.uspGetWhereUsedProductID](#) (Stored procedure)



Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[BillOfMaterials](
```

```
[BillOfMaterialsID] [int] IDENTITY(1,1) NOT NULL,  
[ProductAssemblyID] [int] NULL,  
[ComponentID] [int] NOT NULL,  
[StartDate] [datetime] NOT NULL,  
[EndDate] [datetime] NULL,  
[UnitMeasureCode] [nchar](3) COLLATE Latin1_General_CS_AS NOT NULL,  
[BOMLevel] [smallint] NOT NULL,  
[PerAssemblyQty] [decimal](8, 2) NOT NULL,  
[ModifiedDate] [datetime] NOT NULL  
) ON [PRIMARY]
```

Production.Culture

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

I F P	Column name	Data type	Nulls	Default	Description
	CultureID	nchar(6)	NO		Primary key for Culture records.
	Name	Name	NO		Culture description.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_Culture_Name	Name	ASC	Yes	NONCLUSTERED
PK_Culture_CultureID	CultureID	ASC	Yes	CLUSTERED

Referencing tables




Table name	Foreign key	Primary key or unique constraint
Production.ProductModelProductDescriptionCulture	FK_ProductModelProductDescriptionCulture_Culture_CultureID	PK_Culture_CultureID

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[Culture](
[CultureID] [nchar](6) COLLATE Latin1_General_CS_AS NOT NULL,
[Name] [dbo].[Name] NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Production.Document

Product maintenance documents.

I F P	Column name	Data type	Nulls	Default	Description
	DocumentID	int	NO		Primary key for Document records.
	Title	nvarchar(50)	NO		Title of the document.
	FileName	nvarchar(400)	NO		Directory path and file name of the document
	FileExtension	nvarchar(8)	NO		File extension indicating the document type. For example, .doc or .txt.
	Revision	nchar(5)	NO		Revision number of the document.
	ChangeNumber	int	NO	((0))	Engineering change approval number.
	Status	tinyint	NO		1 = Pending approval, 2 = Approved, 3 = Obsolete
	DocumentSummary	nvarchar	YES		Document abstract.
	Document	varbinary	YES		Complete document.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_Document_FileName_Revision	FileName	ASC	Yes	NONCLUSTERED
AK_Document_FileName_Revision	Revision	ASC	Yes	NONCLUSTERED
PK_Document_DocumentID	DocumentID	ASC	Yes	CLUSTERED

Referencing tables

Table name	Foreign key	Primary key or unique constraint
Production.ProductDocument	FK_ProductDocument_Document_DocumentID	PK_Document_DocumentID

Check constraints



Check name	Column name	Expression
CK_Document_Status	Status	([Status]>=(1) AND [Status]<=(3))

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[Document](
[DocumentID] [int] IDENTITY(1,1) NOT NULL,
[Title] [nvarchar](50) COLLATE Latin1_General_CS_AS NOT NULL,
[FileName] [nvarchar](400) COLLATE Latin1_General_CS_AS NOT NULL,
[FileExtension] [nvarchar](8) COLLATE Latin1_General_CS_AS NOT NULL,
[Revision] [nchar](5) COLLATE Latin1_General_CS_AS NOT NULL,
[ChangeNumber] [int] NOT NULL,
[Status] [tinyint] NOT NULL,
[DocumentSummary] [nvarchar](max) COLLATE Latin1_General_CS_AS NULL,
[Document] [varbinary](max) NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Production.Illustration

Bicycle assembly diagrams.

I F P	Column name	Data type	Nulls	Default	Description
	IllustrationID	int	NO		Primary key for Illustration records.
	Diagram	xml	YES		Illustrations used in manufacturing instructions. Stored as XML.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_Illustration_IllustrationID	IllustrationID	ASC	Yes	CLUSTERED

Referencing tables



Table name	Foreign key	Primary key or unique constraint
Production.ProductModelIllustration	FK_ProductModelIllustration_Illustration_IllustrationID	PK_Illustration_IllustrationID

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[Illustration](
  [IllustrationID] [int] IDENTITY(1,1) NOT NULL,
  [Diagram] [xml] NULL,
  [ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Production.Location

Product inventory and manufacturing locations.

I F P	Column name	Data type	Nulls	Default	Description
	LocationID	smallint	NO		Primary key for Location records.
	Name	Name	NO		Location description.
	CostRate	smallmoney	NO	((0.00))	Standard hourly cost of the manufacturing location.
	Availability	decimal(8,2)	NO	((0.00))	Work capacity (in hours) of the manufacturing location.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_Location_Name	Name	ASC	Yes	NONCLUSTERED
PK_Location_LocationID	LocationID	ASC	Yes	CLUSTERED

Referencing tables

Table name	Foreign key	Primary key or unique constraint
Production.ProductInventory	FK_ProductInventory_Location_LocationID	PK_Location_LocationID
Production.WorkOrderRouting	FK_WorkOrderRouting_Location_LocationID	PK_Location_LocationID

Check constraints









Check name	Column name	Expression
CK_Location_CostRate	CostRate	([CostRate]>=(0.00))
CK_Location_Availability	Availability	([Availability]>=(0.00))

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[Location](
[LocationID] [smallint] IDENTITY(1,1) NOT NULL,
[Name] [dbo].[Name] NOT NULL,
[CostRate] [smallmoney] NOT NULL,
[Availability] [decimal](8, 2) NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Production.Product

Products sold or used in the manufacturing of sold products.

I F P	Column name	Data type	Nulls	Default	Description
	ProductID	int	NO		Primary key for Product records.
	Name	Name	NO		Name of the product.
	ProductNumber	nvarchar(25)	NO		Unique product identification number.
	MakeFlag	Flag	NO	((1))	0 = Product is purchased, 1 = Product is manufactured in-house.
	FinishedGoodsFlag	Flag	NO	((1))	0 = Product is not a salable item. 1 = Product is salable.
	Color	nvarchar(15)	YES		Product color.
	SafetyStockLevel	smallint	NO		Minimum inventory quantity.
	ReorderPoint	smallint	NO		Inventory level that triggers a purchase order or work order.
	StandardCost	money	NO		Standard cost of the product.
	ListPrice	money	NO		Selling price.
	Size	nvarchar(5)	YES		Product size.
	SizeUnitMeasureCode	nchar(3)	YES		Unit of measure for Size column.
	WeightUnitMeasureCode	nchar(3)	YES		Unit of measure for Weight column.
	Weight	decimal(8,2)	YES		Product weight.
	DaysToManufacture	int	NO		Number of days required to manufacture the product.
	ProductLine	nchar(2)	YES		R = Road, M = Mountain, T = Touring, S = Standard
	Class	nchar(2)	YES		H = High, M = Medium, L = Low
	Style	nchar(2)	YES		W = Womens, M = Mens, U = Universal
	ProductSubcategoryID	int	YES		Product is a member of this product subcategory. Foreign key to ProductSubCategory.ProductSubCategoryID.
	ProductModelID	int	YES		Product is a member of this product model. Foreign key to ProductModel.ProductModelID.
	SellStartDate	datetime	NO		Date the product was available for sale.
	SellEndDate	datetime	YES		Date the product was no longer available for sale.
	DiscontinuedDate	datetime	YES		Date the product was discontinued.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_Product_Name	Name	ASC	Yes	NONCLUSTERED
AK_Product_ProductNumber	ProductNumber	ASC	Yes	NONCLUSTERED
AK_Product_rowguid	rowguid	ASC	Yes	NONCLUSTERED
PK_Product_ProductID	ProductID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_Product_UnitMeasure_SizeUnitMeasureCode	SizeUnitMeasureCode	PK_UnitMeasure_UnitMeasureCode (Production.UnitMeasure)	Foreign key constraint referencing UnitMeasure.UnitMeasureCode
FK_Product_UnitMeasure_WeightUnitMeasureCode	WeightUnitMeasureCode	PK_UnitMeasure_UnitMeasureCode (Production.UnitMeasure)	Foreign key constraint referencing UnitMeasure.UnitMeasureCode.
FK_Product_ProductSubcategory_ProductSubcategoryID	ProductSubcategoryID	PK_ProductSubcategory_ProductSubcategoryID (Production.ProductSubcategory)	Foreign key constraint referencing ProductSubcategory.ProductSubcategoryID.
FK_Product_ProductModel_ProductModelID	ProductModelID	PK_ProductModel_ProductModelID (Production.ProductModel)	Foreign key constraint referencing ProductModel.ProductModelID.

Referencing tables

Table name	Foreign key	Primary key or unique constraint
Production.ProductCostHistory	FK_ProductCostHistory_Product_ProductID	PK_Product_ProductID

Production.ProductDocument	FK_ProductDocument_Product_ProductID	PK_Product_ProductID
Production.ProductInventory	FK_ProductInventory_Product_ProductID	PK_Product_ProductID
Production.ProductListPriceHistory	FK_ProductListPriceHistory_Product_ProductID	PK_Product_ProductID
Production.ProductProductPhoto	FK_ProductProductPhoto_Product_ProductID	PK_Product_ProductID
Production.ProductReview	FK_ProductReview_Product_ProductID	PK_Product_ProductID
Purchasing.ProductVendor	FK_ProductVendor_Product_ProductID	PK_Product_ProductID
Sales.SpecialOfferProduct	FK_SpecialOfferProduct_Product_ProductID	PK_Product_ProductID
Production.TransactionHistory	FK_TransactionHistory_Product_ProductID	PK_Product_ProductID
Production.WorkOrder	FK_WorkOrder_Product_ProductID	PK_Product_ProductID
Production.BillOfMaterials	FK_BillOfMaterials_Product_ComponentID	PK_Product_ProductID
Purchasing.PurchaseOrderDetail	FK_PurchaseOrderDetail_Product_ProductID	PK_Product_ProductID
Sales.ShoppingCartItem	FK_ShoppingCartItem_Product_ProductID	PK_Product_ProductID
Production.BillOfMaterials	FK_BillOfMaterials_Product_ProductAssemblyID	PK_Product_ProductID

Check constraints

Check name	Column name	Expression
CK_Product_SafetyStockLevel	SafetyStockLevel	([SafetyStockLevel]>(0))
CK_Product_ReorderPoint	ReorderPoint	([ReorderPoint]>(0))
CK_Product_StandardCost	StandardCost	([StandardCost]>=(0.00))
CK_Product_ListPrice	ListPrice	([ListPrice]>=(0.00))
CK_Product_Weight	Weight	([Weight]>(0.00))
CK_Product_DaysToManufacture	DaysToManufacture	([DaysToManufacture]>=(0))
CK_Product_ProductLine	ProductLine	(upper([ProductLine])='K' OR upper([ProductLine])='M' OR upper([ProductLine])='T' OR upper([ProductLine])='S' OR (upper([Class])='H' OR upper([Class])='M' OR upper([Class])='L' OR [Class] IS NULL))
CK_Product_Class	Class	(upper([Style])='U' OR upper([Style])='M' OR upper([Style])='W' OR [Style] IS NULL)
CK_Product_SellEndDate	SellStartDate	([SellEndDate]>=[SellStartDate] OR [SellEndDate] IS NULL)
CK_Product_SellEndDate	SellEndDate	([SellEndDate]>=[SellStartDate] OR [SellEndDate] IS NULL)

Objects that depend on Production.Product

dbo.ufnGetProductDealerPrice (UDF)

dbo.ufnGetProductListPrice (UDF)

dbo.ufnGetProductStandardCost (UDF)

dbo.uspGetBillOfMaterials (Stored procedure)

dbo.uspGetWhereUsedProductID (Stored procedure)

Production.vProductAndDescription (VIEW)

Table definition

```

SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[Product](
[ProductID] [int] IDENTITY(1,1) NOT NULL,
[Name] [dbo].[Name] NOT NULL,
[ProductNumber] [nvarchar](25) COLLATE Latin1_General_CS_AS NOT NULL,
[MakeFlag] [dbo].[Flag] NOT NULL,
[FinishedGoodsFlag] [dbo].[Flag] NOT NULL,
[Color] [nvarchar](15) COLLATE Latin1_General_CS_AS NULL,
[SafetyStockLevel] [smallint] NOT NULL,
[ReorderPoint] [smallint] NOT NULL,
[StandardCost] [money] NOT NULL,
[ListPrice] [money] NOT NULL,
[Size] [nvarchar](5) COLLATE Latin1_General_CS_AS NULL,
[SizeUnitMeasureCode] [nchar](3) COLLATE Latin1_General_CS_AS NULL,
[WeightUnitMeasureCode] [nchar](3) COLLATE Latin1_General_CS_AS NULL,
[Weight] [decimal](8, 2) NULL,
[DaysToManufacture] [int] NOT NULL,
[ProductLine] [nchar](2) COLLATE Latin1_General_CS_AS NULL,




```



```
[Class] [nchar](2) COLLATE Latin1_General_CS_AS NULL,  
[Style] [nchar](2) COLLATE Latin1_General_CS_AS NULL,  
[ProductSubcategoryID] [int] NULL,  
[ProductModelID] [int] NULL,  
[SellStartDate] [datetime] NOT NULL,  
[SellEndDate] [datetime] NULL,  
[DiscontinuedDate] [datetime] NULL,  
[rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,  
[ModifiedDate] [datetime] NOT NULL  
) ON [PRIMARY]
```

Production.ProductCategory

High-level product categorization.

I F P	Column name	Data type	Nulls	Default	Description
	ProductCategoryID	int	NO		Primary key for ProductCategory records.
	Name	Name	NO		Category description.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_ProductCategory_Name	Name	ASC	Yes	NONCLUSTERED
AK_ProductCategory_rowguid	rowguid	ASC	Yes	NONCLUSTERED
PK_ProductCategory_ProductCategoryID	ProductCategoryID	ASC	Yes	CLUSTERED

Referencing tables




Table name	Foreign key	Primary key or unique constraint
Production.ProductSubcategory	FK_ProductSubcategory_ProductCategory_ProductCategoryID	PK_ProductCategory_ProductCategoryID

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[ProductCategory](
[ProductCategoryID] [int] IDENTITY(1,1) NOT NULL,
[Name] [dbo].[Name] NOT NULL,
[rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Production.ProductCostHistory

Changes in the cost of a product over time.

I F P	Column name	Data type	Nulls	Default	Description
 	ProductID	int	NO		Product identification number. Foreign key to Product.ProductID
	StartDate	datetime	NO		Product cost start date.
	EndDate	datetime	YES		Product cost end date.
	StandardCost	money	NO		Standard cost of the product.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_ProductCostHistory_ProductID_Star tDate	ProductID	ASC	Yes	CLUSTERED
PK_ProductCostHistory_ProductID_Star tDate	StartDate	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_ProductCostHistory_Product _ProductID	ProductID	PK_Product_ProductID (Production.Product)	Foreign key constraint referencing Product.ProductID.

Check constraints

Check name	Column name	Expression
CK_ProductCostHistory_EndDate	StartDate	((EndDate)>=[StartDate] OR [EndDate] IS NULL)
CK_ProductCostHistory_EndDate	EndDate	((EndDate)>=[StartDate] OR [EndDate] IS NULL)
CK_ProductCostHistory_StandardC ost	StandardCost	((StandardCost)>=(0.00))

Objects that depend on Production.ProductCostHistory



[dbo.ufnGetProductStandardCost \(UDF\)](#)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[ProductCostHistory](
  [ProductID] [int] NOT NULL,
  [StartDate] [datetime] NOT NULL,
  [EndDate] [datetime] NULL,
  [StandardCost] [money] NOT NULL,
  [ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Production.ProductDescription

Product descriptions in several languages.

I F P	Column name	Data type	Nulls	Default	Description
	ProductDescriptionID	int	NO		Primary key for ProductDescription records.
	Description	nvarchar(400)	NO		Description of the product.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_ProductDescription_rowguid	rowguid	ASC	Yes	NONCLUSTERED
PK_ProductDescription_ProductDescriptionID	ProductDescriptionID	ASC	Yes	CLUSTERED

Referencing tables

Table name	Foreign key	Primary key or unique constraint
Production.ProductModelProductDescriptionCulture	FK_ProductModelProductDescriptionCulture_ProductDescription_ProductDescriptionID	PK_ProductDescription_ProductDescriptionID

Objects that depend on Production.ProductDescription





[Production.vProductAndDescription \(VIEW\)](#)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[ProductDescription](
[ProductDescriptionID] [int] IDENTITY(1,1) NOT NULL,
[Description] [nvarchar](400) COLLATE Latin1_General_CS_AS NOT NULL,
[rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Production.ProductDocument

Cross-reference table mapping products to related product documents.

I F P	Column name	Data type	Nulls	Default	Description
 	ProductID	int	NO		Product identification number. Foreign key to Product.ProductID.
 	DocumentID	int	NO		Document identification number. Foreign key to Document.DocumentID.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_ProductDocument_ProductID_DocumentID	ProductID	ASC	Yes	CLUSTERED
PK_ProductDocument_ProductID_DocumentID	DocumentID	ASC	Yes	CLUSTERED

Foreign keys





Constraint name	Column name	Reference	Description
FK_ProductDocument_ProductID	ProductID	PK_Product_ProductID (Production.Product)	Foreign key constraint referencing Product.ProductID.
FK_ProductDocument_DocumentID	DocumentID	PK_Document_DocumentID (Production.Document)	Foreign key constraint referencing Document.DocumentID.

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[ProductDocument](
[ProductID] [int] NOT NULL,
[DocumentID] [int] NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Production.ProductInventory

Product inventory information.

I F P	Column name	Data type	Nulls	Default	Description
 	ProductID	int	NO		Product identification number. Foreign key to Product.ProductID.
 	LocationID	smallint	NO		Inventory location identification number. Foreign key to Location.LocationID.
	Shelf	nvarchar(10)	NO		Storage compartment within an inventory location.
	Bin	tinyint	NO		Storage container on a shelf in an inventory location.
	Quantity	smallint	NO	((0))	Quantity of products in the inventory location.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_ProductInventory_ProductID_LocationID	ProductID	ASC	Yes	CLUSTERED
PK_ProductInventory_ProductID_LocationID	LocationID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_ProductInventory_Location_LocationID	LocationID	PK_Location_LocationID (Production.Location)	Foreign key constraint referencing Location.LocationID.
FK_ProductInventory_Product_ProductID	ProductID	PK_Product_ProductID (Production.Product)	Foreign key constraint referencing Product.ProductID.

Check constraints

Check name	Column name	Expression
CK_ProductInventory_Shelf	Shelf	((Shelf) like '[A-Za-z]' OR [Shelf]='N/A')
CK_ProductInventory_Bin	Bin	((Bin)>=(0) AND [Bin]<=(100))

Objects that depend on Production.ProductInventory




[dbo.ufnGetStock \(UDF\)](#)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[ProductInventory](
[ProductID] [int] NOT NULL,
[LocationID] [smallint] NOT NULL,
[Shelf] [nvarchar](10) COLLATE Latin1_General_CS_AS NOT NULL,
[Bin] [tinyint] NOT NULL,
[Quantity] [smallint] NOT NULL,
[rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Production.ProductListPriceHistory

Changes in the list price of a product over time.

I F P	Column name	Data type	Nulls	Default	Description
 	ProductID	int	NO		Product identification number. Foreign key to Product.ProductID
	StartDate	datetime	NO		List price start date.
	EndDate	datetime	YES		List price end date
	ListPrice	money	NO		Product list price.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_ProductListPriceHistory_ProductID_	ProductID	ASC	Yes	CLUSTERED
PK_ProductListPriceHistory_ProductID_	StartDate	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_ProductListPriceHistory_ProductID_	ProductID	PK_Product_ProductID (Production.Product)	Foreign key constraint referencing Product.ProductID.

Check constraints

Check name	Column name	Expression
CK_ProductListPriceHistory_EndDate	StartDate	([EndDate]>=[StartDate] OR [EndDate] IS NULL)
CK_ProductListPriceHistory_EndDate	EndDate	([EndDate]>=[StartDate] OR [EndDate] IS NULL)
CK_ProductListPriceHistory_ListPrice	ListPrice	([ListPrice]>(0.00))

Objects that depend on Production.ProductListPriceHistory

[dbo.ufnGetProductDealerPrice \(UDF\)](#)






[dbo.ufnGetProductListPrice \(UDF\)](#)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[ProductListPriceHistory](
[ProductID] [int] NOT NULL,
[StartDate] [datetime] NOT NULL,
[EndDate] [datetime] NULL,
[ListPrice] [money] NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Production.ProductModel

Product model classification.

I F P	Column name	Data type	Nulls	Default	Description
	ProductModelID	int	NO		Primary key for ProductModel records.
	Name	Name	NO		Product model description.
	CatalogDescription	xml	YES		Detailed product catalog information in xml format.
	Instructions	xml	YES		Manufacturing instructions in xml format.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_ProductModel_Name	Name	ASC	Yes	NONCLUSTERED
AK_ProductModel_rowguid	rowguid	ASC	Yes	NONCLUSTERED
PK_ProductModel_ProductModelID	ProductModelID	ASC	Yes	CLUSTERED
PXML_ProductModel_CatalogDescription	CatalogDescription	ASC		XML
PXML_ProductModel_Instructions	Instructions	ASC		XML

Referencing tables

Table name	Foreign key	Primary key or unique constraint
Production.Product	FK_Product_ProductModel_ProductModelID	PK_ProductModel_ProductModelID
Production.ProductModelIllustration	FK_ProductModelIllustration_ProductModel_ProductModelID	PK_ProductModel_ProductModelID
Production.ProductModelProductDescriptionCulture	FK_ProductModelProductDescriptionCulture_ProductModel_ProductModelID	PK_ProductModel_ProductModelID

Objects that depend on Production.ProductModel

[Production.vProductAndDescription \(VIEW\)](#)

[Production.vProductModelCatalogDescription \(VIEW\)](#)





[Production.vProductModelInstructions \(VIEW\)](#)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[ProductModel](
  [ProductModelID] [int] IDENTITY(1,1) NOT NULL,
  [Name] [dbo].[Name] NOT NULL,
  [CatalogDescription] [xml](CONTENT [Production].[ProductDescriptionSchemaCollection]) NULL,
  [Instructions] [xml](CONTENT [Production].[ManuInstructionsSchemaCollection]) NULL,
  [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
  [ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```


Production.ProductModelIllustration

Cross-reference table mapping product models and illustrations.

I F P	Column name	Data type	Nulls	Default	Description
 	ProductModelID	int	NO		Primary key. Foreign key to ProductModel.ProductModelID.
 	IllustrationID	int	NO		Primary key. Foreign key to Illustration.IllustrationID.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_ProductModelIllustration_ProductModelID_IllustrationID	ProductModelID	ASC	Yes	CLUSTERED
PK_ProductModelIllustration_ProductModelID_IllustrationID	IllustrationID	ASC	Yes	CLUSTERED

Foreign keys







Constraint name	Column name	Reference	Description
FK_ProductModelIllustration_IllustrationID	IllustrationID	PK_Illustration_IllustrationID (Production.Illustration)	Foreign key constraint referencing Illustration.IllustrationID.
FK_ProductModelIllustration_ProductModelID	ProductModelID	PK_ProductModel_ProductModelID (Production.ProductModel)	Foreign key constraint referencing ProductModel.ProductModelID.

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[ProductModelIllustration](
  [ProductModelID] [int] NOT NULL,
  [IllustrationID] [int] NOT NULL,
  [ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Production.ProductModelProductDescriptionCulture

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

I F P	Column name	Data type	Nulls	Default	Description
 	ProductModelID	int	NO		Primary key. Foreign key to ProductModel.ProductModelID.
 	ProductDescriptionID	int	NO		Primary key. Foreign key to ProductDescription.ProductDescriptionID.
 	CultureID	nchar(6)	NO		Culture identification number. Foreign key to Culture.CultureID.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_ProductModelProductDescriptionCulture_ProductModelID_ProductDescriptionID_CultureID	ProductModelID	ASC	Yes	CLUSTERED
PK_ProductModelProductDescriptionCulture_ProductModelID_ProductDescriptionID_CultureID	ProductDescriptionID	ASC	Yes	CLUSTERED
PK_ProductModelProductDescriptionCulture_ProductModelID_ProductDescriptionID_CultureID	CultureID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_ProductModelProductDescriptionCulture_ProductDescriptionID	ProductDescriptionID	PK_ProductDescription_ProductDescriptionID (Production.ProductDescription)	Foreign key constraint referencing ProductDescription.ProductDescriptionID.
FK_ProductModelProductDescriptionCulture_CultureID	CultureID	PK_Culture_CultureID (Production.Culture)	Foreign key constraint referencing Culture.CultureID.
FK_ProductModelProductDescriptionCulture_ProductModelID	ProductModelID	PK_ProductModel_ProductModelID (Production.ProductModel)	Foreign key constraint referencing ProductModel.ProductModelID.

Objects that depend on Production.ProductModelProductDescriptionCulture



[Production.vProductAndDescription \(VIEW\)](#)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[ProductModelProductDescriptionCulture](
[ProductModelID] [int] NOT NULL,
[ProductDescriptionID] [int] NOT NULL,
[CultureID] [nchar](6) COLLATE Latin1_General_CS_AS NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Production.ProductPhoto

Product images.

I F P	Column name	Data type	Nulls	Default	Description
	ProductPhotoID	int	NO		Primary key for ProductPhoto records.
	ThumbNailPhoto	varbinary	YES		Small image of the product.
	ThumbNailPhotoFileName	nvarchar(50)	YES		Small image file name.
	LargePhoto	varbinary	YES		Large image of the product.
	LargePhotoFileName	nvarchar(50)	YES		Large image file name.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_ProductPhoto_ProductPhotoID	ProductPhotoID	ASC	Yes	CLUSTERED

Referencing tables





Table name	Foreign key	Primary key or unique constraint
Production.ProductProductPhoto	FK_ProductProductPhoto_ProductPhoto_Pr oductPhotoID	PK_ProductPhoto_ProductPhotoID

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[ProductPhoto](
  [ProductPhotoID] [int] IDENTITY(1,1) NOT NULL,
  [ThumbNailPhoto] [varbinary](max) NULL,
  [ThumbNailPhotoFileName] [nvarchar](50) COLLATE Latin1_General_CS_AS NULL,
  [LargePhoto] [varbinary](max) NULL,
  [LargePhotoFileName] [nvarchar](50) COLLATE Latin1_General_CS_AS NULL,
  [ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Production.ProductProductPhoto

Cross-reference table mapping products and product photos.

I F P	Column name	Data type	Nulls	Default	Description
 	ProductID	int	NO		Product identification number. Foreign key to Product.ProductID.
 	ProductPhotoID	int	NO		Product photo identification number. Foreign key to ProductPhoto.ProductPhotoID.
	Primary	Flag	NO	((0))	0 = Photo is not the principal image. 1 = Photo is the principal image.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_ProductProductPhoto_ProductID_Pr oductPhotoID	ProductID	ASC	Yes	NONCLUSTERED
PK_ProductProductPhoto_ProductID_Pr oductPhotoID	ProductPhotoID	ASC	Yes	NONCLUSTERED

Foreign keys




Constraint name	Column name	Reference	Description
FK_ProductProductPhoto_Produ ct_ProductID	ProductID	PK_Product_ProductID (Production.Product)	Foreign key constraint referencing Product.ProductID.
FK_ProductProductPhoto_Produ ctPhoto_ProductPhotoID	ProductPhotoID	PK_ProductPhoto_ProductPhotoID (Production.ProductPhoto)	Foreign key constraint referencing ProductPhoto.ProductPhotoID.

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[ProductProductPhoto](
[ProductID] [int] NOT NULL,
[ProductPhotoID] [int] NOT NULL,
[Primary] [dbo].[Flag] NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Production.ProductReview

Customer reviews of products they have purchased.

I F P	Column name	Data type	Nulls	Default	Description
	ProductReviewID	int	NO		Primary key for ProductReview records.
	ProductID	int	NO		Product identification number. Foreign key to Product.ProductID.
	ReviewerName	Name	NO		Name of the reviewer.
	ReviewDate	datetime	NO	(getdate())	Date review was submitted.
	EmailAddress	nvarchar(50)	NO		Reviewer's e-mail address.
	Rating	int	NO		Product rating given by the reviewer. Scale is 1 to 5 with 5 as the highest rating.
	Comments	nvarchar(3850)	YES		Reviewer's comments
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
IX_ProductReview_ProductID_Name	ProductID	ASC		NONCLUSTERED
IX_ProductReview_ProductID_Name	ReviewerName	ASC		NONCLUSTERED
PK_ProductReview_ProductReviewID	ProductReviewID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_ProductReview_Product_ProductID	ProductID	PK_Product_ProductID (Production.Product)	Foreign key constraint referencing Product.ProductID.

Check constraints





Check name	Column name	Expression
CK_ProductReview_Rating	Rating	([Rating]>=(1) AND [Rating]<=(5))

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[ProductReview](
[ProductReviewID] [int] IDENTITY(1,1) NOT NULL,
[ProductID] [int] NOT NULL,
[ReviewerName] [dbo].[Name] NOT NULL,
[ReviewDate] [datetime] NOT NULL,
[EmailAddress] [nvarchar](50) COLLATE Latin1_General_CS_AS NOT NULL,
[Rating] [int] NOT NULL,
[Comments] [nvarchar](3850) COLLATE Latin1_General_CS_AS NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Production.ProductSubcategory

Product subcategories. See ProductCategory table.

I F P	Column name	Data type	Nulls	Default	Description
	ProductSubcategoryID	int	NO		Primary key for ProductSubcategory records.
	ProductCategoryID	int	NO		Product category identification number. Foreign key to ProductCategory.ProductCategoryID.
	Name	Name	NO		Subcategory description.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_ProductSubcategory_Name	Name	ASC	Yes	NONCLUSTERED
AK_ProductSubcategory_rowguid	rowguid	ASC	Yes	NONCLUSTERED
PK_ProductSubcategory_ProductSubcategoryID	ProductSubcategoryID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_ProductSubcategory_ProductCategoryID	ProductCategoryID	PK_ProductCategory_ProductCategoryID (Production.ProductCategory)	Foreign key constraint referencing ProductCategory.ProductCategoryID.

Referencing tables



Table name	Foreign key	Primary key or unique constraint
Production.Product	FK_Product_ProductSubcategory_ProductSubcategoryID	PK_ProductSubcategory_ProductSubcategoryID

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[ProductSubcategory](
[ProductSubcategoryID] [int] IDENTITY(1,1) NOT NULL,
[ProductCategoryID] [int] NOT NULL,
[Name] [dbo].[Name] NOT NULL,
[rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Production.ScrapReason

Manufacturing failure reasons lookup table.

I F P	Column name	Data type	Nulls	Default	Description
	ScrapReasonID	smallint	NO		Primary key for ScrapReason records.
	Name	Name	NO		Failure description.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_ScrapReason_Name	Name	ASC	Yes	NONCLUSTERED
PK_ScrapReason_ScrapReasonID	ScrapReasonID	ASC	Yes	CLUSTERED

Referencing tables





Table name	Foreign key	Primary key or unique constraint
Production.WorkOrder	FK_WorkOrder_ScrapReason_ScrapReasonID	PK_ScrapReason_ScrapReasonID

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[ScrapReason](
[ScrapReasonID] [smallint] IDENTITY(1,1) NOT NULL,
[Name] [dbo].[Name] NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Production.TransactionHistory

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

I F P	Column name	Data type	Nulls	Default	Description
	TransactionID	int	NO		Primary key for TransactionHistory records.
	ProductID	int	NO		Product identification number. Foreign key to Product.ProductID.
	ReferenceOrderID	int	NO		Purchase order, sales order, or work order identification number.
	ReferenceOrderLineID	int	NO	((0))	Line number associated with the purchase order, sales order, or work order.
	TransactionDate	datetime	NO	(getdate())	Date and time of the transaction.
	TransactionType	nchar(1)	NO		W = WorkOrder, S = SalesOrder, P = PurchaseOrder
	Quantity	int	NO		Product quantity.
	ActualCost	money	NO		Product cost.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
IX_TransactionHistory_ProductID	ProductID	ASC		NONCLUSTERED
IX_TransactionHistory_ReferenceOrderID_ReferenceOrderLineID	ReferenceOrderID	ASC		NONCLUSTERED
IX_TransactionHistory_ReferenceOrderID_ReferenceOrderLineID	ReferenceOrderLineID	ASC		NONCLUSTERED
PK_TransactionHistory_TransactionID	TransactionID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_TransactionHistory_ProductID_ProductID	ProductID	PK_Product_ProductID (Production.Product)	Foreign key constraint referencing Product.ProductID.

Check constraints

Check name	Column name	Expression
CK_TransactionHistory_TransactionType	TransactionType	(upper(TransactionType))='P' OR upper(TransactionType)='S' OR upper(TransactionType)='W'

Objects that depend on Production.TransactionHistory

[Production.iWorkOrder \(TRIGGER\)](#)

[Production.uWorkOrder \(TRIGGER\)](#)

[Purchasing.iPurchaseOrderDetail \(TRIGGER\)](#)

[Purchasing.uPurchaseOrderDetail \(TRIGGER\)](#)






[Sales.iduSalesOrderDetail \(TRIGGER\)](#)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[TransactionHistory](
[TransactionID] [int] IDENTITY(100000,1) NOT NULL,
[ProductID] [int] NOT NULL,
[ReferenceOrderID] [int] NOT NULL,
[ReferenceOrderLineID] [int] NOT NULL,
[TransactionDate] [datetime] NOT NULL,
[TransactionType] [nchar](1) COLLATE Latin1_General_CS_AS NOT NULL,
[Quantity] [int] NOT NULL,
[ActualCost] [money] NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```


Production.TransactionHistoryArchive

Transactions for previous years.

I F P	Column name	Data type	Nulls	Default	Description
 	TransactionID	int	NO		Primary key for TransactionHistoryArchive records.
	ProductID	int	NO		Product identification number. Foreign key to Product.ProductID.
	ReferenceOrderID	int	NO		Purchase order, sales order, or work order identification number.
	ReferenceOrderLineID	int	NO	((0))	Line number associated with the purchase order, sales order, or work order.
	TransactionDate	datetime	NO	(getdate())	Date and time of the transaction.
	TransactionType	nchar(1)	NO		W = Work Order, S = Sales Order, P = Purchase Order
	Quantity	int	NO		Product quantity.
	ActualCost	money	NO		Product cost.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
IX_TransactionHistoryArchive_ProductID	ProductID	ASC		NONCLUSTERED
IX_TransactionHistoryArchive_ReferenceOrderID_ReferenceOrderLineID	ReferenceOrderID	ASC		NONCLUSTERED
IX_TransactionHistoryArchive_ReferenceOrderID_ReferenceOrderLineID	ReferenceOrderLineID	ASC		NONCLUSTERED
PK_TransactionHistoryArchive_TransactionID	TransactionID	ASC	Yes	CLUSTERED

Check constraints



Check name	Column name	Expression
CK_TransactionHistoryArchive_TransactionType	TransactionType	(upper([TransactionType])='P' OR upper([TransactionType])='S' OR upper([TransactionType])='W')

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[TransactionHistoryArchive](
[TransactionID] [int] NOT NULL,
[ProductID] [int] NOT NULL,
[ReferenceOrderID] [int] NOT NULL,
[ReferenceOrderLineID] [int] NOT NULL,
[TransactionDate] [datetime] NOT NULL,
[TransactionType] [nchar](1) COLLATE Latin1_General_CS_AS NOT NULL,
[Quantity] [int] NOT NULL,
[ActualCost] [money] NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Production.UnitMeasure

Unit of measure lookup table.

I F P	Column name	Data type	Nulls	Default	Description
	UnitMeasureCode	nchar(3)	NO		Primary key.
	Name	Name	NO		Unit of measure description.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_UnitMeasure_Name	Name	ASC	Yes	NONCLUSTERED
PK_UnitMeasure_UnitMeasureCode	UnitMeasureCode	ASC	Yes	CLUSTERED

Referencing tables




Table name	Foreign key	Primary key or unique constraint
Production.Product	FK_Product_UnitMeasure_SizeUnitMeasureCode	PK_UnitMeasure_UnitMeasureCode
Production.Product	FK_Product_UnitMeasure_WeightUnitMeasureCode	PK_UnitMeasure_UnitMeasureCode
Purchasing.ProductVendor	FK_ProductVendor_UnitMeasure_UnitMeasureCode	PK_UnitMeasure_UnitMeasureCode
Production.BillOfMaterials	FK_BillOfMaterials_UnitMeasure_UnitMeasureCode	PK_UnitMeasure_UnitMeasureCode

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[UnitMeasure](
[UnitMeasureCode] [nchar](3) COLLATE Latin1_General_CS_AS NOT NULL,
[Name] [dbo].[Name] NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Production.WorkOrder

Manufacturing work orders.

I F P	Column name	Data type	Nulls	Default	Description
	WorkOrderID	int	NO		Primary key for WorkOrder records.
	ProductID	int	NO		Product identification number. Foreign key to Product.ProductID.
	OrderQty	int	NO		Product quantity to build.
	StockedQty	int	NO		Quantity built and put in inventory.
	ScrappedQty	smallint	NO		Quantity that failed inspection.
	StartDate	datetime	NO		Work order start date.
	EndDate	datetime	YES		Work order end date.
	DueDate	datetime	NO		Work order due date.
	ScrapReasonID	smallint	YES		Reason for inspection failure.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
IX_WorkOrder_ProductID	ProductID	ASC		NONCLUSTERED
IX_WorkOrder_ScrapReasonID	ScrapReasonID	ASC		NONCLUSTERED
PK_WorkOrder_WorkOrderID	WorkOrderID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_WorkOrder_Product_ProductID	ProductID	PK_Product_ProductID (Production.Product)	Foreign key constraint referencing Product.ProductID.
FK_WorkOrder_ScrapReason_ScrapReasonID	ScrapReasonID	PK_ScrapReason_ScrapReasonID (Production.ScrapReason)	Foreign key constraint referencing ScrapReason.ScrapReasonID.

Referencing tables

Table name	Foreign key	Primary key or unique constraint
Production.WorkOrderRouting	FK_WorkOrderRouting_WorkOrder_WorkOrderID	PK_WorkOrder_WorkOrderID

Check constraints

Check name	Column name	Expression
CK_WorkOrder_OrderQty	OrderQty	([OrderQty]>(0))
CK_WorkOrder_ScrappedQty	ScrappedQty	([ScrappedQty]>=(0))
CK_WorkOrder_EndDate	StartDate	([EndDate]>=[StartDate] OR [EndDate] IS NULL)
CK_WorkOrder_EndDate	EndDate	([EndDate]>=[StartDate] OR [EndDate] IS NULL)

Triggers

Trigger name: Production.iWorkOrder

Created on: 26 Apr 2006

Trigger type: INSERT

Trigger active: Yes

```
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;

```

Trigger name: [Production.uWorkOrder](#)

Created on: [26 Apr 2006](#)

Trigger type: [UPDATE](#)

Trigger active: [Yes](#)

```

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;

```

Objects that depend on Production.WorkOrder

Production.uWorkOrder (TRIGGER)






Production.WorkOrder (TABLE)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Production].[WorkOrder](
[WorkOrderID] [int] IDENTITY(1,1) NOT NULL,
[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
) ON [PRIMARY]
```

Production.WorkOrderRouting

Work order details.

I F P	Column name	Data type	Nulls	Default	Description
 	WorkOrderID	int	NO		Primary key. Foreign key to WorkOrder.WorkOrderID.
	ProductID	int	NO		Primary key. Foreign key to Product.ProductID.
	OperationSequence	smallint	NO		Primary key. Indicates the manufacturing process sequence.
	LocationID	smallint	NO		Manufacturing location where the part is processed. Foreign key to Location.LocationID.
	ScheduledStartDate	datetime	NO		Planned manufacturing start date.
	ScheduledEndDate	datetime	NO		Planned manufacturing end date.
	ActualStartDate	datetime	YES		Actual start date.
	ActualEndDate	datetime	YES		Actual end date.
	ActualResourceHrs	decimal(9,4)	YES		Number of manufacturing hours used.
	PlannedCost	money	NO		Estimated manufacturing cost.
	ActualCost	money	YES		Actual manufacturing cost.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
IX_WorkOrderRouting_ProductID	ProductID	ASC		NONCLUSTERED
PK_WorkOrderRouting_WorkOrderID_ProductID_OperationSequence	WorkOrderID	ASC	Yes	CLUSTERED
PK_WorkOrderRouting_WorkOrderID_ProductID_OperationSequence	ProductID	ASC	Yes	CLUSTERED
PK_WorkOrderRouting_WorkOrderID_ProductID_OperationSequence	OperationSequence	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_WorkOrderRouting_Location_LocationID	LocationID	PK_Location_LocationID (Production.Location)	Foreign key constraint referencing Location.LocationID.
FK_WorkOrderRouting_WorkOrderID_WorkOrderID	WorkOrderID	PK_WorkOrder_WorkOrderID (Production.WorkOrder)	Foreign key constraint referencing WorkOrder.WorkOrderID.

Check constraints

Check name	Column name	Expression
CK_WorkOrderRouting_ScheduledEndDate	ScheduledStartDate	([ScheduledEndDate]>=[ScheduledStartDate])
CK_WorkOrderRouting_ScheduledEndDate	ScheduledEndDate	([ScheduledEndDate]>=[ScheduledStartDate])
CK_WorkOrderRouting_ActualEndDate	ActualStartDate	([ActualEndDate]>=[ActualStartDate] OR [ActualEndDate] IS NULL OR [ActualStartDate] IS NULL)
CK_WorkOrderRouting_ActualEndDate	ActualEndDate	([ActualEndDate]>=[ActualStartDate] OR [ActualEndDate] IS NULL OR [ActualStartDate] IS NULL)
CK_WorkOrderRouting_ActualResourceHrs	ActualResourceHrs	([ActualResourceHrs]>=(0.0000))
CK_WorkOrderRouting_PlannedCost	PlannedCost	([PlannedCost]>(0.00))
CK_WorkOrderRouting_ActualCost	ActualCost	([ActualCost]>(0.00))







Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
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  
) ON [PRIMARY]
```

Purchasing.ProductVendor

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

I F P	Column name	Data type	Nulls	Default	Description
 	ProductID	int	NO		Primary key. Foreign key to Product.ProductID.
 	VendorID	int	NO		Primary key. Foreign key to Vendor.VendorID.
	AverageLeadTime	int	NO		The average span of time (in days) between placing an order with the vendor and receiving the purchased product.
	StandardPrice	money	NO		The vendor's usual selling price.
	LastReceiptCost	money	YES		The selling price when last purchased.
	LastReceiptDate	datetime	YES		Date the product was last received by the vendor.
	MinOrderQty	int	NO		The maximum quantity that should be ordered.
	MaxOrderQty	int	NO		The minimum quantity that should be ordered.
	OnOrderQty	int	YES		The quantity currently on order.
 	UnitMeasureCode	nchar(3)	NO		The product's unit of measure.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
IX_ProductVendor_UnitMeasureCode	UnitMeasureCode	ASC		NONCLUSTERED
IX_ProductVendor_VendorID	VendorID	ASC		NONCLUSTERED
PK_ProductVendor_ProductID_VendorID	ProductID	ASC	Yes	CLUSTERED
PK_ProductVendor_ProductID_VendorID	VendorID	ASC	Yes	CLUSTERED

Foreign keys

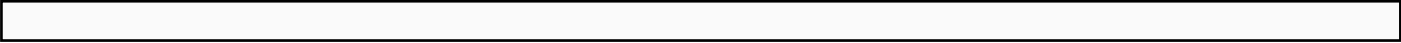
Constraint name	Column name	Reference	Description
FK_ProductVendor_Product_ProductID	ProductID	PK_Product_ProductID (Production.Product)	Foreign key constraint referencing Product.ProductID.
FK_ProductVendor_UnitMeasure_UnitMeasureCode	UnitMeasureCode	PK_UnitMeasure_UnitMeasureCode (Production.UnitMeasure)	Foreign key constraint referencing UnitMeasure.UnitMeasureCode.
FK_ProductVendor_Vendor_VendorID	VendorID	PK_Vendor_VendorID (Purchasing.Vendor)	Foreign key constraint referencing Vendor.VendorID.

Check constraints

Check name	Column name	Expression
CK_ProductVendor_AverageLeadTime	AverageLeadTime	([AverageLeadTime]>=(1))
CK_ProductVendor_StandardPrice	StandardPrice	([StandardPrice]>(0.00))
CK_ProductVendor_LastReceiptCost	LastReceiptCost	([LastReceiptCost]>(0.00))
CK_ProductVendor_MinOrderQty	MinOrderQty	([MinOrderQty]>=(1))
CK_ProductVendor_MaxOrderQty	MaxOrderQty	([MaxOrderQty]>=(1))
CK_ProductVendor_OnOrderQty	OnOrderQty	([OnOrderQty]>=(0))






Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Purchasing].[ProductVendor](
[ProductID] [int] NOT NULL,
[VendorID] [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 Latin1_General_CS_AS NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Purchasing.PurchaseOrderDetail

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

I F P	Column name	Data type	Nulls	Default	Description
 	PurchaseOrderID	int	NO		Primary key. Foreign key to PurchaseOrderHeader.PurchaseOrderID.
	PurchaseOrderDetailID	int	NO		Primary key. One line number per purchased product.
	DueDate	datetime	NO		Date the product is expected to be received.
	OrderQty	smallint	NO		Quantity ordered.
 	ProductID	int	NO		Product identification number. Foreign key to Product.ProductID.
	UnitPrice	money	NO		Vendor's selling price of a single product.
	LineTotal	money	NO		Per product subtotal. Computed as OrderQty * UnitPrice.
	ReceivedQty	decimal(8,2)	NO		Quantity actually received from the vendor.
	RejectedQty	decimal(8,2)	NO		Quantity rejected during inspection.
	StockedQty	decimal(9,2)	NO		Quantity accepted into inventory. Computed as ReceivedQty - RejectedQty.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
IX_PurchaseOrderDetail_ProductID	ProductID	ASC		NONCLUSTERED
PK_PurchaseOrderDetail_PurchaseOrderID_PurchaseOrderDetailID	PurchaseOrderID	ASC	Yes	CLUSTERED
PK_PurchaseOrderDetail_PurchaseOrderID_PurchaseOrderDetailID	PurchaseOrderDetailID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_PurchaseOrderDetail_PurchaseOrderHeader_PurchaseOrderID	PurchaseOrderID	PK_PurchaseOrderHeader_PurchaseOrderID (Purchasing.PurchaseOrderHeader)	Foreign key constraint referencing PurchaseOrderHeader.PurchaseOrderID.
FK_PurchaseOrderDetail_ProductID	ProductID	PK_Product_ProductID (Production.Product)	Foreign key constraint referencing Product.ProductID.

Check constraints

Check name	Column name	Expression
CK_PurchaseOrderDetail_OrderQty	OrderQty	([OrderQty]>(0))
CK_PurchaseOrderDetail_UnitPrice	UnitPrice	([UnitPrice]>=(0.00))
CK_PurchaseOrderDetail_ReceivedQty	ReceivedQty	([ReceivedQty]>=(0.00))
CK_PurchaseOrderDetail_RejectedQty	RejectedQty	([RejectedQty]>=(0.00))

Triggers

Trigger name: Purchasing.iPurchaseOrderDetail Created on: 26 Apr 2006
Trigger type: INSERT Trigger active: Yes

```
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;

```

Trigger name: [Purchasing.uPurchaseOrderDetail](#)
Trigger type: [UPDATE](#)

Created on: [26 Apr 2006](#)
Trigger active: [Yes](#)

```

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;

```

Objects that depend on Purchasing.PurchaseOrderDetail

[Purchasing.iPurchaseOrderDetail \(TRIGGER\)](#)

[Purchasing.PurchaseOrderDetail \(TABLE\)](#)

[Purchasing.uPurchaseOrderDetail \(TRIGGER\)](#)

Table definition






```

SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Purchasing].[PurchaseOrderDetail](
[PurchaseOrderID] [int] NOT NULL,
[PurchaseOrderDetailID] [int] IDENTITY(1,1) NOT NULL,
[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
) ON [PRIMARY]

```

Purchasing.PurchaseOrderHeader

General purchase order information. See *PurchaseOrderDetail*.

I F P	Column name	Data type	Nulls	Default	Description
	PurchaseOrderID	int	NO		Primary key.
	RevisionNumber	tinyint	NO	((0))	Incremental number to track changes to the purchase order over time.
	Status	tinyint	NO	((1))	Order current status. 1 = Pending; 2 = Approved; 3 = Rejected; 4 = Complete
	EmployeeID	int	NO		Employee who created the purchase order. Foreign key to Employee.EmployeeID.
	VendorID	int	NO		Vendor with whom the purchase order is placed. Foreign key to Vendor.VendorID.
	ShipMethodID	int	NO		Shipping method. Foreign key to ShipMethod.ShipMethodID.
	OrderDate	datetime	NO	(getdate())	Purchase order creation date.
	ShipDate	datetime	YES		Estimated shipment date from the vendor.
	SubTotal	money	NO	((0.00))	Purchase order subtotal. Computed as SUM (PurchaseOrderDetail.LineTotal)for the appropriate PurchaseOrderID.
	TaxAmt	money	NO	((0.00))	Tax amount.
	Freight	money	NO	((0.00))	Shipping cost.
	TotalDue	money	NO		Total due to vendor. Computed as Subtotal + TaxAmt + Freight.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
IX_PurchaseOrderHeader_EmployeeID	EmployeeID	ASC		NONCLUSTERED
IX_PurchaseOrderHeader_VendorID	VendorID	ASC		NONCLUSTERED
PK_PurchaseOrderHeader_PurchaseOrderID	PurchaseOrderID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_PurchaseOrderHeader_EmployeeID	EmployeeID	PK_Employee_EmployeeID (HumanResources.Employee)	Foreign key constraint referencing Employee.EmployeeID.
FK_PurchaseOrderHeader_VendorID	VendorID	PK_Vendor_VendorID (Purchasing.Vendor)	Foreign key constraint referencing Vendor.VendorID.
FK_PurchaseOrderHeader_ShipMethodID	ShipMethodID	PK_ShipMethod_ShipMethodID (Purchasing.ShipMethod)	Foreign key constraint referencing ShipMethod.ShipMethodID.

Referencing tables

Table name	Foreign key	Primary key or unique constraint
Purchasing.PurchaseOrderDetail	FK_PurchaseOrderDetail_PurchaseOrderHeader_PurchaseOrderID	PK_PurchaseOrderHeader_PurchaseOrderID

Check constraints

Check name	Column name	Expression
CK_PurchaseOrderHeader_Status	Status	([Status]>=(1) AND [Status]<=(4))
CK_PurchaseOrderHeader_ShipDate	OrderDate	([ShipDate]>=[OrderDate] OR [ShipDate] IS NULL)
CK_PurchaseOrderHeader_ShipDate	ShipDate	([ShipDate]>=[OrderDate] OR [ShipDate] IS NULL)
CK_PurchaseOrderHeader_SubTotal	SubTotal	([SubTotal]>=(0.00))
CK_PurchaseOrderHeader_TaxAmt	TaxAmt	([TaxAmt]>=(0.00))
CK_PurchaseOrderHeader_Freight	Freight	([Freight]>=(0.00))

Triggers

Trigger name:	Purchasing.uPurchaseOrderHeader	Created on:	26 Apr 2006
Trigger type:	UPDATE	Trigger active:	Yes

```

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;

```

Objects that depend on Purchasing.PurchaseOrderHeader

Purchasing.iPurchaseOrderDetail (TRIGGER)

Purchasing.PurchaseOrderHeader (TABLE)

Purchasing.uPurchaseOrderDetail (TRIGGER)

Purchasing.uPurchaseOrderHeader (TRIGGER)

Table definition




```

SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Purchasing].[PurchaseOrderHeader](
[PurchaseOrderID] [int] IDENTITY(1,1) NOT NULL,
[RevisionNumber] [tinyint] NOT NULL,
[Status] [tinyint] NOT NULL,
[EmployeeID] [int] NOT NULL,
[VendorID] [int] NOT NULL,
[ShipMethodID] [int] NOT NULL,
[OrderDate] [datetime] NOT NULL,
[ShipDate] [datetime] NULL,
[SubTotal] [money] NOT NULL,
[TaxAmt] [money] NOT NULL,
[Freight] [money] NOT NULL,
[TotalDue] AS (isnull(([SubTotal]+[TaxAmt])+[Freight],(0))) PERSISTED NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]

```

Purchasing.ShipMethod

Shipping company lookup table.

I F P	Column name	Data type	Nulls	Default	Description
	ShipMethodID	int	NO		Primary key for ShipMethod records.
	Name	Name	NO		Shipping company name.
	ShipBase	money	NO	((0.00))	Minimum shipping charge.
	ShipRate	money	NO	((0.00))	Shipping charge per pound.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_ShipMethod_Name	Name	ASC	Yes	NONCLUSTERED
AK_ShipMethod_rowguid	rowguid	ASC	Yes	NONCLUSTERED
PK_ShipMethod_ShipMethodID	ShipMethodID	ASC	Yes	CLUSTERED

Referencing tables

Table name	Foreign key	Primary key or unique constraint
Sales.SalesOrderHeader	FK_SalesOrderHeader_ShipMethod_ShipMethodID	PK_ShipMethod_ShipMethodID
Purchasing.PurchaseOrderHeader	FK_PurchaseOrderHeader_ShipMethod_ShipMethodID	PK_ShipMethod_ShipMethodID

Check constraints



Check name	Column name	Expression
CK_ShipMethod_ShipBase	ShipBase	([ShipBase]>(0.00))
CK_ShipMethod_ShipRate	ShipRate	([ShipRate]>(0.00))

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Purchasing].[ShipMethod](
[ShipMethodID] [int] IDENTITY(1,1) NOT NULL,
[Name] [dbo].[Name] NOT NULL,
[ShipBase] [money] NOT NULL,
[ShipRate] [money] NOT NULL,
[rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Purchasing.Vendor

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

I F P	Column name	Data type	Nulls	Default	Description
	VendorID	int	NO		Primary key for Vendor records.
	AccountNumber	AccountNumber	NO		Vendor account (identification) number.
	Name	Name	NO		Company name.
	CreditRating	tinyint	NO		1 = Superior, 2 = Excellent, 3 = Above average, 4 = Average, 5 = Below average
	PreferredVendorStatus	Flag	NO	((1))	0 = Do not use if another vendor is available. 1 = Preferred over other vendors supplying the same product.
	ActiveFlag	Flag	NO	((1))	0 = Vendor no longer used. 1 = Vendor is actively used.
	PurchasingWebServiceURL	nvarchar(1024)	YES		Vendor URL.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_Vendor_AccountNumber	AccountNumber	ASC	Yes	NONCLUSTERED
PK_Vendor_VendorID	VendorID	ASC	Yes	CLUSTERED

Referencing tables

Table name	Foreign key	Primary key or unique constraint
Purchasing.ProductVendor	FK_ProductVendor_Vendor_VendorID	PK_Vendor_VendorID
Purchasing.PurchaseOrderHeader	FK_PurchaseOrderHeader_Vendor_VendorID	PK_Vendor_VendorID
Purchasing.VendorAddress	FK_VendorAddress_Vendor_VendorID	PK_Vendor_VendorID
Purchasing.VendorContact	FK_VendorContact_Vendor_VendorID	PK_Vendor_VendorID

Check constraints

Check name	Column name	Expression
CK_Vendor_CreditRating	CreditRating	([CreditRating]>=(1) AND [CreditRating]<=(5))

Triggers

Trigger name: Purchasing.dVendor Created on: 26 Apr 2006
Trigger type: INSTEAD OF DELETE Trigger active: Yes

```
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;

```

Objects that depend on Purchasing.Vendor

[Purchasing.vVendor \(VIEW\)](#)

Table definition






```

SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Purchasing].[Vendor](
[VendorID] [int] IDENTITY(1,1) NOT NULL,
[AccountNumber] [dbo].[AccountNumber] NOT NULL,
[Name] [dbo].[Name] NOT NULL,
[CreditRating] [tinyint] NOT NULL,
[PreferredVendorStatus] [dbo].[Flag] NOT NULL,
[ActiveFlag] [dbo].[Flag] NOT NULL,
[PurchasingWebServiceURL] [nvarchar](1024) COLLATE Latin1_General_CS_AS NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]

```

Purchasing.VendorAddress

Cross-reference mapping vendors and addresses.

I F P	Column name	Data type	Nulls	Default	Description
 	VendorID	int	NO		Primary key. Foreign key to Vendor.VendorID.
 	AddressID	int	NO		Primary key. Foreign key to Address.AddressID.
	AddressTypeID	int	NO		Address type. Foreign key to AddressType.AddressTypeID.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
IX_VendorAddress_AddressID	AddressID	ASC		NONCLUSTERED
PK_VendorAddress_VendorID_AddressID	VendorID	ASC	Yes	CLUSTERED
PK_VendorAddress_VendorID_AddressID	AddressID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_VendorAddress_Address_AddressID	AddressID	PK_Address_AddressID (Person.Address)	Foreign key constraint referencing Address.AddressID.
FK_VendorAddress_Vendor_VendorID	VendorID	PK_Vendor_VendorID (Purchasing.Vendor)	Foreign key constraint referencing Vendor.VendorID.
FK_VendorAddress_AddressType_AddressTypeID	AddressTypeID	PK_AddressType_AddressTypeID (Person.AddressType)	Foreign key constraint referencing AddressType.AddressTypeID.

Objects that depend on Purchasing.VendorAddress




[Purchasing.vVendor \(VIEW\)](#)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Purchasing].[VendorAddress](
[VendorID] [int] NOT NULL,
[AddressID] [int] NOT NULL,
[AddressTypeID] [int] NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Purchasing.VendorContact

Cross-reference table mapping vendors and their employees.

I F P	Column name	Data type	Nulls	Default	Description
	VendorID	int	NO		Primary key.
	ContactID	int	NO		Contact (Vendor employee) identification number. Foreign key to Contact.ContactID.
	ContactTypeID	int	NO		Contact type such as sales manager, or sales agent.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
IX_VendorContact_ContactID	ContactID	ASC		NONCLUSTERED
IX_VendorContact_ContactTypeID	ContactTypeID	ASC		NONCLUSTERED
PK_VendorContact_VendorID_ContactID	VendorID	ASC	Yes	CLUSTERED
PK_VendorContact_VendorID_ContactID	ContactID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_VendorContact_Contact_ContactID	ContactID	PK_Contact_ContactID (Person.Contact)	Foreign key constraint referencing Contact.ContactID.
FK_VendorContact_ContactType_ContactTypeID	ContactTypeID	PK_ContactType_ContactTypeID (Person.ContactType)	Foreign key constraint referencing ContactType.ContactTypeID.
FK_VendorContact_Vendor_VendorID	VendorID	PK_Vendor_VendorID (Purchasing.Vendor)	Foreign key constraint referencing Vendor.VendorID.

Objects that depend on Purchasing.VendorContact





[Purchasing.vVendor \(VIEW\)](#)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Purchasing].[VendorContact](
  [VendorID] [int] NOT NULL,
  [ContactID] [int] NOT NULL,
  [ContactTypeID] [int] NOT NULL,
  [ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Sales.ContactCreditCard

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

I F P	Column name	Data type	Nulls	Default	Description
 	ContactID	int	NO		Customer identification number. Foreign key to Contact.ContactID.
 	CreditCardID	int	NO		Credit card identification number. Foreign key to CreditCard.CreditCardID.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_ContactCreditCard_ContactID_CreditCardID	ContactID	ASC	Yes	CLUSTERED
PK_ContactCreditCard_ContactID_CreditCardID	CreditCardID	ASC	Yes	CLUSTERED

Foreign keys





Constraint name	Column name	Reference	Description
FK_ContactCreditCard_Contact_ContactID	ContactID	PK_Contact_ContactID (Person.Contact)	Foreign key constraint referencing Contact.ContactID.
FK_ContactCreditCard_CreditCard_CreditCardID	CreditCardID	PK_CreditCard_CreditCardID (Sales.CreditCard)	Foreign key constraint referencing CreditCard.CreditCardID.

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Sales].[ContactCreditCard](
[ContactID] [int] NOT NULL,
[CreditCardID] [int] NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Sales.CountryRegionCurrency

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

I F P	Column name	Data type	Nulls	Default	Description
 	CountryRegionCode	nvarchar(3)	NO		ISO code for countries and regions. Foreign key to CountryRegion.CountryRegionCode.
 	CurrencyCode	nchar(3)	NO		ISO standard currency code. Foreign key to Currency.CurrencyCode.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
IX_CountryRegionCurrency_CurrencyCode	CurrencyCode	ASC		NONCLUSTERED
PK_CountryRegionCurrency_CountryRegionCode_CurrencyCode	CountryRegionCode	ASC	Yes	CLUSTERED
PK_CountryRegionCurrency_CountryRegionCode_CurrencyCode	CurrencyCode	ASC	Yes	CLUSTERED

Foreign keys



Constraint name	Column name	Reference	Description
FK_CountryRegionCurrency_CountryRegion_CountryRegionCode	CountryRegionCode	PK_CountryRegion_CountryRegionCode (Person.CountryRegion)	Foreign key constraint referencing CountryRegion.CountryRegionCode.
FK_CountryRegionCurrency_Currency_CurrencyCode	CurrencyCode	PK_Currency_CurrencyCode (Sales.Currency)	Foreign key constraint referencing Currency.CurrencyCode.

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Sales].[CountryRegionCurrency](
[CountryRegionCode] [nvarchar](3) COLLATE Latin1_General_CS_AS NOT NULL,
[CurrencyCode] [nchar](3) COLLATE Latin1_General_CS_AS NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Sales.CreditCard

Customer credit card information.

I F P	Column name	Data type	Nulls	Default	Description
	CreditCardID	int	NO		Primary key for CreditCard records.
	CardType	nvarchar(50)	NO		Credit card name.
	CardNumber	nvarchar(25)	NO		Credit card number.
	ExpMonth	tinyint	NO		Credit card expiration month.
	ExpYear	smallint	NO		Credit card expiration year.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_CreditCard_CardNumber	CardNumber	ASC	Yes	NONCLUSTERED
PK_CreditCard_CreditCardID	CreditCardID	ASC	Yes	CLUSTERED

Referencing tables



Table name	Foreign key	Primary key or unique constraint
Sales.SalesOrderHeader	FK_SalesOrderHeader_CreditCard_CreditCardID	PK_CreditCard_CreditCardID
Sales.ContactCreditCard	FK_ContactCreditCard_CreditCard_CreditCardID	PK_CreditCard_CreditCardID

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Sales].[CreditCard](
  [CreditCardID] [int] IDENTITY(1,1) NOT NULL,
  [CardType] [nvarchar](50) COLLATE Latin1_General_CS_AS NOT NULL,
  [CardNumber] [nvarchar](25) COLLATE Latin1_General_CS_AS NOT NULL,
  [ExpMonth] [tinyint] NOT NULL,
  [ExpYear] [smallint] NOT NULL,
  [ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Sales.Currency

Lookup table containing standard ISO currencies.

I F P	Column name	Data type	Nulls	Default	Description
	CurrencyCode	nchar(3)	NO		The ISO code for the Currency.
	Name	Name	NO		Currency name.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_Currency_Name	Name	ASC	Yes	NONCLUSTERED
PK_Currency_CurrencyCode	CurrencyCode	ASC	Yes	CLUSTERED

Referencing tables





Table name	Foreign key	Primary key or unique constraint
Sales.CountryRegionCurrency	FK_CountryRegionCurrency_Currency_CurrencyCode	PK_Currency_CurrencyCode
Sales.CurrencyRate	FK_CurrencyRate_Currency_FromCurrencyCode	PK_Currency_CurrencyCode
Sales.CurrencyRate	FK_CurrencyRate_Currency_ToCurrencyCode	PK_Currency_CurrencyCode

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Sales].[Currency](
  [CurrencyCode] [nchar](3) COLLATE Latin1_General_CS_AS NOT NULL,
  [Name] [dbo].[Name] NOT NULL,
  [ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Sales.CurrencyRate

Currency exchange rates.

I F P	Column name	Data type	Nulls	Default	Description
	CurrencyRateID	int	NO		Primary key for CurrencyRate records.
	CurrencyRateDate	datetime	NO		Date and time the exchange rate was obtained.
	FromCurrencyCode	nchar(3)	NO		Exchange rate was converted from this currency code.
	ToCurrencyCode	nchar(3)	NO		Exchange rate was converted to this currency code.
	AverageRate	money	NO		Average exchange rate for the day.
	EndOfDayRate	money	NO		Final exchange rate for the day.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_CurrencyRate_CurrencyRateDate_FromCurrencyCode_ToCurrencyCode	CurrencyRateDate	ASC	Yes	NONCLUSTERED
AK_CurrencyRate_CurrencyRateDate_FromCurrencyCode_ToCurrencyCode	FromCurrencyCode	ASC	Yes	NONCLUSTERED
AK_CurrencyRate_CurrencyRateDate_FromCurrencyCode_ToCurrencyCode	ToCurrencyCode	ASC	Yes	NONCLUSTERED
PK_CurrencyRate_CurrencyRateID	CurrencyRateID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_CurrencyRate_Currency_FromCurrencyCode	FromCurrencyCode	PK_Currency_CurrencyCode (Sales.Currency)	Foreign key constraint referencing Currency.FromCurrencyCode.
FK_CurrencyRate_Currency_ToCurrencyCode	ToCurrencyCode	PK_Currency_CurrencyCode (Sales.Currency)	Foreign key constraint referencing Currency.ToCurrencyCode.

Referencing tables





Table name	Foreign key	Primary key or unique constraint
Sales.SalesOrderHeader	FK_SalesOrderHeader_CurrencyRate_CurrencyRateID	PK_CurrencyRate_CurrencyRateID

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Sales].[CurrencyRate](
  [CurrencyRateID] [int] IDENTITY(1,1) NOT NULL,
  [CurrencyRateDate] [datetime] NOT NULL,
  [FromCurrencyCode] [nchar](3) COLLATE Latin1_General_CS_AS NOT NULL,
  [ToCurrencyCode] [nchar](3) COLLATE Latin1_General_CS_AS NOT NULL,
  [AverageRate] [money] NOT NULL,
  [EndOfDayRate] [money] NOT NULL,
  [ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```


Sales.Customer

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

I F P	Column name	Data type	Nulls	Default	Description
	CustomerID	int	NO		Primary key for Customer records.
	TerritoryID	int	YES		ID of the territory in which the customer is located. Foreign key to SalesTerritory.SalesTerritoryID.
	AccountNumber	varchar(10)	NO		Unique number identifying the customer assigned by the accounting system.
	CustomerType	nchar(1)	NO		Customer type: I = Individual, S = Store
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_Customer_AccountNumber	AccountNumber	ASC	Yes	NONCLUSTERED
AK_Customer_rowguid	rowguid	ASC	Yes	NONCLUSTERED
IX_Customer_TerritoryID	TerritoryID	ASC		NONCLUSTERED
PK_Customer_CustomerID	CustomerID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_Customer_SalesTerritory_TerritoryID	TerritoryID	PK_SalesTerritory_TerritoryID (Sales.SalesTerritory)	Foreign key constraint referencing SalesTerritory.TerritoryID.

Referencing tables

Table name	Foreign key	Primary key or unique constraint
Sales.Store	FK_Store_Customer_CustomerID	PK_Customer_CustomerID
Sales.Individual	FK_Individual_Customer_CustomerID	PK_Customer_CustomerID
Sales.SalesOrderHeader	FK_SalesOrderHeader_Customer_CustomerID	PK_Customer_CustomerID
Sales.CustomerAddress	FK_CustomerAddress_Customer_CustomerID	PK_Customer_CustomerID

Check constraints

Check name	Column name	Expression
CK_Customer_CustomerType	CustomerType	(upper([CustomerType])='I' OR upper([CustomerType])='S')

Objects that depend on Sales.Customer

[Sales.Customer \(TABLE\)](#)

[Sales.vIndividualCustomer \(VIEW\)](#)







[Sales.vStoreWithDemographics \(VIEW\)](#)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
SET ARITHABORT ON
CREATE TABLE [Sales].[Customer](
[CustomerID] [int] IDENTITY(1,1) NOT FOR REPLICATION NOT NULL,
[TerritoryID] [int] NULL,
[AccountNumber] AS (isnull('AW'+[dbo].[ufnLeadingZeros]([CustomerID]), '')),
[CustomerType] [nchar](1) COLLATE Latin1_General_CS_AS NOT NULL,
[rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Sales.CustomerAddress

Cross-reference table mapping customers to their address(es).

I F P	Column name	Data type	Nulls	Default	Description
 	CustomerID	int	NO		Primary key. Foreign key to Customer.CustomerID.
 	AddressID	int	NO		Primary key. Foreign key to Address.AddressID.
	AddressTypeID	int	NO		Address type. Foreign key to AddressType.AddressTypeID.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_CustomerAddress_rowguid	rowguid	ASC	Yes	NONCLUSTERED
PK_CustomerAddress_CustomerID_AddressID	CustomerID	ASC	Yes	CLUSTERED
PK_CustomerAddress_CustomerID_AddressID	AddressID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_CustomerAddress_Address_AddressID	AddressID	PK_Address_AddressID (Person.Address)	Foreign key constraint referencing Address.AddressID.
FK_CustomerAddress_AddressType_AddressTypeID	AddressTypeID	PK_AddressType_AddressTypeID (Person.AddressType)	Foreign key constraint referencing AddressType.AddressTypeID.
FK_CustomerAddress_Customer_CustomerID	CustomerID	PK_Customer_CustomerID (Sales.Customer)	Foreign key constraint referencing Customer.CustomerID.

Objects that depend on Sales.CustomerAddress

[Sales.vIndividualCustomer \(VIEW\)](#)





[Sales.vStoreWithDemographics \(VIEW\)](#)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Sales].[CustomerAddress](
[CustomerID] [int] NOT NULL,
[AddressID] [int] NOT NULL,
[AddressTypeID] [int] NOT NULL,
[rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Sales.Individual

Demographic data about customers that purchase Adventure Works products online.

I F P	Column name	Data type	Nulls	Default	Description
 	CustomerID	int	NO		Unique customer identification number. Foreign key to Customer.CustomerID.
	ContactID	int	NO		Identifies the customer in the Contact table. Foreign key to Contact.ContactID.
	Demographics	xml	YES		Personal information such as hobbies, and income collected from online shoppers. Used for sales analysis.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_Individual_CustomerID	CustomerID	ASC	Yes	CLUSTERED
PXML_Individual_Demographics	Demographics	ASC		XML
XMLPATH_Individual_Demographics	Demographics	ASC		XML
XMLPROPERTY_Individual_Demographics	Demographics	ASC		XML
XMLVALUE_Individual_Demographics	Demographics	ASC		XML

Foreign keys

Constraint name	Column name	Reference	Description
FK_Individual_Customer_CustomerID	CustomerID	PK_Customer_CustomerID (Sales.Customer)	Foreign key constraint referencing Customer.CustomerID.
FK_Individual_Contact_ContactID	ContactID	PK_Contact_ContactID (Person.Contact)	Foreign key constraint referencing Contact.ContactID.

Triggers

Trigger name: Sales.iuIndividual

Created on: 26 Apr 2006

Trigger type: INSERT

Trigger active: Yes

```
CREATE TRIGGER [Sales].[iuIndividual] ON [Sales].[Individual]
AFTER INSERT, UPDATE NOT FOR REPLICATION AS
BEGIN
    DECLARE @Count int;
    SET @Count = @@ROWCOUNT;
    IF @Count = 0
        RETURN;
    SET NOCOUNT ON;
    -- Only allow the Customer to be a Store OR Individual
    IF EXISTS (SELECT * FROM inserted INNER JOIN [Sales].[Store]
        ON inserted.[CustomerID] = [Sales].[Store].[CustomerID])
        BEGIN
            -- Rollback any active or uncommittable transactions
            IF @@TRANCOUNT > 0
                BEGIN
                    ROLLBACK TRANSACTION;
                END
            END;
        IF UPDATE([CustomerID]) OR UPDATE([Demographics])
            BEGIN
                UPDATE [Sales].[Individual]
                SET [Sales].[Individual].[Demographics] = N'
0.00
'
                FROM inserted
                WHERE [Sales].[Individual].[CustomerID] = inserted.[CustomerID]
                AND inserted.[Demographics] IS NULL;
                UPDATE [Sales].[Individual]
                SET [Demographics].modify(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey";
```

```

insert 0.00
as first
into (/IndividualSurvey)[1]' )
FROM inserted
WHERE [Sales].[Individual].[CustomerID] = inserted.[CustomerID]
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;
CREATE TRIGGER [Sales].[iuIndividual] ON [Sales].[Individual]
AFTER INSERT, UPDATE NOT FOR REPLICATION AS
BEGIN
DECLARE @Count int;
SET @Count = @@ROWCOUNT;
IF @Count = 0
RETURN;
SET NOCOUNT ON;
-- Only allow the Customer to be a Store OR Individual
IF EXISTS (SELECT * FROM inserted INNER JOIN [Sales].[Store]
ON inserted.[CustomerID] = [Sales].[Store].[CustomerID])
BEGIN
-- Rollback any active or uncommittable transactions
IF @@TRANCOUNT > 0
BEGIN
ROLLBACK TRANSACTION;
END
END;
IF UPDATE([CustomerID]) OR UPDATE([Demographics])
BEGIN
UPDATE [Sales].[Individual]
SET [Sales].[Individual].[Demographics] = N'
0.00
'
FROM inserted
WHERE [Sales].[Individual].[CustomerID] = inserted.[CustomerID]
AND inserted.[Demographics] IS NULL;
UPDATE [Sales].[Individual]
SET [Demographics].modify( N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey";
insert 0.00
as first
into (/IndividualSurvey)[1]' )
FROM inserted
WHERE [Sales].[Individual].[CustomerID] = inserted.[CustomerID]
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;

```

Objects that depend on Sales.Individual

Sales.iduSalesOrderDetail (TRIGGER)

Sales.iStore (TRIGGER)

Sales.iuIndividual (TRIGGER)

Sales.vIndividualCustomer (VIEW)

Sales.vIndividualDemographics (VIEW)

Table definition

```









SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Sales].[Individual](
[CustomerID] [int] NOT NULL,
[ContactID] [int] NOT NULL,
[Demographics] [xml](CONTENT [Sales].[IndividualSurveySchemaCollection]) NULL,

```

```
[ModifiedDate] [datetime] NOT NULL  
) ON [PRIMARY]
```

Sales.SalesOrderDetail

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

I F P	Column name	Data type	Nulls	Default	Description
 	SalesOrderID	int	NO		Primary key. Foreign key to SalesOrderHeader.SalesOrderID.
 	SalesOrderDetailID	int	NO		Primary key. One incremental unique number per product sold.
	CarrierTrackingNumber	nvarchar(25)	YES		Shipment tracking number supplied by the shipper.
	OrderQty	smallint	NO		Quantity ordered per product.
 	ProductID	int	NO		Product sold to customer. Foreign key to Product.ProductID.
	SpecialOfferID	int	NO		Promotional code. Foreign key to SpecialOffer.SpecialOfferID.
	UnitPrice	money	NO		Selling price of a single product.
	UnitPriceDiscount	money	NO	((0.0))	Discount amount.
	LineTotal	numeric(38.6)	NO		Per product subtotal. Computed as UnitPrice * (1 - UnitPriceDiscount) * OrderQty.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_SalesOrderDetail_rowguid	rowguid	ASC	Yes	NONCLUSTERED
IX_SalesOrderDetail_ProductID	ProductID	ASC		NONCLUSTERED
PK_SalesOrderDetail_SalesOrderID_SalesOrderDetailID	SalesOrderID	ASC	Yes	CLUSTERED
PK_SalesOrderDetail_SalesOrderID_SalesOrderDetailID	SalesOrderDetailID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_SalesOrderDetail_SalesOrderHeader_SalesOrderID	SalesOrderID	PK_SalesOrderHeader_SalesOrderID (Sales.SalesOrderHeader)	Foreign key constraint referencing SalesOrderHeader.PurchaseOrderID.
FK_SalesOrderDetail_SpecialOfferProduct_SpecialOfferIDProductID	ProductID	PK_SpecialOfferProduct_SpecialOfferID_ProductID (Sales.SpecialOfferProduct)	Foreign key constraint referencing SpecialOfferProduct.SpecialOfferIDProductID.
FK_SalesOrderDetail_SpecialOfferProduct_SpecialOfferIDProductID	SpecialOfferID	PK_SpecialOfferProduct_SpecialOfferID_ProductID (Sales.SpecialOfferProduct)	Foreign key constraint referencing SpecialOfferProduct.SpecialOfferIDProductID.

Check constraints

Check name	Column name	Expression
CK_SalesOrderDetail_OrderQty	OrderQty	((OrderQty]>(0))
CK_SalesOrderDetail_UnitPrice	UnitPrice	((UnitPrice]>=(0.00))
CK_SalesOrderDetail_UnitPriceDiscount	UnitPriceDiscount	((UnitPriceDiscount]>=(0.00))

Triggers

Trigger name: Sales.iduSalesOrderDetail

Created on: 26 Apr 2006

Trigger type: INSERT

Trigger active: Yes

```
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 [Sales].[Individual]
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]
ON inserted.[SalesOrderID] = [Sales].[SalesOrderHeader].[SalesOrderID]
WHERE [Sales].[SalesOrderHeader].[CustomerID] = [Sales].[Individual].[CustomerID];
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 [Sales].[Individual]
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]
WHERE [Sales].[SalesOrderHeader].[CustomerID] = [Sales].[Individual].[CustomerID];
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;
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 [Sales].[Individual]
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]
ON inserted.[SalesOrderID] = [Sales].[SalesOrderHeader].[SalesOrderID]
WHERE [Sales].[SalesOrderHeader].[CustomerID] = [Sales].[Individual].[CustomerID];
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 [Sales].[Individual]
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]
WHERE [Sales].[SalesOrderHeader].[CustomerID] = [Sales].[Individual].[CustomerID];
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;
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 [Sales].[Individual]
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]
ON inserted.[SalesOrderID] = [Sales].[SalesOrderHeader].[SalesOrderID]
WHERE [Sales].[SalesOrderHeader].[CustomerID] = [Sales].[Individual].[CustomerID];
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 [Sales].[Individual]
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]
WHERE [Sales].[SalesOrderHeader].[CustomerID] = [Sales].[Individual].[CustomerID];
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;

```

Objects that depend on Sales.SalesOrderDetail

[Sales.iduSalesOrderDetail \(TRIGGER\)](#)

[Sales.SalesOrderDetail \(TABLE\)](#)

Table definition

```













SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Sales].[SalesOrderDetail](
[SalesOrderID] [int] NOT NULL,
[SalesOrderDetailID] [int] IDENTITY(1,1) NOT NULL,

```

```
[CarrierTrackingNumber] [nvarchar](25) COLLATE Latin1_General_CS_AS NULL,  
[OrderQty] [smallint] NOT NULL,  
[ProductID] [int] NOT NULL,  
[SpecialOfferID] [int] NOT NULL,  
[UnitPrice] [money] NOT NULL,  
[UnitPriceDiscount] [money] NOT NULL,  
[LineTotal] AS (isnull(([UnitPrice]*((1.0)-[UnitPriceDiscount]))*[OrderQty],(0.0))),  
[rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,  
[ModifiedDate] [datetime] NOT NULL  
) ON [PRIMARY]
```

Sales.SalesOrderHeader

General sales order information.

I F P	Column name	Data type	Nulls	Default	Description
	SalesOrderID	int	NO		Primary key.
	RevisionNumber	tinyint	NO	((0))	Incremental number to track changes to the sales order over time.
	OrderDate	datetime	NO	(getdate())	Dates the sales order was created.
	DueDate	datetime	NO		Date the order is due to the customer.
	ShipDate	datetime	YES		Date the order was shipped to the customer.
	Status	tinyint	NO	((1))	Order current status. 1 = In process; 2 = Approved; 3 = Backordered; 4 = Rejected; 5 = Shipped; 6 = Cancelled
	OnlineOrderFlag	Flag	NO	((1))	0 = Order placed by sales person. 1 = Order placed online by customer.
	SalesOrderNumber	nvarchar(25)	NO		Unique sales order identification number.
	PurchaseOrderNumber	OrderNumber	YES		Customer purchase order number reference.
	AccountNumber	AccountNumber	YES		Financial accounting number reference.
	CustomerID	int	NO		Customer identification number. Foreign key to Customer.CustomerID.
	ContactID	int	NO		Customer contact identification number. Foreign key to Contact.ContactID.
	SalesPersonID	int	YES		Sales person who created the sales order. Foreign key to SalesPerson.SalePersonID.
	TerritoryID	int	YES		Territory in which the sale was made. Foreign key to SalesTerritory.SalesTerritoryID.
	BillToAddressID	int	NO		Customer billing address. Foreign key to Address.AddressID.
	ShipToAddressID	int	NO		Customer shipping address. Foreign key to Address.AddressID.
	ShipMethodID	int	NO		Shipping method. Foreign key to ShipMethod.ShipMethodID.
	CreditCardID	int	YES		Credit card identification number. Foreign key to CreditCard.CreditCardID.
	CreditCardApprovalCode	varchar(15)	YES		Approval code provided by the credit card company.
	CurrencyRateID	int	YES		Currency exchange rate used. Foreign key to CurrencyRate.CurrencyRateID.
	SubTotal	money	NO	((0.00))	Sales subtotal. Computed as SUM (SalesOrderDetail.LineTotal)for the appropriate SalesOrderID.
	TaxAmt	money	NO	((0.00))	Tax amount.
	Freight	money	NO	((0.00))	Shipping cost.
	TotalDue	money	NO		Total due from customer. Computed as Subtotal + TaxAmt + Freight.
	Comment	nvarchar(128)	YES		Sales representative comments.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_SalesOrderHeader_rowguid	rowguid	ASC	Yes	NONCLUSTERED
AK_SalesOrderHeader_SalesOrderNumber	SalesOrderNumber	ASC	Yes	NONCLUSTERED
IX_SalesOrderHeader_CustomerID	CustomerID	ASC		NONCLUSTERED
IX_SalesOrderHeader_SalesPersonID	SalesPersonID	ASC		NONCLUSTERED
PK_SalesOrderHeader_SalesOrderID	SalesOrderID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_SalesOrderHeader_Address_BillToAddressID	BillToAddressID	PK_Address_AddressID (Person.Address)	Foreign key constraint referencing Address.AddressID.
FK_SalesOrderHeader_Contact_ContactID	ContactID	PK_Contact_ContactID (Person.Contact)	Foreign key constraint referencing Contact.ContactID.

FK_SalesOrderHeader_CreditCardID	CreditCardID	PK_CreditCard_CreditCardID (Sales.CreditCard)	Foreign key constraint referencing CreditCard.CreditCardID.
FK_SalesOrderHeader_CurrencyRateID	CurrencyRateID	PK_CurrencyRate_CurrencyRateID (Sales.CurrencyRate)	Foreign key constraint referencing CurrencyRate.CurrencyRateID.
FK_SalesOrderHeader_SalesPersonID	SalesPersonID	PK_SalesPerson_SalesPersonID (Sales.SalesPerson)	Foreign key constraint referencing SalesPerson.SalesPersonID.
FK_SalesOrderHeader_ShipMethodID	ShipMethodID	PK_ShipMethod_ShipMethodID (Purchasing.ShipMethod)	Foreign key constraint referencing ShipMethod.ShipMethodID.
FK_SalesOrderHeader_SalesTerritoryID	TerritoryID	PK_SalesTerritory_TerritoryID (Sales.SalesTerritory)	Foreign key constraint referencing SalesTerritory.TerritoryID.
FK_SalesOrderHeader_AddressID	ShipToAddressID	PK_Address_AddressID (Person.Address)	Foreign key constraint referencing Address.AddressID.
FK_SalesOrderHeader_CustomerID	CustomerID	PK_Customer_CustomerID (Sales.Customer)	Foreign key constraint referencing Customer.CustomerID.

Referencing tables

Table name	Foreign key	Primary key or unique constraint
Sales.SalesOrderDetail	FK_SalesOrderDetail_SalesOrderHeader_SalesOrderID	PK_SalesOrderHeader_SalesOrderID
Sales.SalesOrderHeaderSalesReason	FK_SalesOrderHeaderSalesReason_SalesOrderHeader_SalesOrderID	PK_SalesOrderHeader_SalesOrderID

Check constraints

Check name	Column name	Expression
CK_SalesOrderHeader_Status	Status	([Status]>=(0) AND [Status]<=(8))
CK_SalesOrderHeader_DueDate	OrderDate	([DueDate]>=[OrderDate])
CK_SalesOrderHeader_DueDate	DueDate	([DueDate]>=[OrderDate])
CK_SalesOrderHeader_ShipDate	OrderDate	([ShipDate]>=[OrderDate] OR [ShipDate] IS NULL)
CK_SalesOrderHeader_ShipDate	ShipDate	([ShipDate]>=[OrderDate] OR [ShipDate] IS NULL)
CK_SalesOrderHeader_SubTotal	SubTotal	([SubTotal]>=(0.00))
CK_SalesOrderHeader_TaxAmt	TaxAmt	([TaxAmt]>=(0.00))
CK_SalesOrderHeader_Freight	Freight	([Freight]>=(0.00))

Triggers

Trigger name: Sales.uSalesOrderHeader Created on: 26 Apr 2006
Trigger type: UPDATE Trigger active: Yes

```
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].[SalesPersonID] = [Sales].[SalesOrderHeader].[SalesPersonID]
AND ([Sales].[SalesOrderHeader].[Status] = 5) -- Shipped
AND [Sales].[SalesOrderHeader].[OrderDate] BETWEEN @StartDate AND @EndDate)
WHERE [Sales].[SalesPerson].[SalesPersonID]
  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;

```

Objects that depend on Sales.SalesOrderHeader

Sales.IduSalesOrderDetail (TRIGGER)

Sales.SalesOrderHeader (TABLE)

Sales.uSalesOrderHeader (TRIGGER)

Sales.vSalesPersonSalesByFiscalYears (VIEW)

Table definition

```





SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
SET ARITHABORT ON
CREATE TABLE [Sales].[SalesOrderHeader](
[SalesOrderID] [int] IDENTITY(1,1) NOT FOR REPLICATION NOT NULL,
[RevisionNumber] [tinyint] NOT NULL,
[OrderDate] [datetime] NOT NULL,
[DueDate] [datetime] NOT NULL,
[ShipDate] [datetime] NULL,
[Status] [tinyint] NOT NULL,
[OnlineOrderFlag] [dbo].[Flag] NOT NULL,
[SalesOrderNumber] AS (isnull(N'SO'+CONVERT([nvarchar](23),[SalesOrderID],0), N'*** ERROR ***')),
[PurchaseOrderNumber] [dbo].[OrderNumber] NULL,
[AccountNumber] [dbo].[AccountNumber] NULL,
[CustomerID] [int] NOT NULL,
[ContactID] [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 Latin1_General_CS_AS NULL,
[CurrencyRateID] [int] NULL,
[SubTotal] [money] NOT NULL,
[TaxAmt] [money] NOT NULL,
[Freight] [money] NOT NULL,
[TotalDue] AS (isnull(([SubTotal]+[TaxAmt])+[Freight],(0))),
[Comment] [nvarchar](128) COLLATE Latin1_General_CS_AS NULL,

```

```
[rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,  
[ModifiedDate] [datetime] NOT NULL  
) ON [PRIMARY]
```

Sales.SalesOrderHeaderSalesReason

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

I F P	Column name	Data type	Nulls	Default	Description
 	SalesOrderID	int	NO		Primary key. Foreign key to SalesOrderHeader.SalesOrderID.
 	SalesReasonID	int	NO		Primary key. Foreign key to SalesReason.SalesReasonID.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_SalesOrderHeaderSalesReason_SalesOrderID_SalesReasonID	SalesOrderID	ASC	Yes	CLUSTERED
PK_SalesOrderHeaderSalesReason_SalesReasonID	SalesReasonID	ASC	Yes	CLUSTERED

Foreign keys




Constraint name	Column name	Reference	Description
FK_SalesOrderHeaderSalesReason_SalesOrderHeader_SalesOrderID	SalesOrderID	PK_SalesOrderHeader_SalesOrderID (Sales.SalesOrderHeader)	Foreign key constraint referencing SalesOrderHeader.SalesOrderID.
FK_SalesOrderHeaderSalesReason_SalesReason_SalesReasonID	SalesReasonID	PK_SalesReason_SalesReasonID (Sales.SalesReason)	Foreign key constraint referencing SalesReason.SalesReasonID.

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Sales].[SalesOrderHeaderSalesReason](
[SalesOrderID] [int] NOT NULL,
[SalesReasonID] [int] NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Sales.SalesPerson

Sales representative current information.

I F P	Column name	Data type	Nulls	Default	Description
	SalesPersonID	int	NO		Primary key for SalesPerson records.
	TerritoryID	int	YES		Territory currently assigned to. Foreign key to SalesTerritory.SalesTerritoryID.
	SalesQuota	money	YES		Projected yearly sales.
	Bonus	money	NO	((0.00))	Bonus due if quota is met.
	CommissionPct	smallmoney	NO	((0.00))	Commission percent received per sale.
	SalesYTD	money	NO	((0.00))	Sales total year to date.
	SalesLastYear	money	NO	((0.00))	Sales total of previous year.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_SalesPerson_rowguid	rowguid	ASC	Yes	NONCLUSTERED
PK_SalesPerson_SalesPersonID	SalesPersonID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_SalesPerson_Employee_SalesPersonID	SalesPersonID	PK_Employee_EmployeeID (HumanResources.Employee)	Foreign key constraint referencing Employee.EmployeeID.
FK_SalesPerson_SalesTerritory_TerritoryID	TerritoryID	PK_SalesTerritory_TerritoryID (Sales.SalesTerritory)	Foreign key constraint referencing SalesTerritory.TerritoryID.

Referencing tables

Table name	Foreign key	Primary key or unique constraint
Sales.SalesOrderHeader	FK_SalesOrderHeader_SalesPerson_SalesPersonID	PK_SalesPerson_SalesPersonID
Sales.SalesTerritoryHistory	FK_SalesTerritoryHistory_SalesPerson_SalesPersonID	PK_SalesPerson_SalesPersonID
Sales.Store	FK_Store_SalesPerson_SalesPersonID	PK_SalesPerson_SalesPersonID
Sales.SalesPersonQuotaHistory	FK_SalesPersonQuotaHistory_SalesPerson_SalesPersonID	PK_SalesPerson_SalesPersonID

Check constraints

Check name	Column name	Expression
CK_SalesPerson_SalesQuota	SalesQuota	((SalesQuota)>(0.00))
CK_SalesPerson_Bonus	Bonus	((Bonus]>=(0.00))
CK_SalesPerson_CommissionPct	CommissionPct	((CommissionPct]>=(0.00))
CK_SalesPerson_SalesYTD	SalesYTD	((SalesYTD]>=(0.00))
CK_SalesPerson_SalesLastYear	SalesLastYear	((SalesLastYear]>=(0.00))

Objects that depend on Sales.SalesPerson

[Sales.uSalesOrderHeader \(TRIGGER\)](#)

[Sales.vSalesPerson \(VIEW\)](#)

[Sales.vSalesPersonSalesByFiscalYears \(VIEW\)](#)

Table definition







```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Sales].[SalesPerson](
  [SalesPersonID] [int] NOT NULL,
  [TerritoryID] [int] NULL,
  [SalesQuota] [money] NULL,
```



```
[Bonus] [money] NOT NULL,  
[CommissionPct] [smallmoney] NOT NULL,  
[SalesYTD] [money] NOT NULL,  
[SalesLastYear] [money] NOT NULL,  
[rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,  
[ModifiedDate] [datetime] NOT NULL  
) ON [PRIMARY]
```

Sales.SalesPersonQuotaHistory

Sales performance tracking.

I F P	Column name	Data type	Nulls	Default	Description
 	SalesPersonID	int	NO		Sales person identification number. Foreign key to SalesPerson.SalesPersonID.
 	QuotaDate	datetime	NO		Sales quota date.
	SalesQuota	money	NO		Sales quota amount.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_SalesPersonQuotaHistory_rowguid	rowguid	ASC	Yes	NONCLUSTERED
PK_SalesPersonQuotaHistory_SalesPersonID_QuotaDate	SalesPersonID	ASC	Yes	CLUSTERED
PK_SalesPersonQuotaHistory_SalesPersonID_QuotaDate	QuotaDate	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_SalesPersonQuotaHistory_SalesPersonID	SalesPersonID	PK_SalesPerson_SalesPersonID (Sales.SalesPerson)	Foreign key constraint referencing SalesPerson.SalesPersonID.

Check constraints


Check name	Column name	Expression
CK_SalesPersonQuotaHistory_SalesQuota	SalesQuota	((SalesQuota)>(0.00))

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Sales].[SalesPersonQuotaHistory](
  [SalesPersonID] [int] NOT NULL,
  [QuotaDate] [datetime] NOT NULL,
  [SalesQuota] [money] NOT NULL,
  [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
  [ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Sales.SalesReason

Lookup table of customer purchase reasons.

I F P	Column name	Data type	Nulls	Default	Description
	SalesReasonID	int	NO		Primary key for SalesReason records.
	Name	Name	NO		Sales reason description.
	ReasonType	Name	NO		Category the sales reason belongs to.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PK_SalesReason_SalesReasonID	SalesReasonID	ASC	Yes	CLUSTERED

Referencing tables





Table name	Foreign key	Primary key or unique constraint
Sales.SalesOrderHeaderSalesReason	FK_SalesOrderHeaderSalesReason_SalesReason_SalesReasonID	PK_SalesReason_SalesReasonID

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Sales].[SalesReason](
  [SalesReasonID] [int] IDENTITY(1,1) NOT NULL,
  [Name] [dbo].[Name] NOT NULL,
  [ReasonType] [dbo].[Name] NOT NULL,
  [ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Sales.SalesTaxRate

Tax rate lookup table.

I F P	Column name	Data type	Nulls	Default	Description
	SalesTaxRateID	int	NO		Primary key for SalesTaxRate records.
	StateProvinceID	int	NO		State, province, or country/region the sales tax applies to.
	TaxType	tinyint	NO		1 = Tax applied to retail transactions, 2 = Tax applied to wholesale transactions, 3 = Tax applied to all sales (retail and wholesale) transactions.
	TaxRate	smallmoney	NO	((0.00))	Tax rate amount.
	Name	Name	NO		Tax rate description.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_SalesTaxRate_rowguid	rowguid	ASC	Yes	NONCLUSTERED
AK_SalesTaxRate_StateProvinceID_TaxType	StateProvinceID	ASC	Yes	NONCLUSTERED
AK_SalesTaxRate_StateProvinceID_TaxType	TaxType	ASC	Yes	NONCLUSTERED
PK_SalesTaxRate_SalesTaxRateID	SalesTaxRateID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_SalesTaxRate_StateProvinceID_StateProvinceID	StateProvinceID	PK_StateProvince_StateProvinceID (Person.StateProvince)	Foreign key constraint referencing StateProvince.StateProvinceID.

Check constraints




Check name	Column name	Expression
CK_SalesTaxRate_TaxType	TaxType	((TaxType)>=(1) AND [TaxType]<=(3))

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Sales].[SalesTaxRate](
  [SalesTaxRateID] [int] IDENTITY(1,1) NOT NULL,
  [StateProvinceID] [int] NOT NULL,
  [TaxType] [tinyint] NOT NULL,
  [TaxRate] [smallmoney] NOT NULL,
  [Name] [dbo].[Name] NOT NULL,
  [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
  [ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Sales.SalesTerritory

Sales territory lookup table.

I F P	Column name	Data type	Nulls	Default	Description
	TerritoryID	int	NO		Primary key for SalesTerritory records.
	Name	Name	NO		Sales territory description
	CountryRegionCode	nvarchar(3)	NO		ISO standard country or region code. Foreign key to CountryRegion.CountryRegionCode.
	Group	nvarchar(50)	NO		Geographic area to which the sales territory belong.
	SalesYTD	money	NO	((0.00))	Sales in the territory year to date.
	SalesLastYear	money	NO	((0.00))	Sales in the territory the previous year.
	CostYTD	money	NO	((0.00))	Business costs in the territory year to date.
	CostLastYear	money	NO	((0.00))	Business costs in the territory the previous year.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_SalesTerritory_Name	Name	ASC	Yes	NONCLUSTERED
AK_SalesTerritory_rowguid	rowguid	ASC	Yes	NONCLUSTERED
PK_SalesTerritory_TerritoryID	TerritoryID	ASC	Yes	CLUSTERED

Referencing tables

Table name	Foreign key	Primary key or unique constraint
Sales.SalesOrderHeader	FK_SalesOrderHeader_SalesTerritory_TerritoryID	PK_SalesTerritory_TerritoryID
Sales.SalesPerson	FK_SalesPerson_SalesTerritory_TerritoryID	PK_SalesTerritory_TerritoryID
Sales.SalesTerritoryHistory	FK_SalesTerritoryHistory_SalesTerritory_TerritoryID	PK_SalesTerritory_TerritoryID
Sales.Customer	FK_Customer_SalesTerritory_TerritoryID	PK_SalesTerritory_TerritoryID
Person.StateProvince	FK_StateProvince_SalesTerritory_TerritoryID	PK_SalesTerritory_TerritoryID

Check constraints

Check name	Column name	Expression
CK_SalesTerritory_SalesYTD	SalesYTD	([SalesYTD]>=(0.00))
CK_SalesTerritory_SalesLastYear	SalesLastYear	([SalesLastYear]>=(0.00))
CK_SalesTerritory_CostYTD	CostYTD	([CostYTD]>=(0.00))
CK_SalesTerritory_CostLastYear	CostLastYear	([CostLastYear]>=(0.00))

Objects that depend on Sales.SalesTerritory

[Sales.uSalesOrderHeader \(TRIGGER\)](#)

[Sales.vSalesPerson \(VIEW\)](#)

[Sales.vSalesPersonSalesByFiscalYears \(VIEW\)](#)





Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Sales].[SalesTerritory](
[TerritoryID] [int] IDENTITY(1,1) NOT NULL,
[Name] [dbo].[Name] NOT NULL,
[CountryRegionCode] [nvarchar](3) COLLATE Latin1_General_CS_AS NOT NULL,
[Group] [nvarchar](50) COLLATE Latin1_General_CS_AS NOT NULL,
[SalesYTD] [money] NOT NULL,
[SalesLastYear] [money] NOT NULL,
[CostYTD] [money] NOT NULL,
[CostLastYear] [money] NOT NULL,
```

```
[rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,  
[ModifiedDate] [datetime] NOT NULL  
) ON [PRIMARY]
```

Sales.SalesTerritoryHistory

Sales representative transfers to other sales territories.

I F P	Column name	Data type	Nulls	Default	Description
	SalesPersonID	int	NO		Primary key for SalesTerritoryHistory records.
	TerritoryID	int	NO		Territory identification number. Foreign key to SalesTerritory.SalesTerritoryID.
	StartDate	datetime	NO		Date the sales representative started work in the territory.
	EndDate	datetime	YES		Date the sales representative left work in the territory.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_SalesTerritoryHistory_rowguid	rowguid	ASC	Yes	NONCLUSTERED
PK_SalesTerritoryHistory_SalesPersonID_StartDate_TerritoryID	SalesPersonID	ASC	Yes	CLUSTERED
PK_SalesTerritoryHistory_SalesPersonID_StartDate_TerritoryID	TerritoryID	ASC	Yes	CLUSTERED
PK_SalesTerritoryHistory_SalesPersonID_StartDate_TerritoryID	StartDate	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_SalesTerritoryHistory_SalesPerson_SalesPersonID	SalesPersonID	PK_SalesPerson_SalesPersonID (Sales.SalesPerson)	Foreign key constraint referencing SalesPerson.SalesPersonID.
FK_SalesTerritoryHistory_SalesTerritory_TerritoryID	TerritoryID	PK_SalesTerritory_TerritoryID (Sales.SalesTerritory)	Foreign key constraint referencing SalesTerritory.TerritoryID.

Check constraints




Check name	Column name	Expression
CK_SalesTerritoryHistory_EndDate	StartDate	([EndDate]>=[StartDate] OR [EndDate] IS NULL)
CK_SalesTerritoryHistory_EndDate	EndDate	([EndDate]>=[StartDate] OR [EndDate] IS NULL)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Sales].[SalesTerritoryHistory](
  [SalesPersonID] [int] NOT NULL,
  [TerritoryID] [int] NOT NULL,
  [StartDate] [datetime] NOT NULL,
  [EndDate] [datetime] NULL,
  [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
  [ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Sales.ShoppingCartItem

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

I F P	Column name	Data type	Nulls	Default	Description
	ShoppingCartItemID	int	NO		Primary key for ShoppingCartItem records.
	ShoppingCartID	nvarchar(50)	NO		Shopping cart identification number.
	Quantity	int	NO	((1))	Product quantity ordered.
	ProductID	int	NO		Product ordered. Foreign key to Product.ProductID.
	DateCreated	datetime	NO	(getdate())	Date the time the record was created.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
IX_ShoppingCartItem_ShoppingCartID_ProductID	ShoppingCartID	ASC		NONCLUSTERED
IX_ShoppingCartItem_ShoppingCartID_ProductID	ProductID	ASC		NONCLUSTERED
PK_ShoppingCartItem_ShoppingCartItemID	ShoppingCartItemID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_ShoppingCartItem_ProductID	ProductID	PK_Product_ProductID (Production.Product)	Foreign key constraint referencing Product.ProductID.

Check constraints



Check name	Column name	Expression
CK_ShoppingCartItem_Quantity	Quantity	((Quantity)>=(1))

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Sales].[ShoppingCartItem](
[ShoppingCartItemID] [int] IDENTITY(1,1) NOT NULL,
[ShoppingCartID] [nvarchar](50) COLLATE Latin1_General_CS_AS NOT NULL,
[Quantity] [int] NOT NULL,
[ProductID] [int] NOT NULL,
[DateCreated] [datetime] NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```


Sales.SpecialOffer

Sale discounts lookup table.

I F P	Column name	Data type	Nulls	Default	Description
	SpecialOfferID	int	NO		Primary key for SpecialOffer records.
	Description	nvarchar(255)	NO		Discount description.
	DiscountPct	smallmoney	NO	((0.00))	Discount percentage.
	Type	nvarchar(50)	NO		Discount type category.
	Category	nvarchar(50)	NO		Group the discount applies to such as Reseller or Customer.
	StartDate	datetime	NO		Discount start date.
	EndDate	datetime	NO		Discount end date.
	MinQty	int	NO	((0))	Minimum discount percent allowed.
	MaxQty	int	YES		Maximum discount percent allowed.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_SpecialOffer_rowguid	rowguid	ASC	Yes	NONCLUSTERED
PK_SpecialOffer_SpecialOfferID	SpecialOfferID	ASC	Yes	CLUSTERED

Referencing tables

Table name	Foreign key	Primary key or unique constraint
Sales.SpecialOfferProduct	FK_SpecialOfferProduct_SpecialOffer_SpecialOfferID	PK_SpecialOffer_SpecialOfferID

Check constraints




Check name	Column name	Expression
CK_SpecialOffer_EndDate	StartDate	([EndDate]>=[StartDate])
CK_SpecialOffer_EndDate	EndDate	([EndDate]>=[StartDate])
CK_SpecialOffer_DiscountPct	DiscountPct	([DiscountPct]>=(0.00))
CK_SpecialOffer_MinQty	MinQty	([MinQty]>=(0))
CK_SpecialOffer_MaxQty	MaxQty	([MaxQty]>=(0))

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Sales].[SpecialOffer](
[SpecialOfferID] [int] IDENTITY(1,1) NOT NULL,
[Description] [nvarchar](255) COLLATE Latin1_General_CS_AS NOT NULL,
[DiscountPct] [smallmoney] NOT NULL,
[Type] [nvarchar](50) COLLATE Latin1_General_CS_AS NOT NULL,
[Category] [nvarchar](50) COLLATE Latin1_General_CS_AS NOT NULL,
[StartDate] [datetime] NOT NULL,
[EndDate] [datetime] NOT NULL,
[MinQty] [int] NOT NULL,
[MaxQty] [int] NULL,
[rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Sales.SpecialOfferProduct

Cross-reference table mapping products to special offer discounts.

I F P	Column name	Data type	Nulls	Default	Description
	SpecialOfferID	int	NO		Primary key for SpecialOfferProduct records.
	ProductID	int	NO		Product identification number. Foreign key to Product.ProductID.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_SpecialOfferProduct_rowguid	rowguid	ASC	Yes	NONCLUSTERED
IX_SpecialOfferProduct_ProductID	ProductID	ASC		NONCLUSTERED
PK_SpecialOfferProduct_SpecialOfferID_ProductID	SpecialOfferID	ASC	Yes	CLUSTERED
PK_SpecialOfferProduct_SpecialOfferID_ProductID	ProductID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_SpecialOfferProduct_ProductID	ProductID	PK_Product_ProductID (Production.Product)	Foreign key constraint referencing Product.ProductID.
FK_SpecialOfferProduct_SpecialOfferID	SpecialOfferID	PK_SpecialOffer_SpecialOfferID (Sales.SpecialOffer)	Foreign key constraint referencing SpecialOffer.SpecialOfferID.

Referencing tables





Table name	Foreign key	Primary key or unique constraint
Sales.SalesOrderDetail	FK_SalesOrderDetail_SpecialOfferProduct_SpecialOfferIDProductID	PK_SpecialOfferProduct_SpecialOfferID_ProductID

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Sales].[SpecialOfferProduct](
  [SpecialOfferID] [int] NOT NULL,
  [ProductID] [int] NOT NULL,
  [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
  [ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Sales.Store

Customers (resellers) of Adventure Works products.

I F P	Column name	Data type	Nulls	Default	Description
	CustomerID	int	NO		Primary key. Foreign key to Customer.CustomerID.
	Name	Name	NO		Name of the store.
	SalesPersonID	int	YES		ID of the sales person assigned to the customer. Foreign key to SalesPerson.SalesPersonID.
	Demographics	xml	YES		Demographic information about the store such as the number of employees, annual sales and store type.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_Store_rowguid	rowguid	ASC	Yes	NONCLUSTERED
IX_Store_SalesPersonID	SalesPersonID	ASC		NONCLUSTERED
PK_Store_CustomerID	CustomerID	ASC	Yes	CLUSTERED
PXML_Store_Demographics	Demographics	ASC		XML

Foreign keys

Constraint name	Column name	Reference	Description
FK_Store_Customer_CustomerID	CustomerID	PK_Customer_CustomerID (Sales.Customer)	Foreign key constraint referencing Customer.CustomerID.
FK_Store_SalesPerson_SalesPersonID	SalesPersonID	PK_SalesPerson_SalesPersonID (Sales.SalesPerson)	Foreign key constraint referencing SalesPerson.SalesPersonID

Referencing tables

Table name	Foreign key	Primary key or unique constraint
Sales.StoreContact	FK_StoreContact_Store_CustomerID	PK_Store_CustomerID

Triggers

Trigger name: Sales.iStore

Created on: 26 Apr 2006

Trigger type: INSERT

Trigger active: Yes

```
CREATE TRIGGER [Sales].[iStore] ON [Sales].[Store]
AFTER INSERT AS
BEGIN
DECLARE @Count int;
SET @Count = @@ROWCOUNT;
IF @Count = 0
RETURN;
SET NOCOUNT ON;
BEGIN TRY
-- Only allow the Customer to be a Store OR Individual
IF EXISTS (SELECT * FROM inserted INNER JOIN [Sales].[Individual]
ON inserted.[CustomerID] = [Sales].[Individual].[CustomerID])
BEGIN
-- Rollback any active or uncommittable transactions
IF @@TRANSCOUNT > 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;
```

Objects that depend on Sales.Store

[Sales.iuIndividual \(TRIGGER\)](#)





[Sales.vStoreWithDemographics \(VIEW\)](#)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Sales].[Store](
[CustomerID] [int] NOT NULL,
[Name] [dbo].[Name] NOT NULL,
[SalesPersonID] [int] NULL,
[Demographics] [xml](CONTENT [Sales].[StoreSurveySchemaCollection]) NULL,
[rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

Sales.StoreContact

Cross-reference table mapping stores and their employees.

I F P	Column name	Data type	Nulls	Default	Description
	CustomerID	int	NO		Store identification number. Foreign key to Customer.CustomerID.
	ContactID	int	NO		Contact (store employee) identification number. Foreign key to Contact.ContactID.
	ContactTypeID	int	NO		Contact type such as owner or purchasing agent. Foreign key to ContactType.ContactTypeID.
	rowguid	uniqueidentifier	NO	(newid())	ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.
	ModifiedDate	datetime	NO	(getdate())	Date and time the record was last updated.

Indexes

Index name	Column name	Sort direction	Is unique	Index type
AK_StoreContact_rowguid	rowguid	ASC	Yes	NONCLUSTERED
IX_StoreContact_ContactID	ContactID	ASC		NONCLUSTERED
IX_StoreContact_ContactTypeID	ContactTypeID	ASC		NONCLUSTERED
PK_StoreContact_CustomerID_ContactID	CustomerID	ASC	Yes	CLUSTERED
PK_StoreContact_CustomerID_ContactID	ContactID	ASC	Yes	CLUSTERED

Foreign keys

Constraint name	Column name	Reference	Description
FK_StoreContact_Contact_ContactID	ContactID	PK_Contact_ContactID (Person.Contact)	Foreign key constraint referencing Contact.ContactID.
FK_StoreContact_Store_CustomerID	CustomerID	PK_Store_CustomerID (Sales.Store)	Foreign key constraint referencing Store.CustomerID.
FK_StoreContact_ContactType_ContactTypeID	ContactTypeID	PK_ContactType_ContactTypeID (Person.ContactType)	Foreign key constraint referencing ContactType.ContactTypeID.

Objects that depend on Sales.StoreContact

[Sales.vStoreWithDemographics \(VIEW\)](#)

Table definition

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE TABLE [Sales].[StoreContact](
[CustomerID] [int] NOT NULL,
[ContactID] [int] NOT NULL,
[ContactTypeID] [int] NOT NULL,
[rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
[ModifiedDate] [datetime] NOT NULL
) ON [PRIMARY]
```

VIEWS

HumanResources.vEmployee

Created: 26 Apr 2006 Last updated: 26 Apr 2006

Employee names and addresses.

Resultset

Source	Column name	Data type	Nulls	Description
HumanResources.Employee	EmployeeID	int	NO	
Person.Contact	Title	nvarchar(8)	YES	
Person.Contact	FirstName	Name	NO	
Person.Contact	MiddleName	Name	YES	
Person.Contact	LastName	Name	NO	
Person.Contact	Suffix	nvarchar(10)	YES	
	JobTitle	nvarchar(50)	NO	
Person.Contact	Phone	Phone	YES	
Person.Contact	EmailAddress	nvarchar(50)	YES	
Person.Contact	EmailPromotion	int	NO	
Person.Address	AddressLine1	nvarchar(60)	NO	
Person.Address	AddressLine2	nvarchar(60)	YES	
Person.Address	City	nvarchar(30)	NO	
	StateProvinceName	Name	NO	
Person.Address	PostalCode	nvarchar(15)	NO	
	CountryRegionName	Name	NO	
Person.Contact	AdditionalContactInfo	xml	YES	

```
CREATE VIEW [HumanResources].[vEmployee]
AS
SELECT
    e.[EmployeeID]
    ,c.[Title]
    ,c.[FirstName]
    ,c.[MiddleName]
    ,c.[LastName]
    ,c.[Suffix]
    ,e.[Title] AS [JobTitle]
    ,c.[Phone]
    ,c.[EmailAddress]
    ,c.[EmailPromotion]
    ,a.[AddressLine1]
    ,a.[AddressLine2]
    ,a.[City]
    ,sp.[Name] AS [StateProvinceName]
    ,a.[PostalCode]
    ,cr.[Name] AS [CountryRegionName]
    ,c.[AdditionalContactInfo]
FROM [HumanResources].[Employee] e
INNER JOIN [Person].[Contact] c
    ON c.[ContactID] = e.[ContactID]
INNER JOIN [HumanResources].[EmployeeAddress] ea
    ON e.[EmployeeID] = ea.[EmployeeID]
INNER JOIN [Person].[Address] a
    ON ea.[AddressID] = a.[AddressID]
INNER JOIN [Person].[StateProvince] sp
    ON sp.[StateProvinceID] = a.[StateProvinceID]
INNER JOIN [Person].[CountryRegion] cr
    ON cr.[CountryRegionCode] = sp.[CountryRegionCode];
```

Related objects

[Person.Address\(TABLE\)](#)

[Person.StateProvince\(TABLE\)](#)

[HumanResources.Employee\(TABLE\)](#)

HumanResources.EmployeeAddress(TABLE)
Person.Contact(TABLE)
Person.CountryRegion(TABLE)

HumanResources.vEmployeeDepartment

Created: 26 Apr 2006 Last updated: 26 Apr 2006

Returns employee name, title, and current department.

Resultset

Source	Column name	Data type	Nulls	Description
HumanResources.Employee	EmployeeID	int	NO	
Person.Contact	Title	nvarchar(8)	YES	
Person.Contact	FirstName	Name	NO	
Person.Contact	MiddleName	Name	YES	
Person.Contact	LastName	Name	NO	
Person.Contact	Suffix	nvarchar(10)	YES	
	JobTitle	nvarchar(50)	NO	
	Department	Name	NO	
HumanResources.Department	GroupName	Name	NO	
HumanResources.EmployeeDepartmentHistory	StartDate	datetime	NO	

```
CREATE VIEW [HumanResources].[vEmployeeDepartment]
AS
SELECT
e.[EmployeeID]
,c.[Title]
,c.[FirstName]
,c.[MiddleName]
,c.[LastName]
,c.[Suffix]
,e.[Title] AS [JobTitle]
,d.[Name] AS [Department]
,d.[GroupName]
,edh.[StartDate]
FROM [HumanResources].[Employee] e
INNER JOIN [Person].[Contact] c
ON c.[ContactID] = e.[ContactID]
INNER JOIN [HumanResources].[EmployeeDepartmentHistory] edh
ON e.[EmployeeID] = edh.[EmployeeID]
INNER JOIN [HumanResources].[Department] d
ON edh.[DepartmentID] = d.[DepartmentID]
WHERE GETDATE() BETWEEN edh.[StartDate] AND ISNULL(edh.[EndDate], GETDATE());
```

Related objects

[Person.Contact\(TABLE\)](#)

[HumanResources.EmployeeDepartmentHistory\(TABLE\)](#)

[HumanResources.Department\(TABLE\)](#)

[HumanResources.Employee\(TABLE\)](#)

HumanResources.vEmployeeDepartmentHistory

Created: 26 Apr 2006 Last updated: 26 Apr 2006

Returns employee name and current and previous departments.

Resultset

Source	Column name	Data type	Nulls	Description
HumanResources.Employee	EmployeeID	int	NO	
Person.Contact	Title	nvarchar(8)	YES	
Person.Contact	FirstName	Name	NO	
Person.Contact	MiddleName	Name	YES	
Person.Contact	LastName	Name	NO	
Person.Contact	Suffix	nvarchar(10)	YES	
	Shift	Name	NO	
	Department	Name	NO	
HumanResources.Department	GroupName	Name	NO	
HumanResources.EmployeeDepartmentHistory	StartDate	datetime	NO	
HumanResources.EmployeeDepartmentHistory	EndDate	datetime	YES	

```
CREATE VIEW [HumanResources].[vEmployeeDepartmentHistory]
AS
SELECT
e.[EmployeeID]
,c.[Title]
,c.[FirstName]
,c.[MiddleName]
,c.[LastName]
,c.[Suffix]
,s.[Name] AS [Shift]
,d.[Name] AS [Department]
,d.[GroupName]
,edh.[StartDate]
,edh.[EndDate]
FROM [HumanResources].[Employee] e
INNER JOIN [Person].[Contact] c
ON c.[ContactID] = e.[ContactID]
INNER JOIN [HumanResources].[EmployeeDepartmentHistory] edh
ON e.[EmployeeID] = edh.[EmployeeID]
INNER JOIN [HumanResources].[Department] d
ON edh.[DepartmentID] = d.[DepartmentID]
INNER JOIN [HumanResources].[Shift] s
ON s.[ShiftID] = edh.[ShiftID];
```

Related objects

[Person.Contact\(TABLE\)](#)

[HumanResources.EmployeeDepartmentHistory\(TABLE\)](#)

[HumanResources.Shift\(TABLE\)](#)

[HumanResources.Department\(TABLE\)](#)

[HumanResources.Employee\(TABLE\)](#)

HumanResources.vJobCandidate

Created: 26 Apr 2006 Last updated: 26 Apr 2006

Job candidate names and resumes.

Resultset

Source	Column name	Data type	Nulls	Description
HumanResources.JobCandidate	JobCandidateID	int	NO	
HumanResources.JobCandidate	EmployeeID	int	YES	
	Name.Prefix	nvarchar(30)	YES	
	Name.First	nvarchar(30)	YES	
	Name.Middle	nvarchar(30)	YES	
	Name.Last	nvarchar(30)	YES	
	Name.Suffix	nvarchar(30)	YES	
	Skills	nvarchar	YES	
	Addr.Type	nvarchar(30)	YES	
	Addr.Loc.CountryRegion	nvarchar(100)	YES	
	Addr.Loc.State	nvarchar(100)	YES	
	Addr.Loc.City	nvarchar(100)	YES	
	Addr.PostalCode	nvarchar(20)	YES	
	EMail	nvarchar	YES	
	WebSite	nvarchar	YES	
HumanResources.JobCandidate	ModifiedDate	datetime	NO	

```
CREATE VIEW [HumanResources].[vJobCandidate]
AS
SELECT
jc.[JobCandidateID]
,jc.[EmployeeID]
,[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);
```

Related objects

[HumanResources.JobCandidate\(TABLE\)](#)

HumanResources.vJobCandidateEducation

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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.

Resultset

Source	Column name	Data type	Nulls	Description
HumanResources.JobCandidate	JobCandidateID	int	NO	
	Edu.Level	nvarchar	YES	
	Edu.StartDate	datetime	YES	
	Edu.EndDate	datetime	YES	
	Edu.Degree	nvarchar(50)	YES	
	Edu.Major	nvarchar(50)	YES	
	Edu.Minor	nvarchar(50)	YES	
	Edu.GPA	nvarchar(5)	YES	
	Edu.GPAScale	nvarchar(5)	YES	
	Edu.School	nvarchar(100)	YES	
	Edu.Loc.CountryRegion	nvarchar(100)	YES	
	Edu.Loc.State	nvarchar(100)	YES	
	Edu.Loc.City	nvarchar(100)	YES	

```
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);
```

Related objects

[HumanResources.JobCandidate\(TABLE\)](#)

HumanResources.vJobCandidateEmployment

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Resultset

Source	Column name	Data type	Nulls	Description
HumanResources.JobCandidate	JobCandidateID	int	NO	
	Emp.StartDate	datetime	YES	
	Emp.EndDate	datetime	YES	
	Emp.OrgName	nvarchar(100)	YES	
	Emp.JobTitle	nvarchar(100)	YES	
	Emp.Responsibility	nvarchar	YES	
	Emp.FunctionCategory	nvarchar	YES	
	Emp.IndustryCategory	nvarchar	YES	
	Emp.Loc.CountryRegion	nvarchar	YES	
	Emp.Loc.State	nvarchar	YES	
	Emp.Loc.City	nvarchar	YES	

```
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);
```

Related objects

[HumanResources.JobCandidate\(TABLE\)](#)



Person.vAdditionalContactInfo

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Resultset

Source	Column name	Data type	Nulls	Description
Person.Contact	ContactID	int	NO	
Person.Contact	FirstName	Name	NO	
Person.Contact	MiddleName	Name	YES	
Person.Contact	LastName	Name	NO	
	TelephoneNumber	nvarchar(50)	YES	
	TelephoneSpecialInstructions	nvarchar	YES	
	Street	nvarchar(50)	YES	
	City	nvarchar(50)	YES	
	StateProvince	nvarchar(50)	YES	
	PostalCode	nvarchar(50)	YES	
	CountryRegion	nvarchar(50)	YES	
	HomeAddressSpecialInstructions	nvarchar	YES	
	EEmailAddress	nvarchar(128)	YES	
	EEmailSpecialInstructions	nvarchar	YES	
	EEmailTelephoneNumber	nvarchar(50)	YES	
Person.Contact	rowguid	uniqueidentifier	NO	
Person.Contact	ModifiedDate	datetime	NO	

```

CREATE VIEW [Person].[vAdditionalContactInfo]
AS
SELECT
[ContactID]
,[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].[Contact]
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;

```

Related objects

[Person.Contact\(TABLE\)](#)

Person.vStateProvinceCountryRegion

Created: 26 Apr 2006 Last updated: 26 Apr 2006

Joins StateProvince table with CountryRegion table.

Resultset

Source	Column name	Data type	Nulls	Description
Person.StateProvince	StateProvinceID	int	NO	Clustered index on the view vStateProvinceCountryRegion.
Person.StateProvince	StateProvinceCode	nchar(3)	NO	
Person.StateProvince	IsOnlyStateProvinceFlag	Flag	NO	
	StateProvinceName	Name	NO	
Person.StateProvince	TerritoryID	int	NO	
Person.CountryRegion	CountryRegionCode	nvarchar(3)	NO	
	CountryRegionName	Name	NO	

```
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];
```

Related objects

[Person.CountryRegion\(TABLE\)](#)

[Person.StateProvince\(TABLE\)](#)



Production.vProductAndDescription

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Resultset

Source	Column name	Data type	Nulls	Description
Production.Product	ProductID	int	NO	Clustered index on the view vProductAndDescription.
Production.Product	Name	Name	NO	
	ProductModel	Name	NO	
Production.ProductModelProductDescriptionCulture	CultureID	nchar(6)	NO	
Production.ProductDescription	Description	nvarchar(400)	NO	

```
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];
```

Related objects

[Production.Product\(TABLE\)](#)

[Production.ProductDescription\(TABLE\)](#)

[Production.ProductModelProductDescriptionCulture\(TABLE\)](#)

[Production.ProductModel\(TABLE\)](#)

Production.vProductModelCatalogDescription

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Resultset

Source	Column name	Data type	Nulls	Description
Production.ProductModel	ProductModelID	int	NO	
Production.ProductModel	Name	Name	NO	
	Summary	nvarchar	YES	
	Manufacturer	nvarchar	YES	
	Copyright	nvarchar(30)	YES	
	ProductURL	nvarchar(256)	YES	
	WarrantyPeriod	nvarchar(256)	YES	
	WarrantyDescription	nvarchar(256)	YES	
	NoOfYears	nvarchar(256)	YES	
	MaintenanceDescription	nvarchar(256)	YES	
	Wheel	nvarchar(256)	YES	
	Saddle	nvarchar(256)	YES	
	Pedal	nvarchar(256)	YES	
	BikeFrame	nvarchar	YES	
	Crankset	nvarchar(256)	YES	
	PictureAngle	nvarchar(256)	YES	
	PictureSize	nvarchar(256)	YES	
	ProductPhotoID	nvarchar(256)	YES	
	Material	nvarchar(256)	YES	
	Color	nvarchar(256)	YES	
	ProductLine	nvarchar(256)	YES	
	Style	nvarchar(256)	YES	
	RiderExperience	nvarchar(1024)	YES	
Production.ProductModel	rowguid	uniqueidentifier	NO	
Production.ProductModel	ModifiedDate	datetime	NO	

```
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
pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";
declare namespace
wm="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelWarrAndMain";
(/pl:ProductDescription/pl:Features/wm:Maintenance/wm:NoOfYears)[1]' , 'nvarchar(256)' ) AS [NoOfYears]
,[CatalogDescription].value( N'declare namespace
pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";
declare namespace
wm="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelWarrAndMain";
(/pl:ProductDescription/pl:Features/wm:Maintenance/wm:Description)[1]' , 'nvarchar(256)' ) AS
[MaintenanceDescription]
,[CatalogDescription].value( N'declare namespace
pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";
declare namespace wf="http://www.adventure-works.com/schemas/OtherFeatures";
(/pl:ProductDescription/pl:Features/wf:wheel)[1]' , 'nvarchar(256)' ) AS [Wheel]
,[CatalogDescription].value( N'declare namespace
pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";
declare namespace wf="http://www.adventure-works.com/schemas/OtherFeatures";
(/pl:ProductDescription/pl:Features/wf:saddle)[1]' , 'nvarchar(256)' ) AS [Saddle]
,[CatalogDescription].value( N'declare namespace
pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";
declare namespace wf="http://www.adventure-works.com/schemas/OtherFeatures";
(/pl:ProductDescription/pl:Features/wf:pedal)[1]' , 'nvarchar(256)' ) AS [Pedal]
,[CatalogDescription].value( N'declare namespace
pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";
declare namespace wf="http://www.adventure-works.com/schemas/OtherFeatures";
(/pl:ProductDescription/pl:Features/wf:BikeFrame)[1]' , 'nvarchar(max)' ) AS [BikeFrame]
,[CatalogDescription].value( N'declare namespace
pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";
declare namespace wf="http://www.adventure-works.com/schemas/OtherFeatures";
(/pl:ProductDescription/pl:Features/wf:crankset)[1]' , 'nvarchar(256)' ) AS [Crankset]
,[CatalogDescription].value( N'declare namespace
pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";
(/pl:ProductDescription/pl:Picture/pl:Angle)[1]' , 'nvarchar(256)' ) AS [PictureAngle]
,[CatalogDescription].value( N'declare namespace
pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";
(/pl:ProductDescription/pl:Picture/pl:Size)[1]' , 'nvarchar(256)' ) AS [PictureSize]
,[CatalogDescription].value( N'declare namespace
pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";
(/pl:ProductDescription/pl:Picture/pl:ProductPhotoID)[1]' , 'nvarchar(256)' ) AS [ProductPhotoID]
,[CatalogDescription].value( N'declare namespace
pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";
(/pl:ProductDescription/pl:Specifications/Material)[1]' , 'nvarchar(256)' ) AS [Material]
,[CatalogDescription].value( N'declare namespace
pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";
(/pl:ProductDescription/pl:Specifications/Color)[1]' , 'nvarchar(256)' ) AS [Color]
,[CatalogDescription].value( N'declare namespace
pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";
(/pl:ProductDescription/pl:Specifications/ProductLine)[1]' , 'nvarchar(256)' ) AS [ProductLine]
,[CatalogDescription].value( N'declare namespace
pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";
(/pl:ProductDescription/pl:Specifications/Style)[1]' , 'nvarchar(256)' ) AS [Style]
,[CatalogDescription].value( N'declare namespace
pl="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";
(/pl:ProductDescription/pl:Specifications/RiderExperience)[1]' , 'nvarchar(1024)' ) AS [RiderExperience]
,[rowguid]
,[ModifiedDate]
FROM [Production].[ProductModel]
WHERE [CatalogDescription] IS NOT NULL;

```

Related objects

[Production.ProductModel\(TABLE\)](#)

Production.vProductModelInstructions

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Resultset

Source	Column name	Data type	Nulls	Description
Production.ProductModel	ProductModelID	int	NO	
Production.ProductModel	Name	Name	NO	
Production.ProductModel	Instructions	nvarchar	YES	
	LocationID	int	YES	
	SetupHours	decimal	YES	
	MachineHours	decimal	YES	
	LaborHours	decimal	YES	
	LotSize	int	YES	
	Step	nvarchar(1024)	YES	
Production.ProductModel	rowguid	uniqueidentifier	NO	
Production.ProductModel	ModifiedDate	datetime	NO	

```
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);
```

Related objects

[Production.ProductModel\(TABLE\)](#)



Purchasing.vVendor

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Resultset

Source	Column name	Data type	Nulls	Description
Purchasing.Vendor	VendorID	int	NO	
Purchasing.Vendor	Name	Name	NO	
	ContactType	Name	NO	
Person.Contact	Title	nvarchar(8)	YES	
Person.Contact	FirstName	Name	NO	
Person.Contact	MiddleName	Name	YES	
Person.Contact	LastName	Name	NO	
Person.Contact	Suffix	nvarchar(10)	YES	
Person.Contact	Phone	Phone	YES	
Person.Contact	EmailAddress	nvarchar(50)	YES	
Person.Contact	EmailPromotion	int	NO	
Person.Address	AddressLine1	nvarchar(60)	NO	
Person.Address	AddressLine2	nvarchar(60)	YES	
Person.Address	City	nvarchar(30)	NO	
	StateProvinceName	Name	NO	
Person.Address	PostalCode	nvarchar(15)	NO	
	CountryRegionName	Name	NO	

```
CREATE VIEW [Purchasing].[vVendor] AS
SELECT
    v.[VendorID]
    ,v.[Name]
    ,ct.[Name] AS [ContactType]
    ,c.[Title]
    ,c.[FirstName]
    ,c.[MiddleName]
    ,c.[LastName]
    ,c.[Suffix]
    ,c.[Phone]
    ,c.[EmailAddress]
    ,c.[EmailPromotion]
    ,a.[AddressLine1]
    ,a.[AddressLine2]
    ,a.[City]
    ,[StateProvinceName] = sp.[Name]
    ,a.[PostalCode]
    ,[CountryRegionName] = cr.[Name]
FROM [Purchasing].[Vendor] v
INNER JOIN [Purchasing].[VendorContact] vc
    ON vc.[VendorID] = v.[VendorID]
INNER JOIN [Person].[Contact] c
    ON c.[ContactID] = vc.[ContactID]
INNER JOIN [Person].[ContactType] ct
    ON vc.[ContactTypeID] = ct.[ContactTypeID]
INNER JOIN [Purchasing].[VendorAddress] va
    ON va.[VendorID] = v.[VendorID]
INNER JOIN [Person].[Address] a
    ON a.[AddressID] = va.[AddressID]
INNER JOIN [Person].[StateProvince] sp
    ON sp.[StateProvinceID] = a.[StateProvinceID]
INNER JOIN [Person].[CountryRegion] cr
    ON cr.[CountryRegionCode] = sp.[CountryRegionCode];
```

Related objects

[Person.Address\(TABLE\)](#)

[Person.CountryRegion\(TABLE\)](#)

Person.StateProvince(TABLE)
Purchasing.VendorAddress(TABLE)
Purchasing.VendorContact(TABLE)
Person.ContactType(TABLE)
Person.Contact(TABLE)
Purchasing.Vendor(TABLE)



Sales.vIndividualCustomer

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Resultset

Source	Column name	Data type	Nulls	Description
Sales.Customer	CustomerID	int	NO	
Person.Contact	Title	nvarchar(8)	YES	
Person.Contact	FirstName	Name	NO	
Person.Contact	MiddleName	Name	YES	
Person.Contact	LastName	Name	NO	
Person.Contact	Suffix	nvarchar(10)	YES	
Person.Contact	Phone	Phone	YES	
Person.Contact	EmailAddress	nvarchar(50)	YES	
Person.Contact	EmailPromotion	int	NO	
	AddressType	Name	NO	
Person.Address	AddressLine1	nvarchar(60)	NO	
Person.Address	AddressLine2	nvarchar(60)	YES	
Person.Address	City	nvarchar(30)	NO	
	StateProvinceName	Name	NO	
Person.Address	PostalCode	nvarchar(15)	NO	
	CountryRegionName	Name	NO	
Sales.Individual	Demographics	xml	YES	

```
CREATE VIEW [Sales].[vIndividualCustomer]
AS
SELECT
    i.[CustomerID]
    ,c.[Title]
    ,c.[FirstName]
    ,c.[MiddleName]
    ,c.[LastName]
    ,c.[Suffix]
    ,c.[Phone]
    ,c.[EmailAddress]
    ,c.[EmailPromotion]
    ,at.[Name] AS [AddressType]
    ,a.[AddressLine1]
    ,a.[AddressLine2]
    ,a.[City]
    ,[StateProvinceName] = sp.[Name]
    ,a.[PostalCode]
    ,[CountryRegionName] = cr.[Name]
    ,i.[Demographics]
FROM [Sales].[Individual] i
INNER JOIN [Person].[Contact] c
    ON c.[ContactID] = i.[ContactID]
INNER JOIN [Sales].[CustomerAddress] ca
    ON ca.[CustomerID] = i.[CustomerID]
INNER JOIN [Person].[Address] a
    ON a.[AddressID] = ca.[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 ca.[AddressTypeID] = at.[AddressTypeID]
WHERE i.[CustomerID] IN (SELECT [Sales].[Customer].[CustomerID]
    FROM [Sales].[Customer] WHERE UPPER([Sales].[Customer].[CustomerType]) = 'I');
```

Related objects

[Person.Address\(TABLE\)](#)

[Sales.Individual\(TABLE\)](#)

Person.StateProvince(TABLE)
Person.CountryRegion(TABLE)
Sales.Customer(TABLE)
Sales.CustomerAddress(TABLE)
Person.Contact(TABLE)
Person.AddressType(TABLE)

Sales.vIndividualDemographics

Created: 26 Apr 2006 Last updated: 26 Apr 2006

Displays the content from each element in the xml column Demographics for each customer in the Sales.Individual table.

Resultset

Source	Column name	Data type	Nulls	Description
Sales.Individual	CustomerID	int	NO	
	TotalPurchaseYTD	money	YES	
	DateFirstPurchase	datetime	YES	
	BirthDate	datetime	YES	
	MaritalStatus	nvarchar(1)	YES	
	YearlyIncome	nvarchar(30)	YES	
	Gender	nvarchar(1)	YES	
	TotalChildren	int	YES	
	NumberChildrenAtHome	int	YES	
	Education	nvarchar(30)	YES	
	Occupation	nvarchar(30)	YES	
	HomeOwnerFlag	bit	YES	
	NumberCarsOwned	int	YES	

```
CREATE VIEW [Sales].[vIndividualDemographics]
AS
SELECT
i.[CustomerID]
,[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 [Sales].[Individual] i
CROSS APPLY i.[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;
```

Related objects

[Sales.Individual\(TABLE\)](#)

Sales.vSalesPerson

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Resultset

Source	Column name	Data type	Nulls	Description
Sales.SalesPerson	SalesPersonID	int	NO	
Person.Contact	Title	nvarchar(8)	YES	
Person.Contact	FirstName	Name	NO	
Person.Contact	MiddleName	Name	YES	
Person.Contact	LastName	Name	NO	
Person.Contact	Suffix	nvarchar(10)	YES	
	JobTitle	nvarchar(50)	NO	
Person.Contact	Phone	Phone	YES	
Person.Contact	EmailAddress	nvarchar(50)	YES	
Person.Contact	EmailPromotion	int	NO	
Person.Address	AddressLine1	nvarchar(60)	NO	
Person.Address	AddressLine2	nvarchar(60)	YES	
Person.Address	City	nvarchar(30)	NO	
	StateProvinceName	Name	NO	
Person.Address	PostalCode	nvarchar(15)	NO	
	CountryRegionName	Name	NO	
	TerritoryName	Name	YES	
	TerritoryGroup	nvarchar(50)	YES	
Sales.SalesPerson	SalesQuota	money	YES	
Sales.SalesPerson	SalesYTD	money	NO	
Sales.SalesPerson	SalesLastYear	money	NO	

```
CREATE VIEW [Sales].[vSalesPerson]
AS
SELECT
    s.[SalesPersonID]
    ,c.[Title]
    ,c.[FirstName]
    ,c.[MiddleName]
    ,c.[LastName]
    ,c.[Suffix]
    ,[JobTitle] = e.[Title]
    ,c.[Phone]
    ,c.[EmailAddress]
    ,c.[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.[EmployeeID] = s.[SalesPersonID]
LEFT OUTER JOIN [Sales].[SalesTerritory] st
    ON st.[TerritoryID] = s.[TerritoryID]
INNER JOIN [Person].[Contact] c
    ON c.[ContactID] = e.[ContactID]
INNER JOIN [HumanResources].[EmployeeAddress] ea
    ON e.[EmployeeID] = ea.[EmployeeID]
INNER JOIN [Person].[Address] a
    ON ea.[AddressID] = a.[AddressID]
INNER JOIN [Person].[StateProvince] sp
```

```
ON sp.[StateProvinceID] = a.[StateProvinceID]
INNER JOIN [Person].[CountryRegion] cr
ON cr.[CountryRegionCode] = sp.[CountryRegionCode];
```

Related objects

[Person.Address\(TABLE\)](#)

[Person.StateProvince\(TABLE\)](#)

[Sales.SalesPerson\(TABLE\)](#)

[HumanResources.EmployeeAddress\(TABLE\)](#)

[Sales.SalesTerritory\(TABLE\)](#)

[HumanResources.Employee\(TABLE\)](#)

[Person.Contact\(TABLE\)](#)

[Person.CountryRegion\(TABLE\)](#)

Sales.vSalesPersonSalesByFiscalYears

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Resultset

Source	Column name	Data type	Nulls	Description
Sales.SalesOrderHeader	SalesPersonID	int	YES	
	FullName	nvarchar(152)	YES	
HumanResources.Employee	Title	nvarchar(50)	NO	
	SalesTerritory	Name	NO	
	2002	money	YES	
	2003	money	YES	
	2004	money	YES	

```
CREATE VIEW [Sales].[vSalesPersonSalesByFiscalYears]
AS
SELECT
    pvt.[SalesPersonID]
    ,pvt.[FullName]
    ,pvt.[Title]
    ,pvt.[SalesTerritory]
    ,pvt.[2002]
    ,pvt.[2003]
    ,pvt.[2004]
FROM (SELECT
    soh.[SalesPersonID]
    ,c.[FirstName] + ' ' + COALESCE(c.[MiddleName], '') + ' ' + c.[LastName] AS [FullName]
    ,e.[Title]
    ,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.[SalesPersonID] = soh.[SalesPersonID]
INNER JOIN [Sales].[SalesTerritory] st
ON sp.[TerritoryID] = st.[TerritoryID]
INNER JOIN [HumanResources].[Employee] e
ON soh.[SalesPersonID] = e.[EmployeeID]
INNER JOIN [Person].[Contact] c
ON e.[ContactID] = c.ContactID
) AS soh
PIVOT
(
    SUM([SubTotal])
    FOR [FiscalYear]
    IN ([2002], [2003], [2004])
) AS pvt;
```

Related objects

[Person.Contact\(TABLE\)](#)

[Sales.SalesOrderHeader\(TABLE\)](#)

[Sales.SalesTerritory\(TABLE\)](#)

[HumanResources.Employee\(TABLE\)](#)

[Sales.SalesPerson\(TABLE\)](#)

Sales.vStoreWithDemographics

Created: 26 Apr 2006 Last updated: 26 Apr 2006

Stores (names and addresses) that sell Adventure Works Cycles products to consumers.

Resultset

Source	Column name	Data type	Nulls	Description
Sales.StoreContact	CustomerID	int	NO	
Person.AddressType	Name	Name	NO	
	ContactType	Name	NO	
Person.Contact	Title	nvarchar(8)	YES	
Person.Contact	FirstName	Name	NO	
Person.Contact	MiddleName	Name	YES	
Person.Contact	LastName	Name	NO	
Person.Contact	Suffix	nvarchar(10)	YES	
Person.Contact	Phone	Phone	YES	
Person.Contact	EmailAddress	nvarchar(50)	YES	
Person.Contact	EmailPromotion	int	NO	
	AddressType	Name	NO	
Person.Address	AddressLine1	nvarchar(60)	NO	
Person.Address	AddressLine2	nvarchar(60)	YES	
Person.Address	City	nvarchar(30)	NO	
	StateProvinceName	Name	NO	
Person.Address	PostalCode	nvarchar(15)	NO	
	CountryRegionName	Name	NO	
	AnnualSales	money	YES	
	AnnualRevenue	money	YES	
	BankName	nvarchar(50)	YES	
	BusinessType	nvarchar(5)	YES	
	YearOpened	int	YES	
	Specialty	nvarchar(50)	YES	
	SquareFeet	int	YES	
	Brands	nvarchar(30)	YES	
	Internet	nvarchar(30)	YES	
	NumberEmployees	int	YES	

```
CREATE VIEW [Sales].[vStoreWithDemographics] AS
SELECT
s.[CustomerID]
,s.[Name]
,ct.[Name] AS [ContactType]
,c.[Title]
,c.[FirstName]
,c.[MiddleName]
,c.[LastName]
,c.[Suffix]
,c.[Phone]
,c.[EmailAddress]
,c.[EmailPromotion]
,at.[Name] AS [AddressType]
,a.[AddressLine1]
,a.[AddressLine2]
,a.[City]
,sp.[Name] AS [StateProvinceName]
,a.[PostalCode]
,cr.[Name] AS [CountryRegionName]
,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
INNER JOIN [Sales].[StoreContact] sc
ON sc.[CustomerID] = s.[CustomerID]
INNER JOIN [Person].[Contact] c
ON c.[ContactID] = sc.[ContactID]
INNER JOIN [Person].[ContactType] ct
ON sc.[ContactTypeID] = ct.[ContactTypeID]
INNER JOIN [Sales].[CustomerAddress] ca
ON ca.[CustomerID] = s.[CustomerID]
INNER JOIN [Person].[Address] a
ON a.[AddressID] = ca.[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 ca.[AddressTypeID] = at.[AddressTypeID]
WHERE s.[CustomerID] IN (SELECT [Sales].[Customer].[CustomerID]
FROM [Sales].[Customer] WHERE UPPER([Sales].[Customer].[CustomerType]) = 'S');

```

Related objects

[Sales.StoreContact\(TABLE\)](#)

[Person.Address\(TABLE\)](#)

[Person.StateProvince\(TABLE\)](#)

[Sales.Store\(TABLE\)](#)

[Sales.Customer\(TABLE\)](#)

[Sales.CustomerAddress\(TABLE\)](#)

[Person.Contact\(TABLE\)](#)

[Person.ContactType\(TABLE\)](#)

[Person.CountryRegion\(TABLE\)](#)

[Person.AddressType\(TABLE\)](#)

Stored procedures

dbo.uspGetBillOfMaterials

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@StartProductID	int	IN	Input parameter for the stored procedure uspGetBillOfMaterials. Enter a valid ProductID from the Production.Product table.
@CheckDate	datetime	IN	Input parameter for the stored procedure uspGetBillOfMaterials used to eliminate components not used after that date. Enter a valid date.

```
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;
```

Related objects

[Production.BillOfMaterials\(TABLE\)](#)

[Production.Product\(TABLE\)](#)

dbo.uspGetEmployeeManagers

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@EmployeeID	int	IN	Input parameter for the stored procedure uspGetEmployeeManagers. Enter a valid EmployeeID from the HumanResources.Employee table.

```
CREATE PROCEDURE [dbo].[uspGetEmployeeManagers]
    @EmployeeID [int]
AS
BEGIN
    SET NOCOUNT ON;

    -- Use recursive query to list out all Employees required for a particular Manager
    WITH [EMP_cte]([EmployeeID], [ManagerID], [FirstName], [LastName], [Title], [RecursionLevel]) -- CTE
name and columns
    AS (
        SELECT e.[EmployeeID], e.[ManagerID], c.[FirstName], c.[LastName], e.[Title], 0 -- Get the
initial Employee
        FROM [HumanResources].[Employee] e
            INNER JOIN [Person].[Contact] c
                ON e.[ContactID] = c.[ContactID]
        WHERE e.[EmployeeID] = @EmployeeID
        UNION ALL
        SELECT e.[EmployeeID], e.[ManagerID], c.[FirstName], c.[LastName], e.[Title], [RecursionLevel] +
1 -- Join recursive member to anchor
        FROM [HumanResources].[Employee] e
            INNER JOIN [EMP_cte]
                ON e.[EmployeeID] = [EMP_cte].[ManagerID]
            INNER JOIN [Person].[Contact] c
                ON e.[ContactID] = c.[ContactID]
    )
    -- Join back to Employee to return the manager name
    SELECT [EMP_cte].[RecursionLevel], [EMP_cte].[EmployeeID], [EMP_cte].[FirstName],
[EMP_cte].[LastName],
        [EMP_cte].[ManagerID], c.[FirstName] AS 'ManagerFirstName', c.[LastName] AS 'ManagerLastName'
-- Outer select from the CTE
    FROM [EMP_cte]
        INNER JOIN [HumanResources].[Employee] e
            ON [EMP_cte].[ManagerID] = e.[EmployeeID]
        INNER JOIN [Person].[Contact] c
            ON e.[ContactID] = c.[ContactID]
    ORDER BY [RecursionLevel], [ManagerID], [EmployeeID]
    OPTION (MAXRECURSION 25)
END;
```

Related objects

[Person.Contact\(TABLE\)](#)

[HumanResources.Employee\(TABLE\)](#)

dbo.uspGetManagerEmployees

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@ManagerID	int	IN	Input parameter for the stored procedure uspGetManagerEmployees. Enter a valid ManagerID from the HumanResources.Employee table.

```
CREATE PROCEDURE [dbo].[uspGetManagerEmployees]
    @ManagerID [int]
AS
BEGIN
    SET NOCOUNT ON;

    -- Use recursive query to list out all Employees required for a particular Manager
    WITH [EMP_cte]([EmployeeID], [ManagerID], [FirstName], [LastName], [RecursionLevel]) -- CTE name and
columns
    AS (
        SELECT e.[EmployeeID], e.[ManagerID], c.[FirstName], c.[LastName], 0 -- Get the initial list of
Employees for Manager n
        FROM [HumanResources].[Employee] e
            INNER JOIN [Person].[Contact] c
                ON e.[ContactID] = c.[ContactID]
        WHERE [ManagerID] = @ManagerID
        UNION ALL
        SELECT e.[EmployeeID], e.[ManagerID], c.[FirstName], c.[LastName], [RecursionLevel] + 1 -- Join
recursive member to anchor
        FROM [HumanResources].[Employee] e
            INNER JOIN [EMP_cte]
                ON e.[ManagerID] = [EMP_cte].[EmployeeID]
            INNER JOIN [Person].[Contact] c
                ON e.[ContactID] = c.[ContactID]
    )
    -- Join back to Employee to return the manager name
    SELECT [EMP_cte].[RecursionLevel], [EMP_cte].[ManagerID], c.[FirstName] AS 'ManagerFirstName',
c.[LastName] AS 'ManagerLastName',
        [EMP_cte].[EmployeeID], [EMP_cte].[FirstName], [EMP_cte].[LastName] -- Outer select from the CTE
    FROM [EMP_cte]
        INNER JOIN [HumanResources].[Employee] e
            ON [EMP_cte].[ManagerID] = e.[EmployeeID]
        INNER JOIN [Person].[Contact] c
            ON e.[ContactID] = c.[ContactID]
    ORDER BY [RecursionLevel], [ManagerID], [EmployeeID]
    OPTION (MAXRECURSION 25)
END;
```

Related objects

[Person.Contact\(TABLE\)](#)

[HumanResources.Employee\(TABLE\)](#)

dbo.uspGetWhereUsedProductID

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@StartProductID	int	IN	Input parameter for the stored procedure uspGetWhereUsedProductID. Enter a valid ProductID from the Production.Product table.
@CheckDate	datetime	IN	Input parameter for the stored procedure uspGetWhereUsedProductID used to eliminate components not used after that date. Enter a valid date.

```
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;
```

Related objects

[Production.BillOfMaterials\(TABLE\)](#)

[Production.Product\(TABLE\)](#)

dbo.uspLogError

Created: 26 Apr 2006 Last updated: 26 Oct 2008

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.

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@ErrorLogID	int	INOUT	Output parameter for the stored procedure uspLogError. Contains the ErrorLogID value corresponding to the row inserted by uspLogError in the ErrorLog table.

```
-- 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;
```

Related objects

[dbo.uspPrintError](#)(Stored procedure)

[dbo.ErrorLog](#)(TABLE)

Object that depends on [dbo.uspLogError](#)

[HumanResources.uspUpdateEmployeeHireInfo](#)(Stored procedure)

[HumanResources.uspUpdateEmployeeLogin](#)(Stored procedure)

[HumanResources.uspUpdateEmployeePersonalInfo](#)(Stored procedure)

[Production.iWorkOrder](#)(TRIGGER)

[Production.uWorkOrder](#)(TRIGGER)

[Purchasing.dVendor](#)(TRIGGER)

[Purchasing.iPurchaseOrderDetail](#)(TRIGGER)

[Purchasing.uPurchaseOrderDetail](#)(TRIGGER)

[Purchasing.uPurchaseOrderHeader](#)(TRIGGER)

[Sales.iduSalesOrderDetail](#)(TRIGGER)

[Sales.iStore](#)(TRIGGER)

[Sales.uSalesOrderHeader](#)(TRIGGER)

dbo.uspPrintError

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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.

```
-- 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;
```

Object that depends on dbo.uspPrintError

dbo.ufnGetContactInformation()
dbo.uspLogError(Stored procedure)
Production.iWorkOrder(TRIGGER)
Production.uWorkOrder(TRIGGER)
Purchasing.dVendor(TRIGGER)
Purchasing.iPurchaseOrderDetail(TRIGGER)
Purchasing.uPurchaseOrderDetail(TRIGGER)
Purchasing.uPurchaseOrderHeader(TRIGGER)
Sales.iduSalesOrderDetail(TRIGGER)
Sales.iStore(TRIGGER)
Sales.uSalesOrderHeader(TRIGGER)

HumanResources.uspUpdateEmployeeHireInfo

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@EmployeeID	int	IN	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a valid EmployeeID from the Employee table.
@Title	nvarchar	IN	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a title for the employee.
@HireDate	datetime	IN	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a hire date for the employee.
@RateChangeDate	datetime	IN	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the date the rate changed for the employee.
@Rate	money	IN	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the new rate for the employee.
@PayFrequency	tinyint	IN	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the pay frequency for the employee.
@CurrentFlag	bit	IN	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the current flag for the employee.

```
CREATE PROCEDURE [HumanResources].[uspUpdateEmployeeHireInfo]
    @EmployeeID [int],
    @Title [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 [Title] = @Title
            ,[HireDate] = @HireDate
            ,[CurrentFlag] = @CurrentFlag
        WHERE [EmployeeID] = @EmployeeID;

        INSERT INTO [HumanResources].[EmployeePayHistory]
            ([EmployeeID]
            ,[RateChangeDate]
            ,[Rate]
            ,[PayFrequency])
        VALUES (@EmployeeID, @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;
```


Related objects

- dbo.uspLogError(Stored procedure)
- HumanResources.Employee(TABLE)
- HumanResources.EmployeePayHistory(TABLE)

HumanResources.uspUpdateEmployeeLogin

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@EmployeeID	int	IN	Input parameter for the stored procedure uspUpdateEmployeeLogin. Enter a valid EmployeeID from the Employee table.
@ManagerID	int	IN	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a valid ManagerID for the employee.
@LoginID	nvarchar	IN	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a valid login for the employee.
@Title	nvarchar	IN	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a title for the employee.
@HireDate	datetime	IN	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a hire date for the employee.
@CurrentFlag	bit	IN	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the current flag for the employee.

```
CREATE PROCEDURE [HumanResources].[uspUpdateEmployeeLogin]
    @EmployeeID [int],
    @ManagerID [int],
    @LoginID [nvarchar](256),
    @Title [nvarchar](50),
    @HireDate [datetime],
    @CurrentFlag [dbo].[Flag]
WITH EXECUTE AS CALLER
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        UPDATE [HumanResources].[Employee]
        SET [ManagerID] = @ManagerID
            ,[LoginID] = @LoginID
            ,[Title] = @Title
            ,[HireDate] = @HireDate
            ,[CurrentFlag] = @CurrentFlag
        WHERE [EmployeeID] = @EmployeeID;
    END TRY
    BEGIN CATCH
        EXECUTE [dbo].[uspLogError];
    END CATCH;
END;
```

Related objects

[dbo.uspLogError\(Stored procedure\)](#)

[HumanResources.Employee\(TABLE\)](#)

HumanResources.uspUpdateEmployeePersonalInfo

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@EmployeeID	int	IN	Input parameter for the stored procedure uspUpdateEmployeePersonalInfo. Enter a valid EmployeeID from the HumanResources.Employee table.
@NationalIDNumber	nvarchar	IN	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a national ID for the employee.
@BirthDate	datetime	IN	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a birth date for the employee.
@MaritalStatus	nchar	IN	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a marital status for the employee.
@Gender	nchar	IN	Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a gender for the employee.

```
CREATE PROCEDURE [HumanResources].[uspUpdateEmployeePersonalInfo]
    @EmployeeID [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 [EmployeeID] = @EmployeeID;
    END TRY
    BEGIN CATCH
        EXECUTE [dbo].[uspLogError];
    END CATCH;
END;
```

Related objects

[dbo.uspLogError](#)(Stored procedure)

[HumanResources.Employee](#)(TABLE)

User defined functions

dbo.ufnGetAccountingEndDate

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
	datetime	OUT	

```
CREATE FUNCTION [dbo].[ufnGetAccountingEndDate]()
RETURNS [datetime]
AS
BEGIN
RETURN DATEADD(millisecond, -2, CONVERT(datetime, '20040701', 112));
END;
```

dbo.ufnGetAccountingStartDate

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
	datetime	OUT	

```
CREATE FUNCTION [dbo].[ufnGetAccountingStartDate]()  
RETURNS [datetime]  
AS  
BEGIN  
RETURN CONVERT(datetime, '20030701', 112);  
END;
```



dbo.ufnGetContactInformation

Created: 26 Apr 2006 Last updated: 26 Oct 2008



Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
@ContactID	int	IN	Input parameter for the table value function ufnGetContactInformation. Enter a valid ContactID from the Person.Contact table.

```

CREATE FUNCTION [dbo].[ufnGetContactInformation](@ContactID int)
RETURNS @retContactInformation TABLE
(
  -- Columns returned by the function
  [ContactID] int PRIMARY KEY NOT NULL,
  [FirstName] [nvarchar](50) NULL,
  [LastName] [nvarchar](50) NULL,
  [JobTitle] [nvarchar](50) NULL,
  [ContactType] [nvarchar](50) NULL
)
AS
-- Returns the first name, last name, job title and contact type for the specified contact.
BEGIN
DECLARE
@FirstName [nvarchar](50),
@LastName [nvarchar](50),
@JobTitle [nvarchar](50),
@ContactType [nvarchar](50);
-- Get common contact information
SELECT
@ContactID = ContactID,
@FirstName = FirstName,
@LastName = LastName
FROM [Person].[Contact]
WHERE [ContactID] = @ContactID;
SET @JobTitle =
CASE
-- Check for employee
WHEN EXISTS(SELECT * FROM [HumanResources].[Employee] e
WHERE e.[ContactID] = @ContactID)
THEN (SELECT [Title]
FROM [HumanResources].[Employee]
WHERE [ContactID] = @ContactID)
-- Check for vendor
WHEN EXISTS(SELECT * FROM [Purchasing].[VendorContact] vc
INNER JOIN [Person].[ContactType] ct
ON vc.[ContactTypeID] = ct.[ContactTypeID]
WHERE vc.[ContactID] = @ContactID)
THEN (SELECT ct.[Name]
FROM [Purchasing].[VendorContact] vc
INNER JOIN [Person].[ContactType] ct
ON vc.[ContactTypeID] = ct.[ContactTypeID]
WHERE vc.[ContactID] = @ContactID)
-- Check for store
WHEN EXISTS(SELECT * FROM [Sales].[StoreContact] sc
INNER JOIN [Person].[ContactType] ct
ON sc.[ContactTypeID] = ct.[ContactTypeID]
WHERE sc.[ContactID] = @ContactID)
THEN (SELECT ct.[Name]
FROM [Sales].[StoreContact] sc
INNER JOIN [Person].[ContactType] ct
ON sc.[ContactTypeID] = ct.[ContactTypeID]
WHERE [ContactID] = @ContactID)
ELSE NULL
END;
SET @ContactType =
CASE
-- Check for employee

```

```

WHEN EXISTS(SELECT * FROM [HumanResources].[Employee] e
WHERE e.[ContactID] = @ContactID)
THEN 'Employee'
-- Check for vendor
WHEN EXISTS(SELECT * FROM [Purchasing].[VendorContact] vc
INNER JOIN [Person].[ContactType] ct
ON vc.[ContactTypeID] = ct.[ContactTypeID]
WHERE vc.[ContactID] = @ContactID)
THEN 'Vendor Contact'
-- Check for store
WHEN EXISTS(SELECT * FROM [Sales].[StoreContact] sc
INNER JOIN [Person].[ContactType] ct
ON sc.[ContactTypeID] = ct.[ContactTypeID]
WHERE sc.[ContactID] = @ContactID)
THEN 'Store Contact'
-- Check for individual consumer
WHEN EXISTS(SELECT * FROM [Sales].[Individual] i
WHERE i.[ContactID] = @ContactID)
THEN 'Consumer'
END;
-- Return the information to the caller
IF @ContactID IS NOT NULL
BEGIN
EXECUTE [dbo].[uspPrintError];
INSERT @retContactInformation
SELECT @ContactID, @FirstName, @LastName, @JobTitle, @ContactType;
END;
RETURN;
END;

```

Related objects

[Production.BillOfMaterials\(TABLE\)](#)

[Production.BillOfMaterials\(TABLE\)](#)

[Production.BillOfMaterials\(TABLE\)](#)

[Production.BillOfMaterials\(TABLE\)](#)

[Production.BillOfMaterials\(TABLE\)](#)

[Production.BillOfMaterials\(TABLE\)](#)

[Production.BillOfMaterials\(TABLE\)](#)

dbo.ufnGetDocumentStatusText

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
	nvarchar	OUT	
@Status	tinyint	IN	Input parameter for the scalar function ufnGetDocumentStatusText. Enter a valid integer.

```
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;
```




dbo.ufnGetProductDealerPrice

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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



Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
	money	OUT	
@ProductID	int	IN	Input parameter for the scalar function ufnGetProductDealerPrice. Enter a valid ProductID from the Production.Product table.
@OrderDate	datetime	IN	Input parameter for the scalar function ufnGetProductDealerPrice. Enter a valid order date.

```
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;
```

Related objects

[Production.BillOfMaterials\(TABLE\)](#)

[Production.BillOfMaterials\(TABLE\)](#)

dbo.ufnGetProductListPrice

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
	money	OUT	
@ProductID	int	IN	Input parameter for the scalar function ufnGetProductListPrice. Enter a valid ProductID from the Production.Product table.
@OrderDate	datetime	IN	Input parameter for the scalar function ufnGetProductListPrice. Enter a valid order date.

```
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;
```

Related objects

[Production.BillOfMaterials\(TABLE\)](#)

[Production.BillOfMaterials\(TABLE\)](#)

dbo.ufnGetProductStandardCost

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
	money	OUT	
@ProductID	int	IN	Input parameter for the scalar function ufnGetProductStandardCost. Enter a valid ProductID from the Production.Product table.
@OrderDate	datetime	IN	Input parameter for the scalar function ufnGetProductStandardCost. Enter a valid order date.

```
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;
```

Related objects

[Production.BillOfMaterials\(TABLE\)](#)

[Production.BillOfMaterials\(TABLE\)](#)

dbo.ufnGetPurchaseOrderStatusText

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
	nvarchar	OUT	
@Status	tinyint	IN	Input parameter for the scalar function ufnGetPurchaseOrderStatusText. Enter a valid integer.

```
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;
```

dbo.ufnGetSalesOrderStatusText

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
	nvarchar	OUT	
@Status	tinyint	IN	Input parameter for the scalar function ufnGetSalesOrderStatusText. Enter a valid integer.

```
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;
```

dbo.ufnGetStock

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
	int	OUT	
@ProductID	int	IN	Input parameter for the scalar function ufnGetStock. Enter a valid ProductID from the Production.ProductInventory table.

```
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;
```

Related objects

[Production.BillOfMaterials\(TABLE\)](#)

dbo.ufnLeadingZeros

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Parameters

PARAMETER NAME	DATA TYPE	MODE	DESCRIPTION
	varchar	OUT	
@Value	int	IN	Input parameter for the scalar function ufnLeadingZeros. Enter a valid integer.

```
CREATE FUNCTION [dbo].[ufnLeadingZeros](
    @Value int
)
RETURNS varchar(8)
WITH SCHEMABINDING
AS
BEGIN
    DECLARE @ReturnValue varchar(8);

    SET @ReturnValue = CONVERT(varchar(8), @Value);
    SET @ReturnValue = REPLICATE('0', 8 - DATALENGTH(@ReturnValue)) + @ReturnValue;

    RETURN (@ReturnValue);
END;
```

Object that depends on dbo.ufnLeadingZeros

[Production.BillOfMaterials\(TABLE\)](#)



User-defined data types

UDT Name	Data type	Length	Allow nulls	Description
AccountNumber	nvarchar	15	null	
Flag	bit		not null	
Name	nvarchar	50	null	
NameStyle	bit		not null	
OrderNumber	nvarchar	25	null	
Phone	nvarchar	25	null	

XML Schemas

HumanResources.HRResumeSchemaCollection

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Components

COMPONENT NAME	TYPE	DERIVATION	NAMESPACE
AddressType	COMPLEX_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
EducationType	COMPLEX_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
EmploymentType	COMPLEX_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
LocationType	COMPLEX_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
NameType	COMPLEX_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
ResumeType	COMPLEX_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
TelephoneType	COMPLEX_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Addr.Location	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Addr.OrgName	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Addr.PostalCode	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Addr.Street	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Addr.Telephone	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Addr.Type	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Address	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
E-Mail	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Edu.Degree	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Edu.EndDate	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Edu.GPA	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Edu.GPAAAlternate	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Edu.GPAScale	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Edu.Level	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Edu.Location	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Edu.Major	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Edu.Minor	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Edu.School	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Edu.StartDate	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Education	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Emp.EndDate	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Emp.FunctionCategory	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Emp.IndustryCategory	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Emp.JobTitle	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Emp.Location	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Emp.OrgName	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Emp.Responsibility	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Emp.StartDate	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Employment	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Loc.City	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Loc.CountryRegion	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Loc.State	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Location	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Name	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Name.First	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Name.Last	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume

Name.Middle	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Name.Prefix	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Name.Suffix	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Resume	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Skills	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Tel.AreaCode	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Tel.Extension	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Tel.IntlCode	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Tel.Number	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Tel.Type	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
Telephone	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume
WebSite	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/Resume

XML Schemas

Person.AdditionalContactInfoSchemaCollection

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Components

COMPONENT NAME	TYPE	DERIVATION	NAMESPACE
date	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactRecord
addressType	COMPLEX_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
eMailType	COMPLEX_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
phoneNumberType	COMPLEX_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
specialInstructionsType	COMPLEX_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
AdditionalContactInfo	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactInfo
City	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
ContactRecord	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactRecord
CountryRegion	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
PostalCode	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
SpecialInstructions	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
SpecialInstructions	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
SpecialInstructions	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
StateProvince	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
Street	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
eMail	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
eMailAddress	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
facsimileTelephoneNumber	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
homePostalAddress	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
internationaliSDNNumber	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
mobile	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
number	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
pager	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
physicalDeliveryOfficeName	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
registeredAddress	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
telephoneNumber	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes
telexNumber	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ContactTypes

XML Schemas

Production.ManulInstructionsSchemaCollection

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Components

COMPONENT NAME	TYPE	DERIVATION	NAMESPACE
LaborHours	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelManulInstructions
LocationID	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelManulInstructions
LotSize	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelManulInstructions
MachineHours	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelManulInstructions
SetupHours	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelManulInstructions
StepType	COMPLEX_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelManulInstructions
Location	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelManulInstructions
blueprint	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelManulInstructions
diag	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelManulInstructions
material	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelManulInstructions
root	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelManulInstructions
specs	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelManulInstructions
step	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelManulInstructions
tool	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelManulInstructions

XML Schemas

Production.ProductDescriptionSchemaCollection

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Components

COMPONENT NAME	TYPE	DERIVATION	NAMESPACE
ProductModelID	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
ProductModelName	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
Category	COMPLEX_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
Features	COMPLEX_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
Manufacturer	COMPLEX_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
Picture	COMPLEX_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
ProductDescription	COMPLEX_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
Specifications	COMPLEX_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
Summary	COMPLEX_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
Angle	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
Category	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
Code	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
Copyright	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
CopyrightURL	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
Description	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
Description	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelWarrAndMain
Description	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelWarrAndMain
Features	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
Maintenance	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelWarrAndMain
Manufacturer	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
Name	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
Name	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
NoOfYears	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelWarrAndMain
Picture	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
ProductDescription	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
ProductPhotoID	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
ProductURL	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
Size	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
Specifications	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
Summary	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
Taxonomy	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelDescription
Warranty	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelWarrAndMain
WarrantyPeriod	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/ProductModelWarrAndMain

XML Schemas

Sales.IndividualSurveySchemaCollection

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Components

COMPONENT NAME	TYPE	DERIVATION	NAMESPACE
BirthDate	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/IndividualSurvey
Comments	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/IndividualSurvey
CommuteDistance	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/IndividualSurvey
DateFirstPurchase	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/IndividualSurvey
Education	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/IndividualSurvey
Gender	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/IndividualSurvey
Hobby	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/IndividualSurvey
HomeOwnerFlag	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/IndividualSurvey
IndividualSurvey	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/IndividualSurvey
MaritalStatus	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/IndividualSurvey
NumberCarsOwned	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/IndividualSurvey
NumberChildrenAtHome	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/IndividualSurvey
Occupation	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/IndividualSurvey
TotalChildren	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/IndividualSurvey
TotalPurchaseYTD	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/IndividualSurvey
YearlyIncome	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/IndividualSurvey
MileRangeType	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/IndividualSurvey
SalaryType	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/IndividualSurvey

XML Schemas

Sales.StoreSurveySchemaCollection

Created: 26 Apr 2006 Last updated: 26 Apr 2006

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

Components

COMPONENT NAME	TYPE	DERIVATION	NAMESPACE
AnnualRevenue	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/StoreSurvey
AnnualSales	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/StoreSurvey
BankName	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/StoreSurvey
Brands	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/StoreSurvey
BusinessType	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/StoreSurvey
Comments	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/StoreSurvey
ContactName	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/StoreSurvey
Internet	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/StoreSurvey
JobTitle	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/StoreSurvey
NumberEmployees	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/StoreSurvey
Specialty	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/StoreSurvey
SquareFeet	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/StoreSurvey
StoreSurvey	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/StoreSurvey
YearOpened	ELEMENT	NONE	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/StoreSurvey
BrandType	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/StoreSurvey
BusinessType	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/StoreSurvey
InternetType	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/StoreSurvey
SpecialtyType	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/07/advent-ure-works/StoreSurvey

XML Schemas

 sys.sys


Created: 14 Oct 2005 Last updated: 14 Oct 2005

Components

COMPONENT NAME	TYPE	DERIVATION	NAMESPACE
anySimpleType	ANY_SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
anyType	ANY_TYPE	NONE	http://www.w3.org/2001/XMLSchema
base	ATTRIBUTE	NONE	http://www.w3.org/XML/1998/namespace
clrTypeName	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
lang	ATTRIBUTE	NONE	http://www.w3.org/XML/1998/namespace
localeId	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
maxLength	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
precision	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
scale	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
space	ATTRIBUTE	NONE	http://www.w3.org/XML/1998/namespace
sqlCollationVersion	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
sqlCompareOptions	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
sqlDbType	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
sqlSortId	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
sqlTypeAlias	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
xmlSchemaCollection	ATTRIBUTE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
xml	COMPLEX_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
ENTITIES	LIST_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
IDREFS	LIST_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
NMTOKENS	LIST_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
sqlCompareOptionsList	LIST_TYPE	NONE	http://schemas.microsoft.com/sqlserver/2004/sqltypes
NOTATION	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
QName	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
anyURI	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
base64Binary	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
boolean	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
date	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
dateTime	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
decimal	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
double	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
duration	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
float	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
gDay	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
gMonth	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
gMonthDay	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
gYear	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
gYearMonth	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
hexBinary	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
string	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
time	PRIMITIVE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
ENTITY	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
ID	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
IDREF	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
NCName	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema

NMTOKEN	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
Name	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
bigint	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
binary	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
bit	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
byte	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
char	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
datetime	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
dbobject	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
decimal	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
float	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
image	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
int	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
int	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
integer	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
language	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
long	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
money	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
nchar	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
negativeInteger	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
nonNegativeInteger	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
nonPositiveInteger	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
normalizedString	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
ntext	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
numeric	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
nvarchar	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
positiveInteger	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
real	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
short	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
smalldatetime	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
smallint	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
smallmoney	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
sqlCompareOptionsEnum	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
sqlDbTypeEnum	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
text	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
timestamp	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
timestampNumeric	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
tinyint	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
token	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
uniqueidentifier	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
unsignedByte	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
unsignedInt	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
unsignedLong	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
unsignedShort	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/2001/XMLSchema
varbinary	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
varchar	SIMPLE_TYPE	RESTRICTION	http://schemas.microsoft.com/sqlserver/2004/sqltypes
xmlSpaceEnum	SIMPLE_TYPE	RESTRICTION	http://www.w3.org/XML/1998/namespace

Users

	User name	Login name	Created	Roles	User description
	dbo	sa	08/04/2003	db_owner	
	TestUser	panel	03/06/2008	db_owner	zxcv

Roles

Role name	App role	Users with this role	Role description
db_accessadmin			
db_backupoperator			
db_datareader			
db_datawriter			
db_ddladmin			
db_denydatareader			
db_denydatawriter			
db_owner		dbo, TestUser	
db_securityadmin			
public			