[Execute the Script](#ExecuteScript)

[Drop the Database](#DropDB)

[Create Database](#CreateDataBase)

[Create DDL Trigger for Database](#CreateDDLTriggerforDB)

[Create Error Log Objects](#CreateErrorLogObjects)

[Create Data Types](#CreateDataTypes)

[Add Pre-Table DataBase Functions](#CreatePreTableDBFunctions)

[Create Five Database Schemas](#CreateDBSchemas)

[Create Six XML Schemas](#CreateSixXMLSchemas)

[Create Tables](#CreateTables)

[Load Data into Tables](#LoaddataintoTables)

[Add Primary Keys](#AddPrimaryKeys)

[Add Indexes](#AddIndexes)

[Add XML Index for each XML Column](#AddXMLIndexforeachXMLColumn)

[Create Fulltext Catalog & Indexes](#CreateFullTextCatalogandIndexes)

[Create Foreign Key Constraints](#CreateForeignKeyConstraints)

[Add Table Triggers](#AddTableTriggers)

[Add Database Views](#AddDatabaseViews)

[Add Database Functions](#AddDatabaseFunctions)

[Create Stored Procedures](#CreateStoredProcedures)

[Add Extended Properties](#AddExtendedProperties)

[Drop DDL Trigger for Database](#DropDDLTriggerforDatabase)

[Output DB Obj Creation Messages](#OutputDatabaseObjectCreationMessages)

**MICROSOFT**

**ADVENTUREWORKS DATABASE DDL**

**A review of the entire script developed by Microsoft**

**And used to create the AdventureWorks database.**

**The table of contents is to the right. Use CTL-Click**

**To quickly moved to those sections. “TOP” hyper-links are**

**Located at the end of each section to take you back here.**

**Technical information is scattered throughout this**

**Document addressing different topics.**

George Tuccio

March 5, 2018

/\*============================================================================

File: instawdb.sql

Summary: Creates the AdventureWorks sample database. Run this on

any version of SQL Server (2008R2 or later) to get AdventureWorks for your

current version.

Date: October 26, 2017

Updated: October 26, 2017

------------------------------------------------------------------------------

This file is part of the Microsoft SQL Server Code Samples.

Copyright (C) Microsoft Corporation. All rights reserved.

This source code is intended only as a supplement to Microsoft

Development Tools and/or on-line documentation. See these other

materials for detailed information regarding Microsoft code samples.

All data in this database is ficticious.

THIS CODE AND INFORMATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY

KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE

IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A

PARTICULAR PURPOSE.

============================================================================\*/

/\*

\* HOW TO RUN THIS SCRIPT:

\*

\* 1. Enable full-text search on your SQL Server instance.

\*

\* 2. Open the script inside SQL Server Management Studio and enable SQLCMD mode.

\* This option is in the Query menu.

\*

\* 3. Copy this script and the install files to C:\Samples\AdventureWorks, or

\* set the following environment variable to your own data path.

\*/

**:setvar SqlSamplesSourceDataPath "C:\Samples\AdventureWorks\"**

/\*

\* 4. Append the SQL Server version number to database name if you want to

\* differentiate it from other installs of AdventureWorks.

\*/

**:setvar DatabaseName "AdventureWorks"**

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Execute the Script

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF '$(SqlSamplesSourceDataPath)' IS NULL OR '$(SqlSamplesSourceDataPath)' = ''

BEGIN

RAISERROR(N'The variable SqlSamplesSourceDataPath must be defined.', 16, 127) WITH NOWAIT

RETURN

END;

Double @@ represents system variables

SET NOCOUNT OFF;

GO

PRINT CONVERT(varchar(1000), @@VERSION);

GO

PRINT '';

PRINT 'Started - ' + CONVERT(varchar, GETDATE(), 121);

GO

USE [master];

GO

[TOP](#Top)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Drop Database

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

:SetVar Variable Reference

PRINT '';

PRINT '\*\*\* Dropping Database';

GO

IF EXISTS (SELECT [name] FROM [master].[sys].[databases] WHERE [name] = N'$(DatabaseName)')

DROP DATABASE $(DatabaseName);

-- If the database has any other open connections close the network connection.

IF @@ERROR = 3702

RAISERROR('$(DatabaseName) database cannot be dropped because there are still other open connections', 127, 127) WITH NOWAIT, LOG;

GO

[TOP](#Top)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Create Database

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PRINT '';

PRINT '\*\*\* Creating Database';

GO

**CREATE DATABASE $(DatabaseName);**

**GO**

PRINT '';

PRINT '\*\*\* Checking for $(DatabaseName) Database';

/\* CHECK FOR DATABASE IF IT DOESN'T EXISTS, DO NOT RUN THE REST OF THE SCRIPT \*/

IF NOT EXISTS (SELECT TOP 1 1 FROM sys.databases WHERE name = N'$(DatabaseName)')

BEGIN

PRINT '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'

+char(10)+'\*\*\*\*\*\*\*\*$(DatabaseName) Database does not exist. Make sure that the script is being run in SQLCMD mode and that the variables have been correctly set.\*\*\*\*\*\*\*\*\*'

+char(10)+'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*';

SET NOEXEC ON;

END

GO

**ALTER DATABASE** $(DatabaseName)

SET RECOVERY SIMPLE,

ANSI\_NULLS ON,

ANSI\_PADDING ON,

ANSI\_WARNINGS ON,

ARITHABORT ON,

CONCAT\_NULL\_YIELDS\_NULL ON,

QUOTED\_IDENTIFIER ON,

NUMERIC\_ROUNDABORT OFF,

PAGE\_VERIFY CHECKSUM,

ALLOW\_SNAPSHOT\_ISOLATION OFF;

GO

USE $(DatabaseName);

GO

[TOP](#Top)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Create DDL Trigger for Database

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DDL triggers are a special kind of trigger that fire in response to Data Definition Language (DDL) statements. They can be used to perform administrative tasks in the database such as auditing and regulating database operations.

DDL triggers, like regular triggers, fire stored procedures in response to an event. However, unlike DML triggers, they do not fire in response to UPDATE, INSERT, or DELETE statements on a table or view. Instead, they fire in response to a variety of Data Definition Language (DDL) events. These events primarily correspond to Transact-SQL statements that start with the keywords **CREATE**, **ALTER**, and **DROP**. Certain system stored procedures that perform DDL-like operations can also fire DDL triggers.

* You want to prevent certain changes to your database schema.
* You want something to occur in the database in response to a change in your database schema.
* You want to record changes or events in the database schema.

DDL triggers fire only after the DDL statements that trigger them are run. DDL triggers cannot be used as INSTEAD OF triggers.

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PRINT '';

PRINT '\*\*\* Creating DDL Trigger for Database';

GO

SET QUOTED\_IDENTIFIER ON;

GO

-- Create table to store database object creation messages

-- \*\*\* WARNING: THIS TABLE IS INTENTIONALLY A HEAP - DO NOT ADD A PRIMARY KEY \*\*\*

CREATE **TABLE** [dbo].[DatabaseLog](

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

[PostTime] [datetime] NOT NULL,

[DatabaseUser] [sysname] NOT NULL,

[Event] [sysname] NOT NULL,

[Schema] [sysname] NULL,

[Object] [sysname] NULL,

[TSQL] [nvarchar](max) NOT NULL,

[XmlEvent] [xml] NOT NULL

) ON [PRIMARY];

GO

CREATE **TRIGGER** [ddlDatabaseTriggerLog] ON DATABASE

FOR DDL\_DATABASE\_LEVEL\_EVENTS AS 🡨seems this is a group of events or event types

BEGIN

SET NOCOUNT ON;

DECLARE @data XML;

DECLARE @schema sysname;

DECLARE @object sysname;

DECLARE @eventType sysname;

SET @data = EVENTDATA();

SET @eventType = @data.value('(/EVENT\_INSTANCE/EventType)[1]', 'sysname');

SET @schema = @data.value('(/EVENT\_INSTANCE/SchemaName)[1]', 'sysname');

SET @object = @data.value('(/EVENT\_INSTANCE/ObjectName)[1]', 'sysname')

IF @object IS NOT NULL

PRINT ' ' + @eventType + ' - ' + @schema + '.' + @object;

ELSE

PRINT ' ' + @eventType + ' - ' + @schema;

IF @eventType IS NULL

PRINT CONVERT(nvarchar(max), @data);

INSERT [dbo].[DatabaseLog]

(

[PostTime],

[DatabaseUser],

[Event],

[Schema],

[Object],

[TSQL],

[XmlEvent]

)

VALUES

(

GETDATE(),

CONVERT(sysname, CURRENT\_USER),

@eventType,

CONVERT(sysname, @schema),

CONVERT(sysname, @object),

@data.value('(/EVENT\_INSTANCE/TSQLCommand)[1]', 'nvarchar(max)'),

@data

);

END;

GO

[TOP](#Top)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Create Error Log objects

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PRINT '';

PRINT '\*\*\* Creating Error Log objects';

GO

-- Create table to store error information

**CREATE TABLE** [dbo].[ErrorLog](

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

[ErrorTime] [datetime] NOT NULL CONSTRAINT [DF\_ErrorLog\_ErrorTime] DEFAULT (GETDATE()),

[UserName] [sysname] NOT NULL,

[ErrorNumber] [int] NOT NULL,

[ErrorSeverity] [int] NULL,

[ErrorState] [int] NULL,

[ErrorProcedure] [nvarchar](126) NULL,

[ErrorLine] [int] NULL,

[ErrorMessage] [nvarchar](4000) NOT NULL

) ON [PRIMARY];

GO

**ALTER TABLE** [dbo].[ErrorLog] WITH CHECK

ADD CONSTRAINT [PK\_ErrorLog\_ErrorLogID] PRIMARY KEY CLUSTERED

(

[ErrorLogID]

) ON [PRIMARY];

When you create a database in Microsoft SQL Server you can have multiple file groups, where storage is created in multiple places, directories or disks. Each file group can be named. The PRIMARY file group is the default one, which is always created, and so the SQL you've given creates your table ON the PRIMARY file group.

GO

-- uspPrintError prints error information about the error that caused

-- execution to jump to the CATCH block of a TRY...CATCH construct.

-- Should be executed from within the scope of a CATCH block otherwise

-- it will return without printing any error information.

**CREATE PROCEDURE** [dbo].[uspPrintError]

AS

BEGIN

SET NOCOUNT ON;

-- Print error information.

PRINT 'Error ' + CONVERT(varchar(50), ERROR\_NUMBER()) +

', Severity ' + CONVERT(varchar(5), ERROR\_SEVERITY()) +

', State ' + CONVERT(varchar(5), ERROR\_STATE()) +

', Procedure ' + ISNULL(ERROR\_PROCEDURE(), '-') +

', Line ' + CONVERT(varchar(5), ERROR\_LINE());

PRINT ERROR\_MESSAGE();

END;

GO

-- uspLogError logs error information in the ErrorLog table about the

-- error that caused execution to jump to the CATCH block of a

-- TRY...CATCH construct. This should be executed from within the scope

-- of a CATCH block otherwise it will return without inserting error

-- information.

**CREATE PROCEDURE** [dbo].[uspLogError]

@ErrorLogID [int] = 0 OUTPUT -- contains the ErrorLogID of the row inserted

AS -- by uspLogError in the ErrorLog table

BEGIN

SET NOCOUNT ON;

-- Output parameter value of 0 indicates that error

-- information was not logged

SET @ErrorLogID = 0;

BEGIN TRY

-- Return if there is no error information to log

IF ERROR\_NUMBER() IS NULL

RETURN;

-- Return if inside an uncommittable transaction.

-- Data insertion/modification is not allowed when

-- a transaction is in an uncommittable state.

IF XACT\_STATE() = -1

BEGIN

PRINT 'Cannot log error since the current transaction is in an uncommittable state. '

+ 'Rollback the transaction before executing uspLogError in order to successfully log error information.';

RETURN;

END

INSERT [dbo].[ErrorLog]

(

[UserName],

[ErrorNumber],

[ErrorSeverity],

[ErrorState],

[ErrorProcedure],

[ErrorLine],

[ErrorMessage]

)

VALUES

(

CONVERT(sysname, CURRENT\_USER),

ERROR\_NUMBER(),

ERROR\_SEVERITY(),

ERROR\_STATE(),

ERROR\_PROCEDURE(),

ERROR\_LINE(),

ERROR\_MESSAGE()

);

-- Pass back the ErrorLogID of the row inserted

SET @ErrorLogID = @@IDENTITY;

END TRY

BEGIN CATCH

PRINT 'An error occurred in stored procedure uspLogError: ';

EXECUTE [dbo].[uspPrintError];

RETURN -1;

END CATCH

END;

GO

[TOP](#Top)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Create Data Types

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. **user-defined data types** (HIGHEST IN ORDER OF PRECEDENCE)
2. **sql\_variant**
3. **xml**
4. **datetimeoffset**
5. **datetime2**
6. **datetime**
7. **smalldatetime**
8. **date**
9. **time**
10. **float**
11. **real**
12. **decimal**
13. **money**
14. **smallmoney**
15. **bigint**
16. **int**
17. **smallint**
18. **tinyint**
19. **bit**
20. **ntext**
21. **text**
22. **image**
23. **timestamp**
24. **uniqueidentifier**
25. **nvarchar** (including **nvarchar(max)** )
26. **nchar**
27. **varchar** (including **varchar(max)** )
28. **char**
29. **varbinary** (including **varbinary(max)** )
30. **binary** (lowest)

PRINT '';

PRINT '\*\*\* Creating Data Types';

GO

CREATE TYPE [AccountNumber] FROM nvarchar(15) NULL;

CREATE TYPE [Flag] FROM bit NOT NULL;

CREATE TYPE [NameStyle] FROM bit NOT NULL;

CREATE TYPE [Name] FROM nvarchar(50) NULL;

CREATE TYPE [OrderNumber] FROM nvarchar(25) NULL;

CREATE TYPE [Phone] FROM nvarchar(25) NULL;

GO

[TOP](#Top)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Add pre-table database functions.

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Built-In functions are used in SQL SELECT expressions to calculate values and manipulate data.  These functions can be used anywhere expressions are allowed.  Common uses of functions include to change a name to all upper case

In SQL a built-in function is a piece for programming that takes zero or more inputs and returns a value.  An example of a built-in functions is ABS(), which when given a value calculates the absolute (non-negative) value of the number.

There are several things to note regarding function:

1. The inputs to a function are called parameters.  Not all function have parameters, and some functions have more than one.
2. Parameters are enclosed in parenthesis.
3. We use functions in the SELECT clause as well as WHERE filter condition. A function can be used anywhere in a SELECT statement that you can use an expression.
4. Function are reserved words. I would avoid using them as column or table names.  If you do, then expect to qualify your names with brackets [].

Most folk’s first impression is that functions are used to return numeric values. Sure, they are used for these, but functions can return many other [**DATA TYPES**](http://www.essentialsql.com/commonly-used-sql-server-data-types/) as well. As you’ll see, functions are used to manipulate DATETIME, VARCHAR, and REAL types.

In SQL server the built-in functions return one value.  These are called **SCALAR** functions, and are used wherever expressions are allowed.

Scalar is just a fancy work for “single value.” You will also learn about functions that can return table rows, these functions are called table value functions.  You’ll use these later on when you create your own [**USER**](https://msdn.microsoft.com/en-us/library/ms191320%28v=sql.120%29.aspx) **DEFINED FUNCTIONS**.

Function Categories

There are over a hundred built-in functions in SQL server.  To understand their breadth of application, I would recommend visiting the [Built-In Functions (Transact SQL)](https://msdn.microsoft.com/en-us/library/ms174318.aspx) page on the MDSN site.  You find the functions are categorized into major categories.  The categories we’ll cover next are:

* **Conversion Functions** – [Convert data with **CAST** and **CONVERT**](http://www.essentialsql.com/datatype-conversion-in-sql-using-cast-and-convert/) between types.
* **Logical Functions** – [execute one expression versus another depending on the outcome of a logical comparison](http://www.essentialsql.com/introduction-to-sql-servers-built-in-logical-functions/).
* **Math Functions** – [perform advanced calculations and round numbers](http://www.essentialsql.com/introduction-to-sql-servers-mathematical-functions/).
* **String Functions** –[change text values to all upper case, or remove the trailing spaces from values.](http://www.essentialsql.com/introduction-to-sql-servers-common-string-functions/)
* **Date Functions** – [add days or months to a date. Calculate the day of week from the date.](http://www.essentialsql.com/introduction-to-sql-servers-date-functions/)

You may be wondering why Quantity was converted to Money and not StandardCost to SMALLINT.

The reason is that order values are implicitly converted from one data type to another is determined by [**TYPE**](https://msdn.microsoft.com/en-us/library/ms190309.aspx) **PRECEDENCE**. Data type precedence determines the direction or order implicit datatype occur.  Here is the order of precedence for the [**COMMON**](http://www.essentialsql.com/commonly-used-sql-server-data-types/) **DATA TYPES** we have previously covered:

DATETIME (highest)

FLOAT

Data types of lower precedence will attempt to convert to one of higher precedence, but not the other way around.

DECIMAL

INT

BIT

NVARCHAR

VARCHAR (lowest)

PRINT '';

PRINT '\*\*\* Creating Pre-Table Database Functions';

GO

**CREATE FUNCTION** [dbo].[ufnLeadingZeros] ( @Value int )

RETURNS varchar(8)

WITH SCHEMABINDING

AS

BEGIN

DECLARE @ReturnValue varchar(8); **🡨 If you don’t specify a length it defaults to (1)**

SET @ReturnValue = CONVERT(varchar(8), @Value);

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

RETURN (@ReturnValue);

END;

GO

[TOP](#Top)

**SCHEMABINDING** is an option that is available for objects in T-**SQL** which contain user defined code. Examples include, stored procedures, indexes, and **functions**. Straight from MSDN, **SCHEMABINDING**… Specifies that the schema is bound to the database objects that it references.

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Create **FIVE** SQL Database Schemas

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- These SQL Schemas are only containers for objects such as tables, stored procedures and functions that they contain. They are also a security boundary.

PRINT '';

PRINT '\*\*\* Creating Database Schemas';

GO

CREATE SCHEMA [HumanResources] AUTHORIZATION [dbo];

GO

CREATE SCHEMA [Person] AUTHORIZATION [dbo];

GO

CREATE SCHEMA [Production] AUTHORIZATION [dbo];

GO

CREATE SCHEMA [Purchasing] AUTHORIZATION [dbo];

GO

CREATE SCHEMA [Sales] AUTHORIZATION [dbo];

GO

[TOP](#Top)

Ownership and User-Schema Separation in SQL Server

A core concept of SQL Server security is that owners of objects have irrevocable permissions to administer them. You cannot remove privileges from an object owner, and you cannot drop users from a database if they own objects in it.

User-Schema Separation

User-schema separation allows for more flexibility in managing database object permissions. A *schema* is a named container for database objects, which allows you to group objects into separate namespaces. For example, the AdventureWorks sample database contains schemas for Production, Sales, and HumanResources.

The four-part naming syntax for referring to objects specifies the schema name:

Server . Database . DatabaseSchema. DatabaseObject

1 2 3 4

Schema Owners and Permissions

Schemas can be owned by any database principal, and a single principal can own multiple schemas. You can apply security rules to a schema, which are inherited by all objects in the schema. Once you set up access permissions for a schema, those permissions are automatically applied as new objects are added to the schema. Users can be assigned a default schema, and multiple database users can share the same schema.

By default, when developers create objects in a schema, the objects are owned by the security principal that owns the schema, not the developer. Object ownership can be transferred with ALTER AUTHORIZATION Transact-SQL statement. A schema can also contain objects that are owned by different users and have more granular permissions than those assigned to the schema, although this is not recommended because it adds complexity to managing permissions. Objects can be moved between schemas, and schema ownership can be transferred between principals. Database users can be dropped without affecting schemas.+

Built-In Schemas

SQL Server ships with ten pre-defined schemas that have the same names as the built-in database users and roles. These exist mainly for backward compatibility. You can drop the schemas that have the same names as the fixed database roles if you do not need them. You cannot drop the following schemas:

* dbo
* guest
* sys
* INFORMATION\_SCHEMA

The dbo Schema

The dbo schema is the default schema for a newly created database. The dboschema is owned by the dbo user account. By default, users created with the CREATE USER Transact-SQL command have dbo as their default schema.

Users who are assigned the dbo schema do not inherit the permissions of the dbouser account. No permissions are inherited from a schema by users; schema permissions are inherited by the database objects contained in the schema.

Note

When database objects are referenced by using a one-part name, SQL Server first looks in the user's default schema. If the object is not found there, SQL Server looks next in the dbo schema. If the object is not in the dbo schema, an error is returned.

[TOP](#Top)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Create **SIX** XML Schemas

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- These SIX XML Schemas are IN ADDITION to the above FIVE SQL Schemas and are meant to store XML Data. Typically they would be used for inter-database, inter-file, SQL to XML or XML to SQL work. XML data can be stored and queried within SQL Server. You can return existing relational data as XML, can query data that is already XML and shred XML data into a relational format.

-- These Schemas are also here for the XML data in this database. They are used to validate and enforce the structure of the XML. It is important not to overuse XML within an SQL system. It is there for a minority role, to process XML data that is either INSIDE the SQL system or interplays with the SQL system from another system.

-- For the SQL Schema, it’s just a container for tables, functions, stored procedures, etc. Definitions and constraints take place in the definitions of those objects it contains.It’s also a security boundary. Here, however, the definitions take place in the XML Schema itself.

-- XML is Case Sensitive.

-- Well-formed XML has only one top-level element and, when it does, it’s called a “Document”. In other words, the Document has to have a single root element. Documents that have multiple top-level elements are fragments.

“?xml” is a Processing Instruction – These instructions are not a part of the data, but determine the details of encoding.

PRINT '';

PRINT '\*\*\* Creating XML Schemas';

GO

-- Create AdditionalContactInfo schema

PRINT '';

The Object

The

PRINT 'Create AdditionalContactInfo schema';

The Database Schema

GO

**CREATE** XML SCHEMA COLLECTION [Person].[AdditionalContactInfoSchemaCollection] AS

'<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema targetNamespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo"

xmlns="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo"

URIs are used as unique identifiers for a namespace. They do not have to lead to any real resources at all.

elementFormDefault="qualified"

xmlns:xsd="http://www.w3.org/2001/XMLSchema" >

<!-- the following imports are not needed. They simply provide readability -->

“xmlns” defines an XML namespace

<xsd:import

namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactRecord" />

Alias for the namespace (a prefix)

<xsd:import

namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes" />

The any element enables the author to extend the XML document with elements not specified by the schema. Parent elements must be either “choice” or “sequence”

<xsd:element name="AdditionalContactInfo" >

<xsd:complexType mixed="true" >

<xsd:sequence>

<xsd:any processContents="strict"

namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactRecord

--------------------------------------------------------Start of ContactRecord.xsd--------------------------------------------------------------------------------

<?xml version="1.0" encoding="UTF-8"?>

[<xsd:schema xmlns="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactRecord" xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" targetNamespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-orks/ContactRecord">](%3cxsd:schema%20xmlns=%22http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactRecord%22%20xmlns:xsd=%22http://www.w3.org/2001/XMLSchema%22%20elementFormDefault=%22qualified%22%20targetNamespace=%22http://schemas.microsoft.com/sqlserver/2004/07/adventure-orks/ContactRecord%22%3e)

[<xsd:annotation>](http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactRecord/ContactRecord.xsd)

<xsd:documentation> (c) 2008 Microsoft Corporation. All rights reserved. The following schema for Microsoft SQL Server is presented in XML format and is for informational purposes only. Microsoft Corporation ("Microsoft") may have trademarks, copyrights, or other intellectual property rights covering subject matter in the schema. Microsoft does not make any representation or warranty regarding the schema or any product or item developed based on the schema. The schema is provided to you on an AS IS basis. Microsoft disclaims all express, implied and statutory warranties, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and freedom from infringement. Without limiting the generality of the foregoing, Microsoft does not make any warranty of any kind that any item developed based on the schema, or any portion of the schema, will not infringe any copyright, patent, trade secret, or other intellectual property right of any person or entity in any country. It is your responsibility to seek licenses for such intellectual property rights where appropriate. MICROSOFT SHALL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE SCHEMA, INCLUDING WITHOUT LIMITATION, ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL (INCLUDING ANY LOST PROFITS), PUNITIVE OR SPECIAL DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN ADVISED OF SUCH DAMAGES. </xsd:documentation>

</xsd:annotation>

[<xsd:element name="**ContactRecord**">](http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactRecord/ContactRecord.xsd)

[<xsd:complexType mixed="**true**">](http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactRecord/ContactRecord.xsd)

[<xsd:choice maxOccurs="**unbounded**" minOccurs="**0**">](http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactRecord/ContactRecord.xsd)

<xsd:any namespace="**http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes**" processContents="**strict**"/>

</xsd:choice>

<xsd:attribute type="**xsd:date**" name="**date**"/>

</xsd:complexType>

</xsd:element>

</xsd:schema>

--------------------------------------------------------end of ContactRecord.xsd---------------------------------------------------------------------------------

<http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes>"

--------------------------------------------------------Start of ContactTypes.xsd--------------------------------------------------------------------------------

<?xml version="1.0" encoding="UTF-8"?>

[<xsd:schema xmlns="**http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes**" xmlns:xsd="**http://www.w3.org/2001/XMLSchema**"](http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes/ContactTypes.xsd)  elementFormDefault="qualified"

targetNamespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes">

<xsd:annotation>

<xsd:documentation> **(c) 2008 Microsoft Corporation. All rights reserved. The following schema for Microsoft SQL Server is presented in XML format and is for informational purposes only. Microsoft Corporation ("Microsoft") may have trademarks, copyrights, or other intellectual property rights covering subject matter in the schema. Microsoft does not make any representation or warranty regarding the schema or any product or item developed based on the schema. The schema is provided to you on an AS IS basis. Microsoft disclaims all express, implied and statutory warranties, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and freedom from infringement. Without limiting the generality of the foregoing, Microsoft does not make any warranty of any kind that any item developed based on the schema, or any portion of the schema, will not infringe any copyright, patent, trade secret, or other intellectual property right of any person or entity in any country. It is your responsibility to seek licenses for such intellectual property rights where appropriate. MICROSOFT SHALL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE SCHEMA, INCLUDING WITHOUT LIMITATION, ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL (INCLUDING ANY LOST PROFITS), PUNITIVE OR SPECIAL DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN ADVISED OF SUCH DAMAGES.** </xsd:documentation>

</xsd:annotation>

<xsd:complexType mixed="true" name="specialInstructionsType">

<xsd:sequence>

<xsd:any maxOccurs="unbounded" minOccurs="0" namespace="##targetNamespace" processContents="strict"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="phoneNumberType">

<xsd:sequence>

<xsd:element name="number">

<xsd:simpleType>

<xsd:restriction base="xsd:string">

<xsd:pattern value="[0-9\(\)\-]\*"/>

</xsd:restriction>

</xsd:simpleType>

</xsd:element>

<xsd:element type="specialInstructionsType" name="SpecialInstructions" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="eMailType">

<xsd:sequence>

<xsd:element type="xsd:string" name="eMailAddress"/>

<xsd:element type="specialInstructionsType" name="SpecialInstructions" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="addressType">

<xsd:sequence>

<xsd:element type="xsd:string" name="Street" maxOccurs="2" minOccurs="1"/>

<xsd:element type="xsd:string" name="City" maxOccurs="1" minOccurs="1"/>

<xsd:element type="xsd:string" name="StateProvince" maxOccurs="1" minOccurs="1"/>

<xsd:element type="xsd:string" name="PostalCode" maxOccurs="1" minOccurs="0"/>

<xsd:element type="xsd:string" name="CountryRegion" maxOccurs="1" minOccurs="1"/>

<xsd:element type="specialInstructionsType" name="SpecialInstructions" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

<xsd:element type="phoneNumberType" name="telephoneNumber"/>

<xsd:element type="phoneNumberType" name="mobile"/>

<xsd:element type="phoneNumberType" name="pager"/>

<xsd:element type="phoneNumberType" name="facsimileTelephoneNumber"/>

<xsd:element type="phoneNumberType" name="telexNumber"/>

<xsd:element type="phoneNumberType" name="internationaliSDNNumber"/>

<xsd:element type="eMailType" name="eMail"/>

<xsd:element type="addressType" name="homePostalAddress"/>

<xsd:element type="addressType" name="physicalDeliveryOfficeName"/>

<xsd:element type="addressType" name="registeredAddress"/>

</xsd:schema>

-------------------------------------end of ContactTypes.xs--------------------------------------------

minOccurs="0" maxOccurs="unbounded" />

</xsd:sequence>

</xsd:complexType>

An XML schema defines:

* Elements that can or must appear in a document
* Atributes that can or must appear in a document
* Which elements are child elements
* The order of child elements
* The number of child elements
* Whether an element is empty or can include text
* Data types for elements and attributes
* Default and fixed values for elements and attributes

</xsd:element>

</xsd:schema>';

GO

[TOP](#Top)

**ALTER** XML SCHEMA COLLECTION [Person].[AdditionalContactInfoSchemaCollection] ADD

'<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema targetNamespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactRecord"

elementFormDefault="qualified"

xmlns="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactRecord"

xmlns:xsd="http://www.w3.org/2001/XMLSchema" >

Name of the Element

<xsd:element name="ContactRecord" >

<xsd:complexType mixed="true" >

<xsd:choice minOccurs="0" maxOccurs="unbounded" >

<xsd:any processContents="strict"

namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes" />

</xsd:choice>

<xsd:attribute name="date" type="xsd:date" />

</xsd:complexType>

</xsd:element>

</xsd:schema>';

GO

[TOP](#Top)

## -- XML Namespaces - The xmlns Attribute

-- When using prefixes in XML, a **namespace** for the prefix must be defined.

-- The namespace can be defined by an **xmlns** attribute in the start tag of an element.

-- The namespace declaration has the following syntax. xmlns:*prefix*="*URI*".

**ALTER** XML SCHEMA COLLECTION [Person].[AdditionalContactInfoSchemaCollection] ADD

'<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema targetNamespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes"

xmlns="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes"

elementFormDefault="qualified"

xmlns:xsd="http://www.w3.org/2001/XMLSchema" >

<xsd:complexType name="specialInstructionsType" mixed="true">

<xsd:sequence>

<xsd:any processContents="strict"

namespace = "##targetNamespace"

Typing throughout this alter. First the type definitions and then the type usages below.

minOccurs="0" maxOccurs="unbounded" />

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="phoneNumberType">

<xsd:sequence>

<xsd:element name="number" >

<xsd:simpleType>

<xsd:restriction base="xsd:string">

<xsd:pattern value="[0-9\(\)\-]\*"/>

</xsd:restriction>

</xsd:simpleType>

</xsd:element>

<xsd:element name="SpecialInstructions" minOccurs="0" type="specialInstructionsType" />

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="eMailType">

<xsd:sequence>

<xsd:element name="eMailAddress" type="xsd:string" />

<xsd:element name="SpecialInstructions" minOccurs="0" type="specialInstructionsType" />

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="addressType">

<xsd:sequence>

<xsd:element name="Street" type="xsd:string" minOccurs="1" maxOccurs="2" />

<xsd:element name="City" type="xsd:string" minOccurs="1" maxOccurs="1" />

<xsd:element name="StateProvince" type="xsd:string" minOccurs="1" maxOccurs="1" />

<xsd:element name="PostalCode" type="xsd:string" minOccurs="0" maxOccurs="1" />

<xsd:element name="CountryRegion" type="xsd:string" minOccurs="1" maxOccurs="1" />

<xsd:element name="SpecialInstructions" type="specialInstructionsType" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

<xsd:element name="telephoneNumber" type="phoneNumberType" />

<xsd:element name="mobile" type="phoneNumberType" />

<xsd:element name="pager" type="phoneNumberType" />

<xsd:element name="facsimileTelephoneNumber" type="phoneNumberType" />

<xsd:element name="telexNumber" type="phoneNumberType" />

<xsd:element name="internationaliSDNNumber" type="phoneNumberType" />

<xsd:element name="eMail" type="eMailType" />

<xsd:element name="homePostalAddress" type="addressType" />

<xsd:element name="physicalDeliveryOfficeName" type="addressType" />

<xsd:element name="registeredAddress" type="addressType" />

</xsd:schema>';

GO

[**TOP**](#Top)

-- Create Individual survey schema.

PRINT '';

PRINT 'Create Individual survey schema';

GO

Syntax

CREATE XML SCHEMA COLLECTION [ <relational\_schema>. ]sql\_identifier AS Expression

Arguments

1. *relational\_schema* (See “Person” Below)Identifies the relational schema name. If not specified, default relational schema is assumed. +
2. *sql\_identifier* (See “IndividualSurveySchemaCollection” Below)Is the SQL identifier for the XML schema collection. +
3. *Expression*Is a string constant or scalar variable. Is **varchar**, **varbinary**, **nvarchar**, or **xml** type.

**CREATE** XML SCHEMA COLLECTION [Person].[IndividualSurveySchemaCollection] AS

'<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema targetNamespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey"

xmlns="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey"

elementFormDefault="qualified"

xmlns:xsd="http://www.w3.org/2001/XMLSchema" >

<xsd:simpleType name="SalaryType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="0-25000" />

<xsd:enumeration value="25001-50000" />

<xsd:enumeration value="50001-75000" />

<xsd:enumeration value="75001-100000" />

<xsd:enumeration value="greater than 100000" />

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="MileRangeType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="0-1 Miles" />

<xsd:enumeration value="1-2 Miles" />

<xsd:enumeration value="2-5 Miles" />

<xsd:enumeration value="5-10 Miles" />

<xsd:enumeration value="10+ Miles" />

</xsd:restriction>

</xsd:simpleType>

<xsd:element name="IndividualSurvey">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="TotalPurchaseYTD" type="xsd:decimal" minOccurs="0" maxOccurs="1" />

<xsd:element name="DateFirstPurchase" type="xsd:date" minOccurs="0" maxOccurs="1" />

<xsd:element name="BirthDate" type="xsd:date" minOccurs="0" maxOccurs="1" />

<xsd:element name="MaritalStatus" type="xsd:string" minOccurs="0" maxOccurs="1" />

<xsd:element name="YearlyIncome" type="SalaryType" minOccurs="0" maxOccurs="1" />

<xsd:element name="Gender" type="xsd:string" minOccurs="0" maxOccurs="1" />

<xsd:element name="TotalChildren" type="xsd:int" minOccurs="0" maxOccurs="1" />

<xsd:element name="NumberChildrenAtHome" type="xsd:int" minOccurs="0" maxOccurs="1" />

<xsd:element name="Education" type="xsd:string" minOccurs="0" maxOccurs="1" />

<xsd:element name="Occupation" type="xsd:string" minOccurs="0" maxOccurs="1" />

<xsd:element name="HomeOwnerFlag" type="xsd:string" minOccurs="0" maxOccurs="1" />

<xsd:element name="NumberCarsOwned" type="xsd:int" minOccurs="0" maxOccurs="1" />

<xsd:element name="Hobby" type="xsd:string" minOccurs="0" maxOccurs="unbounded" />

<xsd:element name="CommuteDistance" type="MileRangeType" minOccurs="0" maxOccurs="1" />

<xsd:element name="Comments" type="xsd:string" minOccurs="0" maxOccurs="1" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>';

GO

[**TOP**](#Top)

-- Create resume schema.

PRINT '';

PRINT 'Create Resume schema';

GO

**CREATE** XML SCHEMA COLLECTION [HumanResources].[HRResumeSchemaCollection] AS

'<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema targetNamespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume"

## -- XML Namespaces - The xmlns Attribute

-- When using prefixes in XML, a **namespace** for the prefix must be defined.

-- The namespace can be defined by an **xmlns** attribute in the start tag of an element.

-- The namespace declaration has the following syntax. xmlns:*prefix*="*URI*".

xmlns="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume"

xmlns:xsd="http://www.w3.org/2001/XMLSchema"

elementFormDefault="qualified" >

**XSD** Simple **Elements**. ❮ Previous Next ❯ **XML Schemas Define** the **elements** of your XML files. A simple **element** is an XML **element** that contains only text. It cannot contain any other **elements** or attributes.

<xsd:element name="Resume" type="ResumeType"/>

<xsd:element name="Address" type="AddressType"/>

<xsd:element name="Education" type="EducationType"/>

<xsd:element name="Employment" type="EmploymentType"/>

<xsd:element name="Location" type="LocationType"/>

<xsd:element name="Name" type="NameType"/>

<xsd:element name="Telephone" type="TelephoneType"/>

A **complex type** is essentially a type definition for elements that may contain attributes and elements. An element can be declared with a type attribute that refers to a **complexType** element that defines the structure, content, and attributes of that element.

<xsd:complexType name="ResumeType">

**xsd**:all - "child elements can appear in any order and each child element can occur zero or one time" (ie, maxOccurs can be 0 or 1) **xsd**:**sequence** - "child elements must appear in a **sequence**. Each child element can occur from 0 to any number of times" (ie, maxOccurs can be 0 or any number or 'unbounded')

<xsd:sequence>

<xsd:element ref="Name"/>

<xsd:element name="Skills" type="xsd:string" minOccurs="0"/>

<xsd:element ref="Employment" maxOccurs="unbounded"/>

<xsd:element ref="Education" maxOccurs="unbounded"/>

<xsd:element ref="Address" maxOccurs="unbounded"/>

<xsd:element ref="Telephone" minOccurs="0"/>

<xsd:element name="EMail" type="xsd:string" minOccurs="0"/>

<xsd:element name="WebSite" type="xsd:string" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="AddressType">

<xsd:sequence>

<xsd:element name="Addr.Type" type="xsd:string">

The **annotation** element is a top level element that specifies **schema** comments. The comments serve as inline documentation.

<xsd:annotation>

<xsd:documentation>Home|Work|Permanent</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="Addr.OrgName" type="xsd:string" minOccurs="0"/>

<xsd:element name="Addr.Street" type="xsd:string" maxOccurs="unbounded"/>

<xsd:element name="Addr.Location">

<xsd:complexType>

<xsd:sequence>

<xsd:element ref="Location"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="Addr.PostalCode" type="xsd:string"/>

<xsd:element name="Addr.Telephone" minOccurs="0">

<xsd:complexType>

<xsd:sequence>

<xsd:element ref="Telephone" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="EducationType">

<xsd:sequence>

<xsd:element name="Edu.Level" type="xsd:string">

<xsd:annotation>

<xsd:documentation>High School|Associate|Bachelor|Master|Doctorate</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="Edu.StartDate" type="xsd:date"/>

<xsd:element name="Edu.EndDate" type="xsd:date"/>

<xsd:element name="Edu.Degree" type="xsd:string" minOccurs="0"/>

<xsd:element name="Edu.Major" type="xsd:string" minOccurs="0"/>

<xsd:element name="Edu.Minor" type="xsd:string" minOccurs="0"/>

<xsd:element name="Edu.GPA" type="xsd:string" minOccurs="0"/>

<xsd:element name="Edu.GPAAlternate" type="xsd:decimal" minOccurs="0">

<xsd:annotation>

<xsd:documentation>In case the institution does not follow a GPA system</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="Edu.GPAScale" type="xsd:decimal" minOccurs="0"/>

<xsd:element name="Edu.School" type="xsd:string" minOccurs="0"/>

<xsd:element name="Edu.Location" minOccurs="0">

<xsd:complexType>

<xsd:sequence>

<xsd:element ref="Location"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="EmploymentType">

<xsd:sequence>

<xsd:element name="Emp.StartDate" type="xsd:date" minOccurs="0"/>

<xsd:element name="Emp.EndDate" type="xsd:date" minOccurs="0"/>

<xsd:element name="Emp.OrgName" type="xsd:string"/>

<xsd:element name="Emp.JobTitle" type="xsd:string"/>

<xsd:element name="Emp.Responsibility" type="xsd:string"/>

<xsd:element name="Emp.FunctionCategory" type="xsd:string" minOccurs="0"/>

<xsd:element name="Emp.IndustryCategory" type="xsd:string" minOccurs="0"/>

<xsd:element name="Emp.Location" minOccurs="0">

<xsd:complexType>

<xsd:sequence>

<xsd:element ref="Location"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="LocationType">

<xsd:sequence>

<xsd:element name="Loc.CountryRegion" type="xsd:string">

<xsd:annotation>

<xsd:documentation>ISO 3166 Country Code</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="Loc.State" type="xsd:string" minOccurs="0"/>

<xsd:element name="Loc.City" type="xsd:string" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="NameType">

<xsd:sequence>

<xsd:element name="Name.Prefix" type="xsd:string" minOccurs="0"/>

<xsd:element name="Name.First" type="xsd:string"/>

<xsd:element name="Name.Middle" type="xsd:string" minOccurs="0"/>

<xsd:element name="Name.Last" type="xsd:string"/>

<xsd:element name="Name.Suffix" type="xsd:string" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="TelephoneType">

<xsd:sequence>

<xsd:element name="Tel.Type" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Voice|Fax|Pager</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="Tel.IntlCode" type="xsd:int" minOccurs="0"/>

<xsd:element name="Tel.AreaCode" type="xsd:int" minOccurs="0"/>

<xsd:element name="Tel.Number" type="xsd:string"/>

<xsd:element name="Tel.Extension" type="xsd:int" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

</xsd:schema>';

GO

[**TOP**](#Top)

-- Create Product catalog description schema.

PRINT '';

PRINT 'Create Product catalog description schema';

GO

The XML representation of schema components uses a vocabulary identified by the namespace name <http://www.w3.org/2001/XMLSchema>. For brevity, the text and examples in this specification use the prefix xs: to stand for this namespace; in practice, any prefix can be used. In the end **xs** or **xsd** are only prefixes. XSD is used for example more by Microsoft schemas. By convention people tend to choose either xs: or xsd: and map that to http://www.w3.org/2001/XMLSchema. Having both in a single document is confusing and should be avoided.

**CREATE** XML SCHEMA COLLECTION [Production].[ProductDescriptionSchemaCollection] AS

'<xsd:schema targetNamespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelWarrAndMain"

xmlns="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelWarrAndMain"

elementFormDefault="qualified"

xmlns:xsd="http://www.w3.org/2001/XMLSchema" >

<xsd:element name="Warranty" >

<xsd:complexType>

<xsd:sequence>

<xsd:element name="WarrantyPeriod" type="xsd:string" />

<xsd:element name="Description" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="Maintenance" >

<xsd:complexType>

<xsd:sequence>

<xsd:element name="NoOfYears" type="xsd:string" />

<xsd:element name="Description" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>';

[**TOP**](#Top)

**ALTER** XML SCHEMA COLLECTION [Production].[ProductDescriptionSchemaCollection] ADD

'<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure- works/ProductModelDescription"

xmlns="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription"

elementFormDefault="qualified"

xmlns:mstns="http://tempuri.org/XMLSchema.xsd"

xmlns:xs="http://www.w3.org/2001/XMLSchema"

xmlns:wm="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelWarrAndMain" >

<xs:import

namespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelWarrAndMain" />

An **element** declaration is an association of a name with a type definition, either simple or complex, an (optional) default value and a (possibly empty) set of identity-constraint definitions.

<xs:element name="ProductDescription" type="ProductDescription" />

<xs:complexType name="ProductDescription">

<xs:annotation>

<xs:documentation>Product description has a summary blurb, if its manufactured elsewhere it

includes a link to the manufacturers site for this component.

Then it has optional zero or more sequences of features, pictures, categories

and technical specifications.

</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="Summary" type="Summary" minOccurs="0" />

<xs:element name="Manufacturer" type="Manufacturer" minOccurs="0" />

<xs:element name="Features" type="Features" minOccurs="0" maxOccurs="unbounded" />

<xs:element name="Picture" type="Picture" minOccurs="0" maxOccurs="unbounded" />

<xs:element name="Category" type="Category" minOccurs="0" maxOccurs="unbounded" />

<xs:element name="Specifications" type="Specifications" minOccurs="0" maxOccurs="unbounded" />

</xs:sequence>

<xs:attribute name="ProductModelID" type="xs:string" />

<xs:attribute name="ProductModelName" type="xs:string" />

</xs:complexType>

<xs:complexType name="Summary" mixed="true" >

<xs:sequence>

<xs:any processContents="skip" namespace="http://www.w3.org/1999/xhtml" minOccurs="0" maxOccurs="unbounded" />

</xs:sequence>

</xs:complexType>

<xs:complexType name="Manufacturer">

<xs:sequence>

<xs:element name="Name" type="xs:string" minOccurs="0" />

<xs:element name="CopyrightURL" type="xs:string" minOccurs="0" />

<xs:element name="Copyright" type="xs:string" minOccurs="0" />

<xs:element name="ProductURL" type="xs:string" minOccurs="0" />

</xs:sequence>

</xs:complexType>

<xs:complexType name="Picture">

<xs:annotation>

<xs:documentation>Pictures of the component, some standard sizes are "Large" for zoom in, "Small" for a normal web page and "Thumbnail" for product listing pages.

</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="Name" type="xs:string" minOccurs="0" />

<xs:element name="Angle" type="xs:string" minOccurs="0" />

<xs:element name="Size" type="xs:string" minOccurs="0" />

<xs:element name="ProductPhotoID" type="xs:integer" minOccurs="0" />

</xs:sequence>

</xs:complexType>

<xs:annotation>

<xs:documentation>Features of the component that are more "sales" oriented.</xs:documentation>

</xs:annotation>

<xs:complexType name="Features" mixed="true" >

<xs:sequence>

<xs:element ref="wm:Warranty" />

<xs:element ref="wm:Maintenance" />

<xs:any processContents="skip" namespace="##other" minOccurs="0" maxOccurs="unbounded" />

</xs:sequence>

</xs:complexType>

<xs:complexType name="Specifications" mixed="true">

<xs:annotation>

<xs:documentation>A single technical aspect of the component.</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:any processContents="skip" minOccurs="0" maxOccurs="unbounded" />

</xs:sequence>

</xs:complexType>

<xs:complexType name="Category">

<xs:annotation>

<xs:documentation>A single categorization element that designates a classification taxonomy and a code within that classification type. Optional description for default display if needed.

</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element ref="Taxonomy" />

<xs:element ref="Code" />

<xs:element ref="Description" minOccurs="0" />

</xs:sequence>

</xs:complexType>

<xs:element name="Taxonomy" type="xs:string" />

<xs:element name="Code" type="xs:string" />

<xs:element name="Description" type="xs:string" />

</xs:schema>';

GO

-- Create Manufacturing instructions schema.

PRINT '';

PRINT 'Create Manufacturing instructions schema';

GO

**CREATE** XML SCHEMA COLLECTION [Production].[ManuInstructionsSchemaCollection] AS

'<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema targetNamespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelManuInstructions"

xmlns="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelManuInstructions"

elementFormDefault="qualified" attributeFormDefault="unqualified"

xmlns:xsd="http://www.w3.org/2001/XMLSchema" >

<xsd:annotation>

<xsd:documentation>

SetupHour is the time it takes to set up the machine.

MachineHour is the time the machine is busy manufcturing

LaborHour is the labor hours in the manu process

LotSize is the minimum quanity manufactured. For example,

no. of frames cut from the sheet metal

</xsd:documentation>

</xsd:annotation>

<xsd:complexType name="StepType" mixed="true" >

<xsd:choice minOccurs="0" maxOccurs="unbounded" >

<xsd:element name="tool" type="xsd:string" />

<xsd:element name="material" type="xsd:string" />

<xsd:element name="blueprint" type="xsd:string" />

<xsd:element name="specs" type="xsd:string" />

<xsd:element name="diag" type="xsd:string" />

</xsd:choice>

</xsd:complexType>

<xsd:element name="root">

<xsd:complexType mixed="true">

<xsd:sequence>

<xsd:element name="Location" minOccurs="1" maxOccurs="unbounded">

<xsd:complexType mixed="true">

<xsd:sequence>

<xsd:element name="step" type="StepType" minOccurs="1" maxOccurs="unbounded" />

</xsd:sequence>

<xsd:attribute name="LocationID" type="xsd:integer" use="required"/>

<xsd:attribute name="SetupHours" type="xsd:decimal" use="optional"/>

<xsd:attribute name="MachineHours" type="xsd:decimal" use="optional"/>

<xsd:attribute name="LaborHours" type="xsd:decimal" use="optional"/>

<xsd:attribute name="LotSize" type="xsd:decimal" use="optional"/>

</xsd:complexType>

</xsd:element>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>';

GO

[**TOP**](#Top)

-- Create Store survey schema.

PRINT '';

PRINT 'Create Store survey schema';

GO

**CREATE** XML SCHEMA COLLECTION [Sales].[StoreSurveySchemaCollection] AS

'<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"

targetNamespace="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey"

xmlns="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey"

elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- BM=Bicycle manu BS=bicyle store OS=online store SGS=sporting goods store D=Discount Store -->

<xsd:simpleType name="BusinessType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="BM" />

<xsd:enumeration value="BS" />

<xsd:enumeration value="D" />

<xsd:enumeration value="OS" />

<xsd:enumeration value="SGS" />

</xsd:restriction>

</xsd:simpleType>

<!-- BMX=BMX Racing -->

<xsd:simpleType name="SpecialtyType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="Family" />

<xsd:enumeration value="Kids" />

<xsd:enumeration value="BMX" />

<xsd:enumeration value="Touring" />

<xsd:enumeration value="Road" />

<xsd:enumeration value="Mountain" />

<xsd:enumeration value="All" />

</xsd:restriction>

</xsd:simpleType>

<!-- AW=AdventureWorks only 2= AdvWorks+1 other brand other brand -->

<xsd:simpleType name="BrandType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="AW" />

<xsd:enumeration value="2" />

<xsd:enumeration value="3" />

<xsd:enumeration value="4+" />

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="InternetType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="56kb" />

<xsd:enumeration value="ISDN" />

<xsd:enumeration value="DSL" />

<xsd:enumeration value="T1" />

<xsd:enumeration value="T2" />

<xsd:enumeration value="T3" />

</xsd:restriction>

</xsd:simpleType>

<xsd:element name="StoreSurvey">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="ContactName" type="xsd:string" minOccurs="0" maxOccurs="1" />

<xsd:element name="JobTitle" type="xsd:string" minOccurs="0" maxOccurs="1" />

<xsd:element name="AnnualSales" type="xsd:decimal" minOccurs="0" maxOccurs="1" />

<xsd:element name="AnnualRevenue" type="xsd:decimal" minOccurs="0" maxOccurs="1" />

<xsd:element name="BankName" type="xsd:string" minOccurs="0" maxOccurs="1" />

<xsd:element name="BusinessType" type="BusinessType" minOccurs="0" maxOccurs="1" />

<xsd:element name="YearOpened" type="xsd:gYear" minOccurs="0" maxOccurs="1" />

<xsd:element name="Specialty" type="SpecialtyType" minOccurs="0" maxOccurs="1" />

<xsd:element name="SquareFeet" type="xsd:float" minOccurs="0" maxOccurs="1" />

<xsd:element name="Brands" type="BrandType" minOccurs="0" maxOccurs="1" />

<xsd:element name="Internet" type="InternetType" minOccurs="0" maxOccurs="1" />

<xsd:element name="NumberEmployees" type="xsd:int" minOccurs="0" maxOccurs="1" />

<xsd:element name="Comments" type="xsd:string" minOccurs="0" maxOccurs="1" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>';

GO

[TOP](#Top)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Create tables

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PRINT '';

PRINT '\*\*\* Creating Tables';

GO

**Constraints** are rules and restrictions applied on a column or a table such that unwanted data can't be inserted into tables. This ensures the accuracy and reliability of the data in the database. We can create **constraints** on single or multiple columns of any table.

**CREATE TABLE** [Person].[Address](

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

[AddressLine1] [nvarchar](60) NOT NULL,

[AddressLine2] [nvarchar](60) NULL,

[City] [nvarchar](30) NOT NULL,

[StateProvinceID] [int] NOT NULL,

[PostalCode] [nvarchar](15) NOT NULL,

[SpatialLocation] [geography] NULL,

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_Address\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_Address\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Person].[AddressType](

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

[Name] [Name] NOT NULL,

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_AddressType\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_AddressType\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [dbo].[AWBuildVersion](

[SystemInformationID] [tinyint] IDENTITY (1, 1) NOT NULL,

[Database Version] [nvarchar](25) NOT NULL,

[VersionDate] [datetime] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_AWBuildVersion\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[BillOfMaterials](

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

[ProductAssemblyID] [int] NULL,

[ComponentID] [int] NOT NULL,

[StartDate] [datetime] NOT NULL CONSTRAINT [DF\_BillOfMaterials\_StartDate] DEFAULT (GETDATE()),

[EndDate] [datetime] NULL,

[UnitMeasureCode] [nchar](3) NOT NULL,

[BOMLevel] [smallint] NOT NULL,

[PerAssemblyQty] [decimal](8, 2) NOT NULL CONSTRAINT [DF\_BillOfMaterials\_PerAssemblyQty] DEFAULT (1.00),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_BillOfMaterials\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_BillOfMaterials\_EndDate] CHECK (([EndDate] > [StartDate]) OR ([EndDate] IS NULL)),

CONSTRAINT [CK\_BillOfMaterials\_ProductAssemblyID] CHECK ([ProductAssemblyID] <> [ComponentID]),

CONSTRAINT [CK\_BillOfMaterials\_BOMLevel] CHECK ((([ProductAssemblyID] IS NULL)

AND ([BOMLevel] = 0) AND ([PerAssemblyQty] = 1.00))

OR (([ProductAssemblyID] IS NOT NULL) AND ([BOMLevel] >= 1))),

CONSTRAINT [CK\_BillOfMaterials\_PerAssemblyQty] CHECK ([PerAssemblyQty] >= 1.00)

) ON [PRIMARY];

GO

**CREATE TABLE** [Person].[BusinessEntity](

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

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_BusinessEntity\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_BusinessEntity\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Person].[BusinessEntityAddress](

[BusinessEntityID] [int] NOT NULL,

[AddressID] [int] NOT NULL,

[AddressTypeID] [int] NOT NULL,

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_BusinessEntityAddress\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_BusinessEntityAddress\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

**CREATE TABLE** [Person].[BusinessEntityContact](

[BusinessEntityID] [int] NOT NULL,

[PersonID] [int] NOT NULL,

[ContactTypeID] [int] NOT NULL,

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_BusinessEntityContact\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_BusinessEntityContact\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Person].[ContactType](

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

[Name] [Name] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_ContactType\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Sales].[CountryRegionCurrency](

[CountryRegionCode] [nvarchar](3) NOT NULL,

[CurrencyCode] [nchar](3) NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_CountryRegionCurrency\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Person].[CountryRegion](

[CountryRegionCode] [nvarchar](3) NOT NULL,

[Name] [Name] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_CountryRegion\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Sales].[CreditCard](

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

[CardType] [nvarchar](50) NOT NULL,

[CardNumber] [nvarchar](25) NOT NULL,

[ExpMonth] [tinyint] NOT NULL,

[ExpYear] [smallint] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_CreditCard\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[Culture](

[CultureID] [nchar](6) NOT NULL,

[Name] [Name] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_Culture\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Sales].[Currency](

[CurrencyCode] [nchar](3) NOT NULL,

[Name] [Name] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_Currency\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Sales].[CurrencyRate](

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

[CurrencyRateDate] [datetime] NOT NULL,

[FromCurrencyCode] [nchar](3) NOT NULL,

[ToCurrencyCode] [nchar](3) NOT NULL,

[AverageRate] [money] NOT NULL,

[EndOfDayRate] [money] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_CurrencyRate\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Sales].[Customer](

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

-- A customer may either be a person, a store, or a person who works for a store

[PersonID] [int] NULL, -- If this customer represents a person, this is non-null

[StoreID] [int] NULL, -- If the customer is a store, or is associated with a store then this is non-null.

[TerritoryID] [int] NULL,

[AccountNumber] AS ISNULL('AW' + [dbo].[ufnLeadingZeros](CustomerID), ''),

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_Customer\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_Customer\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [HumanResources].[Department](

[DepartmentID] [smallint] IDENTITY (1, 1) NOT NULL,

[Name] [Name] NOT NULL,

[GroupName] [Name] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_Department\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[Document](

[DocumentNode] [hierarchyid] NOT NULL,

[DocumentLevel] AS DocumentNode.GetLevel(),

[Title] [nvarchar](50) NOT NULL,

[Owner] [int] NOT NULL,

[FolderFlag] [bit] NOT NULL CONSTRAINT [DF\_Document\_FolderFlag] DEFAULT (0),

[FileName] [nvarchar](400) NOT NULL,

[FileExtension] nvarchar(8) NOT NULL,

[Revision] [nchar](5) NOT NULL,

[ChangeNumber] [int] NOT NULL CONSTRAINT [DF\_Document\_ChangeNumber] DEFAULT (0),

[Status] [tinyint] NOT NULL,

[DocumentSummary] [nvarchar](max) NULL,

[Document] [varbinary](max) NULL,

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL UNIQUE CONSTRAINT [DF\_Document\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_Document\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_Document\_Status] CHECK ([Status] BETWEEN 1 AND 3)

) ON [PRIMARY];

GO

**CREATE TABLE** [Person].[EmailAddress](

[BusinessEntityID] [int] NOT NULL,

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

[EmailAddress] [nvarchar](50) NULL,

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_EmailAddress\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_EmailAddress\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [HumanResources].[Employee](

[BusinessEntityID] [int] NOT NULL,

[NationalIDNumber] [nvarchar](15) NOT NULL,

[LoginID] [nvarchar](256) NOT NULL,

[OrganizationNode] [hierarchyid] NULL,

[OrganizationLevel] AS OrganizationNode.GetLevel(),

[JobTitle] [nvarchar](50) NOT NULL,

[BirthDate] [date] NOT NULL,

[MaritalStatus] [nchar](1) NOT NULL,

[Gender] [nchar](1) NOT NULL,

[HireDate] [date] NOT NULL,

[SalariedFlag] [Flag] NOT NULL CONSTRAINT [DF\_Employee\_SalariedFlag] DEFAULT (1),

[VacationHours] [smallint] NOT NULL CONSTRAINT [DF\_Employee\_VacationHours] DEFAULT (0),

[SickLeaveHours] [smallint] NOT NULL CONSTRAINT [DF\_Employee\_SickLeaveHours] DEFAULT (0),

[CurrentFlag] [Flag] NOT NULL CONSTRAINT [DF\_Employee\_CurrentFlag] DEFAULT (1),

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_Employee\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_Employee\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_Employee\_BirthDate] CHECK ([BirthDate] BETWEEN '1930-01-01' AND DATEADD(YEAR, -18, GETDATE())),

CONSTRAINT [CK\_Employee\_MaritalStatus] CHECK (UPPER([MaritalStatus]) IN ('M', 'S')), -- Married or Single

CONSTRAINT [CK\_Employee\_HireDate] CHECK ([HireDate] BETWEEN '1996-07-01' AND DATEADD(DAY, 1, GETDATE())),

CONSTRAINT [CK\_Employee\_Gender] CHECK (UPPER([Gender]) IN ('M', 'F')), -- Male or Female

CONSTRAINT [CK\_Employee\_VacationHours] CHECK ([VacationHours] BETWEEN -40 AND 240),

CONSTRAINT [CK\_Employee\_SickLeaveHours] CHECK ([SickLeaveHours] BETWEEN 0 AND 120)

) ON [PRIMARY];

GO

**CREATE TABLE** [HumanResources].[EmployeeDepartmentHistory](

[BusinessEntityID] [int] NOT NULL,

[DepartmentID] [smallint] NOT NULL,

[ShiftID] [tinyint] NOT NULL,

[StartDate] [date] NOT NULL,

[EndDate] [date] NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_EmployeeDepartmentHistory\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_EmployeeDepartmentHistory\_EndDate] CHECK (([EndDate] >= [StartDate]) OR ([EndDate] IS NULL)),

) ON [PRIMARY];

GO

**CREATE TABLE** [HumanResources].[EmployeePayHistory](

[BusinessEntityID] [int] NOT NULL,

[RateChangeDate] [datetime] NOT NULL,

[Rate] [money] NOT NULL,

[PayFrequency] [tinyint] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_EmployeePayHistory\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_EmployeePayHistory\_PayFrequency] CHECK ([PayFrequency] IN (1, 2)), -- 1 = monthly salary, 2 = biweekly salary

CONSTRAINT [CK\_EmployeePayHistory\_Rate] CHECK ([Rate] BETWEEN 6.50 AND 200.00)

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[Illustration](

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

[Diagram] [XML] NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_Illustration\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [HumanResources].[JobCandidate](

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

[BusinessEntityID] [int] NULL,

[Resume] [XML]([HumanResources].[HRResumeSchemaCollection]) NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_JobCandidate\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[Location](

[LocationID] [smallint] IDENTITY (1, 1) NOT NULL,

[Name] [Name] NOT NULL,

[CostRate] [smallmoney] NOT NULL CONSTRAINT [DF\_Location\_CostRate] DEFAULT (0.00),

[Availability] [decimal](8, 2) NOT NULL CONSTRAINT [DF\_Location\_Availability] DEFAULT (0.00),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_Location\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_Location\_CostRate] CHECK ([CostRate] >= 0.00),

CONSTRAINT [CK\_Location\_Availability] CHECK ([Availability] >= 0.00)

) ON [PRIMARY];

GO

**CREATE TABLE** [Person].[Password](

[BusinessEntityID] [int] NOT NULL,

[PasswordHash] [varchar](128) NOT NULL,

[PasswordSalt] [varchar](10) NOT NULL,

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_Password\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_Password\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Person].[Person](

[BusinessEntityID] [int] NOT NULL,

[PersonType] [nchar](2) NOT NULL,

[NameStyle] [NameStyle] NOT NULL CONSTRAINT [DF\_Person\_NameStyle] DEFAULT (0),

[Title] [nvarchar](8) NULL,

[FirstName] [Name] NOT NULL,

[MiddleName] [Name] NULL,

[LastName] [Name] NOT NULL,

[Suffix] [nvarchar](10) NULL,

[EmailPromotion] [int] NOT NULL CONSTRAINT [DF\_Person\_EmailPromotion] DEFAULT (0),

[AdditionalContactInfo] [XML]([Person].[AdditionalContactInfoSchemaCollection]) NULL,

[Demographics] [XML]([Person].[IndividualSurveySchemaCollection]) NULL,

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_Person\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_Person\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_Person\_EmailPromotion] CHECK ([EmailPromotion] BETWEEN 0 AND 2),

CONSTRAINT [CK\_Person\_PersonType] CHECK ([PersonType] IS NULL OR UPPER([PersonType]) IN ('SC', 'VC', 'IN', 'EM', 'SP', 'GC'))

) ON [PRIMARY];

GO

**CREATE TABLE** [Sales].[PersonCreditCard](

[BusinessEntityID] [int] NOT NULL,

[CreditCardID] [int] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_PersonCreditCard\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Person].[PersonPhone](

[BusinessEntityID] [int] NOT NULL,

[PhoneNumber] [Phone] NOT NULL,

[PhoneNumberTypeID] [int] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_PersonPhone\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Person].[PhoneNumberType](

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

[Name] [Name] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_PhoneNumberType\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[Product](

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

[Name] [Name] NOT NULL,

[ProductNumber] [nvarchar](25) NOT NULL,

[MakeFlag] [Flag] NOT NULL CONSTRAINT [DF\_Product\_MakeFlag] DEFAULT (1),

[FinishedGoodsFlag] [Flag] NOT NULL CONSTRAINT [DF\_Product\_FinishedGoodsFlag] DEFAULT (1),

[Color] [nvarchar](15) NULL,

[SafetyStockLevel] [smallint] NOT NULL,

[ReorderPoint] [smallint] NOT NULL,

[StandardCost] [money] NOT NULL,

[ListPrice] [money] NOT NULL,

[Size] [nvarchar](5) NULL,

[SizeUnitMeasureCode] [nchar](3) NULL,

[WeightUnitMeasureCode] [nchar](3) NULL,

[Weight] [decimal](8, 2) NULL,

[DaysToManufacture] [int] NOT NULL,

[ProductLine] [nchar](2) NULL,

[Class] [nchar](2) NULL,

[Style] [nchar](2) NULL,

[ProductSubcategoryID] [int] NULL,

[ProductModelID] [int] NULL,

[SellStartDate] [datetime] NOT NULL,

[SellEndDate] [datetime] NULL,

[DiscontinuedDate] [datetime] NULL,

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_Product\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_Product\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_Product\_SafetyStockLevel] CHECK ([SafetyStockLevel] > 0),

CONSTRAINT [CK\_Product\_ReorderPoint] CHECK ([ReorderPoint] > 0),

CONSTRAINT [CK\_Product\_StandardCost] CHECK ([StandardCost] >= 0.00),

CONSTRAINT [CK\_Product\_ListPrice] CHECK ([ListPrice] >= 0.00),

CONSTRAINT [CK\_Product\_Weight] CHECK ([Weight] > 0.00),

CONSTRAINT [CK\_Product\_DaysToManufacture] CHECK ([DaysToManufacture] >= 0),

CONSTRAINT [CK\_Product\_ProductLine] CHECK (UPPER([ProductLine]) IN ('S', 'T', 'M', 'R') OR [ProductLine] IS NULL),

CONSTRAINT [CK\_Product\_Class] CHECK (UPPER([Class]) IN ('L', 'M', 'H') OR [Class] IS NULL),

CONSTRAINT [CK\_Product\_Style] CHECK (UPPER([Style]) IN ('W', 'M', 'U') OR [Style] IS NULL),

CONSTRAINT [CK\_Product\_SellEndDate] CHECK (([SellEndDate] >= [SellStartDate]) OR ([SellEndDate] IS NULL)),

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[ProductCategory](

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

[Name] [Name] NOT NULL,

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_ProductCategory\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_ProductCategory\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[ProductCostHistory](

[ProductID] [int] NOT NULL,

[StartDate] [datetime] NOT NULL,

[EndDate] [datetime] NULL,

[StandardCost] [money] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_ProductCostHistory\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_ProductCostHistory\_EndDate] CHECK (([EndDate] >= [StartDate]) OR ([EndDate] IS NULL)),

CONSTRAINT [CK\_ProductCostHistory\_StandardCost] CHECK ([StandardCost] >= 0.00)

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[ProductDescription](

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

[Description] [nvarchar](400) NOT NULL,

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_ProductDescription\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_ProductDescription\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[ProductDocument](

[ProductID] [int] NOT NULL,

[DocumentNode] [hierarchyid] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_ProductDocument\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[ProductInventory](

[ProductID] [int] NOT NULL,

[LocationID] [smallint] NOT NULL,

[Shelf] [nvarchar](10) NOT NULL,

[Bin] [tinyint] NOT NULL,

[Quantity] [smallint] NOT NULL CONSTRAINT [DF\_ProductInventory\_Quantity] DEFAULT (0),

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_ProductInventory\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_ProductInventory\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_ProductInventory\_Shelf] CHECK (([Shelf] LIKE '[A-Za-z]') OR ([Shelf] = 'N/A')),

CONSTRAINT [CK\_ProductInventory\_Bin] CHECK ([Bin] BETWEEN 0 AND 100)

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[ProductListPriceHistory](

[ProductID] [int] NOT NULL,

[StartDate] [datetime] NOT NULL,

[EndDate] [datetime] NULL,

[ListPrice] [money] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_ProductListPriceHistory\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_ProductListPriceHistory\_EndDate] CHECK (([EndDate] >= [StartDate]) OR ([EndDate] IS NULL)),

CONSTRAINT [CK\_ProductListPriceHistory\_ListPrice] CHECK ([ListPrice] > 0.00)

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[ProductModel](

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

[Name] [Name] NOT NULL,

[CatalogDescription] [XML]([Production].[ProductDescriptionSchemaCollection]) NULL,

[Instructions] [XML]([Production].[ManuInstructionsSchemaCollection]) NULL,

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_ProductModel\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_ProductModel\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[ProductModelIllustration](

[ProductModelID] [int] NOT NULL,

[IllustrationID] [int] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_ProductModelIllustration\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[ProductModelProductDescriptionCulture](

[ProductModelID] [int] NOT NULL,

[ProductDescriptionID] [int] NOT NULL,

[CultureID] [nchar](6) NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_ProductModelProductDescriptionCulture\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[ProductPhoto](

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

[ThumbNailPhoto] [varbinary](max) NULL,

[ThumbnailPhotoFileName] [nvarchar](50) NULL,

[LargePhoto] [varbinary](max) NULL,

[LargePhotoFileName] [nvarchar](50) NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_ProductPhoto\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[ProductProductPhoto](

[ProductID] [int] NOT NULL,

[ProductPhotoID] [int] NOT NULL,

[Primary] [Flag] NOT NULL CONSTRAINT [DF\_ProductProductPhoto\_Primary] DEFAULT (0),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_ProductProductPhoto\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[ProductReview](

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

[ProductID] [int] NOT NULL,

[ReviewerName] [Name] NOT NULL,

[ReviewDate] [datetime] NOT NULL CONSTRAINT [DF\_ProductReview\_ReviewDate] DEFAULT (GETDATE()),

[EmailAddress] [nvarchar](50) NOT NULL,

[Rating] [int] NOT NULL,

[Comments] [nvarchar](3850),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_ProductReview\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_ProductReview\_Rating] CHECK ([Rating] BETWEEN 1 AND 5),

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[ProductSubcategory](

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

[ProductCategoryID] [int] NOT NULL,

[Name] [Name] NOT NULL,

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_ProductSubcategory\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_ProductSubcategory\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Purchasing].[ProductVendor](

[ProductID] [int] NOT NULL,

[BusinessEntityID] [int] NOT NULL,

[AverageLeadTime] [int] NOT NULL,

[StandardPrice] [money] NOT NULL,

[LastReceiptCost] [money] NULL,

[LastReceiptDate] [datetime] NULL,

[MinOrderQty] [int] NOT NULL,

[MaxOrderQty] [int] NOT NULL,

[OnOrderQty] [int] NULL,

[UnitMeasureCode] [nchar](3) NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_ProductVendor\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_ProductVendor\_AverageLeadTime] CHECK ([AverageLeadTime] >= 1),

CONSTRAINT [CK\_ProductVendor\_StandardPrice] CHECK ([StandardPrice] > 0.00),

CONSTRAINT [CK\_ProductVendor\_LastReceiptCost] CHECK ([LastReceiptCost] > 0.00),

CONSTRAINT [CK\_ProductVendor\_MinOrderQty] CHECK ([MinOrderQty] >= 1),

CONSTRAINT [CK\_ProductVendor\_MaxOrderQty] CHECK ([MaxOrderQty] >= 1),

CONSTRAINT [CK\_ProductVendor\_OnOrderQty] CHECK ([OnOrderQty] >= 0)

) ON [PRIMARY];

GO

**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 CONSTRAINT [DF\_PurchaseOrderDetail\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_PurchaseOrderDetail\_OrderQty] CHECK ([OrderQty] > 0),

CONSTRAINT [CK\_PurchaseOrderDetail\_UnitPrice] CHECK ([UnitPrice] >= 0.00),

CONSTRAINT [CK\_PurchaseOrderDetail\_ReceivedQty] CHECK ([ReceivedQty] >= 0.00),

CONSTRAINT [CK\_PurchaseOrderDetail\_RejectedQty] CHECK ([RejectedQty] >= 0.00)

) ON [PRIMARY];

GO

**CREATE TABLE** [Purchasing].[PurchaseOrderHeader](

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

[RevisionNumber] [tinyint] NOT NULL CONSTRAINT [DF\_PurchaseOrderHeader\_RevisionNumber] DEFAULT (0),

[Status] [tinyint] NOT NULL CONSTRAINT [DF\_PurchaseOrderHeader\_Status] DEFAULT (1),

[EmployeeID] [int] NOT NULL,

[VendorID] [int] NOT NULL,

[ShipMethodID] [int] NOT NULL,

[OrderDate] [datetime] NOT NULL CONSTRAINT [DF\_PurchaseOrderHeader\_OrderDate] DEFAULT (GETDATE()),

[ShipDate] [datetime] NULL,

[SubTotal] [money] NOT NULL CONSTRAINT [DF\_PurchaseOrderHeader\_SubTotal] DEFAULT (0.00),

[TaxAmt] [money] NOT NULL CONSTRAINT [DF\_PurchaseOrderHeader\_TaxAmt] DEFAULT (0.00),

[Freight] [money] NOT NULL CONSTRAINT [DF\_PurchaseOrderHeader\_Freight] DEFAULT (0.00),

[TotalDue] AS ISNULL([SubTotal] + [TaxAmt] + [Freight], 0) PERSISTED NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_PurchaseOrderHeader\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_PurchaseOrderHeader\_Status] CHECK ([Status] BETWEEN 1 AND 4), -- 1 = Pending; 2 = Approved; 3 = Rejected; 4 = Complete

CONSTRAINT [CK\_PurchaseOrderHeader\_ShipDate] CHECK (([ShipDate] >= [OrderDate]) OR ([ShipDate] IS NULL)),

CONSTRAINT [CK\_PurchaseOrderHeader\_SubTotal] CHECK ([SubTotal] >= 0.00),

CONSTRAINT [CK\_PurchaseOrderHeader\_TaxAmt] CHECK ([TaxAmt] >= 0.00),

CONSTRAINT [CK\_PurchaseOrderHeader\_Freight] CHECK ([Freight] >= 0.00)

) ON [PRIMARY];

GO

**CREATE TABLE** [Sales].[SalesOrderDetail](

[SalesOrderID] [int] NOT NULL,

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

[CarrierTrackingNumber] [nvarchar](25) NULL,

[OrderQty] [smallint] NOT NULL,

[ProductID] [int] NOT NULL,

[SpecialOfferID] [int] NOT NULL,

[UnitPrice] [money] NOT NULL,

[UnitPriceDiscount] [money] NOT NULL CONSTRAINT [DF\_SalesOrderDetail\_UnitPriceDiscount] DEFAULT (0.0),

[LineTotal] AS ISNULL([UnitPrice] \* (1.0 - [UnitPriceDiscount]) \* [OrderQty], 0.0),

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_SalesOrderDetail\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_SalesOrderDetail\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_SalesOrderDetail\_OrderQty] CHECK ([OrderQty] > 0),

CONSTRAINT [CK\_SalesOrderDetail\_UnitPrice] CHECK ([UnitPrice] >= 0.00),

CONSTRAINT [CK\_SalesOrderDetail\_UnitPriceDiscount] CHECK ([UnitPriceDiscount] >= 0.00)

) ON [PRIMARY];

GO

**CREATE TABLE** [Sales].[SalesOrderHeader](

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

[RevisionNumber] [tinyint] NOT NULL CONSTRAINT [DF\_SalesOrderHeader\_RevisionNumber] DEFAULT (0),

[OrderDate] [datetime] NOT NULL CONSTRAINT [DF\_SalesOrderHeader\_OrderDate] DEFAULT (GETDATE()),

[DueDate] [datetime] NOT NULL,

[ShipDate] [datetime] NULL,

[Status] [tinyint] NOT NULL CONSTRAINT [DF\_SalesOrderHeader\_Status] DEFAULT (1),

[OnlineOrderFlag] [Flag] NOT NULL CONSTRAINT [DF\_SalesOrderHeader\_OnlineOrderFlag] DEFAULT (1),

[SalesOrderNumber] AS ISNULL(N'SO' + CONVERT(nvarchar(23), [SalesOrderID]), N'\*\*\* ERROR \*\*\*'),

[PurchaseOrderNumber] [OrderNumber] NULL,

[AccountNumber] [AccountNumber] NULL,

[CustomerID] [int] NOT NULL,

[SalesPersonID] [int] NULL,

[TerritoryID] [int] NULL,

[BillToAddressID] [int] NOT NULL,

[ShipToAddressID] [int] NOT NULL,

[ShipMethodID] [int] NOT NULL,

[CreditCardID] [int] NULL,

[CreditCardApprovalCode] [varchar](15) NULL,

[CurrencyRateID] [int] NULL,

[SubTotal] [money] NOT NULL CONSTRAINT [DF\_SalesOrderHeader\_SubTotal] DEFAULT (0.00),

[TaxAmt] [money] NOT NULL CONSTRAINT [DF\_SalesOrderHeader\_TaxAmt] DEFAULT (0.00),

[Freight] [money] NOT NULL CONSTRAINT [DF\_SalesOrderHeader\_Freight] DEFAULT (0.00),

[TotalDue] AS ISNULL([SubTotal] + [TaxAmt] + [Freight], 0),

[Comment] [nvarchar](128) NULL,

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_SalesOrderHeader\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_SalesOrderHeader\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_SalesOrderHeader\_Status] CHECK ([Status] BETWEEN 0 AND 8),

CONSTRAINT [CK\_SalesOrderHeader\_DueDate] CHECK ([DueDate] >= [OrderDate]),

CONSTRAINT [CK\_SalesOrderHeader\_ShipDate] CHECK (([ShipDate] >= [OrderDate]) OR ([ShipDate] IS NULL)),

CONSTRAINT [CK\_SalesOrderHeader\_SubTotal] CHECK ([SubTotal] >= 0.00),

CONSTRAINT [CK\_SalesOrderHeader\_TaxAmt] CHECK ([TaxAmt] >= 0.00),

CONSTRAINT [CK\_SalesOrderHeader\_Freight] CHECK ([Freight] >= 0.00)

) ON [PRIMARY];

GO

**CREATE TABLE** [Sales].[SalesOrderHeaderSalesReason](

[SalesOrderID] [int] NOT NULL,

[SalesReasonID] [int] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_SalesOrderHeaderSalesReason\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Sales].[SalesPerson](

[BusinessEntityID] [int] NOT NULL,

[TerritoryID] [int] NULL,

[SalesQuota] [money] NULL,

[Bonus] [money] NOT NULL CONSTRAINT [DF\_SalesPerson\_Bonus] DEFAULT (0.00),

[CommissionPct] [smallmoney] NOT NULL CONSTRAINT [DF\_SalesPerson\_CommissionPct] DEFAULT (0.00),

[SalesYTD] [money] NOT NULL CONSTRAINT [DF\_SalesPerson\_SalesYTD] DEFAULT (0.00),

[SalesLastYear] [money] NOT NULL CONSTRAINT [DF\_SalesPerson\_SalesLastYear] DEFAULT (0.00),

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_SalesPerson\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_SalesPerson\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_SalesPerson\_SalesQuota] CHECK ([SalesQuota] > 0.00),

CONSTRAINT [CK\_SalesPerson\_Bonus] CHECK ([Bonus] >= 0.00),

CONSTRAINT [CK\_SalesPerson\_CommissionPct] CHECK ([CommissionPct] >= 0.00),

CONSTRAINT [CK\_SalesPerson\_SalesYTD] CHECK ([SalesYTD] >= 0.00),

CONSTRAINT [CK\_SalesPerson\_SalesLastYear] CHECK ([SalesLastYear] >= 0.00)

) ON [PRIMARY];

GO

**CREATE TABLE** [Sales].[SalesPersonQuotaHistory](

[BusinessEntityID] [int] NOT NULL,

[QuotaDate] [datetime] NOT NULL,

[SalesQuota] [money] NOT NULL,

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_SalesPersonQuotaHistory\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_SalesPersonQuotaHistory\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_SalesPersonQuotaHistory\_SalesQuota] CHECK ([SalesQuota] > 0.00)

) ON [PRIMARY];

GO

**CREATE TABLE** [Sales].[SalesReason](

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

[Name] [Name] NOT NULL,

[ReasonType] [Name] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_SalesReason\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Sales].[SalesTaxRate](

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

[StateProvinceID] [int] NOT NULL,

[TaxType] [tinyint] NOT NULL,

[TaxRate] [smallmoney] NOT NULL CONSTRAINT [DF\_SalesTaxRate\_TaxRate] DEFAULT (0.00),

[Name] [Name] NOT NULL,

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_SalesTaxRate\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_SalesTaxRate\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_SalesTaxRate\_TaxType] CHECK ([TaxType] BETWEEN 1 AND 3)

) ON [PRIMARY];

GO

**CREATE TABLE** [Sales].[SalesTerritory](

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

[Name] [Name] NOT NULL,

[CountryRegionCode] [nvarchar](3) NOT NULL,

[Group] [nvarchar](50) NOT NULL,

[SalesYTD] [money] NOT NULL CONSTRAINT [DF\_SalesTerritory\_SalesYTD] DEFAULT (0.00),

[SalesLastYear] [money] NOT NULL CONSTRAINT [DF\_SalesTerritory\_SalesLastYear] DEFAULT (0.00),

[CostYTD] [money] NOT NULL CONSTRAINT [DF\_SalesTerritory\_CostYTD] DEFAULT (0.00),

[CostLastYear] [money] NOT NULL CONSTRAINT [DF\_SalesTerritory\_CostLastYear] DEFAULT (0.00),

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_SalesTerritory\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_SalesTerritory\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_SalesTerritory\_SalesYTD] CHECK ([SalesYTD] >= 0.00),

CONSTRAINT [CK\_SalesTerritory\_SalesLastYear] CHECK ([SalesLastYear] >= 0.00),

CONSTRAINT [CK\_SalesTerritory\_CostYTD] CHECK ([CostYTD] >= 0.00),

CONSTRAINT [CK\_SalesTerritory\_CostLastYear] CHECK ([CostLastYear] >= 0.00)

) ON [PRIMARY];

GO

**CREATE TABLE** [Sales].[SalesTerritoryHistory](

[BusinessEntityID] [int] NOT NULL, -- A sales person

[TerritoryID] [int] NOT NULL,

[StartDate] [datetime] NOT NULL,

[EndDate] [datetime] NULL,

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_SalesTerritoryHistory\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_SalesTerritoryHistory\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_SalesTerritoryHistory\_EndDate] CHECK (([EndDate] >= [StartDate]) OR ([EndDate] IS NULL))

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[ScrapReason](

[ScrapReasonID] [smallint] IDENTITY (1, 1) NOT NULL,

[Name] [Name] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_ScrapReason\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [HumanResources].[Shift](

[ShiftID] [tinyint] IDENTITY (1, 1) NOT NULL,

[Name] [Name] NOT NULL,

[StartTime] [time] NOT NULL,

[EndTime] [time] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_Shift\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Purchasing].[ShipMethod](

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

[Name] [Name] NOT NULL,

[ShipBase] [money] NOT NULL CONSTRAINT [DF\_ShipMethod\_ShipBase] DEFAULT (0.00),

[ShipRate] [money] NOT NULL CONSTRAINT [DF\_ShipMethod\_ShipRate] DEFAULT (0.00),

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_ShipMethod\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_ShipMethod\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_ShipMethod\_ShipBase] CHECK ([ShipBase] > 0.00),

CONSTRAINT [CK\_ShipMethod\_ShipRate] CHECK ([ShipRate] > 0.00),

) ON [PRIMARY];

GO

**CREATE TABLE** [Sales].[ShoppingCartItem](

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

[ShoppingCartID] [nvarchar](50) NOT NULL,

[Quantity] [int] NOT NULL CONSTRAINT [DF\_ShoppingCartItem\_Quantity] DEFAULT (1),

[ProductID] [int] NOT NULL,

[DateCreated] [datetime] NOT NULL CONSTRAINT [DF\_ShoppingCartItem\_DateCreated] DEFAULT (GETDATE()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_ShoppingCartItem\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_ShoppingCartItem\_Quantity] CHECK ([Quantity] >= 1)

) ON [PRIMARY];

GO

**CREATE TABLE** [Sales].[SpecialOffer](

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

[Description] [nvarchar](255) NOT NULL,

[DiscountPct] [smallmoney] NOT NULL CONSTRAINT [DF\_SpecialOffer\_DiscountPct] DEFAULT (0.00),

[Type] [nvarchar](50) NOT NULL,

[Category] [nvarchar](50) NOT NULL,

[StartDate] [datetime] NOT NULL,

[EndDate] [datetime] NOT NULL,

[MinQty] [int] NOT NULL CONSTRAINT [DF\_SpecialOffer\_MinQty] DEFAULT (0),

[MaxQty] [int] NULL,

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_SpecialOffer\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_SpecialOffer\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_SpecialOffer\_EndDate] CHECK ([EndDate] >= [StartDate]),

CONSTRAINT [CK\_SpecialOffer\_DiscountPct] CHECK ([DiscountPct] >= 0.00),

CONSTRAINT [CK\_SpecialOffer\_MinQty] CHECK ([MinQty] >= 0),

CONSTRAINT [CK\_SpecialOffer\_MaxQty] CHECK ([MaxQty] >= 0)

) ON [PRIMARY];

GO

**CREATE TABLE** [Sales].[SpecialOfferProduct](

[SpecialOfferID] [int] NOT NULL,

[ProductID] [int] NOT NULL,

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_SpecialOfferProduct\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_SpecialOfferProduct\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Person].[StateProvince](

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

[StateProvinceCode] [nchar](3) NOT NULL,

[CountryRegionCode] [nvarchar](3) NOT NULL,

[IsOnlyStateProvinceFlag] [Flag] NOT NULL CONSTRAINT [DF\_StateProvince\_IsOnlyStateProvinceFlag] DEFAULT (1),

[Name] [Name] NOT NULL,

[TerritoryID] [int] NOT NULL,

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_StateProvince\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_StateProvince\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Sales].[Store](

[BusinessEntityID] [int] NOT NULL,

[Name] [Name] NOT NULL,

[SalesPersonID] [int] NULL,

[Demographics] [XML]([Sales].[StoreSurveySchemaCollection]) NULL,

[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_Store\_rowguid] DEFAULT (NEWID()),

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_Store\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[TransactionHistory](

[TransactionID] [int] IDENTITY (100000, 1) NOT NULL,

[ProductID] [int] NOT NULL,

[ReferenceOrderID] [int] NOT NULL,

[ReferenceOrderLineID] [int] NOT NULL CONSTRAINT [DF\_TransactionHistory\_ReferenceOrderLineID] DEFAULT (0),

[TransactionDate] [datetime] NOT NULL CONSTRAINT [DF\_TransactionHistory\_TransactionDate] DEFAULT (GETDATE()),

[TransactionType] [nchar](1) NOT NULL,

[Quantity] [int] NOT NULL,

[ActualCost] [money] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_TransactionHistory\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_TransactionHistory\_TransactionType] CHECK (UPPER([TransactionType]) IN ('W', 'S', 'P'))

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[TransactionHistoryArchive](

[TransactionID] [int] NOT NULL,

[ProductID] [int] NOT NULL,

[ReferenceOrderID] [int] NOT NULL,

[ReferenceOrderLineID] [int] NOT NULL CONSTRAINT [DF\_TransactionHistoryArchive\_ReferenceOrderLineID] DEFAULT (0),

[TransactionDate] [datetime] NOT NULL CONSTRAINT [DF\_TransactionHistoryArchive\_TransactionDate] DEFAULT (GETDATE()),

[TransactionType] [nchar](1) NOT NULL,

[Quantity] [int] NOT NULL,

[ActualCost] [money] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_TransactionHistoryArchive\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_TransactionHistoryArchive\_TransactionType] CHECK (UPPER([TransactionType]) IN ('W', 'S', 'P'))

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[UnitMeasure](

[UnitMeasureCode] [nchar](3) NOT NULL,

[Name] [Name] NOT NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_UnitMeasure\_ModifiedDate] DEFAULT (GETDATE())

) ON [PRIMARY];

GO

**CREATE TABLE** [Purchasing].[Vendor](

[BusinessEntityID] [int] NOT NULL,

[AccountNumber] [AccountNumber] NOT NULL,

[Name] [Name] NOT NULL,

[CreditRating] [tinyint] NOT NULL,

[PreferredVendorStatus] [Flag] NOT NULL CONSTRAINT [DF\_Vendor\_PreferredVendorStatus] DEFAULT (1),

[ActiveFlag] [Flag] NOT NULL CONSTRAINT [DF\_Vendor\_ActiveFlag] DEFAULT (1),

[PurchasingWebServiceURL] [nvarchar](1024) NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_Vendor\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_Vendor\_CreditRating] CHECK ([CreditRating] BETWEEN 1 AND 5)

) ON [PRIMARY];

GO

**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 CONSTRAINT [DF\_WorkOrder\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_WorkOrder\_OrderQty] CHECK ([OrderQty] > 0),

CONSTRAINT [CK\_WorkOrder\_ScrappedQty] CHECK ([ScrappedQty] >= 0),

CONSTRAINT [CK\_WorkOrder\_EndDate] CHECK (([EndDate] >= [StartDate]) OR ([EndDate] IS NULL))

) ON [PRIMARY];

GO

**CREATE TABLE** [Production].[WorkOrderRouting](

[WorkOrderID] [int] NOT NULL,

[ProductID] [int] NOT NULL,

[OperationSequence] [smallint] NOT NULL,

[LocationID] [smallint] NOT NULL,

[ScheduledStartDate] [datetime] NOT NULL,

[ScheduledEndDate] [datetime] NOT NULL,

[ActualStartDate] [datetime] NULL,

[ActualEndDate] [datetime] NULL,

[ActualResourceHrs] [decimal](9, 4) NULL,

[PlannedCost] [money] NOT NULL,

[ActualCost] [money] NULL,

[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_WorkOrderRouting\_ModifiedDate] DEFAULT (GETDATE()),

CONSTRAINT [CK\_WorkOrderRouting\_ScheduledEndDate] CHECK ([ScheduledEndDate] >= [ScheduledStartDate]),

CONSTRAINT [CK\_WorkOrderRouting\_ActualEndDate] CHECK (([ActualEndDate] >= [ActualStartDate])

OR ([ActualEndDate] IS NULL) OR ([ActualStartDate] IS NULL)),

CONSTRAINT [CK\_WorkOrderRouting\_ActualResourceHrs] CHECK ([ActualResourceHrs] >= 0.0000),

CONSTRAINT [CK\_WorkOrderRouting\_PlannedCost] CHECK ([PlannedCost] > 0.00),

CONSTRAINT [CK\_WorkOrderRouting\_ActualCost] CHECK ([ActualCost] > 0.00)

) ON [PRIMARY];

GO

[TOP](#Top)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Load data into Tables

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PRINT '';

PRINT '\*\*\* Loading Data';

GO

PRINT 'Loading [Person].[Address]';

**BULK INSERT** [Person].[Address] FROM '$(SqlSamplesSourceDataPath)Address.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE = 'char',

FIELDTERMINATOR= '\t',

ROWTERMINATOR = '\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Person].[AddressType]';

**BULK INSERT** [Person].[AddressType] FROM '$(SqlSamplesSourceDataPath)AddressType.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE = 'char',

FIELDTERMINATOR= '\t',

ROWTERMINATOR = '\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [dbo].[AWBuildVersion]';

INSERT INTO [dbo].[AWBuildVersion]

VALUES

( CONVERT(nvarchar(25), SERVERPROPERTY('ProductVersion')), CONVERT(datetime, SERVERPROPERTY('ResourceLastUpdateDateTime')), CONVERT(datetime, GETDATE()) );

PRINT 'Loading [Production].[BillOfMaterials]';

**BULK INSERT** [Production].[BillOfMaterials] FROM '$(SqlSamplesSourceDataPath)BillOfMaterials.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE = 'char',

FIELDTERMINATOR= '\t',

ROWTERMINATOR = '\n',

KEEPIDENTITY,

TABLOCK

);

See :SetVar at beginning of script

PRINT 'Loading [Person].[BusinessEntity]';

**BULK INSERT** [Person].[BusinessEntity] FROM '$(SqlSamplesSourceDataPath)BusinessEntity.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='widechar',

FIELDTERMINATOR='+|',

ROWTERMINATOR='&|\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Person].[BusinessEntityAddress]';

**BULK INSERT** [Person].[BusinessEntityAddress] FROM '$(SqlSamplesSourceDataPath)BusinessEntityAddress.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='widechar',

FIELDTERMINATOR='+|',

ROWTERMINATOR='&|\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Person].[BusinessEntityContact]';

**BULK INSERT** [Person].[BusinessEntityContact] FROM '$(SqlSamplesSourceDataPath)BusinessEntityContact.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='widechar',

FIELDTERMINATOR='+|',

ROWTERMINATOR='&|\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Person].[ContactType]';

**BULK INSERT** [Person].[ContactType] FROM '$(SqlSamplesSourceDataPath)ContactType.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Person].[CountryRegion]';

**BULK INSERT** [Person].[CountryRegion] FROM '$(SqlSamplesSourceDataPath)CountryRegion.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='widechar',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Sales].[CountryRegionCurrency]';

**BULK INSERT** [Sales].[CountryRegionCurrency] FROM '$(SqlSamplesSourceDataPath)CountryRegionCurrency.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='widechar',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Sales].[CreditCard]';

**BULK INSERT** [Sales].[CreditCard] FROM '$(SqlSamplesSourceDataPath)CreditCard.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[Culture]';

**BULK INSERT** [Production].[Culture] FROM '$(SqlSamplesSourceDataPath)Culture.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Sales].[Currency]';

**BULK INSERT** [Sales].[Currency] FROM '$(SqlSamplesSourceDataPath)Currency.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Sales].[CurrencyRate]';

**BULK INSERT** [Sales].[CurrencyRate] FROM '$(SqlSamplesSourceDataPath)CurrencyRate.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Sales].[Customer]';

**BULK INSERT** [Sales].[Customer] FROM '$(SqlSamplesSourceDataPath)Customer.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [HumanResources].[Department]';

**BULK INSERT** [HumanResources].[Department] FROM '$(SqlSamplesSourceDataPath)Department.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[Document]';

**BULK INSERT** [Production].[Document] FROM '$(SqlSamplesSourceDataPath)Document.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='widechar',

FIELDTERMINATOR='+|',

ROWTERMINATOR='&|\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Person].[EmailAddress]';

**BULK INSERT** [Person].[EmailAddress] FROM '$(SqlSamplesSourceDataPath)EmailAddress.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='widechar',

FIELDTERMINATOR='+|',

ROWTERMINATOR='&|\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [HumanResources].[Employee]';

**BULK INSERT** [HumanResources].[Employee] FROM '$(SqlSamplesSourceDataPath)Employee.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='widechar',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [HumanResources].[EmployeeDepartmentHistory]';

**BULK INSERT** [HumanResources].[EmployeeDepartmentHistory] FROM '$(SqlSamplesSourceDataPath)EmployeeDepartmentHistory.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [HumanResources].[EmployeePayHistory]';

**BULK INSERT** [HumanResources].[EmployeePayHistory] FROM '$(SqlSamplesSourceDataPath)EmployeePayHistory.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[Illustration]';

**BULK INSERT** [Production].[Illustration] FROM '$(SqlSamplesSourceDataPath)Illustration.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='widechar',

FIELDTERMINATOR='+|',

ROWTERMINATOR='&|\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [HumanResources].[JobCandidate]';

**BULK INSERT** [HumanResources].[JobCandidate] FROM '$(SqlSamplesSourceDataPath)JobCandidate.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='widechar',

FIELDTERMINATOR='+|',

ROWTERMINATOR='&|\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[Location]';

**BULK INSERT** [Production].[Location] FROM '$(SqlSamplesSourceDataPath)Location.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Person].[Password]';

**BULK INSERT** [Person].[Password] FROM '$(SqlSamplesSourceDataPath)Password.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='widechar',

FIELDTERMINATOR='+|',

ROWTERMINATOR='&|\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Person].[Person]';

**BULK INSERT** [Person].[Person] FROM '$(SqlSamplesSourceDataPath)Person.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='widechar',

FIELDTERMINATOR='+|',

ROWTERMINATOR='&|\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Sales].[PersonCreditCard]';

**BULK INSERT** [Sales].[PersonCreditCard] FROM '$(SqlSamplesSourceDataPath)PersonCreditCard.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Person].[PersonPhone]';

**BULK INSERT** [Person].[PersonPhone] FROM '$(SqlSamplesSourceDataPath)PersonPhone.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='widechar',

FIELDTERMINATOR='+|',

ROWTERMINATOR='&|\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Person].[PhoneNumberType]';

**BULK INSERT** [Person].[PhoneNumberType] FROM '$(SqlSamplesSourceDataPath)PhoneNumberType.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='widechar',

FIELDTERMINATOR='+|',

ROWTERMINATOR='&|\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[Product]';

**BULK INSERT** [Production].[Product] FROM '$(SqlSamplesSourceDataPath)Product.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[ProductCategory]';

**BULK INSERT** [Production].[ProductCategory] FROM '$(SqlSamplesSourceDataPath)ProductCategory.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[ProductCostHistory]';

**BULK INSERT** [Production].[ProductCostHistory] FROM '$(SqlSamplesSourceDataPath)ProductCostHistory.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[ProductDescription]';

**BULK INSERT** [Production].[ProductDescription] FROM '$(SqlSamplesSourceDataPath)ProductDescription.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='widechar',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[ProductDocument]';

**BULK INSERT** [Production].[ProductDocument] FROM '$(SqlSamplesSourceDataPath)ProductDocument.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[ProductInventory]';

**BULK INSERT** [Production].[ProductInventory] FROM '$(SqlSamplesSourceDataPath)ProductInventory.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[ProductListPriceHistory]';

**BULK INSERT** [Production].[ProductListPriceHistory] FROM '$(SqlSamplesSourceDataPath)ProductListPriceHistory.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[ProductModel]';

**BULK INSERT** [Production].[ProductModel] FROM '$(SqlSamplesSourceDataPath)ProductModel.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='widechar',

FIELDTERMINATOR='+|',

ROWTERMINATOR='&|\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[ProductModelIllustration]';

**BULK INSERT** [Production].[ProductModelIllustration] FROM '$(SqlSamplesSourceDataPath)ProductModelIllustration.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[ProductModelProductDescriptionCulture]';

**BULK INSERT** [Production].[ProductModelProductDescriptionCulture] FROM '$(SqlSamplesSourceDataPath)ProductModelProductDescriptionCulture.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[ProductPhoto]';

**BULK INSERT** [Production].[ProductPhoto] FROM '$(SqlSamplesSourceDataPath)ProductPhoto.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='widechar',

FIELDTERMINATOR='+|',

ROWTERMINATOR='&|\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[ProductProductPhoto]';

**BULK INSERT** [Production].[ProductProductPhoto] FROM '$(SqlSamplesSourceDataPath)ProductProductPhoto.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[ProductReview]';

**BULK INSERT** [Production].[ProductReview] FROM '$(SqlSamplesSourceDataPath)ProductReview.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[ProductSubcategory]';

**BULK INSERT** [Production].[ProductSubcategory] FROM '$(SqlSamplesSourceDataPath)ProductSubcategory.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Purchasing].[ProductVendor]';

**BULK INSERT** [Purchasing].[ProductVendor] FROM '$(SqlSamplesSourceDataPath)ProductVendor.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Purchasing].[PurchaseOrderDetail]';

**BULK INSERT** [Purchasing].[PurchaseOrderDetail] FROM '$(SqlSamplesSourceDataPath)PurchaseOrderDetail.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Purchasing].[PurchaseOrderHeader]';

**BULK INSERT** [Purchasing].[PurchaseOrderHeader] FROM '$(SqlSamplesSourceDataPath)PurchaseOrderHeader.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Sales].[SalesOrderDetail]';

**BULK INSERT** [Sales].[SalesOrderDetail] FROM '$(SqlSamplesSourceDataPath)SalesOrderDetail.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Sales].[SalesOrderHeader]';

**BULK INSERT** [Sales].[SalesOrderHeader] FROM '$(SqlSamplesSourceDataPath)SalesOrderHeader.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Sales].[SalesOrderHeaderSalesReason]';

**BULK INSERT** [Sales].[SalesOrderHeaderSalesReason] FROM '$(SqlSamplesSourceDataPath)SalesOrderHeaderSalesReason.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Sales].[SalesPerson]';

**BULK INSERT** [Sales].[SalesPerson] FROM '$(SqlSamplesSourceDataPath)SalesPerson.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Sales].[SalesPersonQuotaHistory]';

**BULK INSERT** [Sales].[SalesPersonQuotaHistory] FROM '$(SqlSamplesSourceDataPath)SalesPersonQuotaHistory.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Sales].[SalesReason]';

**BULK INSERT** [Sales].[SalesReason] FROM '$(SqlSamplesSourceDataPath)SalesReason.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Sales].[SalesTaxRate]';

**BULK INSERT** [Sales].[SalesTaxRate] FROM '$(SqlSamplesSourceDataPath)SalesTaxRate.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Sales].[SalesTerritory]';

**BULK INSERT** [Sales].[SalesTerritory] FROM '$(SqlSamplesSourceDataPath)SalesTerritory.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Sales].[SalesTerritoryHistory]';

**BULK INSERT** [Sales].[SalesTerritoryHistory] FROM '$(SqlSamplesSourceDataPath)SalesTerritoryHistory.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[ScrapReason]';

**BULK INSERT** [Production].[ScrapReason] FROM '$(SqlSamplesSourceDataPath)ScrapReason.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [HumanResources].[Shift]';

**BULK INSERT** [HumanResources].[Shift] FROM '$(SqlSamplesSourceDataPath)Shift.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Purchasing].[ShipMethod]';

**BULK INSERT** [Purchasing].[ShipMethod] FROM '$(SqlSamplesSourceDataPath)ShipMethod.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Sales].[ShoppingCartItem]';

**BULK INSERT** [Sales].[ShoppingCartItem] FROM '$(SqlSamplesSourceDataPath)ShoppingCartItem.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Sales].[SpecialOffer]';

**BULK INSERT** [Sales].[SpecialOffer] FROM '$(SqlSamplesSourceDataPath)SpecialOffer.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Sales].[SpecialOfferProduct]';

**BULK INSERT** [Sales].[SpecialOfferProduct] FROM '$(SqlSamplesSourceDataPath)SpecialOfferProduct.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Person].[StateProvince]';

**BULK INSERT** [Person].[StateProvince] FROM '$(SqlSamplesSourceDataPath)StateProvince.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='widechar',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Sales].[Store]';

**BULK INSERT** [Sales].[Store] FROM '$(SqlSamplesSourceDataPath)Store.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='widechar',

FIELDTERMINATOR='+|',

ROWTERMINATOR='&|\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[TransactionHistory]';

**BULK INSERT** [Production].[TransactionHistory] FROM '$(SqlSamplesSourceDataPath)TransactionHistory.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

TABLOCK

);

PRINT 'Loading [Production].[TransactionHistoryArchive]';

**BULK INSERT** [Production].[TransactionHistoryArchive] FROM '$(SqlSamplesSourceDataPath)TransactionHistoryArchive.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[UnitMeasure]';

**BULK INSERT** [Production].[UnitMeasure] FROM '$(SqlSamplesSourceDataPath)UnitMeasure.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Purchasing].[Vendor]';

**BULK INSERT** [Purchasing].[Vendor] FROM '$(SqlSamplesSourceDataPath)Vendor.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[WorkOrder]';

**BULK INSERT** [Production].[WorkOrder] FROM '$(SqlSamplesSourceDataPath)WorkOrder.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

PRINT 'Loading [Production].[WorkOrderRouting]';

**BULK INSERT** [Production].[WorkOrderRouting] FROM '$(SqlSamplesSourceDataPath)WorkOrderRouting.csv'

WITH (

CHECK\_CONSTRAINTS,

CODEPAGE='ACP',

DATAFILETYPE='char',

FIELDTERMINATOR='\t',

ROWTERMINATOR='\n',

KEEPIDENTITY,

TABLOCK

);

GO

[TOP](#Top)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Add Primary Keys

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PRINT '';

PRINT '\*\*\* Adding Primary Keys';

GO

SET QUOTED\_IDENTIFIER ON;

When **SET QUOTED\_IDENTIFIER** is ON, identifiers can be delimited by double quotation marks, and literals must be delimited by single quotation marks. When**SET QUOTED\_IDENTIFIER** is OFF, identifiers cannot be quoted and must follow all Transact-**SQL** rules for identifiers.

**ALTER TABLE** [Person].[Address] WITH CHECK

{ **CHECK** | **NOCHECK** } CONSTRAINT  
Specifies that constraint\_name is enabled or disabled. This option can only be used with FOREIGN KEY and CHECK constraints. When NOCHECK is specified, the constraint is disabled and future inserts or updates to the column are not validated against the constraint conditions. DEFAULT, PRIMARY KEY, and UNIQUE constraints cannot be disabled.

ADD **CONSTRAINT** [PK\_Address\_AddressID] PRIMARY KEY CLUSTERED

(

[AddressID]

) ON [PRIMARY];

GO

**ALTER TABLE** [Person].[AddressType] WITH CHECK

ADD CONSTRAINT [PK\_AddressType\_AddressTypeID] PRIMARY KEY CLUSTERED

(

[AddressTypeID]

) ON [PRIMARY];

GO

**ALTER TABLE** [dbo].[AWBuildVersion] WITH CHECK

ADD CONSTRAINT [PK\_AWBuildVersion\_SystemInformationID] PRIMARY KEY CLUSTERED

(

[SystemInformationID]

) ON [PRIMARY];

GO

**ALTER TABLE** [Production].[BillOfMaterials] WITH CHECK

ADD CONSTRAINT [PK\_BillOfMaterials\_BillOfMaterialsID] PRIMARY KEY NONCLUSTERED

(

[BillOfMaterialsID]

) ON [PRIMARY];

GO

**ALTER TABLE** [Person].[BusinessEntity] WITH CHECK

ADD CONSTRAINT [PK\_BusinessEntity\_BusinessEntityID] PRIMARY KEY CLUSTERED

(

[BusinessEntityID]

) ON [PRIMARY];

GO

**ALTER TABLE** [Person].[BusinessEntityAddress] WITH CHECK

ADD CONSTRAINT [PK\_BusinessEntityAddress\_BusinessEntityID\_AddressID\_AddressTypeID] PRIMARY KEY CLUSTERED

(

[BusinessEntityID],

[AddressID],

[AddressTypeID]

) ON [PRIMARY];

GO

**ALTER TABLE** [Person].[BusinessEntityContact] WITH CHECK

ADD CONSTRAINT [PK\_BusinessEntityContact\_BusinessEntityID\_PersonID\_ContactTypeID] PRIMARY KEY CLUSTERED

(

[BusinessEntityID],

[PersonID],

[ContactTypeID]

) ON [PRIMARY];

GO

**ALTER TABLE** [Person].[ContactType] WITH CHECK

ADD CONSTRAINT [PK\_ContactType\_ContactTypeID] PRIMARY KEY CLUSTERED

(

[ContactTypeID]

) ON [PRIMARY];

GO

**ALTER TABLE** [Sales].[CountryRegionCurrency] WITH CHECK

ADD CONSTRAINT [PK\_CountryRegionCurrency\_CountryRegionCode\_CurrencyCode] PRIMARY KEY CLUSTERED

(

[CountryRegionCode],

[CurrencyCode]

) ON [PRIMARY];

GO

**ALTER TABLE** [Person].[CountryRegion] WITH CHECK

ADD CONSTRAINT [PK\_CountryRegion\_CountryRegionCode] PRIMARY KEY CLUSTERED

(

[CountryRegionCode]

) ON [PRIMARY];

GO

**ALTER TABLE** [Sales].[CreditCard] WITH CHECK

ADD CONSTRAINT [PK\_CreditCard\_CreditCardID] PRIMARY KEY CLUSTERED

(

[CreditCardID]

) ON [PRIMARY];

GO

**ALTER TABLE** [Production].[Culture] WITH CHECK

ADD CONSTRAINT [PK\_Culture\_CultureID] PRIMARY KEY CLUSTERED

(

[CultureID]

) ON [PRIMARY];

GO

**ALTER TABLE** [Sales].[Currency] WITH CHECK

ADD CONSTRAINT [PK\_Currency\_CurrencyCode] PRIMARY KEY CLUSTERED

(

[CurrencyCode]

) ON [PRIMARY];

GO

**ALTER TABLE** [Sales].[CurrencyRate] WITH CHECK

ADD CONSTRAINT [PK\_CurrencyRate\_CurrencyRateID] PRIMARY KEY CLUSTERED

(

[CurrencyRateID]

) ON [PRIMARY];

GO

**ALTER TABLE** [Sales].[Customer] WITH CHECK

ADD CONSTRAINT [PK\_Customer\_CustomerID] PRIMARY KEY CLUSTERED

(

[CustomerID]

) ON [PRIMARY];

GO

**ALTER TABLE** [dbo].[DatabaseLog] WITH CHECK

ADD CONSTRAINT [PK\_DatabaseLog\_DatabaseLogID] PRIMARY KEY NONCLUSTERED

(

[DatabaseLogID]

) ON [PRIMARY];

GO

**ALTER TABLE** [HumanResources].[Department] WITH CHECK

ADD CONSTRAINT [PK\_Department\_DepartmentID] PRIMARY KEY CLUSTERED

(

[DepartmentID]

) ON [PRIMARY];

GO

**ALTER TABLE** [Production].[Document] WITH CHECK

ADD CONSTRAINT [PK\_Document\_DocumentNode] PRIMARY KEY CLUSTERED

(

[DocumentNode]

) ON [PRIMARY];

GO

**ALTER TABLE** [Person].[EmailAddress] WITH CHECK

ADD CONSTRAINT [PK\_EmailAddress\_BusinessEntityID\_EmailAddressID] PRIMARY KEY CLUSTERED

(

[BusinessEntityID],

[EmailAddressID]

) ON [PRIMARY];

GO

**ALTER TABLE** [HumanResources].[Employee] WITH CHECK

ADD CONSTRAINT [PK\_Employee\_BusinessEntityID] PRIMARY KEY CLUSTERED

(

[BusinessEntityID]

) ON [PRIMARY];

GO

**ALTER TABLE** [HumanResources].[EmployeeDepartmentHistory] WITH CHECK

ADD CONSTRAINT [PK\_EmployeeDepartmentHistory\_BusinessEntityID\_StartDate\_DepartmentID] PRIMARY KEY CLUSTERED

(

[BusinessEntityID],

[StartDate],

[DepartmentID],

[ShiftID]

) ON [PRIMARY];

GO

**ALTER TABLE** [HumanResources].[EmployeePayHistory] WITH CHECK

ADD CONSTRAINT [PK\_EmployeePayHistory\_BusinessEntityID\_RateChangeDate] PRIMARY KEY CLUSTERED

(

[BusinessEntityID],

[RateChangeDate]

) ON [PRIMARY];

GO

**ALTER TABLE** [Production].[Illustration] WITH CHECK

ADD CONSTRAINT [PK\_Illustration\_IllustrationID] PRIMARY KEY CLUSTERED

(

[IllustrationID]

) ON [PRIMARY];

GO

**ALTER TABLE** [HumanResources].[JobCandidate] WITH CHECK

ADD CONSTRAINT [PK\_JobCandidate\_JobCandidateID] PRIMARY KEY CLUSTERED

(

[JobCandidateID]

) ON [PRIMARY];

GO

**ALTER TABLE** [Production].[Location] WITH CHECK

ADD CONSTRAINT [PK\_Location\_LocationID] PRIMARY KEY CLUSTERED

(

[LocationID]

) ON [PRIMARY];

GO

**ALTER TABLE** [Person].[Password] WITH CHECK

ADD CONSTRAINT [PK\_Password\_BusinessEntityID] PRIMARY KEY CLUSTERED

(

[BusinessEntityID]

) ON [PRIMARY];

GO

**ALTER TABLE** [Person].[Person] WITH CHECK

ADD CONSTRAINT [PK\_Person\_BusinessEntityID]

**PRIMARY KEY CLUSTERED**

(

[BusinessEntityID]

) ON [PRIMARY];

GO

**ALTER TABLE** [Sales].[PersonCreditCard] WITH CHECK

ADD CONSTRAINT [PK\_PersonCreditCard\_BusinessEntityID\_CreditCardID]

**PRIMARY KEY CLUSTERED**

(

[BusinessEntityID],

[CreditCardID]

) ON [PRIMARY];

GO

**ALTER TABLE** [Person].[PersonPhone] WITH CHECK

ADD CONSTRAINT [PK\_PersonPhone\_BusinessEntityID\_PhoneNumber\_PhoneNumberTypeID]

**PRIMARY KEY CLUSTERED**

(

[BusinessEntityID],

[PhoneNumber],

[PhoneNumberTypeID]

) ON [PRIMARY];

GO

**ALTER TABLE** [Person].[PhoneNumberType] WITH CHECK

ADD CONSTRAINT [PK\_PhoneNumberType\_PhoneNumberTypeID]

**PRIMARY KEY CLUSTERED**

(

[PhoneNumberTypeID]

) ON [PRIMARY];

GO

**ALTER TABLE** [Production].[Product] WITH CHECK

ADD CONSTRAINT [PK\_Product\_ProductID]

**PRIMARY KEY CLUSTERED**

(

[ProductID]

) ON [PRIMARY];

GO

**ALTER TABLE** [Production].[ProductCategory] WITH CHECK

ADD CONSTRAINT [PK\_ProductCategory\_ProductCategoryID]

**PRIMARY KEY CLUSTERED**

(

[ProductCategoryID]

) ON [PRIMARY];

GO

**ALTER TABLE** [Production].[ProductCostHistory] WITH CHECK

ADD CONSTRAINT [PK\_ProductCostHistory\_ProductID\_StartDate] PRIMARY KEY CLUSTERED

(

[ProductID],

[StartDate]

) ON [PRIMARY];

GO

ALTER TABLE [Production].[ProductDescription] WITH CHECK

ADD CONSTRAINT [PK\_ProductDescription\_ProductDescriptionID] PRIMARY KEY CLUSTERED

(

[ProductDescriptionID]

) ON [PRIMARY];

GO

ALTER TABLE [Production].[ProductDocument] WITH CHECK

ADD CONSTRAINT [PK\_ProductDocument\_ProductID\_DocumentNode] PRIMARY KEY CLUSTERED

(

[ProductID],

[DocumentNode]

) ON [PRIMARY];

GO

ALTER TABLE [Production].[ProductInventory] WITH CHECK

ADD CONSTRAINT [PK\_ProductInventory\_ProductID\_LocationID] PRIMARY KEY CLUSTERED

(

[ProductID],

[LocationID]

) ON [PRIMARY];

GO

ALTER TABLE [Production].[ProductListPriceHistory] WITH CHECK

ADD CONSTRAINT [PK\_ProductListPriceHistory\_ProductID\_StartDate] PRIMARY KEY CLUSTERED

(

[ProductID],

[StartDate]

) ON [PRIMARY];

GO

ALTER TABLE [Production].[ProductModel] WITH CHECK

ADD CONSTRAINT [PK\_ProductModel\_ProductModelID] PRIMARY KEY CLUSTERED

(

[ProductModelID]

) ON [PRIMARY];

GO

ALTER TABLE [Production].[ProductModelIllustration] WITH CHECK ADD

CONSTRAINT [PK\_ProductModelIllustration\_ProductModelID\_IllustrationID] PRIMARY KEY CLUSTERED

(

[ProductModelID],

[IllustrationID]

) ON [PRIMARY];

GO

ALTER TABLE [Production].[ProductModelProductDescriptionCulture] WITH CHECK ADD

CONSTRAINT [PK\_ProductModelProductDescriptionCulture\_ProductModelID\_ProductDescriptionID\_CultureID] PRIMARY KEY CLUSTERED

(

[ProductModelID],

[ProductDescriptionID],

[CultureID]

) ON [PRIMARY];

GO

ALTER TABLE [Production].[ProductPhoto] WITH CHECK ADD

CONSTRAINT [PK\_ProductPhoto\_ProductPhotoID] PRIMARY KEY CLUSTERED

(

[ProductPhotoID]

) ON [PRIMARY];

GO

ALTER TABLE [Production].[ProductProductPhoto] WITH CHECK ADD

CONSTRAINT [PK\_ProductProductPhoto\_ProductID\_ProductPhotoID] PRIMARY KEY NONCLUSTERED

(

[ProductID],

[ProductPhotoID]

) ON [PRIMARY];

GO

ALTER TABLE [Production].[ProductReview] WITH CHECK ADD

CONSTRAINT [PK\_ProductReview\_ProductReviewID] PRIMARY KEY CLUSTERED

(

[ProductReviewID]

) ON [PRIMARY];

GO

ALTER TABLE [Production].[ProductSubcategory] WITH CHECK ADD

CONSTRAINT [PK\_ProductSubcategory\_ProductSubcategoryID] PRIMARY KEY CLUSTERED

(

[ProductSubcategoryID]

) ON [PRIMARY];

GO

ALTER TABLE [Purchasing].[ProductVendor] WITH CHECK ADD

CONSTRAINT [PK\_ProductVendor\_ProductID\_BusinessEntityID] PRIMARY KEY CLUSTERED

(

[ProductID],

[BusinessEntityID]

) ON [PRIMARY];

GO

ALTER TABLE [Purchasing].[PurchaseOrderDetail] WITH CHECK ADD

CONSTRAINT [PK\_PurchaseOrderDetail\_PurchaseOrderID\_PurchaseOrderDetailID] PRIMARY KEY CLUSTERED

(

[PurchaseOrderID],

[PurchaseOrderDetailID]

) ON [PRIMARY];

GO

ALTER TABLE [Purchasing].[PurchaseOrderHeader] WITH CHECK ADD

CONSTRAINT [PK\_PurchaseOrderHeader\_PurchaseOrderID] PRIMARY KEY CLUSTERED

(

[PurchaseOrderID]

) ON [PRIMARY];

GO

ALTER TABLE [Sales].[SalesOrderDetail] WITH CHECK ADD

CONSTRAINT [PK\_SalesOrderDetail\_SalesOrderID\_SalesOrderDetailID] PRIMARY KEY CLUSTERED

(

[SalesOrderID],

[SalesOrderDetailID]

) ON [PRIMARY];

GO

ALTER TABLE [Sales].[SalesOrderHeader] WITH CHECK ADD

CONSTRAINT [PK\_SalesOrderHeader\_SalesOrderID] PRIMARY KEY CLUSTERED

(

[SalesOrderID]

) ON [PRIMARY];

GO

ALTER TABLE [Sales].[SalesOrderHeaderSalesReason] WITH CHECK ADD

CONSTRAINT [PK\_SalesOrderHeaderSalesReason\_SalesOrderID\_SalesReasonID] PRIMARY KEY CLUSTERED

(

[SalesOrderID],

[SalesReasonID]

) ON [PRIMARY];

GO

ALTER TABLE [Sales].[SalesPerson] WITH CHECK ADD

CONSTRAINT [PK\_SalesPerson\_BusinessEntityID] PRIMARY KEY CLUSTERED

(

[BusinessEntityID]

) ON [PRIMARY];

GO

ALTER TABLE [Sales].[SalesPersonQuotaHistory] WITH CHECK ADD

CONSTRAINT [PK\_SalesPersonQuotaHistory\_BusinessEntityID\_QuotaDate] PRIMARY KEY CLUSTERED

(

[BusinessEntityID],

[QuotaDate] --,

-- [ProductCategoryID]

) ON [PRIMARY];

GO

ALTER TABLE [Sales].[SalesReason] WITH CHECK ADD

CONSTRAINT [PK\_SalesReason\_SalesReasonID] PRIMARY KEY CLUSTERED

(

[SalesReasonID]

) ON [PRIMARY];

GO

ALTER TABLE [Sales].[SalesTaxRate] WITH CHECK ADD

CONSTRAINT [PK\_SalesTaxRate\_SalesTaxRateID] PRIMARY KEY CLUSTERED

(

[SalesTaxRateID]

) ON [PRIMARY];

GO

ALTER TABLE [Sales].[SalesTerritory] WITH CHECK ADD

CONSTRAINT [PK\_SalesTerritory\_TerritoryID] PRIMARY KEY CLUSTERED

(

[TerritoryID]

) ON [PRIMARY];

GO

ALTER TABLE [Sales].[SalesTerritoryHistory] WITH CHECK ADD

CONSTRAINT [PK\_SalesTerritoryHistory\_BusinessEntityID\_StartDate\_TerritoryID] PRIMARY KEY CLUSTERED

(

[BusinessEntityID], --Sales person

[StartDate],

[TerritoryID]

) ON [PRIMARY];

GO

ALTER TABLE [Production].[ScrapReason] WITH CHECK ADD

CONSTRAINT [PK\_ScrapReason\_ScrapReasonID] PRIMARY KEY CLUSTERED

(

[ScrapReasonID]

) ON [PRIMARY];

GO

ALTER TABLE [HumanResources].[Shift] WITH CHECK ADD

CONSTRAINT [PK\_Shift\_ShiftID] PRIMARY KEY CLUSTERED

(

[ShiftID]

) ON [PRIMARY];

GO

ALTER TABLE [Purchasing].[ShipMethod] WITH CHECK ADD

CONSTRAINT [PK\_ShipMethod\_ShipMethodID] PRIMARY KEY CLUSTERED

(

[ShipMethodID]

) ON [PRIMARY];

GO

ALTER TABLE [Sales].[ShoppingCartItem] WITH CHECK ADD

CONSTRAINT [PK\_ShoppingCartItem\_ShoppingCartItemID] PRIMARY KEY CLUSTERED

(

[ShoppingCartItemID]

) ON [PRIMARY];

GO

ALTER TABLE [Sales].[SpecialOffer] WITH CHECK ADD

CONSTRAINT [PK\_SpecialOffer\_SpecialOfferID] PRIMARY KEY CLUSTERED

(

[SpecialOfferID]

) ON [PRIMARY];

GO

ALTER TABLE [Sales].[SpecialOfferProduct] WITH CHECK ADD

CONSTRAINT [PK\_SpecialOfferProduct\_SpecialOfferID\_ProductID] PRIMARY KEY CLUSTERED

(

[SpecialOfferID],

[ProductID]

) ON [PRIMARY];

GO

GO

ALTER TABLE [Person].[StateProvince] WITH CHECK ADD

CONSTRAINT [PK\_StateProvince\_StateProvinceID] PRIMARY KEY CLUSTERED

(

[StateProvinceID]

) ON [PRIMARY];

GO

ALTER TABLE [Sales].[Store] WITH CHECK ADD

CONSTRAINT [PK\_Store\_BusinessEntityID] PRIMARY KEY CLUSTERED

(

[BusinessEntityID]

) ON [PRIMARY];

GO

ALTER TABLE [Production].[TransactionHistory] WITH CHECK ADD

CONSTRAINT [PK\_TransactionHistory\_TransactionID] PRIMARY KEY CLUSTERED

(

[TransactionID]

) ON [PRIMARY];

GO

ALTER TABLE [Production].[TransactionHistoryArchive] WITH CHECK ADD

CONSTRAINT [PK\_TransactionHistoryArchive\_TransactionID] PRIMARY KEY CLUSTERED

(

[TransactionID]

) ON [PRIMARY];

GO

ALTER TABLE [Production].[UnitMeasure] WITH CHECK ADD

CONSTRAINT [PK\_UnitMeasure\_UnitMeasureCode] PRIMARY KEY CLUSTERED

(

[UnitMeasureCode]

) ON [PRIMARY];

GO

ALTER TABLE [Purchasing].[Vendor] WITH CHECK ADD

CONSTRAINT [PK\_Vendor\_BusinessEntityID] PRIMARY KEY CLUSTERED

(

[BusinessEntityID]

) ON [PRIMARY];

GO

ALTER TABLE [Production].[WorkOrder] WITH CHECK ADD

CONSTRAINT [PK\_WorkOrder\_WorkOrderID] PRIMARY KEY CLUSTERED

(

[WorkOrderID]

) ON [PRIMARY];

GO

ALTER TABLE [Production].[WorkOrderRouting] WITH CHECK ADD

CONSTRAINT [PK\_WorkOrderRouting\_WorkOrderID\_ProductID\_OperationSequence] PRIMARY KEY CLUSTERED

(

[WorkOrderID],

[ProductID],

[OperationSequence]

) ON [PRIMARY];

GO

[TOP](#Top)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Add Indexes

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PRINT '';

PRINT '\*\*\* Adding Indexes';

GO

Index Name

Table Name

Column(s)

**CREATE UNIQUE INDEX [AK\_Address\_rowguid] ON [Person].[Address]([rowguid]) ON [PRIMARY];**

CREATE UNIQUE INDEX [IX\_Address\_AddressLine1\_AddressLine2\_City\_StateProvinceID\_PostalCode] ON [Person].[Address] ([AddressLine1], [AddressLine2], [City], [StateProvinceID], [PostalCode]) ON [PRIMARY];

CREATE INDEX [IX\_Address\_StateProvinceID] ON [Person].[Address]([StateProvinceID]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_AddressType\_rowguid] ON [Person].[AddressType]([rowguid]) ON [PRIMARY];

CREATE UNIQUE INDEX [AK\_AddressType\_Name] ON [Person].[AddressType]([Name]) ON [PRIMARY];

GO

CREATE INDEX [IX\_BillOfMaterials\_UnitMeasureCode] ON [Production].[BillOfMaterials]([UnitMeasureCode]) ON [PRIMARY];

CREATE UNIQUE CLUSTERED INDEX [AK\_BillOfMaterials\_ProductAssemblyID\_ComponentID\_StartDate] ON [Production].[BillOfMaterials]([ProductAssemblyID], [ComponentID], [StartDate]) ON [PRIMARY];

GO

**A table or view can contain the following types of indexes:**

* **Clustered**
  + Clustered indexes sort and store the data rows in the table or view based on their key values. These are the columns included in the index definition. There can be only one clustered index per table, because the data rows themselves can be sorted in only one order.
  + The only time the data rows in a table are stored in sorted order is when the table contains a clustered index. When a table has a clustered index, the table is called a clustered table. If a table has no clustered index, its data rows are stored in an unordered structure called a heap.
* **Nonclustered**
  + Nonclustered indexes have a structure separate from the data rows. A nonclustered index contains the nonclustered index key values and each key value entry has a pointer to the data row that contains the key value.
  + The pointer from an index row in a nonclustered index to a data row is called a row locator. The structure of the row locator depends on whether the data pages are stored in a heap or a clustered table. For a heap, a row locator is a pointer to the row. For a clustered table, the row locator is the clustered index key.
  + You can add nonkey columns to the leaf level of the nonclustered index to by-pass existing index key limits, and execute fully covered, indexed, queries. For more information, see [Create Indexes with Included Columns](https://docs.microsoft.com/en-us/sql/relational-databases/indexes/create-indexes-with-included-columns). For details about index key limits see [Maximum Capacity Specifications for SQL Server](https://docs.microsoft.com/en-us/sql/sql-server/maximum-capacity-specifications-for-sql-server).

Both clustered and nonclustered indexes can be unique. This means no two rows can have the same value for the index key. Otherwise, the index is not unique and multiple rows can share the same key value. For more information, see [Create Unique Indexes](https://docs.microsoft.com/en-us/sql/relational-databases/indexes/create-unique-indexes).

Indexes are automatically maintained for a table or view whenever the table data is modified.

**Create Indexes with Included Columns**

This topic describes how to add included (or nonkey) columns to extend the functionality of nonclustered indexes in SQL Server by using SQL Server Management Studio or Transact-SQL. By including nonkey columns, you can create nonclustered indexes that cover more queries. This is because the nonkey columns have the following benefits:+

* They can be data types not allowed as index key columns.
* They are not considered by the Database Engine when calculating the number of index key columns or index key size.

An index with nonkey columns can significantly improve query performance when all columns in the query are included in the index either as key or nonkey columns. Performance gains are achieved because the query optimizer can locate all the column values within the index; table or clustered index data is not accessed resulting in fewer disk I/O operations.

Note

When an index contains all the columns referenced by a query it is typically referred to as *covering the query*.

**Before You Begin**

Design Recommendations

* Redesign nonclustered indexes with a large index key size so that only columns used for searching and lookups are key columns. Make all other columns that cover the query into nonkey columns. In this way, you will have all columns needed to cover the query, but the index key itself is small and efficient.
* Include nonkey columns in a nonclustered index to avoid exceeding the current index size limitations of a maximum of 32 key columns and a maximum index key size of 1,700 bytes (16 key columns and 900 bytes prior to SQL Server 2016). The Database Engine does not consider nonkey columns when calculating the number of index key columns or index key size.

Limitations and Restrictions

* Nonkey columns can only be defined on nonclustered indexes.
* All data types except **text**, **ntext**, and **image** can be used as nonkey columns.
* Computed columns that are deterministic and either precise or imprecise can be nonkey columns. For more information, see [Indexes on Computed Columns](https://docs.microsoft.com/en-us/sql/relational-databases/indexes/indexes-on-computed-columns).
* Computed columns derived from **image**, **ntext**, and **text** data types can be nonkey columns as long as the computed column data type is allowed as a nonkey index column.
* Nonkey columns cannot be dropped from a table unless that table’s index is dropped first.
* Nonkey columns cannot be changed, except to do the following:
  + Change the nullability of the column from NOT NULL to NULL.
  + Increase the length of **varchar**, **nvarchar**, or **varbinary** columns.

CREATE UNIQUE INDEX [AK\_BusinessEntity\_rowguid] ON [Person].[BusinessEntity]([rowguid]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_BusinessEntityAddress\_rowguid] ON [Person].[BusinessEntityAddress]([rowguid]) ON [PRIMARY];

CREATE INDEX [IX\_BusinessEntityAddress\_AddressID] ON [Person].[BusinessEntityAddress]([AddressID]) ON [PRIMARY];

CREATE INDEX [IX\_BusinessEntityAddress\_AddressTypeID] ON [Person].[BusinessEntityAddress]([AddressTypeID]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_BusinessEntityContact\_rowguid] ON [Person].[BusinessEntityContact]([rowguid]) ON [PRIMARY];

CREATE INDEX [IX\_BusinessEntityContact\_PersonID] ON [Person].[BusinessEntityContact]([PersonID]) ON [PRIMARY];

CREATE INDEX [IX\_BusinessEntityContact\_ContactTypeID] ON [Person].[BusinessEntityContact]([ContactTypeID]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_ContactType\_Name] ON [Person].[ContactType]([Name]) ON [PRIMARY];

GO

CREATE INDEX [IX\_CountryRegionCurrency\_CurrencyCode] ON [Sales].[CountryRegionCurrency]([CurrencyCode]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_CountryRegion\_Name] ON [Person].[CountryRegion]([Name]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_CreditCard\_CardNumber] ON [Sales].[CreditCard]([CardNumber]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_Culture\_Name] ON [Production].[Culture]([Name]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_Currency\_Name] ON [Sales].[Currency]([Name]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_CurrencyRate\_CurrencyRateDate\_FromCurrencyCode\_ToCurrencyCode] ON [Sales].[CurrencyRate]([CurrencyRateDate], [FromCurrencyCode], [ToCurrencyCode]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_Customer\_rowguid] ON [Sales].[Customer]([rowguid]) ON [PRIMARY];

CREATE UNIQUE INDEX [AK\_Customer\_AccountNumber] ON [Sales].[Customer]([AccountNumber]) ON [PRIMARY];

CREATE INDEX [IX\_Customer\_TerritoryID] ON [Sales].[Customer]([TerritoryID]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_Department\_Name] ON [HumanResources].[Department]([Name]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_Document\_DocumentLevel\_DocumentNode] ON [Production].[Document] ([DocumentLevel], [DocumentNode]);

CREATE UNIQUE INDEX [AK\_Document\_rowguid] ON [Production].[Document]([rowguid]) ON [PRIMARY];

CREATE INDEX [IX\_Document\_FileName\_Revision] ON [Production].[Document]([FileName], [Revision]) ON [PRIMARY];

GO

CREATE INDEX [IX\_EmailAddress\_EmailAddress] ON [Person].[EmailAddress]([EmailAddress]) ON [PRIMARY];

GO

CREATE INDEX [IX\_Employee\_OrganizationNode] ON [HumanResources].[Employee] ([OrganizationNode]);

CREATE INDEX [IX\_Employee\_OrganizationLevel\_OrganizationNode] ON [HumanResources].[Employee] ([OrganizationLevel], [OrganizationNode]);

CREATE UNIQUE INDEX [AK\_Employee\_LoginID] ON [HumanResources].[Employee]([LoginID]) ON [PRIMARY];

CREATE UNIQUE INDEX [AK\_Employee\_NationalIDNumber] ON [HumanResources].[Employee]([NationalIDNumber]) ON [PRIMARY];

CREATE UNIQUE INDEX [AK\_Employee\_rowguid] ON [HumanResources].[Employee]([rowguid]) ON [PRIMARY];

GO

CREATE INDEX [IX\_EmployeeDepartmentHistory\_DepartmentID] ON [HumanResources].[EmployeeDepartmentHistory]([DepartmentID]) ON [PRIMARY];

CREATE INDEX [IX\_EmployeeDepartmentHistory\_ShiftID] ON [HumanResources].[EmployeeDepartmentHistory]([ShiftID]) ON [PRIMARY];

GO

CREATE INDEX [IX\_JobCandidate\_BusinessEntityID] ON [HumanResources].[JobCandidate]([BusinessEntityID]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_Location\_Name] ON [Production].[Location]([Name]) ON [PRIMARY];

GO

CREATE INDEX [IX\_Person\_LastName\_FirstName\_MiddleName] ON [Person].[Person] ([LastName], [FirstName], [MiddleName]) ON [PRIMARY];

CREATE UNIQUE INDEX [AK\_Person\_rowguid] ON [Person].[Person]([rowguid]) ON [PRIMARY];

CREATE INDEX [IX\_PersonPhone\_PhoneNumber] on [Person].[PersonPhone] ([PhoneNumber]) ON [PRIMARY];

CREATE UNIQUE INDEX [AK\_Product\_ProductNumber] ON [Production].[Product]([ProductNumber]) ON [PRIMARY];

CREATE UNIQUE INDEX [AK\_Product\_Name] ON [Production].[Product]([Name]) ON [PRIMARY];

CREATE UNIQUE INDEX [AK\_Product\_rowguid] ON [Production].[Product]([rowguid]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_ProductCategory\_Name] ON [Production].[ProductCategory]([Name]) ON [PRIMARY];

CREATE UNIQUE INDEX [AK\_ProductCategory\_rowguid] ON [Production].[ProductCategory]([rowguid]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_ProductDescription\_rowguid] ON [Production].[ProductDescription]([rowguid]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_ProductModel\_Name] ON [Production].[ProductModel]([Name]) ON [PRIMARY];

CREATE UNIQUE INDEX [AK\_ProductModel\_rowguid] ON [Production].[ProductModel]([rowguid]) ON [PRIMARY];

GO

CREATE NONCLUSTERED INDEX [IX\_ProductReview\_ProductID\_Name] ON [Production].[ProductReview]([ProductID], [ReviewerName]) INCLUDE ([Comments]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_ProductSubcategory\_Name] ON [Production].[ProductSubcategory]([Name]) ON [PRIMARY];

CREATE UNIQUE INDEX [AK\_ProductSubcategory\_rowguid] ON [Production].[ProductSubcategory]([rowguid]) ON [PRIMARY];

GO

CREATE INDEX [IX\_ProductVendor\_UnitMeasureCode] ON [Purchasing].[ProductVendor]([UnitMeasureCode]) ON [PRIMARY];

CREATE INDEX [IX\_ProductVendor\_BusinessEntityID] ON [Purchasing].[ProductVendor]([BusinessEntityID]) ON [PRIMARY];

GO

CREATE INDEX [IX\_PurchaseOrderDetail\_ProductID] ON [Purchasing].[PurchaseOrderDetail]([ProductID]) ON [PRIMARY];

GO

CREATE INDEX [IX\_PurchaseOrderHeader\_VendorID] ON [Purchasing].[PurchaseOrderHeader]([VendorID]) ON [PRIMARY];

CREATE INDEX [IX\_PurchaseOrderHeader\_EmployeeID] ON [Purchasing].[PurchaseOrderHeader]([EmployeeID]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_SalesOrderDetail\_rowguid] ON [Sales].[SalesOrderDetail]([rowguid]) ON [PRIMARY];

CREATE INDEX [IX\_SalesOrderDetail\_ProductID] ON [Sales].[SalesOrderDetail]([ProductID]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_SalesOrderHeader\_rowguid] ON [Sales].[SalesOrderHeader]([rowguid]) ON [PRIMARY];

CREATE UNIQUE INDEX [AK\_SalesOrderHeader\_SalesOrderNumber] ON [Sales].[SalesOrderHeader]([SalesOrderNumber]) ON [PRIMARY];

CREATE INDEX [IX\_SalesOrderHeader\_CustomerID] ON [Sales].[SalesOrderHeader]([CustomerID]) ON [PRIMARY];

CREATE INDEX [IX\_SalesOrderHeader\_SalesPersonID] ON [Sales].[SalesOrderHeader]([SalesPersonID]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_SalesPerson\_rowguid] ON [Sales].[SalesPerson]([rowguid]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_SalesPersonQuotaHistory\_rowguid] ON [Sales].[SalesPersonQuotaHistory]([rowguid]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_SalesTaxRate\_StateProvinceID\_TaxType] ON [Sales].[SalesTaxRate]([StateProvinceID], [TaxType]) ON [PRIMARY];

CREATE UNIQUE INDEX [AK\_SalesTaxRate\_rowguid] ON [Sales].[SalesTaxRate]([rowguid]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_SalesTerritory\_Name] ON [Sales].[SalesTerritory]([Name]) ON [PRIMARY];

CREATE UNIQUE INDEX [AK\_SalesTerritory\_rowguid] ON [Sales].[SalesTerritory]([rowguid]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_SalesTerritoryHistory\_rowguid] ON [Sales].[SalesTerritoryHistory]([rowguid]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_ScrapReason\_Name] ON [Production].[ScrapReason]([Name]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_Shift\_Name] ON [HumanResources].[Shift]([Name]) ON [PRIMARY];

CREATE UNIQUE INDEX [AK\_Shift\_StartTime\_EndTime] ON [HumanResources].[Shift]([StartTime], [EndTime]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_ShipMethod\_Name] ON [Purchasing].[ShipMethod]([Name]) ON [PRIMARY];

CREATE UNIQUE INDEX [AK\_ShipMethod\_rowguid] ON [Purchasing].[ShipMethod]([rowguid]) ON [PRIMARY];

GO

CREATE INDEX [IX\_ShoppingCartItem\_ShoppingCartID\_ProductID] ON [Sales].[ShoppingCartItem]([ShoppingCartID], [ProductID]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_SpecialOffer\_rowguid] ON [Sales].[SpecialOffer]([rowguid]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_SpecialOfferProduct\_rowguid] ON [Sales].[SpecialOfferProduct]([rowguid]) ON [PRIMARY];

CREATE INDEX [IX\_SpecialOfferProduct\_ProductID] ON [Sales].[SpecialOfferProduct]([ProductID]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_StateProvince\_Name] ON [Person].[StateProvince]([Name]) ON [PRIMARY];

CREATE UNIQUE INDEX [AK\_StateProvince\_StateProvinceCode\_CountryRegionCode] ON [Person].[StateProvince]([StateProvinceCode], [CountryRegionCode]) ON [PRIMARY];

CREATE UNIQUE INDEX [AK\_StateProvince\_rowguid] ON [Person].[StateProvince]([rowguid]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_Store\_rowguid] ON [Sales].[Store]([rowguid]) ON [PRIMARY];

CREATE INDEX [IX\_Store\_SalesPersonID] ON [Sales].[Store]([SalesPersonID]) ON [PRIMARY];

GO

CREATE INDEX [IX\_TransactionHistory\_ProductID] ON [Production].[TransactionHistory]([ProductID]) ON [PRIMARY];

CREATE INDEX [IX\_TransactionHistory\_ReferenceOrderID\_ReferenceOrderLineID] ON [Production].[TransactionHistory]([ReferenceOrderID], [ReferenceOrderLineID]) ON [PRIMARY];

GO

CREATE INDEX [IX\_TransactionHistoryArchive\_ProductID] ON [Production].[TransactionHistoryArchive]([ProductID]) ON [PRIMARY];

CREATE INDEX [IX\_TransactionHistoryArchive\_ReferenceOrderID\_ReferenceOrderLineID] ON [Production].[TransactionHistoryArchive]([ReferenceOrderID], [ReferenceOrderLineID]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_UnitMeasure\_Name] ON [Production].[UnitMeasure]([Name]) ON [PRIMARY];

GO

CREATE UNIQUE INDEX [AK\_Vendor\_AccountNumber] ON [Purchasing].[Vendor]([AccountNumber]) ON [PRIMARY];

GO

CREATE INDEX [IX\_WorkOrder\_ScrapReasonID] ON [Production].[WorkOrder]([ScrapReasonID]) ON [PRIMARY];

CREATE INDEX [IX\_WorkOrder\_ProductID] ON [Production].[WorkOrder]([ProductID]) ON [PRIMARY];

GO

CREATE INDEX [IX\_WorkOrderRouting\_ProductID] ON [Production].[WorkOrderRouting]([ProductID]) ON [PRIMARY];

GO

[TOP](#Top)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Create XML index for each XML column

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PRINT '';

PRINT '\*\*\* Creating XML index for each XML column';

GO

SET ARITHABORT ON;

SET QUOTED\_IDENTIFIER ON;

SET ANSI\_NULLS ON;

SET ANSI\_WARNINGS ON;

SET CONCAT\_NULL\_YIELDS\_NULL ON;

SET NUMERIC\_ROUNDABORT OFF;

CREATE PRIMARY XML INDEX [PXML\_Person\_AddContact] ON [Person].[Person]([AdditionalContactInfo]);

GO

CREATE PRIMARY XML INDEX [PXML\_Person\_Demographics] ON [Person].[Person]([Demographics]);

GO

CREATE XML INDEX [XMLPATH\_Person\_Demographics] ON [Person].[Person]([Demographics])

USING XML INDEX [PXML\_Person\_Demographics] FOR PATH;

GO

CREATE XML INDEX [XMLPROPERTY\_Person\_Demographics] ON [Person].[Person]([Demographics])

USING XML INDEX [PXML\_Person\_Demographics] FOR PROPERTY;

GO

CREATE XML INDEX [XMLVALUE\_Person\_Demographics] ON [Person].[Person]([Demographics])

USING XML INDEX [PXML\_Person\_Demographics] FOR VALUE;

GO

CREATE PRIMARY XML INDEX [PXML\_Store\_Demographics] ON [Sales].[Store]([Demographics]);

GO

CREATE PRIMARY XML INDEX [PXML\_ProductModel\_CatalogDescription] ON [Production].[ProductModel]([CatalogDescription]);

GO

CREATE PRIMARY XML INDEX [PXML\_ProductModel\_Instructions] ON [Production].[ProductModel]([Instructions]);

GO

[TOP](#Top)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Create Full Text catalog and indexes

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PRINT '';

PRINT '\*\*\* Creating Full Text catalog and indexes';

GO

--This creates a default FULLTEXT CATALOG where to logically store all the FTIndexes going to be created

**CREATE FULLTEXT CATALOG AW2016FullTextCatalog AS DEFAULT;**

GO

--This creates a FULLTEXT INDEX on ProductReview table. The index will cover the column 'Comments' which contains plain text data.

CREATE FULLTEXT INDEX ON Production.ProductReview(Comments) KEY INDEX PK\_ProductReview\_ProductReviewID;

GO

--This creates a FULLTEXT INDEX on JobCandidate table. The index will cover the column 'Resume' which contains XML data related with the candidates

--resumes.This is a good example of how iFTS will automatically call the XML filter in order to parse the data and store the information into the FTIndex

--created. No data type column is needed in this case as the datatype already provides the needed information

CREATE FULLTEXT INDEX ON HumanResources.JobCandidate(Resume) KEY INDEX PK\_JobCandidate\_JobCandidateID;

GO

--This creates a FULLTEXT INDEX on Document table. The index will cover the columns 'Document' and ‘DocumentSummary’. Note that the column ‘Document’

--contains binary data on a format specified by the 'FileExtension' column.This is a good example of how iFTS will automatically call the need

--iFilter associated with the 'FileExtension'associated with each row/document (in this case, all are .doc, which should be loaded into SQL from the OS by default)

CREATE FULLTEXT INDEX ON Production.Document(Document TYPE COLUMN FileExtension, DocumentSummary) KEY INDEX PK\_Document\_DocumentNode;

GO

[TOP](#Top)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Create Foreign key constraints

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PRINT '';

PRINT '\*\*\* Creating Foreign Key Constraints';

GO

**ALTER TABLE** [Person].[Address] ADD

CONSTRAINT [FK\_Address\_StateProvince\_StateProvinceID]

**FOREIGN KEY**

(

[StateProvinceID]

) REFERENCES [Person].[StateProvince](

[StateProvinceID]

);

GO

ALTER TABLE [Production].[BillOfMaterials] ADD

CONSTRAINT [FK\_BillOfMaterials\_Product\_ProductAssemblyID] FOREIGN KEY

(

[ProductAssemblyID]

) REFERENCES [Production].[Product](

[ProductID]

),

CONSTRAINT [FK\_BillOfMaterials\_Product\_ComponentID] FOREIGN KEY

(

[ComponentID]

) REFERENCES [Production].[Product](

[ProductID]

),

CONSTRAINT [FK\_BillOfMaterials\_UnitMeasure\_UnitMeasureCode] FOREIGN KEY

(

[UnitMeasureCode]

) REFERENCES [Production].[UnitMeasure](

[UnitMeasureCode]

);

GO

ALTER TABLE [Person].[BusinessEntityAddress] ADD

CONSTRAINT [FK\_BusinessEntityAddress\_Address\_AddressID] FOREIGN KEY

(

[AddressID]

) REFERENCES [Person].[Address](

[AddressID]

),

CONSTRAINT [FK\_BusinessEntityAddress\_AddressType\_AddressTypeID] FOREIGN KEY

(

[AddressTypeID]

) REFERENCES [Person].[AddressType](

[AddressTypeID]

),

CONSTRAINT [FK\_BusinessEntityAddress\_BusinessEntity\_BusinessEntityID] FOREIGN KEY

(

[BusinessEntityID]

) REFERENCES [Person].[BusinessEntity](

[BusinessEntityID]

);

GO

ALTER TABLE [Person].[BusinessEntityContact] ADD

CONSTRAINT [FK\_BusinessEntityContact\_Person\_PersonID] FOREIGN KEY

(

[PersonID]

) REFERENCES [Person].[Person](

[BusinessEntityID]

),

CONSTRAINT [FK\_BusinessEntityContact\_ContactType\_ContactTypeID] FOREIGN KEY

(

[ContactTypeID]

) REFERENCES [Person].[ContactType](

[ContactTypeID]

),

CONSTRAINT [FK\_BusinessEntityContact\_BusinessEntity\_BusinessEntityID] FOREIGN KEY

(

[BusinessEntityID]

) REFERENCES [Person].[BusinessEntity](

[BusinessEntityID]

);

GO

ALTER TABLE [Sales].[CountryRegionCurrency] ADD

CONSTRAINT [FK\_CountryRegionCurrency\_CountryRegion\_CountryRegionCode] FOREIGN KEY

(

[CountryRegionCode]

) REFERENCES [Person].[CountryRegion](

[CountryRegionCode]

),

CONSTRAINT [FK\_CountryRegionCurrency\_Currency\_CurrencyCode] FOREIGN KEY

(

[CurrencyCode]

) REFERENCES [Sales].[Currency](

[CurrencyCode]

);

GO

ALTER TABLE [Sales].[CurrencyRate] ADD

CONSTRAINT [FK\_CurrencyRate\_Currency\_FromCurrencyCode] FOREIGN KEY

(

[FromCurrencyCode]

) REFERENCES [Sales].[Currency](

[CurrencyCode]

),

CONSTRAINT [FK\_CurrencyRate\_Currency\_ToCurrencyCode] FOREIGN KEY

(

[ToCurrencyCode]

) REFERENCES [Sales].[Currency](

[CurrencyCode]

);

GO

ALTER TABLE [Sales].[Customer] ADD

CONSTRAINT [FK\_Customer\_Person\_PersonID] FOREIGN KEY

(

[PersonID]

) REFERENCES [Person].[Person](

[BusinessEntityID]

),

CONSTRAINT [FK\_Customer\_Store\_StoreID] FOREIGN KEY

(

[StoreID]

) REFERENCES [Sales].[Store](

[BusinessEntityID]

),

CONSTRAINT [FK\_Customer\_SalesTerritory\_TerritoryID] FOREIGN KEY

(

[TerritoryID]

) REFERENCES [Sales].[SalesTerritory](

[TerritoryID]

);

GO

ALTER TABLE [Production].[Document] ADD

CONSTRAINT [FK\_Document\_Employee\_Owner] FOREIGN KEY

(

[Owner]

) REFERENCES [HumanResources].[Employee](

[BusinessEntityID]

);

GO

ALTER TABLE [Person].[EmailAddress] ADD

CONSTRAINT [FK\_EmailAddress\_Person\_BusinessEntityID] FOREIGN KEY

(

[BusinessEntityID]

) REFERENCES [Person].[Person](

[BusinessEntityID]

);

GO

ALTER TABLE [HumanResources].[Employee] ADD

CONSTRAINT [FK\_Employee\_Person\_BusinessEntityID] FOREIGN KEY

(

[BusinessEntityID]

) REFERENCES [Person].[Person](

[BusinessEntityID]

);

GO

ALTER TABLE [HumanResources].[EmployeeDepartmentHistory] ADD

CONSTRAINT [FK\_EmployeeDepartmentHistory\_Department\_DepartmentID] FOREIGN KEY

(

[DepartmentID]

) REFERENCES [HumanResources].[Department](

[DepartmentID]

),

CONSTRAINT [FK\_EmployeeDepartmentHistory\_Employee\_BusinessEntityID] FOREIGN KEY

(

[BusinessEntityID]

) REFERENCES [HumanResources].[Employee](

[BusinessEntityID]

),

CONSTRAINT [FK\_EmployeeDepartmentHistory\_Shift\_ShiftID] FOREIGN KEY

(

[ShiftID]

) REFERENCES [HumanResources].[Shift](

[ShiftID]

);

GO

ALTER TABLE [HumanResources].[EmployeePayHistory] ADD

CONSTRAINT [FK\_EmployeePayHistory\_Employee\_BusinessEntityID] FOREIGN KEY

(

[BusinessEntityID]

) REFERENCES [HumanResources].[Employee](

[BusinessEntityID]

);

GO

ALTER TABLE [HumanResources].[JobCandidate] ADD

CONSTRAINT [FK\_JobCandidate\_Employee\_BusinessEntityID] FOREIGN KEY

(

[BusinessEntityID]

) REFERENCES [HumanResources].[Employee](

[BusinessEntityID]

);

GO

ALTER TABLE [Person].[Password] ADD

CONSTRAINT [FK\_Password\_Person\_BusinessEntityID] FOREIGN KEY

(

[BusinessEntityID]

) REFERENCES [Person].[Person](

[BusinessEntityID]

);

GO

ALTER TABLE [Person].[Person] ADD

CONSTRAINT [FK\_Person\_BusinessEntity\_BusinessEntityID] FOREIGN KEY

(

[BusinessEntityID]

) REFERENCES [Person].[BusinessEntity](

[BusinessEntityID]

);

GO

ALTER TABLE [Sales].[PersonCreditCard] ADD

CONSTRAINT [FK\_PersonCreditCard\_Person\_BusinessEntityID] FOREIGN KEY

(

[BusinessEntityID]

) REFERENCES [Person].[Person](

[BusinessEntityID]

),

CONSTRAINT [FK\_PersonCreditCard\_CreditCard\_CreditCardID] FOREIGN KEY

(

[CreditCardID]

) REFERENCES [Sales].[CreditCard](

[CreditCardID]

);

GO

ALTER TABLE [Person].[PersonPhone] ADD

CONSTRAINT [FK\_PersonPhone\_Person\_BusinessEntityID] FOREIGN KEY

(

[BusinessEntityID]

) REFERENCES [Person].[Person](

[BusinessEntityID]

),

CONSTRAINT [FK\_PersonPhone\_PhoneNumberType\_PhoneNumberTypeID] FOREIGN KEY

(

[PhoneNumberTypeID]

) REFERENCES [Person].[PhoneNumberType](

[PhoneNumberTypeID]

);

GO

ALTER TABLE [Production].[Product] ADD

CONSTRAINT [FK\_Product\_UnitMeasure\_SizeUnitMeasureCode] FOREIGN KEY

(

[SizeUnitMeasureCode]

) REFERENCES [Production].[UnitMeasure](

[UnitMeasureCode]

),

CONSTRAINT [FK\_Product\_UnitMeasure\_WeightUnitMeasureCode] FOREIGN KEY

(

[WeightUnitMeasureCode]

) REFERENCES [Production].[UnitMeasure](

[UnitMeasureCode]

),

CONSTRAINT [FK\_Product\_ProductModel\_ProductModelID] FOREIGN KEY

(

[ProductModelID]

) REFERENCES [Production].[ProductModel](

[ProductModelID]

),

CONSTRAINT [FK\_Product\_ProductSubcategory\_ProductSubcategoryID] FOREIGN KEY

(

[ProductSubcategoryID]

) REFERENCES [Production].[ProductSubcategory](

[ProductSubcategoryID]

);

GO

ALTER TABLE [Production].[ProductCostHistory] ADD

CONSTRAINT [FK\_ProductCostHistory\_Product\_ProductID] FOREIGN KEY

(

[ProductID]

) REFERENCES [Production].[Product](

[ProductID]

);

GO

ALTER TABLE [Production].[ProductDocument] ADD

CONSTRAINT [FK\_ProductDocument\_Product\_ProductID] FOREIGN KEY

(

[ProductID]

) REFERENCES [Production].[Product](

[ProductID]

),

CONSTRAINT [FK\_ProductDocument\_Document\_DocumentNode] FOREIGN KEY

(

[DocumentNode]

) REFERENCES [Production].[Document](

[DocumentNode]

);

GO

ALTER TABLE [Production].[ProductInventory] ADD

CONSTRAINT [FK\_ProductInventory\_Location\_LocationID] FOREIGN KEY

(

[LocationID]

) REFERENCES [Production].[Location](

[LocationID]

),

CONSTRAINT [FK\_ProductInventory\_Product\_ProductID] FOREIGN KEY

(

[ProductID]

) REFERENCES [Production].[Product](

[ProductID]

);

GO

ALTER TABLE [Production].[ProductListPriceHistory] ADD

CONSTRAINT [FK\_ProductListPriceHistory\_Product\_ProductID] FOREIGN KEY

(

[ProductID]

) REFERENCES [Production].[Product](

[ProductID]

);

GO

ALTER TABLE [Production].[ProductModelIllustration] ADD

CONSTRAINT [FK\_ProductModelIllustration\_ProductModel\_ProductModelID] FOREIGN KEY

(

[ProductModelID]

) REFERENCES [Production].[ProductModel](

[ProductModelID]

),

CONSTRAINT [FK\_ProductModelIllustration\_Illustration\_IllustrationID] FOREIGN KEY

(

[IllustrationID]

) REFERENCES [Production].[Illustration](

[IllustrationID]

);

GO

ALTER TABLE [Production].[ProductModelProductDescriptionCulture] ADD

CONSTRAINT [FK\_ProductModelProductDescriptionCulture\_ProductDescription\_ProductDescriptionID] FOREIGN KEY

(

[ProductDescriptionID]

) REFERENCES [Production].[ProductDescription](

[ProductDescriptionID]

),

CONSTRAINT [FK\_ProductModelProductDescriptionCulture\_Culture\_CultureID] FOREIGN KEY

(

[CultureID]

) REFERENCES [Production].[Culture]

(

[CultureID]

),

CONSTRAINT [FK\_ProductModelProductDescriptionCulture\_ProductModel\_ProductModelID] FOREIGN KEY

(

[ProductModelID]

) REFERENCES [Production].[ProductModel](

[ProductModelID]

);

GO

ALTER TABLE [Production].[ProductProductPhoto] ADD

CONSTRAINT [FK\_ProductProductPhoto\_Product\_ProductID] FOREIGN KEY

(

[ProductID]

) REFERENCES [Production].[Product](

[ProductID]

),

CONSTRAINT [FK\_ProductProductPhoto\_ProductPhoto\_ProductPhotoID] FOREIGN KEY

(

[ProductPhotoID]

) REFERENCES [Production].[ProductPhoto](

[ProductPhotoID]

);

GO

ALTER TABLE [Production].[ProductReview] ADD

CONSTRAINT [FK\_ProductReview\_Product\_ProductID] FOREIGN KEY

(

[ProductID]

) REFERENCES [Production].[Product](

[ProductID]

);

GO

ALTER TABLE [Production].[ProductSubcategory] ADD

CONSTRAINT [FK\_ProductSubcategory\_ProductCategory\_ProductCategoryID] FOREIGN KEY

(

[ProductCategoryID]

) REFERENCES [Production].[ProductCategory](

[ProductCategoryID]

);

GO

ALTER TABLE [Purchasing].[ProductVendor] ADD

CONSTRAINT [FK\_ProductVendor\_Product\_ProductID] FOREIGN KEY

(

[ProductID]

) REFERENCES [Production].[Product](

[ProductID]

),

CONSTRAINT [FK\_ProductVendor\_UnitMeasure\_UnitMeasureCode] FOREIGN KEY

(

[UnitMeasureCode]

) REFERENCES [Production].[UnitMeasure](

[UnitMeasureCode]

),

CONSTRAINT [FK\_ProductVendor\_Vendor\_BusinessEntityID] FOREIGN KEY

(

[BusinessEntityID]

) REFERENCES [Purchasing].[Vendor](

[BusinessEntityID]

);

GO

ALTER TABLE [Purchasing].[PurchaseOrderDetail] ADD

CONSTRAINT [FK\_PurchaseOrderDetail\_Product\_ProductID] FOREIGN KEY

(

[ProductID]

) REFERENCES [Production].[Product](

[ProductID]

),

CONSTRAINT [FK\_PurchaseOrderDetail\_PurchaseOrderHeader\_PurchaseOrderID] FOREIGN KEY

(

[PurchaseOrderID]

) REFERENCES [Purchasing].[PurchaseOrderHeader](

[PurchaseOrderID]

);

GO

ALTER TABLE [Purchasing].[PurchaseOrderHeader] ADD

CONSTRAINT [FK\_PurchaseOrderHeader\_Employee\_EmployeeID] FOREIGN KEY

(

[EmployeeID]

) REFERENCES [HumanResources].[Employee](

[BusinessEntityID]

),

CONSTRAINT [FK\_PurchaseOrderHeader\_Vendor\_VendorID] FOREIGN KEY

(

[VendorID]

) REFERENCES [Purchasing].[Vendor](

[BusinessEntityID]

),

CONSTRAINT [FK\_PurchaseOrderHeader\_ShipMethod\_ShipMethodID] FOREIGN KEY

(

[ShipMethodID]

) REFERENCES [Purchasing].[ShipMethod](

[ShipMethodID]

);

GO

ALTER TABLE [Sales].[SalesOrderDetail] ADD

CONSTRAINT [FK\_SalesOrderDetail\_SalesOrderHeader\_SalesOrderID] FOREIGN KEY

(

[SalesOrderID]

) REFERENCES [Sales].[SalesOrderHeader](

[SalesOrderID]

) ON DELETE CASCADE,

CONSTRAINT [FK\_SalesOrderDetail\_SpecialOfferProduct\_SpecialOfferIDProductID] FOREIGN KEY

(

[SpecialOfferID],

[ProductID]

) REFERENCES [Sales].[SpecialOfferProduct](

[SpecialOfferID],

[ProductID]

);

GO

ALTER TABLE [Sales].[SalesOrderHeader] ADD

CONSTRAINT [FK\_SalesOrderHeader\_Address\_BillToAddressID] FOREIGN KEY

(

[BillToAddressID]

) REFERENCES [Person].[Address](

[AddressID]

),

CONSTRAINT [FK\_SalesOrderHeader\_Address\_ShipToAddressID] FOREIGN KEY

(

[ShipToAddressID]

) REFERENCES [Person].[Address](

[AddressID]

),

CONSTRAINT [FK\_SalesOrderHeader\_CreditCard\_CreditCardID] FOREIGN KEY

(

[CreditCardID]

) REFERENCES [Sales].[CreditCard](

[CreditCardID]

),

CONSTRAINT [FK\_SalesOrderHeader\_CurrencyRate\_CurrencyRateID] FOREIGN KEY

(

[CurrencyRateID]

) REFERENCES [Sales].[CurrencyRate](

[CurrencyRateID]

),

CONSTRAINT [FK\_SalesOrderHeader\_Customer\_CustomerID] FOREIGN KEY

(

[CustomerID]

) REFERENCES [Sales].[Customer](

[CustomerID]

),

CONSTRAINT [FK\_SalesOrderHeader\_SalesPerson\_SalesPersonID] FOREIGN KEY

(

[SalesPersonID]

) REFERENCES [Sales].[SalesPerson](

[BusinessEntityID]

),

CONSTRAINT [FK\_SalesOrderHeader\_ShipMethod\_ShipMethodID] FOREIGN KEY

(

[ShipMethodID]

) REFERENCES [Purchasing].[ShipMethod](

[ShipMethodID]

),

CONSTRAINT [FK\_SalesOrderHeader\_SalesTerritory\_TerritoryID] FOREIGN KEY

(

[TerritoryID]

) REFERENCES [Sales].[SalesTerritory](

[TerritoryID]

);

GO

ALTER TABLE [Sales].[SalesOrderHeaderSalesReason] ADD

CONSTRAINT [FK\_SalesOrderHeaderSalesReason\_SalesReason\_SalesReasonID] FOREIGN KEY

(

[SalesReasonID]

) REFERENCES [Sales].[SalesReason](

[SalesReasonID]

),

CONSTRAINT [FK\_SalesOrderHeaderSalesReason\_SalesOrderHeader\_SalesOrderID] FOREIGN KEY

(

[SalesOrderID]

) REFERENCES [Sales].[SalesOrderHeader](

[SalesOrderID]

) ON DELETE CASCADE;

GO

ALTER TABLE [Sales].[SalesPerson] ADD

CONSTRAINT [FK\_SalesPerson\_Employee\_BusinessEntityID] FOREIGN KEY

(

[BusinessEntityID]

) REFERENCES [HumanResources].[Employee](

[BusinessEntityID]

),

CONSTRAINT [FK\_SalesPerson\_SalesTerritory\_TerritoryID] FOREIGN KEY

(

[TerritoryID]

) REFERENCES [Sales].[SalesTerritory](

[TerritoryID]

);

GO

ALTER TABLE [Sales].[SalesPersonQuotaHistory] ADD

CONSTRAINT [FK\_SalesPersonQuotaHistory\_SalesPerson\_BusinessEntityID] FOREIGN KEY

(

[BusinessEntityID]

) REFERENCES [Sales].[SalesPerson](

[BusinessEntityID]

);

GO

ALTER TABLE [Sales].[SalesTaxRate] ADD

CONSTRAINT [FK\_SalesTaxRate\_StateProvince\_StateProvinceID] FOREIGN KEY

(

[StateProvinceID]

) REFERENCES [Person].[StateProvince](

[StateProvinceID]

);

GO

ALTER TABLE [Sales].[SalesTerritory] ADD

CONSTRAINT [FK\_SalesTerritory\_CountryRegion\_CountryRegionCode] FOREIGN KEY

(

[CountryRegionCode]

) REFERENCES [Person].[CountryRegion] (

[CountryRegionCode]

);

GO

ALTER TABLE [Sales].[SalesTerritoryHistory] ADD

CONSTRAINT [FK\_SalesTerritoryHistory\_SalesPerson\_BusinessEntityID] FOREIGN KEY

(

[BusinessEntityID]

) REFERENCES [Sales].[SalesPerson](

[BusinessEntityID]

),

CONSTRAINT [FK\_SalesTerritoryHistory\_SalesTerritory\_TerritoryID] FOREIGN KEY

(

[TerritoryID]

) REFERENCES [Sales].[SalesTerritory](

[TerritoryID]

);

GO

ALTER TABLE [Sales].[ShoppingCartItem] ADD

CONSTRAINT [FK\_ShoppingCartItem\_Product\_ProductID] FOREIGN KEY

(

[ProductID]

) REFERENCES [Production].[Product](

[ProductID]

);

GO

ALTER TABLE [Sales].[SpecialOfferProduct] ADD

CONSTRAINT [FK\_SpecialOfferProduct\_Product\_ProductID] FOREIGN KEY

(

[ProductID]

) REFERENCES [Production].[Product](

[ProductID]

),

CONSTRAINT [FK\_SpecialOfferProduct\_SpecialOffer\_SpecialOfferID] FOREIGN KEY

(

[SpecialOfferID]

) REFERENCES [Sales].[SpecialOffer](

[SpecialOfferID]

);

GO

ALTER TABLE [Person].[StateProvince] ADD

CONSTRAINT [FK\_StateProvince\_CountryRegion\_CountryRegionCode] FOREIGN KEY

(

[CountryRegionCode]

) REFERENCES [Person].[CountryRegion](

[CountryRegionCode]

),

CONSTRAINT [FK\_StateProvince\_SalesTerritory\_TerritoryID] FOREIGN KEY

(

[TerritoryID]

) REFERENCES [Sales].[SalesTerritory](

[TerritoryID]

);

GO

ALTER TABLE [Sales].[Store] ADD

CONSTRAINT [FK\_Store\_BusinessEntity\_BusinessEntityID] FOREIGN KEY

(

[BusinessEntityID]

) REFERENCES [Person].[BusinessEntity](

[BusinessEntityID]

),

CONSTRAINT [FK\_Store\_SalesPerson\_SalesPersonID] FOREIGN KEY

(

[SalesPersonID]

) REFERENCES [Sales].[SalesPerson](

[BusinessEntityID]

);

GO

ALTER TABLE [Production].[TransactionHistory] ADD

CONSTRAINT [FK\_TransactionHistory\_Product\_ProductID] FOREIGN KEY

(

[ProductID]

) REFERENCES [Production].[Product](

[ProductID]

);

GO

ALTER TABLE [Purchasing].[Vendor] ADD

CONSTRAINT [FK\_Vendor\_BusinessEntity\_BusinessEntityID] FOREIGN KEY

(

[BusinessEntityID]

) REFERENCES [Person].[BusinessEntity](

[BusinessEntityID]

);

GO

ALTER TABLE [Production].[WorkOrder] ADD

CONSTRAINT [FK\_WorkOrder\_Product\_ProductID] FOREIGN KEY

(

[ProductID]

) REFERENCES [Production].[Product](

[ProductID]

),

CONSTRAINT [FK\_WorkOrder\_ScrapReason\_ScrapReasonID] FOREIGN KEY

(

[ScrapReasonID]

) REFERENCES [Production].[ScrapReason](

[ScrapReasonID]

);

GO

ALTER TABLE [Production].[WorkOrderRouting] ADD

CONSTRAINT [FK\_WorkOrderRouting\_Location\_LocationID] FOREIGN KEY

(

[LocationID]

) REFERENCES [Production].[Location](

[LocationID]

),

CONSTRAINT [FK\_WorkOrderRouting\_WorkOrder\_WorkOrderID] FOREIGN KEY

(

[WorkOrderID]

) REFERENCES [Production].[WorkOrder](

[WorkOrderID]

);

GO

[TOP](#Top)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Add table triggers.

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PRINT '';

PRINT '\*\*\* Creating Table Triggers';

GO

CREATE TRIGGER [HumanResources].[dEmployee] ON [HumanResources].[Employee]

INSTEAD OF DELETE NOT FOR REPLICATION AS

BEGIN

DECLARE @Count int;

SET @Count = @@ROWCOUNT;

IF @Count = 0

RETURN;

SET NOCOUNT ON;

BEGIN

RAISERROR

(N'Employees cannot be deleted. They can only be marked as not current.', -- Message

10, -- Severity.

1); -- State.

-- Rollback any active or uncommittable transactions

IF @@TRANCOUNT > 0

BEGIN

ROLLBACK TRANSACTION;

END

END;

END;

GO

CREATE TRIGGER [Person].[iuPerson] ON [Person].[Person]

AFTER INSERT, UPDATE NOT FOR REPLICATION AS

BEGIN

DECLARE @Count int;

SET @Count = @@ROWCOUNT;

IF @Count = 0

RETURN;

SET NOCOUNT ON;

IF UPDATE([BusinessEntityID]) OR UPDATE([Demographics])

BEGIN

UPDATE [Person].[Person]

SET [Person].[Person].[Demographics] = N'<IndividualSurvey xmlns="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey">

<TotalPurchaseYTD>0.00</TotalPurchaseYTD>

</IndividualSurvey>'

FROM inserted

WHERE [Person].[Person].[BusinessEntityID] = inserted.[BusinessEntityID]

AND inserted.[Demographics] IS NULL;

UPDATE [Person].[Person]

SET [Demographics].modify(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey";

insert <TotalPurchaseYTD>0.00</TotalPurchaseYTD>

as first

into (/IndividualSurvey)[1]')

FROM inserted

WHERE [Person].[Person].[BusinessEntityID] = inserted.[BusinessEntityID]

AND inserted.[Demographics] IS NOT NULL

AND inserted.[Demographics].exist(N'declare default element namespace

"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey";

/IndividualSurvey/TotalPurchaseYTD') <> 1;

END;

END;

GO

CREATE TRIGGER [Purchasing].[iPurchaseOrderDetail] ON [Purchasing].[PurchaseOrderDetail]

AFTER INSERT AS

BEGIN

DECLARE @Count int;

SET @Count = @@ROWCOUNT;

IF @Count = 0

RETURN;

SET NOCOUNT ON;

BEGIN TRY

INSERT INTO [Production].[TransactionHistory]

([ProductID]

,[ReferenceOrderID]

,[ReferenceOrderLineID]

,[TransactionType]

,[TransactionDate]

,[Quantity]

,[ActualCost])

SELECT

inserted.[ProductID]

,inserted.[PurchaseOrderID]

,inserted.[PurchaseOrderDetailID]

,'P'

,GETDATE()

,inserted.[OrderQty]

,inserted.[UnitPrice]

FROM inserted

INNER JOIN [Purchasing].[PurchaseOrderHeader]

ON inserted.[PurchaseOrderID] = [Purchasing].[PurchaseOrderHeader].[PurchaseOrderID];

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

-- PurchaseOrderHeader trigger to fire which will update the RevisionNumber.

UPDATE [Purchasing].[PurchaseOrderHeader]

SET [Purchasing].[PurchaseOrderHeader].[SubTotal] =

(SELECT SUM([Purchasing].[PurchaseOrderDetail].[LineTotal])

FROM [Purchasing].[PurchaseOrderDetail]

WHERE [Purchasing].[PurchaseOrderHeader].[PurchaseOrderID] = [Purchasing].[PurchaseOrderDetail].[PurchaseOrderID])

WHERE [Purchasing].[PurchaseOrderHeader].[PurchaseOrderID] IN (SELECT inserted.[PurchaseOrderID] FROM inserted);

END TRY

BEGIN CATCH

EXECUTE [dbo].[uspPrintError];

-- Rollback any active or uncommittable transactions before

-- inserting information in the ErrorLog

IF @@TRANCOUNT > 0

BEGIN

ROLLBACK TRANSACTION;

END

EXECUTE [dbo].[uspLogError];

END CATCH;

END;

GO

CREATE TRIGGER [Purchasing].[uPurchaseOrderDetail] ON [Purchasing].[PurchaseOrderDetail]

AFTER UPDATE AS

BEGIN

DECLARE @Count int;

SET @Count = @@ROWCOUNT;

IF @Count = 0

RETURN;

SET NOCOUNT ON;

BEGIN TRY

IF UPDATE([ProductID]) OR UPDATE([OrderQty]) OR UPDATE([UnitPrice])

-- Insert record into TransactionHistory

BEGIN

INSERT INTO [Production].[TransactionHistory]

([ProductID]

,[ReferenceOrderID]

,[ReferenceOrderLineID]

,[TransactionType]

,[TransactionDate]

,[Quantity]

,[ActualCost])

SELECT

inserted.[ProductID]

,inserted.[PurchaseOrderID]

,inserted.[PurchaseOrderDetailID]

,'P'

,GETDATE()

,inserted.[OrderQty]

,inserted.[UnitPrice]

FROM inserted

INNER JOIN [Purchasing].[PurchaseOrderDetail]

ON inserted.[PurchaseOrderID] = [Purchasing].[PurchaseOrderDetail].[PurchaseOrderID];

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

-- PurchaseOrderHeader trigger to fire which will update the RevisionNumber.

UPDATE [Purchasing].[PurchaseOrderHeader]

SET [Purchasing].[PurchaseOrderHeader].[SubTotal] =

(SELECT SUM([Purchasing].[PurchaseOrderDetail].[LineTotal])

FROM [Purchasing].[PurchaseOrderDetail]

WHERE [Purchasing].[PurchaseOrderHeader].[PurchaseOrderID]

= [Purchasing].[PurchaseOrderDetail].[PurchaseOrderID])

WHERE [Purchasing].[PurchaseOrderHeader].[PurchaseOrderID]

IN (SELECT inserted.[PurchaseOrderID] FROM inserted);

UPDATE [Purchasing].[PurchaseOrderDetail]

SET [Purchasing].[PurchaseOrderDetail].[ModifiedDate] = GETDATE()

FROM inserted

WHERE inserted.[PurchaseOrderID] = [Purchasing].[PurchaseOrderDetail].[PurchaseOrderID]

AND inserted.[PurchaseOrderDetailID] = [Purchasing].[PurchaseOrderDetail].[PurchaseOrderDetailID];

END;

END TRY

BEGIN CATCH

EXECUTE [dbo].[uspPrintError];

-- Rollback any active or uncommittable transactions before

-- inserting information in the ErrorLog

IF @@TRANCOUNT > 0

BEGIN

ROLLBACK TRANSACTION;

END

EXECUTE [dbo].[uspLogError];

END CATCH;

END;

GO

CREATE TRIGGER [Purchasing].[uPurchaseOrderHeader] ON [Purchasing].[PurchaseOrderHeader]

AFTER UPDATE AS

BEGIN

DECLARE @Count int;

SET @Count = @@ROWCOUNT;

IF @Count = 0

RETURN;

SET NOCOUNT ON;

BEGIN TRY

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

IF NOT UPDATE([Status])

BEGIN

UPDATE [Purchasing].[PurchaseOrderHeader]

SET [Purchasing].[PurchaseOrderHeader].[RevisionNumber] =

[Purchasing].[PurchaseOrderHeader].[RevisionNumber] + 1

WHERE [Purchasing].[PurchaseOrderHeader].[PurchaseOrderID] IN

(SELECT inserted.[PurchaseOrderID] FROM inserted);

END;

END TRY

BEGIN CATCH

EXECUTE [dbo].[uspPrintError];

-- Rollback any active or uncommittable transactions before

-- inserting information in the ErrorLog

IF @@TRANCOUNT > 0

BEGIN

ROLLBACK TRANSACTION;

END

EXECUTE [dbo].[uspLogError];

END CATCH;

END;

GO

CREATE TRIGGER [Sales].[iduSalesOrderDetail] ON [Sales].[SalesOrderDetail]

AFTER INSERT, DELETE, UPDATE AS

BEGIN

DECLARE @Count int;

SET @Count = @@ROWCOUNT;

IF @Count = 0

RETURN;

SET NOCOUNT ON;

BEGIN TRY

-- If inserting or updating these columns

IF UPDATE([ProductID]) OR UPDATE([OrderQty]) OR UPDATE([UnitPrice]) OR UPDATE([UnitPriceDiscount])

-- Insert record into TransactionHistory

BEGIN

INSERT INTO [Production].[TransactionHistory]

([ProductID]

,[ReferenceOrderID]

,[ReferenceOrderLineID]

,[TransactionType]

,[TransactionDate]

,[Quantity]

,[ActualCost])

SELECT

inserted.[ProductID]

,inserted.[SalesOrderID]

,inserted.[SalesOrderDetailID]

,'S'

,GETDATE()

,inserted.[OrderQty]

,inserted.[UnitPrice]

FROM inserted

INNER JOIN [Sales].[SalesOrderHeader]

ON inserted.[SalesOrderID] = [Sales].[SalesOrderHeader].[SalesOrderID];

UPDATE [Person].[Person]

SET [Demographics].modify('declare default element namespace

"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey";

replace value of (/IndividualSurvey/TotalPurchaseYTD)[1]

with data(/IndividualSurvey/TotalPurchaseYTD)[1] + sql:column ("inserted.LineTotal")')

FROM inserted

INNER JOIN [Sales].[SalesOrderHeader] AS SOH

ON inserted.[SalesOrderID] = SOH.[SalesOrderID]

INNER JOIN [Sales].[Customer] AS C

ON SOH.[CustomerID] = C.[CustomerID]

WHERE C.[PersonID] = [Person].[Person].[BusinessEntityID];

END;

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

-- SalesOrderHeader trigger to fire which will update the RevisionNumber.

UPDATE [Sales].[SalesOrderHeader]

SET [Sales].[SalesOrderHeader].[SubTotal] =

(SELECT SUM([Sales].[SalesOrderDetail].[LineTotal])

FROM [Sales].[SalesOrderDetail]

WHERE [Sales].[SalesOrderHeader].[SalesOrderID] = [Sales].[SalesOrderDetail].[SalesOrderID])

WHERE [Sales].[SalesOrderHeader].[SalesOrderID] IN (SELECT inserted.[SalesOrderID] FROM inserted);

UPDATE [Person].[Person]

SET [Demographics].modify('declare default element namespace

"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey";

replace value of (/IndividualSurvey/TotalPurchaseYTD)[1]

with data(/IndividualSurvey/TotalPurchaseYTD)[1] - sql:column("deleted.LineTotal")')

FROM deleted

INNER JOIN [Sales].[SalesOrderHeader]

ON deleted.[SalesOrderID] = [Sales].[SalesOrderHeader].[SalesOrderID]

INNER JOIN [Sales].[Customer]

ON [Sales].[Customer].[CustomerID] = [Sales].[SalesOrderHeader].[CustomerID]

WHERE [Sales].[Customer].[PersonID] = [Person].[Person].[BusinessEntityID];

END TRY

BEGIN CATCH

EXECUTE [dbo].[uspPrintError];

-- Rollback any active or uncommittable transactions before

-- inserting information in the ErrorLog

IF @@TRANCOUNT > 0

BEGIN

ROLLBACK TRANSACTION;

END

EXECUTE [dbo].[uspLogError];

END CATCH;

END;

GO

CREATE TRIGGER [Sales].[uSalesOrderHeader] ON [Sales].[SalesOrderHeader]

AFTER UPDATE NOT FOR REPLICATION AS

BEGIN

DECLARE @Count int;

SET @Count = @@ROWCOUNT;

IF @Count = 0

RETURN;

SET NOCOUNT ON;

BEGIN TRY

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

IF NOT UPDATE([Status])

BEGIN

UPDATE [Sales].[SalesOrderHeader]

SET [Sales].[SalesOrderHeader].[RevisionNumber] =

[Sales].[SalesOrderHeader].[RevisionNumber] + 1

WHERE [Sales].[SalesOrderHeader].[SalesOrderID] IN

(SELECT inserted.[SalesOrderID] FROM inserted);

END;

-- Update the SalesPerson SalesYTD when SubTotal is updated

IF UPDATE([SubTotal])

BEGIN

DECLARE @StartDate datetime,

@EndDate datetime

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

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

UPDATE [Sales].[SalesPerson]

SET [Sales].[SalesPerson].[SalesYTD] =

(SELECT SUM([Sales].[SalesOrderHeader].[SubTotal])

FROM [Sales].[SalesOrderHeader]

WHERE [Sales].[SalesPerson].[BusinessEntityID] = [Sales].[SalesOrderHeader].[SalesPersonID]

AND ([Sales].[SalesOrderHeader].[Status] = 5) -- Shipped

AND [Sales].[SalesOrderHeader].[OrderDate] BETWEEN @StartDate AND @EndDate)

WHERE [Sales].[SalesPerson].[BusinessEntityID]

IN (SELECT DISTINCT inserted.[SalesPersonID] FROM inserted

WHERE inserted.[OrderDate] BETWEEN @StartDate AND @EndDate);

-- Update the SalesTerritory SalesYTD when SubTotal is updated

UPDATE [Sales].[SalesTerritory]

SET [Sales].[SalesTerritory].[SalesYTD] =

(SELECT SUM([Sales].[SalesOrderHeader].[SubTotal])

FROM [Sales].[SalesOrderHeader]

WHERE [Sales].[SalesTerritory].[TerritoryID] = [Sales].[SalesOrderHeader].[TerritoryID]

AND ([Sales].[SalesOrderHeader].[Status] = 5) -- Shipped

AND [Sales].[SalesOrderHeader].[OrderDate] BETWEEN @StartDate AND @EndDate)

WHERE [Sales].[SalesTerritory].[TerritoryID]

IN (SELECT DISTINCT inserted.[TerritoryID] FROM inserted

WHERE inserted.[OrderDate] BETWEEN @StartDate AND @EndDate);

END;

END TRY

BEGIN CATCH

EXECUTE [dbo].[uspPrintError];

-- Rollback any active or uncommittable transactions before

-- inserting information in the ErrorLog

IF @@TRANCOUNT > 0

BEGIN

ROLLBACK TRANSACTION;

END

EXECUTE [dbo].[uspLogError];

END CATCH;

END;

GO

CREATE TRIGGER [Purchasing].[dVendor] ON [Purchasing].[Vendor]

INSTEAD OF DELETE NOT FOR REPLICATION AS

BEGIN

DECLARE @Count int;

SET @Count = @@ROWCOUNT;

IF @Count = 0

RETURN;

SET NOCOUNT ON;

BEGIN TRY

DECLARE @DeleteCount int;

SELECT @DeleteCount = COUNT(\*) FROM deleted;

IF @DeleteCount > 0

BEGIN

RAISERROR

(N'Vendors cannot be deleted. They can only be marked as not active.', -- Message

10, -- Severity.

1); -- State.

-- Rollback any active or uncommittable transactions

IF @@TRANCOUNT > 0

BEGIN

ROLLBACK TRANSACTION;

END

END;

END TRY

BEGIN CATCH

EXECUTE [dbo].[uspPrintError];

-- Rollback any active or uncommittable transactions before

-- inserting information in the ErrorLog

IF @@TRANCOUNT > 0

BEGIN

ROLLBACK TRANSACTION;

END

EXECUTE [dbo].[uspLogError];

END CATCH;

END;

GO

CREATE TRIGGER [Production].[iWorkOrder] ON [Production].[WorkOrder]

AFTER INSERT AS

BEGIN

DECLARE @Count int;

SET @Count = @@ROWCOUNT;

IF @Count = 0

RETURN;

SET NOCOUNT ON;

BEGIN TRY

INSERT INTO [Production].[TransactionHistory](

[ProductID]

,[ReferenceOrderID]

,[TransactionType]

,[TransactionDate]

,[Quantity]

,[ActualCost])

SELECT

inserted.[ProductID]

,inserted.[WorkOrderID]

,'W'

,GETDATE()

,inserted.[OrderQty]

,0

FROM inserted;

END TRY

BEGIN CATCH

EXECUTE [dbo].[uspPrintError];

-- Rollback any active or uncommittable transactions before

-- inserting information in the ErrorLog

IF @@TRANCOUNT > 0

BEGIN

ROLLBACK TRANSACTION;

END

EXECUTE [dbo].[uspLogError];

END CATCH;

END;

GO

CREATE TRIGGER [Production].[uWorkOrder] ON [Production].[WorkOrder]

AFTER UPDATE AS

BEGIN

DECLARE @Count int;

SET @Count = @@ROWCOUNT;

IF @Count = 0

RETURN;

SET NOCOUNT ON;

BEGIN TRY

IF UPDATE([ProductID]) OR UPDATE([OrderQty])

BEGIN

INSERT INTO [Production].[TransactionHistory](

[ProductID]

,[ReferenceOrderID]

,[TransactionType]

,[TransactionDate]

,[Quantity])

SELECT

inserted.[ProductID]

,inserted.[WorkOrderID]

,'W'

,GETDATE()

,inserted.[OrderQty]

FROM inserted;

END;

END TRY

BEGIN CATCH

EXECUTE [dbo].[uspPrintError];

-- Rollback any active or uncommittable transactions before

-- inserting information in the ErrorLog

IF @@TRANCOUNT > 0

BEGIN

ROLLBACK TRANSACTION;

END

EXECUTE [dbo].[uspLogError];

END CATCH;

END;

GO

[TOP](#Top)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Add database views.

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PRINT '';

PRINT '\*\*\* Creating Table Views';

GO

**CREATE VIEW** [Person].[vAdditionalContactInfo]

AS

SELECT

[BusinessEntityID]

,[FirstName]

,[MiddleName]

,[LastName]

,[ContactInfo].ref.value

(N'declare namespace ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";

declare namespace act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";

(act:telephoneNumber)[1]/act:number', 'nvarchar(50)')

AS [TelephoneNumber]

,LTRIM(RTRIM([ContactInfo].ref.value

(N'declare namespace ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";

declare namespace act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";

act:telephoneNumber/act:SpecialInstructions/text())[1]', 'nvarchar(max)')))

AS [TelephoneSpecialInstructions]

,[ContactInfo].ref.value

(N'declare namespace ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";

declare namespace act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";

act:homePostalAddress/act:Street)[1]', 'nvarchar(50)')

AS [Street]

,[ContactInfo].ref.value

(N'declare namespace ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";

declare namespace act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";

act:homePostalAddress/act:City)[1]', 'nvarchar(50)')

AS [City]

,[ContactInfo].ref.value

(N'declare namespace ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";

declare namespace act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";

act:homePostalAddress/act:StateProvince)[1]', 'nvarchar(50)')

AS [StateProvince]

,[ContactInfo].ref.value

(N'declare namespace ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";

declare namespace act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";

act:homePostalAddress/act:PostalCode)[1]', 'nvarchar(50)')

AS [PostalCode]

,[ContactInfo].ref.value

(N'declare namespace ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";

declare namespace act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";

act:homePostalAddress/act:CountryRegion)[1]', 'nvarchar(50)')

AS [CountryRegion]

,[ContactInfo].ref.value

(N'declare namespace ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";

declare namespace act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";

(act:homePostalAddress/act:SpecialInstructions/text())[1]', 'nvarchar(max)')

AS [HomeAddressSpecialInstructions]

,[ContactInfo].ref.value

(N'declare namespace ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";

declare namespace act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";

act:eMail/act:eMailAddress)[1]', 'nvarchar(128)')

AS [EMailAddress]

,LTRIM(RTRIM([ContactInfo].ref.value

(N'declare namespace ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";

declare namespace act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";

(act:eMail/act:SpecialInstructions/text())[1]', 'nvarchar(max)')))

AS [EMailSpecialInstructions]

,[ContactInfo].ref.value

(N'declare namespace ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";

declare namespace act="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactTypes";

(act:eMail/act:SpecialInstructions/act:telephoneNumber/act:number)[1]', 'nvarchar(50)')

AS [EMailTelephoneNumber]

,[rowguid]

,[ModifiedDate]

FROM [Person].[Person]

OUTER APPLY [AdditionalContactInfo].nodes

('declare namespace ci="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ContactInfo";

/ci:AdditionalContactInfo')

AS ContactInfo(ref)

WHERE [AdditionalContactInfo] IS NOT NULL;

GO

**CREATE VIEW** [HumanResources].[vEmployee]

AS

SELECT

e.[BusinessEntityID]

,p.[Title]

,p.[FirstName]

,p.[MiddleName]

,p.[LastName]

,p.[Suffix]

,e.[JobTitle]

,pp.[PhoneNumber]

,pnt.[Name] AS [PhoneNumberType]

,ea.[EmailAddress]

,p.[EmailPromotion]

,a.[AddressLine1]

,a.[AddressLine2]

,a.[City]

,sp.[Name] AS [StateProvinceName]

,a.[PostalCode]

,cr.[Name] AS [CountryRegionName]

,p.[AdditionalContactInfo]

FROM [HumanResources].[Employee] e

INNER JOIN [Person].[Person] p

ON p.[BusinessEntityID] = e.[BusinessEntityID]

INNER JOIN [Person].[BusinessEntityAddress] bea

ON bea.[BusinessEntityID] = e.[BusinessEntityID]

INNER JOIN [Person].[Address] a

ON a.[AddressID] = bea.[AddressID]

INNER JOIN [Person].[StateProvince] sp

ON sp.[StateProvinceID] = a.[StateProvinceID]

INNER JOIN [Person].[CountryRegion] cr

ON cr.[CountryRegionCode] = sp.[CountryRegionCode]

LEFT OUTER JOIN [Person].[PersonPhone] pp

ON pp.BusinessEntityID = p.[BusinessEntityID]

LEFT OUTER JOIN [Person].[PhoneNumberType] pnt

ON pp.[PhoneNumberTypeID] = pnt.[PhoneNumberTypeID]

LEFT OUTER JOIN [Person].[EmailAddress] ea

ON p.[BusinessEntityID] = ea.[BusinessEntityID];

GO

CREATE VIEW [HumanResources].[vEmployeeDepartment]

AS

SELECT

e.[BusinessEntityID]

,p.[Title]

,p.[FirstName]

,p.[MiddleName]

,p.[LastName]

,p.[Suffix]

,e.[JobTitle]

,d.[Name] AS [Department]

,d.[GroupName]

,edh.[StartDate]

FROM [HumanResources].[Employee] e

INNER JOIN [Person].[Person] p

ON p.[BusinessEntityID] = e.[BusinessEntityID]

INNER JOIN [HumanResources].[EmployeeDepartmentHistory] edh

ON e.[BusinessEntityID] = edh.[BusinessEntityID]

INNER JOIN [HumanResources].[Department] d

ON edh.[DepartmentID] = d.[DepartmentID]

WHERE edh.EndDate IS NULL

GO

CREATE VIEW [HumanResources].[vEmployeeDepartmentHistory]

AS

SELECT

e.[BusinessEntityID]

,p.[Title]

,p.[FirstName]

,p.[MiddleName]

,p.[LastName]

,p.[Suffix]

,s.[Name] AS [Shift]

,d.[Name] AS [Department]

,d.[GroupName]

,edh.[StartDate]

,edh.[EndDate]

FROM [HumanResources].[Employee] e

INNER JOIN [Person].[Person] p

ON p.[BusinessEntityID] = e.[BusinessEntityID]

INNER JOIN [HumanResources].[EmployeeDepartmentHistory] edh

ON e.[BusinessEntityID] = edh.[BusinessEntityID]

INNER JOIN [HumanResources].[Department] d

ON edh.[DepartmentID] = d.[DepartmentID]

INNER JOIN [HumanResources].[Shift] s

ON s.[ShiftID] = edh.[ShiftID];

GO

CREATE VIEW [Sales].[vIndividualCustomer]

AS

SELECT

p.[BusinessEntityID]

,p.[Title]

,p.[FirstName]

,p.[MiddleName]

,p.[LastName]

,p.[Suffix]

,pp.[PhoneNumber]

,pnt.[Name] AS [PhoneNumberType]

,ea.[EmailAddress]

,p.[EmailPromotion]

,at.[Name] AS [AddressType]

,a.[AddressLine1]

,a.[AddressLine2]

,a.[City]

,[StateProvinceName] = sp.[Name]

,a.[PostalCode]

,[CountryRegionName] = cr.[Name]

,p.[Demographics]

FROM [Person].[Person] p

INNER JOIN [Person].[BusinessEntityAddress] bea

ON bea.[BusinessEntityID] = p.[BusinessEntityID]

INNER JOIN [Person].[Address] a

ON a.[AddressID] = bea.[AddressID]

INNER JOIN [Person].[StateProvince] sp

ON sp.[StateProvinceID] = a.[StateProvinceID]

INNER JOIN [Person].[CountryRegion] cr

ON cr.[CountryRegionCode] = sp.[CountryRegionCode]

INNER JOIN [Person].[AddressType] at

ON at.[AddressTypeID] = bea.[AddressTypeID]

INNER JOIN [Sales].[Customer] c

ON c.[PersonID] = p.[BusinessEntityID]

LEFT OUTER JOIN [Person].[EmailAddress] ea

ON ea.[BusinessEntityID] = p.[BusinessEntityID]

LEFT OUTER JOIN [Person].[PersonPhone] pp

ON pp.[BusinessEntityID] = p.[BusinessEntityID]

LEFT OUTER JOIN [Person].[PhoneNumberType] pnt

ON pnt.[PhoneNumberTypeID] = pp.[PhoneNumberTypeID]

WHERE c.StoreID IS NULL;

GO

**CREATE VIEW** [Sales].[vPersonDemographics]

AS

SELECT

p.[BusinessEntityID]

,[IndividualSurvey].[ref].[value]

(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey"; **TotalPurchaseYTD**[1]', 'money') AS [TotalPurchaseYTD]

,CONVERT(datetime, REPLACE([IndividualSurvey].[ref].[value]

(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey"; **DateFirstPurchase**[1]', 'nvarchar(20)') ,'Z', ''), 101) AS [DateFirstPurchase]

,CONVERT(datetime, REPLACE([IndividualSurvey].[ref].[value]

(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey"; **BirthDate**[1]', 'nvarchar(20)') ,'Z', ''), 101) AS [BirthDate]

,[IndividualSurvey].[ref].[value]

(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey"; **MaritalStatus**[1]', 'nvarchar(1)') AS [MaritalStatus]

,[IndividualSurvey].[ref].[value]

(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey"; **YearlyIncome**[1]', 'nvarchar(30)') AS [YearlyIncome]

,[IndividualSurvey].[ref].[value]

(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey"; **Gender**[1]', 'nvarchar(1)') AS [Gender]

,[IndividualSurvey].[ref].[value]

(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey"; **TotalChildren**[1]', 'integer') AS [TotalChildren]

,[IndividualSurvey].[ref].[value]

(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey"; **NumberChildrenAtHome**[1]', 'integer') AS [NumberChildrenAtHome]

,[IndividualSurvey].[ref].[value]

(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey"; **Education**[1]', 'nvarchar(30)') AS [Education]

,[IndividualSurvey].[ref].[value]

(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey"; **Occupation**[1]', 'nvarchar(30)') AS [Occupation]

,[IndividualSurvey].[ref].[value]

(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey"; **HomeOwnerFlag**[1]', 'bit') AS [HomeOwnerFlag]

,[IndividualSurvey].[ref].[value]

(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey"; **NumberCarsOwned**[1]', 'integer') AS [NumberCarsOwned]

FROM [Person].[Person] p

CROSS APPLY p.[Demographics].nodes

(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey"; /IndividualSurvey') AS [IndividualSurvey](ref)

WHERE [Demographics] IS NOT NULL;

GO

-------------------------------------------- EXCERPT ----------------------------------------------------------

**http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey/IndividualSurvey.xsd**

<?xml version="1.0" encoding="UTF-8"?>

[<xsd:schema xmlns="**http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey**" xmlns:xsd="**http://www.w3.org/2001/XMLSchema**" elementFormDefault="**qualified**" targetNamespace="**http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey**">](http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey/IndividualSurvey.xsd)

[<xsd:annotation>](http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey/IndividualSurvey.xsd)

<xsd:documentation> (c) 2008 Microsoft Corporation. All rights reserved. The following schema for Microsoft SQL Server is presented in XML format and is for informational purposes only. Microsoft Corporation ("Microsoft") may have trademarks, copyrights, or other intellectual property rights covering subject matter in the schema. Microsoft does not make any representation or warranty regarding the schema or any product or item developed based on the schema. The schema is provided to you on an AS IS basis. Microsoft disclaims all express, implied and statutory warranties, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and freedom from infringement. Without limiting the generality of the foregoing, Microsoft does not make any warranty of any kind that any item developed based on the schema, or any portion of the schema, will not infringe any copyright, patent, trade secret, or other intellectual property right of any person or entity in any country. It is your responsibility to seek licenses for such intellectual property rights where appropriate. MICROSOFT SHALL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE SCHEMA, INCLUDING WITHOUT LIMITATION, ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL (INCLUDING ANY LOST PROFITS), PUNITIVE OR SPECIAL DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN ADVISED OF SUCH DAMAGES. </xsd:documentation>

</xsd:annotation>

[<xsd:simpleType name="**SalaryType**">](http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey/IndividualSurvey.xsd)

[<xsd:restriction base="**xsd:string**">](http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey/IndividualSurvey.xsd)

<xsd:enumeration value="**0-25000**"/>

<xsd:enumeration value="**25001-50000**"/>

<xsd:enumeration value="**50001-75000**"/>

<xsd:enumeration value="**75001-100000**"/>

<xsd:enumeration value="**greater than 100000**"/>

</xsd:restriction>

</xsd:simpleType>

[<xsd:simpleType name="**MileRangeType**">](http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey/IndividualSurvey.xsd)

[<xsd:restriction base="**xsd:string**">](http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey/IndividualSurvey.xsd)

<xsd:enumeration value="**0-1 Miles**"/>

<xsd:enumeration value="**1-2 Miles**"/>

<xsd:enumeration value="**2-5 Miles**"/>

<xsd:enumeration value="**5-10 Miles**"/>

<xsd:enumeration value="**10+ Miles**"/>

</xsd:restriction>

</xsd:simpleType>

[<xsd:element name="**IndividualSurvey**">](http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey/IndividualSurvey.xsd)

[<xsd:complexType>](http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey/IndividualSurvey.xsd)

[<xsd:sequence>](http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey/IndividualSurvey.xsd)

<xsd:element type="**xsd:decimal**" name="**TotalPurchaseYTD**" maxOccurs="**1**" minOccurs="**0**"/>

<xsd:element type="**xsd:date**" name="**DateFirstPurchase**" maxOccurs="**1**" minOccurs="**0**"/>

<xsd:element type="**xsd:date**" name="**BirthDate**" maxOccurs="**1**" minOccurs="**0**"/>

<xsd:element type="**xsd:string**" name="**MaritalStatus**" maxOccurs="**1**" minOccurs="**0**"/>

<xsd:element type="**SalaryType**" name="**YearlyIncome**" maxOccurs="**1**" minOccurs="**0**"/>

<xsd:element type="**xsd:string**" name="**Gender**" maxOccurs="**1**" minOccurs="**0**"/>

<xsd:element type="**xsd:int**" name="**TotalChildren**" maxOccurs="**1**" minOccurs="**0**"/>

<xsd:element type="**xsd:int**" name="**NumberChildrenAtHome**" maxOccurs="**1**" minOccurs="**0**"/>

<xsd:element type="**xsd:string**" name="**Education**" maxOccurs="**1**" minOccurs="**0**"/>

<xsd:element type="**xsd:string**" name="**Occupation**" maxOccurs="**1**" minOccurs="**0**"/>

<xsd:element type="**xsd:string**" name="**HomeOwnerFlag**" maxOccurs="**1**" minOccurs="**0**"/>

<xsd:element type="**xsd:int**" name="**NumberCarsOwned**" maxOccurs="**1**" minOccurs="**0**"/>

<xsd:element type="**xsd:string**" name="**Hobby**" maxOccurs="**unbounded**" minOccurs="**0**"/>

<xsd:element type="**MileRangeType**" name="**CommuteDistance**" maxOccurs="**1**" minOccurs="**0**"/>

<xsd:element type="**xsd:string**" name="**Comments**" maxOccurs="**1**" minOccurs="**0**"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

------------------------------------------------------------------------------------------------------------------

CREATE VIEW [HumanResources].[vJobCandidate]

AS

SELECT

jc.[JobCandidateID]

,jc.[BusinessEntityID]

,[Resume].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(/Resume/Name/Name.Prefix)[1]', 'nvarchar(30)') AS [Name.Prefix]

,[Resume].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(/Resume/Name/Name.First)[1]', 'nvarchar(30)') AS [Name.First]

,[Resume].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(/Resume/Name/Name.Middle)[1]', 'nvarchar(30)') AS [Name.Middle]

,[Resume].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(/Resume/Name/Name.Last)[1]', 'nvarchar(30)') AS [Name.Last]

,[Resume].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(/Resume/Name/Name.Suffix)[1]', 'nvarchar(30)') AS [Name.Suffix]

,[Resume].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(/Resume/Skills)[1]', 'nvarchar(max)') AS [Skills]

,[Resume].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Address/Addr.Type)[1]', 'nvarchar(30)') AS [Addr.Type]

,[Resume].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Address/Addr.Location/Location/Loc.CountryRegion)[1]', 'nvarchar(100)') AS [Addr.Loc.CountryRegion]

,[Resume].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Address/Addr.Location/Location/Loc.State)[1]', 'nvarchar(100)') AS [Addr.Loc.State]

,[Resume].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Address/Addr.Location/Location/Loc.City)[1]', 'nvarchar(100)') AS [Addr.Loc.City]

,[Resume].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Address/Addr.PostalCode)[1]', 'nvarchar(20)') AS [Addr.PostalCode]

,[Resume].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(/Resume/EMail)[1]', 'nvarchar(max)') AS [EMail]

,[Resume].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(/Resume/WebSite)[1]', 'nvarchar(max)') AS [WebSite]

,jc.[ModifiedDate]

FROM [HumanResources].[JobCandidate] jc

CROSS APPLY jc.[Resume].nodes(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

/Resume') AS Resume(ref);

GO

CREATE VIEW [HumanResources].[vJobCandidateEmployment]

AS

SELECT

jc.[JobCandidateID]

,CONVERT(datetime, REPLACE([Employment].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Emp.StartDate)[1]', 'nvarchar(20)') ,'Z', ''), 101) AS [Emp.StartDate]

,CONVERT(datetime, REPLACE([Employment].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Emp.EndDate)[1]', 'nvarchar(20)') ,'Z', ''), 101) AS [Emp.EndDate]

,[Employment].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Emp.OrgName)[1]', 'nvarchar(100)') AS [Emp.OrgName]

,[Employment].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Emp.JobTitle)[1]', 'nvarchar(100)') AS [Emp.JobTitle]

,[Employment].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Emp.Responsibility)[1]', 'nvarchar(max)') AS [Emp.Responsibility]

,[Employment].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Emp.FunctionCategory)[1]', 'nvarchar(max)') AS [Emp.FunctionCategory]

,[Employment].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Emp.IndustryCategory)[1]', 'nvarchar(max)') AS [Emp.IndustryCategory]

,[Employment].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Emp.Location/Location/Loc.CountryRegion)[1]', 'nvarchar(max)') AS [Emp.Loc.CountryRegion]

,[Employment].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Emp.Location/Location/Loc.State)[1]', 'nvarchar(max)') AS [Emp.Loc.State]

,[Employment].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Emp.Location/Location/Loc.City)[1]', 'nvarchar(max)') AS [Emp.Loc.City]

FROM [HumanResources].[JobCandidate] jc

CROSS APPLY jc.[Resume].nodes(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

/Resume/Employment') AS Employment(ref);

GO

CREATE VIEW [HumanResources].[vJobCandidateEducation]

AS

SELECT

jc.[JobCandidateID]

,[Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Edu.Level)[1]', 'nvarchar(max)') AS [Edu.Level]

,CONVERT(datetime, REPLACE([Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Edu.StartDate)[1]', 'nvarchar(20)') ,'Z', ''), 101) AS [Edu.StartDate]

,CONVERT(datetime, REPLACE([Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Edu.EndDate)[1]', 'nvarchar(20)') ,'Z', ''), 101) AS [Edu.EndDate]

,[Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Edu.Degree)[1]', 'nvarchar(50)') AS [Edu.Degree]

,[Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Edu.Major)[1]', 'nvarchar(50)') AS [Edu.Major]

,[Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Edu.Minor)[1]', 'nvarchar(50)') AS [Edu.Minor]

,[Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Edu.GPA)[1]', 'nvarchar(5)') AS [Edu.GPA]

,[Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Edu.GPAScale)[1]', 'nvarchar(5)') AS [Edu.GPAScale]

,[Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Edu.School)[1]', 'nvarchar(100)') AS [Edu.School]

,[Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Edu.Location/Location/Loc.CountryRegion)[1]', 'nvarchar(100)') AS [Edu.Loc.CountryRegion]

,[Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Edu.Location/Location/Loc.State)[1]', 'nvarchar(100)') AS [Edu.Loc.State]

,[Education].ref.value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

(Edu.Location/Location/Loc.City)[1]', 'nvarchar(100)') AS [Edu.Loc.City]

FROM [HumanResources].[JobCandidate] jc

CROSS APPLY jc.[Resume].nodes(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/Resume";

/Resume/Education') AS [Education](ref);

GO

CREATE VIEW [Production].[vProductAndDescription]

WITH SCHEMABINDING

AS

-- View (indexed or standard) to display products and product descriptions by language.

SELECT

p.[ProductID]

,p.[Name]

,pm.[Name] AS [ProductModel]

,pmx.[CultureID]

,pd.[Description]

FROM [Production].[Product] p

INNER JOIN [Production].[ProductModel] pm

ON p.[ProductModelID] = pm.[ProductModelID]

INNER JOIN [Production].[ProductModelProductDescriptionCulture] pmx

ON pm.[ProductModelID] = pmx.[ProductModelID]

INNER JOIN [Production].[ProductDescription] pd

ON pmx.[ProductDescriptionID] = pd.[ProductDescriptionID];

GO

-- Index the vProductAndDescription view

CREATE UNIQUE CLUSTERED INDEX [IX\_vProductAndDescription] ON [Production].[vProductAndDescription]([CultureID], [ProductID]);

GO

CREATE VIEW [Production].[vProductModelCatalogDescription]

AS

SELECT

[ProductModelID]

,[Name]

,[CatalogDescription].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";

declare namespace html="http://www.w3.org/1999/xhtml";

(/p1:ProductDescription/p1:Summary/html:p)[1]', 'nvarchar(max)') AS [Summary]

,[CatalogDescription].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";

(/p1:ProductDescription/p1:Manufacturer/p1:Name)[1]', 'nvarchar(max)') AS [Manufacturer]

,[CatalogDescription].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";

(/p1:ProductDescription/p1:Manufacturer/p1:Copyright)[1]', 'nvarchar(30)') AS [Copyright]

,[CatalogDescription].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";

(/p1:ProductDescription/p1:Manufacturer/p1:ProductURL)[1]', 'nvarchar(256)') AS [ProductURL]

,[CatalogDescription].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";

declare namespace wm="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelWarrAndMain";

(/p1:ProductDescription/p1:Features/wm:Warranty/wm:WarrantyPeriod)[1]', 'nvarchar(256)') AS [WarrantyPeriod]

,[CatalogDescription].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";

declare namespace wm="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelWarrAndMain";

(/p1:ProductDescription/p1:Features/wm:Warranty/wm:Description)[1]', 'nvarchar(256)') AS [WarrantyDescription]

,[CatalogDescription].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";

declare namespace wm="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelWarrAndMain";

(/p1:ProductDescription/p1:Features/wm:Maintenance/wm:NoOfYears)[1]', 'nvarchar(256)') AS [NoOfYears]

,[CatalogDescription].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";

declare namespace wm="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelWarrAndMain";

(/p1:ProductDescription/p1:Features/wm:Maintenance/wm:Description)[1]', 'nvarchar(256)') AS [MaintenanceDescription]

,[CatalogDescription].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";

declare namespace wf="http://www.adventure-works.com/schemas/OtherFeatures";

(/p1:ProductDescription/p1:Features/wf:wheel)[1]', 'nvarchar(256)') AS [Wheel]

,[CatalogDescription].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";

declare namespace wf="http://www.adventure-works.com/schemas/OtherFeatures";

(/p1:ProductDescription/p1:Features/wf:saddle)[1]', 'nvarchar(256)') AS [Saddle]

,[CatalogDescription].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";

declare namespace wf="http://www.adventure-works.com/schemas/OtherFeatures";

(/p1:ProductDescription/p1:Features/wf:pedal)[1]', 'nvarchar(256)') AS [Pedal]

,[CatalogDescription].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";

declare namespace wf="http://www.adventure-works.com/schemas/OtherFeatures";

(/p1:ProductDescription/p1:Features/wf:BikeFrame)[1]', 'nvarchar(max)') AS [BikeFrame]

,[CatalogDescription].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";

declare namespace wf="http://www.adventure-works.com/schemas/OtherFeatures";

(/p1:ProductDescription/p1:Features/wf:crankset)[1]', 'nvarchar(256)') AS [Crankset]

,[CatalogDescription].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";

(/p1:ProductDescription/p1:Picture/p1:Angle)[1]', 'nvarchar(256)') AS [PictureAngle]

,[CatalogDescription].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";

(/p1:ProductDescription/p1:Picture/p1:Size)[1]', 'nvarchar(256)') AS [PictureSize]

,[CatalogDescription].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";

(/p1:ProductDescription/p1:Picture/p1:ProductPhotoID)[1]', 'nvarchar(256)') AS [ProductPhotoID]

,[CatalogDescription].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";

(/p1:ProductDescription/p1:Specifications/Material)[1]', 'nvarchar(256)') AS [Material]

,[CatalogDescription].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";

(/p1:ProductDescription/p1:Specifications/Color)[1]', 'nvarchar(256)') AS [Color]

,[CatalogDescription].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";

(/p1:ProductDescription/p1:Specifications/ProductLine)[1]', 'nvarchar(256)') AS [ProductLine]

,[CatalogDescription].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";

(/p1:ProductDescription/p1:Specifications/Style)[1]', 'nvarchar(256)') AS [Style]

,[CatalogDescription].value(N'declare namespace p1="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelDescription";

(/p1:ProductDescription/p1:Specifications/RiderExperience)[1]', 'nvarchar(1024)') AS [RiderExperience]

,[rowguid]

,[ModifiedDate]

FROM [Production].[ProductModel]

WHERE [CatalogDescription] IS NOT NULL;

GO

CREATE VIEW [Production].[vProductModelInstructions]

AS

SELECT

[ProductModelID]

,[Name]

,[Instructions].value(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelManuInstructions";

(/root/text())[1]', 'nvarchar(max)') AS [Instructions]

,[MfgInstructions].ref.value('@LocationID[1]', 'int') AS [LocationID]

,[MfgInstructions].ref.value('@SetupHours[1]', 'decimal(9, 4)') AS [SetupHours]

,[MfgInstructions].ref.value('@MachineHours[1]', 'decimal(9, 4)') AS [MachineHours]

,[MfgInstructions].ref.value('@LaborHours[1]', 'decimal(9, 4)') AS [LaborHours]

,[MfgInstructions].ref.value('@LotSize[1]', 'int') AS [LotSize]

,[Steps].ref.value('string(.)[1]', 'nvarchar(1024)') AS [Step]

,[rowguid]

,[ModifiedDate]

FROM [Production].[ProductModel]

CROSS APPLY [Instructions].nodes(N'declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelManuInstructions";

/root/Location') MfgInstructions(ref)

CROSS APPLY [MfgInstructions].ref.nodes('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ProductModelManuInstructions";

step') Steps(ref);

GO

CREATE VIEW [Sales].[vSalesPerson]

AS

SELECT

s.[BusinessEntityID]

,p.[Title]

,p.[FirstName]

,p.[MiddleName]

,p.[LastName]

,p.[Suffix]

,e.[JobTitle]

,pp.[PhoneNumber]

,pnt.[Name] AS [PhoneNumberType]

,ea.[EmailAddress]

,p.[EmailPromotion]

,a.[AddressLine1]

,a.[AddressLine2]

,a.[City]

,[StateProvinceName] = sp.[Name]

,a.[PostalCode]

,[CountryRegionName] = cr.[Name]

,[TerritoryName] = st.[Name]

,[TerritoryGroup] = st.[Group]

,s.[SalesQuota]

,s.[SalesYTD]

,s.[SalesLastYear]

FROM [Sales].[SalesPerson] s

INNER JOIN [HumanResources].[Employee] e

ON e.[BusinessEntityID] = s.[BusinessEntityID]

INNER JOIN [Person].[Person] p

ON p.[BusinessEntityID] = s.[BusinessEntityID]

INNER JOIN [Person].[BusinessEntityAddress] bea

ON bea.[BusinessEntityID] = s.[BusinessEntityID]

INNER JOIN [Person].[Address] a

ON a.[AddressID] = bea.[AddressID]

INNER JOIN [Person].[StateProvince] sp

ON sp.[StateProvinceID] = a.[StateProvinceID]

INNER JOIN [Person].[CountryRegion] cr

ON cr.[CountryRegionCode] = sp.[CountryRegionCode]

LEFT OUTER JOIN [Sales].[SalesTerritory] st

ON st.[TerritoryID] = s.[TerritoryID]

LEFT OUTER JOIN [Person].[EmailAddress] ea

ON ea.[BusinessEntityID] = p.[BusinessEntityID]

LEFT OUTER JOIN [Person].[PersonPhone] pp

ON pp.[BusinessEntityID] = p.[BusinessEntityID]

LEFT OUTER JOIN [Person].[PhoneNumberType] pnt

ON pnt.[PhoneNumberTypeID] = pp.[PhoneNumberTypeID];

GO

CREATE VIEW [Sales].[vSalesPersonSalesByFiscalYears]

AS

SELECT

pvt.[SalesPersonID]

,pvt.[FullName]

,pvt.[JobTitle]

,pvt.[SalesTerritory]

,pvt.[2002]

,pvt.[2003]

,pvt.[2004]

FROM (SELECT

soh.[SalesPersonID]

,p.[FirstName] + ' ' + COALESCE(p.[MiddleName], '') + ' ' + p.[LastName] AS [FullName]

,e.[JobTitle]

,st.[Name] AS [SalesTerritory]

,soh.[SubTotal]

,YEAR(DATEADD(m, 6, soh.[OrderDate])) AS [FiscalYear]

FROM [Sales].[SalesPerson] sp

INNER JOIN [Sales].[SalesOrderHeader] soh

ON sp.[BusinessEntityID] = soh.[SalesPersonID]

INNER JOIN [Sales].[SalesTerritory] st

ON sp.[TerritoryID] = st.[TerritoryID]

INNER JOIN [HumanResources].[Employee] e

ON soh.[SalesPersonID] = e.[BusinessEntityID]

INNER JOIN [Person].[Person] p

ON p.[BusinessEntityID] = sp.[BusinessEntityID]

) AS soh

PIVOT

(

SUM([SubTotal])

FOR [FiscalYear]

IN ([2002], [2003], [2004])

) AS pvt;

GO

CREATE VIEW [Person].[vStateProvinceCountryRegion]

WITH SCHEMABINDING

AS

SELECT

sp.[StateProvinceID]

,sp.[StateProvinceCode]

,sp.[IsOnlyStateProvinceFlag]

,sp.[Name] AS [StateProvinceName]

,sp.[TerritoryID]

,cr.[CountryRegionCode]

,cr.[Name] AS [CountryRegionName]

FROM [Person].[StateProvince] sp

INNER JOIN [Person].[CountryRegion] cr

ON sp.[CountryRegionCode] = cr.[CountryRegionCode];

GO

-- Index the vStateProvinceCountryRegion view

CREATE UNIQUE CLUSTERED INDEX [IX\_vStateProvinceCountryRegion] ON [Person].[vStateProvinceCountryRegion]([StateProvinceID], [CountryRegionCode]);

GO

CREATE VIEW [Sales].[vStoreWithDemographics] AS

SELECT

s.[BusinessEntityID]

,s.[Name]

,s.[Demographics].value('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey";

(/StoreSurvey/AnnualSales)[1]', 'money') AS [AnnualSales]

,s.[Demographics].value('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey";

(/StoreSurvey/AnnualRevenue)[1]', 'money') AS [AnnualRevenue]

,s.[Demographics].value('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey";

(/StoreSurvey/BankName)[1]', 'nvarchar(50)') AS [BankName]

,s.[Demographics].value('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey";

(/StoreSurvey/BusinessType)[1]', 'nvarchar(5)') AS [BusinessType]

,s.[Demographics].value('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey";

(/StoreSurvey/YearOpened)[1]', 'integer') AS [YearOpened]

,s.[Demographics].value('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey";

(/StoreSurvey/Specialty)[1]', 'nvarchar(50)') AS [Specialty]

,s.[Demographics].value('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey";

(/StoreSurvey/SquareFeet)[1]', 'integer') AS [SquareFeet]

,s.[Demographics].value('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey";

(/StoreSurvey/Brands)[1]', 'nvarchar(30)') AS [Brands]

,s.[Demographics].value('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey";

(/StoreSurvey/Internet)[1]', 'nvarchar(30)') AS [Internet]

,s.[Demographics].value('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/StoreSurvey";

(/StoreSurvey/NumberEmployees)[1]', 'integer') AS [NumberEmployees]

FROM [Sales].[Store] s;

GO

CREATE VIEW [Sales].[vStoreWithContacts] AS

SELECT

s.[BusinessEntityID]

,s.[Name]

,ct.[Name] AS [ContactType]

,p.[Title]

,p.[FirstName]

,p.[MiddleName]

,p.[LastName]

,p.[Suffix]

,pp.[PhoneNumber]

,pnt.[Name] AS [PhoneNumberType]

,ea.[EmailAddress]

,p.[EmailPromotion]

FROM [Sales].[Store] s

INNER JOIN [Person].[BusinessEntityContact] bec

ON bec.[BusinessEntityID] = s.[BusinessEntityID]

INNER JOIN [Person].[ContactType] ct

ON ct.[ContactTypeID] = bec.[ContactTypeID]

INNER JOIN [Person].[Person] p

ON p.[BusinessEntityID] = bec.[PersonID]

LEFT OUTER JOIN [Person].[EmailAddress] ea

ON ea.[BusinessEntityID] = p.[BusinessEntityID]

LEFT OUTER JOIN [Person].[PersonPhone] pp

ON pp.[BusinessEntityID] = p.[BusinessEntityID]

LEFT OUTER JOIN [Person].[PhoneNumberType] pnt

ON pnt.[PhoneNumberTypeID] = pp.[PhoneNumberTypeID];

GO

CREATE VIEW [Sales].[vStoreWithAddresses] AS

SELECT

s.[BusinessEntityID]

,s.[Name]

,at.[Name] AS [AddressType]

,a.[AddressLine1]

,a.[AddressLine2]

,a.[City]

,sp.[Name] AS [StateProvinceName]

,a.[PostalCode]

,cr.[Name] AS [CountryRegionName]

FROM [Sales].[Store] s

INNER JOIN [Person].[BusinessEntityAddress] bea

ON bea.[BusinessEntityID] = s.[BusinessEntityID]

INNER JOIN [Person].[Address] a

ON a.[AddressID] = bea.[AddressID]

INNER JOIN [Person].[StateProvince] sp

ON sp.[StateProvinceID] = a.[StateProvinceID]

INNER JOIN [Person].[CountryRegion] cr

ON cr.[CountryRegionCode] = sp.[CountryRegionCode]

INNER JOIN [Person].[AddressType] at

ON at.[AddressTypeID] = bea.[AddressTypeID];

GO

CREATE VIEW [Purchasing].[vVendorWithContacts] AS

SELECT

v.[BusinessEntityID]

,v.[Name]

,ct.[Name] AS [ContactType]

,p.[Title]

,p.[FirstName]

,p.[MiddleName]

,p.[LastName]

,p.[Suffix]

,pp.[PhoneNumber]

,pnt.[Name] AS [PhoneNumberType]

,ea.[EmailAddress]

,p.[EmailPromotion]

FROM [Purchasing].[Vendor] v

INNER JOIN [Person].[BusinessEntityContact] bec

ON bec.[BusinessEntityID] = v.[BusinessEntityID]

INNER JOIN [Person].ContactType ct

ON ct.[ContactTypeID] = bec.[ContactTypeID]

INNER JOIN [Person].[Person] p

ON p.[BusinessEntityID] = bec.[PersonID]

LEFT OUTER JOIN [Person].[EmailAddress] ea

ON ea.[BusinessEntityID] = p.[BusinessEntityID]

LEFT OUTER JOIN [Person].[PersonPhone] pp

ON pp.[BusinessEntityID] = p.[BusinessEntityID]

LEFT OUTER JOIN [Person].[PhoneNumberType] pnt

ON pnt.[PhoneNumberTypeID] = pp.[PhoneNumberTypeID];

GO

CREATE VIEW [Purchasing].[vVendorWithAddresses] AS

SELECT

v.[BusinessEntityID]

,v.[Name]

,at.[Name] AS [AddressType]

,a.[AddressLine1]

,a.[AddressLine2]

,a.[City]

,sp.[Name] AS [StateProvinceName]

,a.[PostalCode]

,cr.[Name] AS [CountryRegionName]

FROM [Purchasing].[Vendor] v

INNER JOIN [Person].[BusinessEntityAddress] bea

ON bea.[BusinessEntityID] = v.[BusinessEntityID]

INNER JOIN [Person].[Address] a

ON a.[AddressID] = bea.[AddressID]

INNER JOIN [Person].[StateProvince] sp

ON sp.[StateProvinceID] = a.[StateProvinceID]

INNER JOIN [Person].[CountryRegion] cr

ON cr.[CountryRegionCode] = sp.[CountryRegionCode]

INNER JOIN [Person].[AddressType] at

ON at.[AddressTypeID] = bea.[AddressTypeID];

GO

[TOP](#Top)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Add database functions.

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PRINT '';

PRINT '\*\*\* Creating Database Functions';

GO

CREATE FUNCTION [dbo].[ufnGetAccountingStartDate]()

RETURNS [datetime]

AS

BEGIN

RETURN CONVERT(datetime, '20030701', 112);

END;

GO

CREATE FUNCTION [dbo].[ufnGetAccountingEndDate]()

RETURNS [datetime]

AS

BEGIN

RETURN DATEADD(millisecond, -2, CONVERT(datetime, '20040701', 112));

END;

GO

CREATE FUNCTION [dbo].[ufnGetContactInformation](@PersonID int)

RETURNS @retContactInformation TABLE

(

-- Columns returned by the function

[PersonID] int NOT NULL,

[FirstName] [nvarchar](50) NULL,

[LastName] [nvarchar](50) NULL,

[JobTitle] [nvarchar](50) NULL,

[BusinessEntityType] [nvarchar](50) NULL

)

AS

-- Returns the first name, last name, job title and business entity type for the specified contact.

-- Since a contact can serve multiple roles, more than one row may be returned.

BEGIN

IF @PersonID IS NOT NULL

BEGIN

IF EXISTS(SELECT \* FROM [HumanResources].[Employee] e

WHERE e.[BusinessEntityID] = @PersonID)

INSERT INTO @retContactInformation

SELECT @PersonID, p.FirstName, p.LastName, e.[JobTitle], 'Employee'

FROM [HumanResources].[Employee] AS e

INNER JOIN [Person].[Person] p

ON p.[BusinessEntityID] = e.[BusinessEntityID]

WHERE e.[BusinessEntityID] = @PersonID;

IF EXISTS(SELECT \* FROM [Purchasing].[Vendor] AS v

INNER JOIN [Person].[BusinessEntityContact] bec

ON bec.[BusinessEntityID] = v.[BusinessEntityID]

WHERE bec.[PersonID] = @PersonID)

INSERT INTO @retContactInformation

SELECT @PersonID, p.FirstName, p.LastName, ct.[Name], 'Vendor Contact'

FROM [Purchasing].[Vendor] AS v

INNER JOIN [Person].[BusinessEntityContact] bec

ON bec.[BusinessEntityID] = v.[BusinessEntityID]

INNER JOIN [Person].ContactType ct

ON ct.[ContactTypeID] = bec.[ContactTypeID]

INNER JOIN [Person].[Person] p

ON p.[BusinessEntityID] = bec.[PersonID]

WHERE bec.[PersonID] = @PersonID;

IF EXISTS(SELECT \* FROM [Sales].[Store] AS s

INNER JOIN [Person].[BusinessEntityContact] bec

ON bec.[BusinessEntityID] = s.[BusinessEntityID]

WHERE bec.[PersonID] = @PersonID)

INSERT INTO @retContactInformation

SELECT @PersonID, p.FirstName, p.LastName, ct.[Name], 'Store Contact'

FROM [Sales].[Store] AS s

INNER JOIN [Person].[BusinessEntityContact] bec

ON bec.[BusinessEntityID] = s.[BusinessEntityID]

INNER JOIN [Person].ContactType ct

ON ct.[ContactTypeID] = bec.[ContactTypeID]

INNER JOIN [Person].[Person] p

ON p.[BusinessEntityID] = bec.[PersonID]

WHERE bec.[PersonID] = @PersonID;

IF EXISTS(SELECT \* FROM [Person].[Person] AS p

INNER JOIN [Sales].[Customer] AS c

ON c.[PersonID] = p.[BusinessEntityID]

WHERE p.[BusinessEntityID] = @PersonID AND c.[StoreID] IS NULL)

INSERT INTO @retContactInformation

SELECT @PersonID, p.FirstName, p.LastName, NULL, 'Consumer'

FROM [Person].[Person] AS p

INNER JOIN [Sales].[Customer] AS c

ON c.[PersonID] = p.[BusinessEntityID]

WHERE p.[BusinessEntityID] = @PersonID AND c.[StoreID] IS NULL;

END

RETURN;

END;

GO

CREATE FUNCTION [dbo].[ufnGetProductDealerPrice](@ProductID [int], @OrderDate [datetime])

RETURNS [money]

AS

-- Returns the dealer price for the product on a specific date.

BEGIN

DECLARE @DealerPrice money;

DECLARE @DealerDiscount money;

SET @DealerDiscount = 0.60 -- 60% of list price

SELECT @DealerPrice = plph.[ListPrice] \* @DealerDiscount

FROM [Production].[Product] p

INNER JOIN [Production].[ProductListPriceHistory] plph

ON p.[ProductID] = plph.[ProductID]

AND p.[ProductID] = @ProductID

AND @OrderDate BETWEEN plph.[StartDate] AND COALESCE(plph.[EndDate], CONVERT(datetime, '99991231', 112)); -- Make sure we get all the prices!

RETURN @DealerPrice;

END;

GO

CREATE FUNCTION [dbo].[ufnGetProductListPrice](@ProductID [int], @OrderDate [datetime])

RETURNS [money]

AS

BEGIN

DECLARE @ListPrice money;

SELECT @ListPrice = plph.[ListPrice]

FROM [Production].[Product] p

INNER JOIN [Production].[ProductListPriceHistory] plph

ON p.[ProductID] = plph.[ProductID]

AND p.[ProductID] = @ProductID

AND @OrderDate BETWEEN plph.[StartDate] AND COALESCE(plph.[EndDate], CONVERT(datetime, '99991231', 112)); -- Make sure we get all the prices!

RETURN @ListPrice;

END;

GO

CREATE FUNCTION [dbo].[ufnGetProductStandardCost](@ProductID [int], @OrderDate [datetime])

RETURNS [money]

AS

-- Returns the standard cost for the product on a specific date.

BEGIN

DECLARE @StandardCost money;

SELECT @StandardCost = pch.[StandardCost]

FROM [Production].[Product] p

INNER JOIN [Production].[ProductCostHistory] pch

ON p.[ProductID] = pch.[ProductID]

AND p.[ProductID] = @ProductID

AND @OrderDate BETWEEN pch.[StartDate] AND COALESCE(pch.[EndDate], CONVERT(datetime, '99991231', 112)); -- Make sure we get all the prices!

RETURN @StandardCost;

END;

GO

CREATE FUNCTION [dbo].[ufnGetStock](@ProductID [int])

RETURNS [int]

AS

-- Returns the stock level for the product. This function is used internally only

BEGIN

DECLARE @ret int;

SELECT @ret = SUM(p.[Quantity])

FROM [Production].[ProductInventory] p

WHERE p.[ProductID] = @ProductID

AND p.[LocationID] = '6'; -- Only look at inventory in the misc storage

IF (@ret IS NULL)

SET @ret = 0

RETURN @ret

END;

GO

CREATE FUNCTION [dbo].[ufnGetDocumentStatusText](@Status [tinyint])

RETURNS [nvarchar](16)

AS

-- Returns the sales order status text representation for the status value.

BEGIN

DECLARE @ret [nvarchar](16);

SET @ret =

CASE @Status

WHEN 1 THEN N'Pending approval'

WHEN 2 THEN N'Approved'

WHEN 3 THEN N'Obsolete'

ELSE N'\*\* Invalid \*\*'

END;

RETURN @ret

END;

GO

CREATE FUNCTION [dbo].[ufnGetPurchaseOrderStatusText](@Status [tinyint])

RETURNS [nvarchar](15)

AS

-- Returns the sales order status text representation for the status value.

BEGIN

DECLARE @ret [nvarchar](15);

SET @ret =

CASE @Status

WHEN 1 THEN 'Pending'

WHEN 2 THEN 'Approved'

WHEN 3 THEN 'Rejected'

WHEN 4 THEN 'Complete'

ELSE '\*\* Invalid \*\*'

END;

RETURN @ret

END;

GO

CREATE FUNCTION [dbo].[ufnGetSalesOrderStatusText](@Status [tinyint])

RETURNS [nvarchar](15)

AS

-- Returns the sales order status text representation for the status value.

BEGIN

DECLARE @ret [nvarchar](15);

SET @ret =

CASE @Status

WHEN 1 THEN 'In process'

WHEN 2 THEN 'Approved'

WHEN 3 THEN 'Backordered'

WHEN 4 THEN 'Rejected'

WHEN 5 THEN 'Shipped'

WHEN 6 THEN 'Cancelled'

ELSE '\*\* Invalid \*\*'

END;

RETURN @ret

END;

GO

[TOP](#Top)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Create stored procedures

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PRINT '';

PRINT '\*\*\* Creating Stored Procedures';

GO

----------------------------------------------------------------------------------------------------------

CREATE PROCEDURE [dbo].[uspGetBillOfMaterials]

@StartProductID [int],

Stored Procedure Variables passed to SP by Caller

@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 (Common Table Expression – temporary result dataset) name and columns

AS (

SELECT b.[ProductAssemblyID], b.[ComponentID], p.[Name], b.[PerAssemblyQty], p.[StandardCost], p.[ListPrice], b.[BOMLevel], 0 -- Get the initial list of components for the bike assembly

FROM [Production].[BillOfMaterials] b

INNER JOIN [Production].[Product] p

ON b.[ComponentID] = p.[ProductID]

WHERE b.[ProductAssemblyID] = @StartProductID

AND @CheckDate >= b.[StartDate]

AND @CheckDate <= ISNULL(b.[EndDate], @CheckDate)

UNION ALL

SELECT b.[ProductAssemblyID], b.[ComponentID], p.[Name], b.[PerAssemblyQty], p.[StandardCost], p.[ListPrice], b.[BOMLevel], [RecursionLevel] + 1 -- Join recursive member to anchor

FROM [BOM\_cte] cte

INNER JOIN [Production].[BillOfMaterials] b

ON b.[ProductAssemblyID] = cte.[ComponentID]

INNER JOIN [Production].[Product] p

ON b.[ComponentID] = p.[ProductID]

WHERE @CheckDate >= b.[StartDate]

AND @CheckDate <= ISNULL(b.[EndDate], @CheckDate)

)

-- Outer select from the CTE

SELECT b.[ProductAssemblyID], b.[ComponentID], b.[ComponentDesc], SUM(b.[PerAssemblyQty]) AS [TotalQuantity] , b.[StandardCost], b.[ListPrice], b.[BOMLevel], b.[RecursionLevel]

FROM [BOM\_cte] b

GROUP BY b.[ComponentID], b.[ComponentDesc], b.[ProductAssemblyID], b.[BOMLevel], b.[RecursionLevel], b.[StandardCost], b.[ListPrice]

ORDER BY b.[BOMLevel], b.[ProductAssemblyID], b.[ComponentID]

OPTION (MAXRECURSION 25)

END;

GO

----------------------------------------------------------------------------------------------------------(Co

CREATE PROCEDURE [dbo].[uspGetEmployeeManagers]

@BusinessEntityID [int]

AS

BEGIN

SET NOCOUNT ON;

-- Use recursive query to list out all Employees required for a particular Manager

WITH [EMP\_cte]([BusinessEntityID], [OrganizationNode], [FirstName], [LastName], [JobTitle], [RecursionLevel])

-- CTE name and columns

AS (

SELECT e.[BusinessEntityID], e.[OrganizationNode], p.[FirstName], p.[LastName], e.[JobTitle], 0

-- Get the initial Employee

FROM [HumanResources].[Employee] e

INNER JOIN [Person].[Person] as p

ON p.[BusinessEntityID] = e.[BusinessEntityID]

WHERE e.[BusinessEntityID] = @BusinessEntityID

UNION ALL

SELECT e.[BusinessEntityID], e.[OrganizationNode], p.[FirstName], p.[LastName], e.[JobTitle], [RecursionLevel] + 1 -- Join recursive member to anchor

FROM [HumanResources].[Employee] e

INNER JOIN [EMP\_cte]

ON e.[OrganizationNode] = [EMP\_cte].[OrganizationNode].GetAncestor(1)

INNER JOIN [Person].[Person] p

ON p.[BusinessEntityID] = e.[BusinessEntityID]

)

-- Join back to Employee to return the manager name

SELECT [EMP\_cte].[RecursionLevel], [EMP\_cte].[BusinessEntityID], [EMP\_cte].[FirstName], [EMP\_cte].[LastName],

[EMP\_cte].[OrganizationNode].ToString() AS [OrganizationNode], p.[FirstName] AS 'ManagerFirstName', p.[LastName] AS 'ManagerLastName' -- Outer select from the CTE

FROM [EMP\_cte]

INNER JOIN [HumanResources].[Employee] e

ON EMP\_cte].[OrganizationNode].GetAncestor(1) = e.[OrganizationNode]

INNER JOIN [Person].[Person] p

ON p.[BusinessEntityID] = e.[BusinessEntityID]

ORDER BY [RecursionLevel], [EMP\_cte].[OrganizationNode].ToString()

OPTION (MAXRECURSION 25)

END;

GO

------------------------------------------------------------------------------------------------------------

CREATE PROCEDURE [dbo].[uspGetManagerEmployees]

@BusinessEntityID [int]

AS

BEGIN

SET NOCOUNT ON;

-- Use recursive query to list out all Employees required for a particular Manager

WITH [EMP\_cte]([BusinessEntityID], [OrganizationNode], [FirstName], [LastName], [RecursionLevel])

-- CTE name and columns

AS (

SELECT e.[BusinessEntityID], e.[OrganizationNode], p.[FirstName], p.[LastName], 0

-- Get the initial list of Employees for Manager n

FROM [HumanResources].[Employee] e

INNER JOIN [Person].[Person] p

ON p.[BusinessEntityID] = e.[BusinessEntityID]

WHERE e.[BusinessEntityID] = @BusinessEntityID

UNION ALL

SELECT e.[BusinessEntityID], e.[OrganizationNode], p.[FirstName], p.[LastName], [RecursionLevel] + 1

-- Join recursive member to anchor

FROM [HumanResources].[Employee] e

INNER JOIN [EMP\_cte]

ON e.[OrganizationNode].GetAncestor(1) = [EMP\_cte].[OrganizationNode]

INNER JOIN [Person].[Person] p

ON p.[BusinessEntityID] = e.[BusinessEntityID]

)

-- Join back to Employee to return the manager name

SELECT [EMP\_cte].[RecursionLevel], [EMP\_cte].[OrganizationNode].ToString() as [OrganizationNode], p.[FirstName] AS 'ManagerFirstName', p.[LastName] AS 'ManagerLastName',

[EMP\_cte].[BusinessEntityID], [EMP\_cte].[FirstName], [EMP\_cte].[LastName]

-- Outer select from the CTE

FROM [EMP\_cte]

INNER JOIN [HumanResources].[Employee] e

ON [EMP\_cte].[OrganizationNode].GetAncestor(1) = e.[OrganizationNode]

INNER JOIN [Person].[Person] p

ON p.[BusinessEntityID] = e.[BusinessEntityID]

ORDER BY [RecursionLevel], [EMP\_cte].[OrganizationNode].ToString()

OPTION (MAXRECURSION 25)

END;

GO

CREATE PROCEDURE [dbo].[uspGetWhereUsedProductID]

@StartProductID [int],

@CheckDate [datetime]

AS

BEGIN

SET NOCOUNT ON;

--Use recursive query to generate a multi-level Bill of Material (i.e. all level 1 components of a level 0 assembly, all level 2 components of a level 1 assembly)

WITH [BOM\_cte]([ProductAssemblyID], [ComponentID], [ComponentDesc], [PerAssemblyQty], [StandardCost], [ListPrice], [BOMLevel], [RecursionLevel])

-- CTE name and columns

AS (

SELECT b.[ProductAssemblyID], b.[ComponentID], p.[Name], b.[PerAssemblyQty], p.[StandardCost], p.[ListPrice], b.[BOMLevel], 0

-- Get the initial list of components for the bike assembly

FROM [Production].[BillOfMaterials] b

INNER JOIN [Production].[Product] p

ON b.[ProductAssemblyID] = p.[ProductID]

WHERE b.[ComponentID] = @StartProductID

AND @CheckDate >= b.[StartDate]

AND @CheckDate <= ISNULL(b.[EndDate], @CheckDate)

UNION ALL

SELECT b.[ProductAssemblyID], b.[ComponentID], p.[Name], b.[PerAssemblyQty], p.[StandardCost], p.[ListPrice], b.[BOMLevel], [RecursionLevel] + 1

-- Join recursive member to anchor

FROM [BOM\_cte] cte

INNER JOIN [Production].[BillOfMaterials] b

ON cte.[ProductAssemblyID] = b.[ComponentID]

INNER JOIN [Production].[Product] p

ON b.[ProductAssemblyID] = p.[ProductID]

WHERE @CheckDate >= b.[StartDate]

AND @CheckDate <= ISNULL(b.[EndDate], @CheckDate)

)

-- Outer select from the CTE

SELECT b.[ProductAssemblyID], b.[ComponentID], b.[ComponentDesc], SUM(b.[PerAssemblyQty]) AS [TotalQuantity] , b.[StandardCost], b.[ListPrice], b.[BOMLevel], b.[RecursionLevel]

FROM [BOM\_cte] b

GROUP BY b.[ComponentID], b.[ComponentDesc], b.[ProductAssemblyID], b.[BOMLevel], b.[RecursionLevel], b.[StandardCost], b.[ListPrice]

ORDER BY b.[BOMLevel], b.[ProductAssemblyID], b.[ComponentID]

OPTION (MAXRECURSION 25)

END;

GO

-----------------------------------------------------------------------------------------------------

CREATE PROCEDURE [HumanResources].[uspUpdateEmployeeHireInfo]

@BusinessEntityID [int],

@JobTitle [nvarchar](50),

@HireDate [datetime],

@RateChangeDate [datetime],

@Rate [money],

@PayFrequency [tinyint],

@CurrentFlag [dbo].[Flag]

WITH EXECUTE AS CALLER

AS

BEGIN

SET NOCOUNT ON;

BEGIN TRY

BEGIN TRANSACTION;

UPDATE [HumanResources].[Employee]

SET [JobTitle] = @JobTitle

,[HireDate] = @HireDate

,[CurrentFlag] = @CurrentFlag

WHERE [BusinessEntityID] = @BusinessEntityID;

INSERT INTO [HumanResources].[EmployeePayHistory]

([BusinessEntityID]

,[RateChangeDate]

,[Rate]

,[PayFrequency])

VALUES (@BusinessEntityID, @RateChangeDate, @Rate, @PayFrequency);

COMMIT TRANSACTION;

END TRY

BEGIN CATCH

-- Rollback any active or uncommittable transactions before

-- inserting information in the ErrorLog

IF @@TRANCOUNT > 0

BEGIN

ROLLBACK TRANSACTION;

END

EXECUTE [dbo].[uspLogError];

END CATCH;

END;

GO

CREATE PROCEDURE [HumanResources].[uspUpdateEmployeeLogin]

@BusinessEntityID [int],

@OrganizationNode [hierarchyid],

@LoginID [nvarchar](256),

@JobTitle [nvarchar](50),

@HireDate [datetime],

@CurrentFlag [dbo].[Flag]

WITH EXECUTE AS CALLER

AS

BEGIN

SET NOCOUNT ON;

BEGIN TRY

UPDATE [HumanResources].[Employee]

SET [OrganizationNode] = @OrganizationNode

,[LoginID] = @LoginID

,[JobTitle] = @JobTitle

,[HireDate] = @HireDate

,[CurrentFlag] = @CurrentFlag

WHERE [BusinessEntityID] = @BusinessEntityID;

END TRY

BEGIN CATCH

EXECUTE [dbo].[uspLogError];

END CATCH;

END;

GO

CREATE PROCEDURE [HumanResources].[uspUpdateEmployeePersonalInfo]

@BusinessEntityID [int],

@NationalIDNumber [nvarchar](15),

@BirthDate [datetime],

@MaritalStatus [nchar](1),

@Gender [nchar](1)

WITH EXECUTE AS CALLER

AS

BEGIN

SET NOCOUNT ON;

BEGIN TRY

UPDATE [HumanResources].[Employee]

SET [NationalIDNumber] = @NationalIDNumber

,[BirthDate] = @BirthDate

,[MaritalStatus] = @MaritalStatus

,[Gender] = @Gender

WHERE [BusinessEntityID] = @BusinessEntityID;

END TRY

BEGIN CATCH

EXECUTE [dbo].[uspLogError];

END CATCH;

END;

GO

--A stored procedure which demonstrates integrated full text search

CREATE PROCEDURE [dbo].[uspSearchCandidateResumes]

@searchString [nvarchar](1000),

@useInflectional [bit]=0,

@useThesaurus [bit]=0,

@language[int]=0

WITH EXECUTE AS CALLER

AS

BEGIN

SET NOCOUNT ON;

DECLARE @string nvarchar(1050)

--setting the lcid to the default instance LCID if needed

IF @language = NULL OR @language = 0

BEGIN

SELECT @language =CONVERT(int, serverproperty('lcid'))

END

--FREETEXTTABLE case as inflectional and Thesaurus were required

IF @useThesaurus = 1 AND @useInflectional = 1

BEGIN

SELECT FT\_TBL.[JobCandidateID], KEY\_TBL.[RANK] FROM [HumanResources].[JobCandidate] AS FT\_TBL

INNER JOIN FREETEXTTABLE([HumanResources].[JobCandidate],\*, @searchString,LANGUAGE @language) AS KEY\_TBL

ON FT\_TBL.[JobCandidateID] =KEY\_TBL.[KEY]

END

ELSE IF @useThesaurus = 1

BEGIN

SELECT @string ='FORMSOF(THESAURUS,"'+@searchString +'"'+')'

SELECT FT\_TBL.[JobCandidateID], KEY\_TBL.[RANK] FROM [HumanResources].[JobCandidate] AS FT\_TBL

INNER JOIN CONTAINSTABLE([HumanResources].[JobCandidate],\*, @string,LANGUAGE @language) AS KEY\_TBL

ON FT\_TBL.[JobCandidateID] =KEY\_TBL.[KEY]

END

ELSE IF @useInflectional = 1

BEGIN

SELECT @string ='FORMSOF(INFLECTIONAL,"'+@searchString +'"'+')'

SELECT FT\_TBL.[JobCandidateID], KEY\_TBL.[RANK] FROM [HumanResources].[JobCandidate] AS FT\_TBL

INNER JOIN CONTAINSTABLE([HumanResources].[JobCandidate],\*, @string,LANGUAGE @language) AS KEY\_TBL

ON FT\_TBL.[JobCandidateID] =KEY\_TBL.[KEY]

END

ELSE --base case, plain CONTAINSTABLE

BEGIN

SELECT @string='"'+@searchString +'"'

SELECT FT\_TBL.[JobCandidateID],KEY\_TBL.[RANK] FROM [HumanResources].[JobCandidate] AS FT\_TBL

INNER JOIN CONTAINSTABLE([HumanResources].[JobCandidate],\*,@string,LANGUAGE @language) AS KEY\_TBL

ON FT\_TBL.[JobCandidateID] =KEY\_TBL.[KEY]

END

END;

GO

[TOP](#Top)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Add Extended Properties

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Can add extended properties to a database, table, column, filegroup, schema, table, role, type and user

<https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-addextendedproperty-transact-sql>

PRINT '';

PRINT '\*\*\* Creating Extended Properties';

GO

SET NOCOUNT ON;

GO

PRINT ' Database';

GO

-- Database

EXECUTE [sys].[sp\_addextendedproperty]

N'MS\_Description', N'AdventureWorks 2016 Sample OLTP Database',

NULL, NULL,

NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty]

N'MS\_Description', N'Database trigger to audit all of the DDL changes made to the AdventureWorks 2016 database.',

N'TRIGGER', [ddlDatabaseTriggerLog],

NULL, NULL,

NULL, NULL;

GO

PRINT ' Files and Filegroups';

GO

-- Files and Filegroups

EXECUTE [sys].[sp\_addextendedproperty]

N'MS\_Description', N'Primary filegroup for the AdventureWorks 2016 sample database.',

N'FILEGROUP', [PRIMARY];

GO

PRINT ' Schemas';

GO

-- Schemas

EXECUTE [sys].[sp\_addextendedproperty]

N'MS\_Description', N'Contains objects related to employees and departments.',

N'SCHEMA', [HumanResources],

NULL, NULL,

NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty]

N'MS\_Description', N'Contains objects related to products, inventory, and manufacturing.',

N'SCHEMA', [Production],

NULL, NULL,

NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty]

N'MS\_Description', N'Contains objects related to vendors and purchase orders.',

N'SCHEMA', [Purchasing],

NULL, NULL,

NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty]

N'MS\_Description', N'Contains objects related to customers, sales orders, and sales territories.',

N'SCHEMA', [Sales],

NULL, NULL,

NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty]

N'MS\_Description', N'Contains objects related to names and addresses of customers, vendors, and employees',

N'SCHEMA', [Person],

NULL, NULL,

NULL, NULL;

GO

PRINT ' Tables and Columns';

GO

-- Tables and Columns

---------------------------------------------- EXCERPT ----------------------------------------------------------

Can add extended properties to a database, table, column, filegroup, schema, table, role, type and user

<https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-addextendedproperty-transact-sql>

F. Adding an extended property to a table

1. The following example adds an extended property to the Address table in the Person schema. +

USE AdventureWorks2012;

GO

EXEC sys.sp\_addextendedproperty

@name = N'MS\_DescriptionExample',

@value = N'Street address information for customers, employees, and vendors.',

@level0type = N'SCHEMA', @level0name = 'Person',

@level1type = N'TABLE', @level1name = 'Address';

GO

------------------------------------------------------------------------------------------------------------------

EXECUTE [sys].[sp\_addextendedproperty]

N'MS\_Description', N'Street address information for customers, employees, and vendors.',

N'SCHEMA', [Person],

N'TABLE', [Address],

NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty]

N'MS\_Description', N'Primary key for Address records.',

N'SCHEMA', [Person],

N'TABLE', [Address],

N'COLUMN', [AddressID];

EXECUTE [sys].[sp\_addextendedproperty]

N'MS\_Description', N'First street address line.',

N'SCHEMA', [Person],

N'TABLE', [Address],

N'COLUMN', [AddressLine1];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Second street address line.', N'SCHEMA', [Person], N'TABLE', [Address], N'COLUMN', [AddressLine2];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Name of the city.', N'SCHEMA', [Person], N'TABLE', [Address], N'COLUMN', [City];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique identification number for the state or province. Foreign key to StateProvince table.', N'SCHEMA', [Person], N'TABLE', [Address], N'COLUMN', [StateProvinceID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Postal code for the street address.', N'SCHEMA', [Person], N'TABLE', [Address], N'COLUMN', [PostalCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Latitude and longitude of this address.', N'SCHEMA', [Person], N'TABLE', [Address], N'COLUMN', [SpatialLocation];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Person], N'TABLE', [Address], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Person], N'TABLE', [Address], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Types of addresses stored in the Address table. ', N'SCHEMA', [Person], N'TABLE', [AddressType], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for AddressType records.', N'SCHEMA', [Person], N'TABLE', [AddressType], N'COLUMN', [AddressTypeID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Address type description. For example, Billing, Home, or Shipping.', N'SCHEMA', [Person], N'TABLE', [AddressType], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Person], N'TABLE', [AddressType], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Person], N'TABLE', [AddressType], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Current version number of the AdventureWorks 2016 sample database. ', N'SCHEMA', [dbo], N'TABLE', [AWBuildVersion], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for AWBuildVersion records.', N'SCHEMA', [dbo], N'TABLE', [AWBuildVersion], N'COLUMN', [SystemInformationID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Version number of the database in 9.yy.mm.dd.00 format.', N'SCHEMA', [dbo], N'TABLE', [AWBuildVersion], N'COLUMN', [Database Version];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [dbo], N'TABLE', [AWBuildVersion], N'COLUMN', [VersionDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [dbo], N'TABLE', [AWBuildVersion], N'COLUMN', [ModifiedDate];

GO

--- BillOfMaterials

EXECUTE [sys].[sp\_addextendedproperty]

N'MS\_Description', N'Items required to make bicycles and bicycle subassemblies. It identifies the heirarchical relationship between a parent product and its components.',

N'SCHEMA', [Production],

N'TABLE', [BillOfMaterials],

NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty]

N'MS\_Description', N'Primary key for BillOfMaterials records.',

N'SCHEMA', [Production],

N'TABLE', [BillOfMaterials],

N'COLUMN', [BillOfMaterialsID];

EXECUTE [sys].[sp\_addextendedproperty]

N'MS\_Description', N'Parent product identification number. Foreign key to Product.ProductID.',

N'SCHEMA', [Production],

N'TABLE', [BillOfMaterials],

N'COLUMN', [ProductAssemblyID];

EXECUTE [sys].[sp\_addextendedproperty]

N'MS\_Description', N'Component identification number. Foreign key to Product.ProductID.',

N'SCHEMA', [Production],

N'TABLE', [BillOfMaterials],

N'COLUMN', [ComponentID];

EXECUTE [sys].[sp\_addextendedproperty]

N'MS\_Description', N'Date the component started being used in the assembly item.',

N'SCHEMA', [Production],

N'TABLE', [BillOfMaterials],

N'COLUMN', [StartDate];

EXECUTE [sys].[sp\_addextendedproperty]

N'MS\_Description', N'Date the component stopped being used in the assembly item.',

N'SCHEMA', [Production],

N'TABLE', [BillOfMaterials],

N'COLUMN', [EndDate];

EXECUTE [sys].[sp\_addextendedproperty]

N'MS\_Description', N'Standard code identifying the unit of measure for the quantity.',

N'SCHEMA', [Production],

N'TABLE', [BillOfMaterials],

N'COLUMN', [UnitMeasureCode];

EXECUTE [sys].[sp\_addextendedproperty]

N'MS\_Description', N'Indicates the depth the component is from its parent (AssemblyID).',

N'SCHEMA', [Production],

N'TABLE', [BillOfMaterials],

N'COLUMN', [BOMLevel];

EXECUTE [sys].[sp\_addextendedproperty]

N'MS\_Description', N'Quantity of the component needed to create the assembly.',

N'SCHEMA', [Production],

N'TABLE', [BillOfMaterials],

N'COLUMN', [PerAssemblyQty];

EXECUTE [sys].[sp\_addextendedproperty]

N'MS\_Description', N'Date and time the record was last updated.',

N'SCHEMA', [Production],

N'TABLE', [BillOfMaterials],

N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Source of the ID that connects vendors, customers, and employees with address and contact information.', N'SCHEMA', [Person], N'TABLE', [BusinessEntity], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for all customers, vendors, and employees.', N'SCHEMA', [Person], N'TABLE', [BusinessEntity], N'COLUMN', [BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Person], N'TABLE', [BusinessEntity], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Person], N'TABLE', [BusinessEntity], N'COLUMN', [ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Cross-reference table mapping customers, vendors, and employees to their addresses.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityAddress], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Foreign key to BusinessEntity.BusinessEntityID.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityAddress], N'COLUMN', [BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Foreign key to Address.AddressID.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityAddress], N'COLUMN', [AddressID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Foreign key to AddressType.AddressTypeID.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityAddress], N'COLUMN', [AddressTypeID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityAddress], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityAddress], N'COLUMN', [ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Cross-reference table mapping stores, vendors, and employees to people', N'SCHEMA', [Person], N'TABLE', [BusinessEntityContact], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Foreign key to BusinessEntity.BusinessEntityID.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityContact], N'COLUMN', [BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Foreign key to Person.BusinessEntityID.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityContact], N'COLUMN', [PersonID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Foreign key to ContactType.ContactTypeID.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityContact], N'COLUMN', [ContactTypeID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityContact], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityContact], N'COLUMN', [ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Lookup table containing the types of business entity contacts.', N'SCHEMA', [Person], N'TABLE', [ContactType], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for ContactType records.', N'SCHEMA', [Person], N'TABLE', [ContactType], N'COLUMN', [ContactTypeID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Contact type description.', N'SCHEMA', [Person], N'TABLE', [ContactType], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Person], N'TABLE', [ContactType], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Cross-reference table mapping ISO currency codes to a country or region.', N'SCHEMA', [Sales], N'TABLE', [CountryRegionCurrency], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ISO code for countries and regions. Foreign key to CountryRegion.CountryRegionCode.', N'SCHEMA', [Sales], N'TABLE', [CountryRegionCurrency], N'COLUMN', [CountryRegionCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ISO standard currency code. Foreign key to Currency.CurrencyCode.', N'SCHEMA', [Sales], N'TABLE', [CountryRegionCurrency], N'COLUMN', [CurrencyCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Sales], N'TABLE', [CountryRegionCurrency], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Lookup table containing the ISO standard codes for countries and regions.', N'SCHEMA', [Person], N'TABLE', [CountryRegion], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ISO standard code for countries and regions.', N'SCHEMA', [Person], N'TABLE', [CountryRegion], N'COLUMN', [CountryRegionCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Country or region name.', N'SCHEMA', [Person], N'TABLE', [CountryRegion], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Person], N'TABLE', [CountryRegion], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Customer credit card information.', N'SCHEMA', [Sales], N'TABLE', [CreditCard], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for CreditCard records.', N'SCHEMA', [Sales], N'TABLE', [CreditCard], N'COLUMN', [CreditCardID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Credit card name.', N'SCHEMA', [Sales], N'TABLE', [CreditCard], N'COLUMN', [CardType];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Credit card number.', N'SCHEMA', [Sales], N'TABLE', [CreditCard], N'COLUMN', [CardNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Credit card expiration month.', N'SCHEMA', [Sales], N'TABLE', [CreditCard], N'COLUMN', [ExpMonth];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Credit card expiration year.', N'SCHEMA', [Sales], N'TABLE', [CreditCard], N'COLUMN', [ExpYear];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Sales], N'TABLE', [CreditCard], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Lookup table containing the languages in which some AdventureWorks data is stored.', N'SCHEMA', [Production], N'TABLE', [Culture], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for Culture records.', N'SCHEMA', [Production], N'TABLE', [Culture], N'COLUMN', [CultureID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Culture description.', N'SCHEMA', [Production], N'TABLE', [Culture], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [Culture], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Lookup table containing standard ISO currencies.', N'SCHEMA', [Sales], N'TABLE', [Currency], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The ISO code for the Currency.', N'SCHEMA', [Sales], N'TABLE', [Currency], N'COLUMN', [CurrencyCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Currency name.', N'SCHEMA', [Sales], N'TABLE', [Currency], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Sales], N'TABLE', [Currency], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Currency exchange rates.', N'SCHEMA', [Sales], N'TABLE', [CurrencyRate], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for CurrencyRate records.', N'SCHEMA', [Sales], N'TABLE', [CurrencyRate], N'COLUMN', [CurrencyRateID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the exchange rate was obtained.', N'SCHEMA', [Sales], N'TABLE', [CurrencyRate], N'COLUMN', [CurrencyRateDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Exchange rate was converted from this currency code.', N'SCHEMA', [Sales], N'TABLE', [CurrencyRate], N'COLUMN', [FromCurrencyCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Exchange rate was converted to this currency code.', N'SCHEMA', [Sales], N'TABLE', [CurrencyRate], N'COLUMN', [ToCurrencyCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Average exchange rate for the day.', N'SCHEMA', [Sales], N'TABLE', [CurrencyRate], N'COLUMN', [AverageRate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Final exchange rate for the day.', N'SCHEMA', [Sales], N'TABLE', [CurrencyRate], N'COLUMN', [EndOfDayRate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Sales], N'TABLE', [CurrencyRate], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Current customer information. Also see the Person and Store tables.', N'SCHEMA', [Sales], N'TABLE', [Customer], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key.', N'SCHEMA', [Sales], N'TABLE', [Customer], N'COLUMN', [CustomerID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key to Person.BusinessEntityID', N'SCHEMA', [Sales], N'TABLE', [Customer], N'COLUMN', [PersonID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key to Store.BusinessEntityID', N'SCHEMA', [Sales], N'TABLE', [Customer], N'COLUMN', [StoreID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ID of the territory in which the customer is located. Foreign key to SalesTerritory.SalesTerritoryID.', N'SCHEMA', [Sales], N'TABLE', [Customer], N'COLUMN', [TerritoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique number identifying the customer assigned by the accounting system.', N'SCHEMA', [Sales], N'TABLE', [Customer], N'COLUMN', [AccountNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Sales], N'TABLE', [Customer], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Sales], N'TABLE', [Customer], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Audit table tracking all DDL changes made to the AdventureWorks database. Data is captured by the database trigger ddlDatabaseTriggerLog.', N'SCHEMA', [dbo], N'TABLE', [DatabaseLog], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for DatabaseLog records.', N'SCHEMA', [dbo], N'TABLE', [DatabaseLog], N'COLUMN', [DatabaseLogID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The date and time the DDL change occurred.', N'SCHEMA', [dbo], N'TABLE', [DatabaseLog], N'COLUMN', [PostTime];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The user who implemented the DDL change.', N'SCHEMA', [dbo], N'TABLE', [DatabaseLog], N'COLUMN', [DatabaseUser];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The type of DDL statement that was executed.', N'SCHEMA', [dbo], N'TABLE', [DatabaseLog], N'COLUMN', [Event];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The schema to which the changed object belongs.', N'SCHEMA', [dbo], N'TABLE', [DatabaseLog], N'COLUMN', [Schema];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The object that was changed by the DDL statment.', N'SCHEMA', [dbo], N'TABLE', [DatabaseLog], N'COLUMN', [Object];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The exact Transact-SQL statement that was executed.', N'SCHEMA', [dbo], N'TABLE', [DatabaseLog], N'COLUMN', [TSQL];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The raw XML data generated by database trigger.', N'SCHEMA', [dbo], N'TABLE', [DatabaseLog], N'COLUMN', [XmlEvent];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Lookup table containing the departments within the Adventure Works Cycles company.', N'SCHEMA', [HumanResources], N'TABLE', [Department], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for Department records.', N'SCHEMA', [HumanResources], N'TABLE', [Department], N'COLUMN', [DepartmentID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Name of the department.', N'SCHEMA', [HumanResources], N'TABLE', [Department], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Name of the group to which the department belongs.', N'SCHEMA', [HumanResources], N'TABLE', [Department], N'COLUMN', [GroupName];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [HumanResources], N'TABLE', [Department], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product maintenance documents.', N'SCHEMA', [Production], N'TABLE', [Document], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for Document records.', N'SCHEMA', [Production], N'TABLE', [Document], N'COLUMN', [DocumentNode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Depth in the document hierarchy.', N'SCHEMA', [Production], N'TABLE', [Document], N'COLUMN', [DocumentLevel];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Title of the document.', N'SCHEMA', [Production], N'TABLE', [Document], N'COLUMN', [Title];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Employee who controls the document. Foreign key to Employee.BusinessEntityID', N'SCHEMA', [Production], N'TABLE', [Document], N'COLUMN', [Owner];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'0 = This is a folder, 1 = This is a document.', N'SCHEMA', [Production], N'TABLE', [Document], N'COLUMN', [FolderFlag];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'File name of the document', N'SCHEMA', [Production], N'TABLE', [Document], N'COLUMN', [FileName];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'File extension indicating the document type. For example, .doc or .txt.', N'SCHEMA', [Production], N'TABLE', [Document], N'COLUMN', [FileExtension];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Revision number of the document. ', N'SCHEMA', [Production], N'TABLE', [Document], N'COLUMN', [Revision];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Engineering change approval number.', N'SCHEMA', [Production], N'TABLE', [Document], N'COLUMN', [ChangeNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'1 = Pending approval, 2 = Approved, 3 = Obsolete', N'SCHEMA', [Production], N'TABLE', [Document], N'COLUMN', [Status];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Document abstract.', N'SCHEMA', [Production], N'TABLE', [Document], N'COLUMN', [DocumentSummary];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Complete document.', N'SCHEMA', [Production], N'TABLE', [Document], N'COLUMN', [Document];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Required for FileStream.', N'SCHEMA', [Production], N'TABLE', [Document], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [Document], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Where to send a person email.', N'SCHEMA', [Person], N'TABLE', [EmailAddress], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Person associated with this email address. Foreign key to Person.BusinessEntityID', N'SCHEMA', [Person], N'TABLE', [EmailAddress], N'COLUMN', [BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. ID of this email address.', N'SCHEMA', [Person], N'TABLE', [EmailAddress], N'COLUMN', [EmailAddressID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'E-mail address for the person.', N'SCHEMA', [Person], N'TABLE', [EmailAddress], N'COLUMN', [EmailAddress];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Person], N'TABLE', [EmailAddress], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Person], N'TABLE', [EmailAddress], N'COLUMN', [ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Employee information such as salary, department, and title.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for Employee records. Foreign key to BusinessEntity.BusinessEntityID.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'COLUMN', [BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique national identification number such as a social security number.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'COLUMN', [NationalIDNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Network login.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'COLUMN', [LoginID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Where the employee is located in corporate hierarchy.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'COLUMN', [OrganizationNode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The depth of the employee in the corporate hierarchy.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'COLUMN', [OrganizationLevel];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Work title such as Buyer or Sales Representative.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'COLUMN', [JobTitle];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date of birth.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'COLUMN', [BirthDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'M = Married, S = Single', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'COLUMN', [MaritalStatus];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'M = Male, F = Female', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'COLUMN', [Gender];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Employee hired on this date.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'COLUMN', [HireDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Job classification. 0 = Hourly, not exempt from collective bargaining. 1 = Salaried, exempt from collective bargaining.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'COLUMN', [SalariedFlag];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Number of available vacation hours.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'COLUMN', [VacationHours];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Number of available sick leave hours.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'COLUMN', [SickLeaveHours];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'0 = Inactive, 1 = Active', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'COLUMN', [CurrentFlag];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Employee department transfers.', N'SCHEMA', [HumanResources], N'TABLE', [EmployeeDepartmentHistory], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Employee identification number. Foreign key to Employee.BusinessEntityID.', N'SCHEMA', [HumanResources], N'TABLE', [EmployeeDepartmentHistory], N'COLUMN', [BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Department in which the employee worked including currently. Foreign key to Department.DepartmentID.', N'SCHEMA', [HumanResources], N'TABLE', [EmployeeDepartmentHistory], N'COLUMN', [DepartmentID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Identifies which 8-hour shift the employee works. Foreign key to Shift.Shift.ID.', N'SCHEMA', [HumanResources], N'TABLE', [EmployeeDepartmentHistory], N'COLUMN', [ShiftID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date the employee started work in the department.', N'SCHEMA', [HumanResources], N'TABLE', [EmployeeDepartmentHistory], N'COLUMN', [StartDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date the employee left the department. NULL = Current department.', N'SCHEMA', [HumanResources], N'TABLE', [EmployeeDepartmentHistory], N'COLUMN', [EndDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [HumanResources], N'TABLE', [EmployeeDepartmentHistory], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Employee pay history.', N'SCHEMA', [HumanResources], N'TABLE', [EmployeePayHistory], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Employee identification number. Foreign key to Employee.BusinessEntityID.', N'SCHEMA', [HumanResources], N'TABLE', [EmployeePayHistory], N'COLUMN', [BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date the change in pay is effective', N'SCHEMA', [HumanResources], N'TABLE', [EmployeePayHistory], N'COLUMN', [RateChangeDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Salary hourly rate.', N'SCHEMA', [HumanResources], N'TABLE', [EmployeePayHistory], N'COLUMN', [Rate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'1 = Salary received monthly, 2 = Salary received biweekly', N'SCHEMA', [HumanResources], N'TABLE', [EmployeePayHistory], N'COLUMN', [PayFrequency];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [HumanResources], N'TABLE', [EmployeePayHistory], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Audit table tracking errors in the the AdventureWorks database that are caught by the CATCH block of a TRY...CATCH construct. Data is inserted by stored procedure dbo.uspLogError when it is executed from inside the CATCH block of a TRY...CATCH construct.', N'SCHEMA', [dbo], N'TABLE', [ErrorLog], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for ErrorLog records.', N'SCHEMA', [dbo], N'TABLE', [ErrorLog], N'COLUMN', [ErrorLogID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The date and time at which the error occurred.', N'SCHEMA', [dbo], N'TABLE', [ErrorLog], N'COLUMN', [ErrorTime];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The user who executed the batch in which the error occurred.', N'SCHEMA', [dbo], N'TABLE', [ErrorLog], N'COLUMN', [UserName];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The error number of the error that occurred.', N'SCHEMA', [dbo], N'TABLE', [ErrorLog], N'COLUMN', [ErrorNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The severity of the error that occurred.', N'SCHEMA', [dbo], N'TABLE', [ErrorLog], N'COLUMN', [ErrorSeverity];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The state number of the error that occurred.', N'SCHEMA', [dbo], N'TABLE', [ErrorLog], N'COLUMN', [ErrorState];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The name of the stored procedure or trigger where the error occurred.', N'SCHEMA', [dbo], N'TABLE', [ErrorLog], N'COLUMN', [ErrorProcedure];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The line number at which the error occurred.', N'SCHEMA', [dbo], N'TABLE', [ErrorLog], N'COLUMN', [ErrorLine];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The message text of the error that occurred.', N'SCHEMA', [dbo], N'TABLE', [ErrorLog], N'COLUMN', [ErrorMessage];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Bicycle assembly diagrams.', N'SCHEMA', [Production], N'TABLE', [Illustration], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for Illustration records.', N'SCHEMA', [Production], N'TABLE', [Illustration], N'COLUMN', [IllustrationID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Illustrations used in manufacturing instructions. Stored as XML.', N'SCHEMA', [Production], N'TABLE', [Illustration], N'COLUMN', [Diagram];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [Illustration], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Résumés submitted to Human Resources by job applicants.', N'SCHEMA', [HumanResources], N'TABLE', [JobCandidate], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for JobCandidate records.', N'SCHEMA', [HumanResources], N'TABLE', [JobCandidate], N'COLUMN', [JobCandidateID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Employee identification number if applicant was hired. Foreign key to Employee.BusinessEntityID.', N'SCHEMA', [HumanResources], N'TABLE', [JobCandidate], N'COLUMN', [BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Résumé in XML format.', N'SCHEMA', [HumanResources], N'TABLE', [JobCandidate], N'COLUMN', [Resume];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [HumanResources], N'TABLE', [JobCandidate], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product inventory and manufacturing locations.', N'SCHEMA', [Production], N'TABLE', [Location], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for Location records.', N'SCHEMA', [Production], N'TABLE', [Location], N'COLUMN', [LocationID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Location description.', N'SCHEMA', [Production], N'TABLE', [Location], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Standard hourly cost of the manufacturing location.', N'SCHEMA', [Production], N'TABLE', [Location], N'COLUMN', [CostRate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Work capacity (in hours) of the manufacturing location.', N'SCHEMA', [Production], N'TABLE', [Location], N'COLUMN', [Availability];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [Location], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'One way hashed authentication information', N'SCHEMA', [Person], N'TABLE', [Password], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Password for the e-mail account.', N'SCHEMA', [Person], N'TABLE', [Password], N'COLUMN', [PasswordHash];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Random value concatenated with the password string before the password is hashed.', N'SCHEMA', [Person], N'TABLE', [Password], N'COLUMN', [PasswordSalt];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Person], N'TABLE', [Password], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Person], N'TABLE', [Password], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Human beings involved with AdventureWorks: employees, customer contacts, and vendor contacts.', N'SCHEMA', [Person], N'TABLE', [Person], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for Person records.', N'SCHEMA', [Person], N'TABLE', [Person], N'COLUMN', [BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary type of person: SC = Store Contact, IN = Individual (retail) customer, SP = Sales person, EM = Employee (non-sales), VC = Vendor contact, GC = General contact', N'SCHEMA', [Person], N'TABLE', [Person], N'COLUMN', [PersonType];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'0 = The data in FirstName and LastName are stored in western style (first name, last name) order. 1 = Eastern style (last name, first name) order.', N'SCHEMA', [Person], N'TABLE', [Person], N'COLUMN', [NameStyle];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'A courtesy title. For example, Mr. or Ms.', N'SCHEMA', [Person], N'TABLE', [Person], N'COLUMN', [Title];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'First name of the person.', N'SCHEMA', [Person], N'TABLE', [Person], N'COLUMN', [FirstName];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Middle name or middle initial of the person.', N'SCHEMA', [Person], N'TABLE', [Person], N'COLUMN', [MiddleName];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Last name of the person.', N'SCHEMA', [Person], N'TABLE', [Person], N'COLUMN', [LastName];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Surname suffix. For example, Sr. or Jr.', N'SCHEMA', [Person], N'TABLE', [Person], N'COLUMN', [Suffix];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'0 = Contact does not wish to receive e-mail promotions, 1 = Contact does wish to receive e-mail promotions from AdventureWorks, 2 = Contact does wish to receive e-mail promotions from AdventureWorks and selected partners. ', N'SCHEMA', [Person], N'TABLE', [Person], N'COLUMN', [EmailPromotion];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Personal information such as hobbies, and income collected from online shoppers. Used for sales analysis.', N'SCHEMA', [Person], N'TABLE', [Person], N'COLUMN', [Demographics];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Additional contact information about the person stored in xml format. ', N'SCHEMA', [Person], N'TABLE', [Person], N'COLUMN', [AdditionalContactInfo];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Person], N'TABLE', [Person], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Person], N'TABLE', [Person], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Cross-reference table mapping people to their credit card information in the CreditCard table. ', N'SCHEMA', [Sales], N'TABLE', [PersonCreditCard], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Business entity identification number. Foreign key to Person.BusinessEntityID.', N'SCHEMA', [Sales], N'TABLE', [PersonCreditCard], N'COLUMN', [BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Credit card identification number. Foreign key to CreditCard.CreditCardID.', N'SCHEMA', [Sales], N'TABLE', [PersonCreditCard], N'COLUMN', [CreditCardID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Sales], N'TABLE', [PersonCreditCard], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Telephone number and type of a person.', N'SCHEMA', [Person], N'TABLE', [PersonPhone], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Business entity identification number. Foreign key to Person.BusinessEntityID.', N'SCHEMA', [Person], N'TABLE', [PersonPhone], N'COLUMN', [BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Telephone number identification number.', N'SCHEMA', [Person], N'TABLE', [PersonPhone], N'COLUMN', [PhoneNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Kind of phone number. Foreign key to PhoneNumberType.PhoneNumberTypeID.', N'SCHEMA', [Person], N'TABLE', [PersonPhone], N'COLUMN', [PhoneNumberTypeID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Person], N'TABLE', [PersonPhone], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Type of phone number of a person.', N'SCHEMA', [Person], N'TABLE', [PhoneNumberType], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for telephone number type records.', N'SCHEMA', [Person], N'TABLE', [PhoneNumberType], N'COLUMN', [PhoneNumberTypeID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Name of the telephone number type', N'SCHEMA', [Person], N'TABLE', [PhoneNumberType], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Person], N'TABLE', [PhoneNumberType], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Products sold or used in the manfacturing of sold products.', N'SCHEMA', [Production], N'TABLE', [Product], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for Product records.', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Name of the product.', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique product identification number.', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [ProductNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'0 = Product is purchased, 1 = Product is manufactured in-house.', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [MakeFlag];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'0 = Product is not a salable item. 1 = Product is salable.', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [FinishedGoodsFlag];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product color.', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [Color];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Minimum inventory quantity. ', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [SafetyStockLevel];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Inventory level that triggers a purchase order or work order. ', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [ReorderPoint];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Standard cost of the product.', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [StandardCost];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Selling price.', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [ListPrice];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product size.', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [Size];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unit of measure for Size column.', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [SizeUnitMeasureCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unit of measure for Weight column.', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [WeightUnitMeasureCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product weight.', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [Weight];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Number of days required to manufacture the product.', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [DaysToManufacture];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'R = Road, M = Mountain, T = Touring, S = Standard', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [ProductLine];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'H = High, M = Medium, L = Low', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [Class];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'W = Womens, M = Mens, U = Universal', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [Style];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product is a member of this product subcategory. Foreign key to ProductSubCategory.ProductSubCategoryID. ', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [ProductSubcategoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product is a member of this product model. Foreign key to ProductModel.ProductModelID.', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [ProductModelID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date the product was available for sale.', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [SellStartDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date the product was no longer available for sale.', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [SellEndDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date the product was discontinued.', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [DiscontinuedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [Product], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'High-level product categorization.', N'SCHEMA', [Production], N'TABLE', [ProductCategory], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for ProductCategory records.', N'SCHEMA', [Production], N'TABLE', [ProductCategory], N'COLUMN', [ProductCategoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Category description.', N'SCHEMA', [Production], N'TABLE', [ProductCategory], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Production], N'TABLE', [ProductCategory], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [ProductCategory], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Changes in the cost of a product over time.', N'SCHEMA', [Production], N'TABLE', [ProductCostHistory], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product identification number. Foreign key to Product.ProductID', N'SCHEMA', [Production], N'TABLE', [ProductCostHistory], N'COLUMN', [ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product cost start date.', N'SCHEMA', [Production], N'TABLE', [ProductCostHistory], N'COLUMN', [StartDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product cost end date.', N'SCHEMA', [Production], N'TABLE', [ProductCostHistory], N'COLUMN', [EndDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Standard cost of the product.', N'SCHEMA', [Production], N'TABLE', [ProductCostHistory], N'COLUMN', [StandardCost];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [ProductCostHistory], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product descriptions in several languages.', N'SCHEMA', [Production], N'TABLE', [ProductDescription], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for ProductDescription records.', N'SCHEMA', [Production], N'TABLE', [ProductDescription], N'COLUMN', [ProductDescriptionID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Description of the product.', N'SCHEMA', [Production], N'TABLE', [ProductDescription], N'COLUMN', [Description];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Production], N'TABLE', [ProductDescription], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [ProductDescription], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Cross-reference table mapping products to related product documents.', N'SCHEMA', [Production], N'TABLE', [ProductDocument], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product identification number. Foreign key to Product.ProductID.', N'SCHEMA', [Production], N'TABLE', [ProductDocument], N'COLUMN', [ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Document identification number. Foreign key to Document.DocumentNode.', N'SCHEMA', [Production], N'TABLE', [ProductDocument], N'COLUMN', [DocumentNode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [ProductDocument], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product inventory information.', N'SCHEMA', [Production], N'TABLE', [ProductInventory], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product identification number. Foreign key to Product.ProductID.', N'SCHEMA', [Production], N'TABLE', [ProductInventory], N'COLUMN', [ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Inventory location identification number. Foreign key to Location.LocationID. ', N'SCHEMA', [Production], N'TABLE', [ProductInventory], N'COLUMN', [LocationID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Storage compartment within an inventory location.', N'SCHEMA', [Production], N'TABLE', [ProductInventory], N'COLUMN', [Shelf];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Storage container on a shelf in an inventory location.', N'SCHEMA', [Production], N'TABLE', [ProductInventory], N'COLUMN', [Bin];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Quantity of products in the inventory location.', N'SCHEMA', [Production], N'TABLE', [ProductInventory], N'COLUMN', [Quantity];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Production], N'TABLE', [ProductInventory], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [ProductInventory], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Changes in the list price of a product over time.', N'SCHEMA', [Production], N'TABLE', [ProductListPriceHistory], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product identification number. Foreign key to Product.ProductID', N'SCHEMA', [Production], N'TABLE', [ProductListPriceHistory], N'COLUMN', [ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'List price start date.', N'SCHEMA', [Production], N'TABLE', [ProductListPriceHistory], N'COLUMN', [StartDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'List price end date', N'SCHEMA', [Production], N'TABLE', [ProductListPriceHistory], N'COLUMN', [EndDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product list price.', N'SCHEMA', [Production], N'TABLE', [ProductListPriceHistory], N'COLUMN', [ListPrice];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [ProductListPriceHistory], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product model classification.', N'SCHEMA', [Production], N'TABLE', [ProductModel], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for ProductModel records.', N'SCHEMA', [Production], N'TABLE', [ProductModel], N'COLUMN', [ProductModelID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product model description.', N'SCHEMA', [Production], N'TABLE', [ProductModel], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Detailed product catalog information in xml format.', N'SCHEMA', [Production], N'TABLE', [ProductModel], N'COLUMN', [CatalogDescription];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Manufacturing instructions in xml format.', N'SCHEMA', [Production], N'TABLE', [ProductModel], N'COLUMN', [Instructions];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Production], N'TABLE', [ProductModel], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [ProductModel], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Cross-reference table mapping product models and illustrations.', N'SCHEMA', [Production], N'TABLE', [ProductModelIllustration], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Foreign key to ProductModel.ProductModelID.', N'SCHEMA', [Production], N'TABLE', [ProductModelIllustration], N'COLUMN', [ProductModelID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Foreign key to Illustration.IllustrationID.', N'SCHEMA', [Production], N'TABLE', [ProductModelIllustration], N'COLUMN', [IllustrationID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [ProductModelIllustration], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Cross-reference table mapping product descriptions and the language the description is written in.', N'SCHEMA', [Production], N'TABLE', [ProductModelProductDescriptionCulture], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Foreign key to ProductModel.ProductModelID.', N'SCHEMA', [Production], N'TABLE', [ProductModelProductDescriptionCulture], N'COLUMN', [ProductModelID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Foreign key to ProductDescription.ProductDescriptionID.', N'SCHEMA', [Production], N'TABLE', [ProductModelProductDescriptionCulture], N'COLUMN', [ProductDescriptionID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Culture identification number. Foreign key to Culture.CultureID.', N'SCHEMA', [Production], N'TABLE', [ProductModelProductDescriptionCulture], N'COLUMN', [CultureID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [ProductModelProductDescriptionCulture], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product images.', N'SCHEMA', [Production], N'TABLE', [ProductPhoto], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for ProductPhoto records.', N'SCHEMA', [Production], N'TABLE', [ProductPhoto], N'COLUMN', [ProductPhotoID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Small image of the product.', N'SCHEMA', [Production], N'TABLE', [ProductPhoto], N'COLUMN', [ThumbNailPhoto];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Small image file name.', N'SCHEMA', [Production], N'TABLE', [ProductPhoto], N'COLUMN', [ThumbnailPhotoFileName];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Large image of the product.', N'SCHEMA', [Production], N'TABLE', [ProductPhoto], N'COLUMN', [LargePhoto];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Large image file name.', N'SCHEMA', [Production], N'TABLE', [ProductPhoto], N'COLUMN', [LargePhotoFileName];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [ProductPhoto], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Cross-reference table mapping products and product photos.', N'SCHEMA', [Production], N'TABLE', [ProductProductPhoto], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product identification number. Foreign key to Product.ProductID.', N'SCHEMA', [Production], N'TABLE', [ProductProductPhoto], N'COLUMN', [ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product photo identification number. Foreign key to ProductPhoto.ProductPhotoID.', N'SCHEMA', [Production], N'TABLE', [ProductProductPhoto], N'COLUMN', [ProductPhotoID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'0 = Photo is not the principal image. 1 = Photo is the principal image.', N'SCHEMA', [Production], N'TABLE', [ProductProductPhoto], N'COLUMN', [Primary];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [ProductProductPhoto], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Customer reviews of products they have purchased.', N'SCHEMA', [Production], N'TABLE', [ProductReview], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for ProductReview records.', N'SCHEMA', [Production], N'TABLE', [ProductReview], N'COLUMN', [ProductReviewID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product identification number. Foreign key to Product.ProductID.', N'SCHEMA', [Production], N'TABLE', [ProductReview], N'COLUMN', [ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Name of the reviewer.', N'SCHEMA', [Production], N'TABLE', [ProductReview], N'COLUMN', [ReviewerName];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date review was submitted.', N'SCHEMA', [Production], N'TABLE', [ProductReview], N'COLUMN', [ReviewDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Reviewer''s e-mail address.', N'SCHEMA', [Production], N'TABLE', [ProductReview], N'COLUMN', [EmailAddress];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product rating given by the reviewer. Scale is 1 to 5 with 5 as the highest rating.', N'SCHEMA', [Production], N'TABLE', [ProductReview], N'COLUMN', [Rating];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Reviewer''s comments', N'SCHEMA', [Production], N'TABLE', [ProductReview], N'COLUMN', [Comments];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [ProductReview], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product subcategories. See ProductCategory table.', N'SCHEMA', [Production], N'TABLE', [ProductSubcategory], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for ProductSubcategory records.', N'SCHEMA', [Production], N'TABLE', [ProductSubcategory], N'COLUMN', [ProductSubcategoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product category identification number. Foreign key to ProductCategory.ProductCategoryID.', N'SCHEMA', [Production], N'TABLE', [ProductSubcategory], N'COLUMN', [ProductCategoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Subcategory description.', N'SCHEMA', [Production], N'TABLE', [ProductSubcategory], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Production], N'TABLE', [ProductSubcategory], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [ProductSubcategory], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Cross-reference table mapping vendors with the products they supply.', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Foreign key to Product.ProductID.', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'COLUMN', [ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Foreign key to Vendor.BusinessEntityID.', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'COLUMN', [BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The average span of time (in days) between placing an order with the vendor and receiving the purchased product.', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'COLUMN', [AverageLeadTime];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The vendor''s usual selling price.', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'COLUMN', [StandardPrice];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The selling price when last purchased.', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'COLUMN', [LastReceiptCost];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date the product was last received by the vendor.', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'COLUMN', [LastReceiptDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The maximum quantity that should be ordered.', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'COLUMN', [MinOrderQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The minimum quantity that should be ordered.', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'COLUMN', [MaxOrderQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The quantity currently on order.', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'COLUMN', [OnOrderQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'The product''s unit of measure.', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'COLUMN', [UnitMeasureCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Individual products associated with a specific purchase order. See PurchaseOrderHeader.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Foreign key to PurchaseOrderHeader.PurchaseOrderID.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'COLUMN', [PurchaseOrderID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. One line number per purchased product.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'COLUMN', [PurchaseOrderDetailID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date the product is expected to be received.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'COLUMN', [DueDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Quantity ordered.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'COLUMN', [OrderQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product identification number. Foreign key to Product.ProductID.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'COLUMN', [ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Vendor''s selling price of a single product.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'COLUMN', [UnitPrice];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Per product subtotal. Computed as OrderQty \* UnitPrice.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'COLUMN', [LineTotal];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Quantity actually received from the vendor.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'COLUMN', [ReceivedQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Quantity rejected during inspection.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'COLUMN', [RejectedQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Quantity accepted into inventory. Computed as ReceivedQty - RejectedQty.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'COLUMN', [StockedQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'General purchase order information. See PurchaseOrderDetail.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'COLUMN', [PurchaseOrderID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Incremental number to track changes to the purchase order over time.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'COLUMN', [RevisionNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Order current status. 1 = Pending; 2 = Approved; 3 = Rejected; 4 = Complete', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'COLUMN', [Status];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Employee who created the purchase order. Foreign key to Employee.BusinessEntityID.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'COLUMN', [EmployeeID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Vendor with whom the purchase order is placed. Foreign key to Vendor.BusinessEntityID.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'COLUMN', [VendorID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Shipping method. Foreign key to ShipMethod.ShipMethodID.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'COLUMN', [ShipMethodID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Purchase order creation date.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'COLUMN', [OrderDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Estimated shipment date from the vendor.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'COLUMN', [ShipDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Purchase order subtotal. Computed as SUM(PurchaseOrderDetail.LineTotal)for the appropriate PurchaseOrderID.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'COLUMN', [SubTotal];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Tax amount.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'COLUMN', [TaxAmt];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Shipping cost.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'COLUMN', [Freight];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Total due to vendor. Computed as Subtotal + TaxAmt + Freight.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'COLUMN', [TotalDue];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Individual products associated with a specific sales order. See SalesOrderHeader.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Foreign key to SalesOrderHeader.SalesOrderID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'COLUMN', [SalesOrderID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. One incremental unique number per product sold.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'COLUMN', [SalesOrderDetailID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Shipment tracking number supplied by the shipper.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'COLUMN', [CarrierTrackingNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Quantity ordered per product.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'COLUMN', [OrderQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product sold to customer. Foreign key to Product.ProductID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'COLUMN', [ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Promotional code. Foreign key to SpecialOffer.SpecialOfferID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'COLUMN', [SpecialOfferID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Selling price of a single product.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'COLUMN', [UnitPrice];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Discount amount.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'COLUMN', [UnitPriceDiscount];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Per product subtotal. Computed as UnitPrice \* (1 - UnitPriceDiscount) \* OrderQty.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'COLUMN', [LineTotal];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'General sales order information.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [SalesOrderID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Incremental number to track changes to the sales order over time.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [RevisionNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Dates the sales order was created.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [OrderDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date the order is due to the customer.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [DueDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date the order was shipped to the customer.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [ShipDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Order current status. 1 = In process; 2 = Approved; 3 = Backordered; 4 = Rejected; 5 = Shipped; 6 = Cancelled', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [Status];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'0 = Order placed by sales person. 1 = Order placed online by customer.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [OnlineOrderFlag];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique sales order identification number.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [SalesOrderNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Customer purchase order number reference. ', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [PurchaseOrderNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Financial accounting number reference.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [AccountNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Customer identification number. Foreign key to Customer.BusinessEntityID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [CustomerID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Sales person who created the sales order. Foreign key to SalesPerson.BusinessEntityID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [SalesPersonID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Territory in which the sale was made. Foreign key to SalesTerritory.SalesTerritoryID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [TerritoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Customer billing address. Foreign key to Address.AddressID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [BillToAddressID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Customer shipping address. Foreign key to Address.AddressID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [ShipToAddressID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Shipping method. Foreign key to ShipMethod.ShipMethodID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [ShipMethodID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Credit card identification number. Foreign key to CreditCard.CreditCardID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [CreditCardID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Approval code provided by the credit card company.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [CreditCardApprovalCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Currency exchange rate used. Foreign key to CurrencyRate.CurrencyRateID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [CurrencyRateID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Sales subtotal. Computed as SUM(SalesOrderDetail.LineTotal)for the appropriate SalesOrderID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [SubTotal];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Tax amount.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [TaxAmt];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Shipping cost.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [Freight];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Total due from customer. Computed as Subtotal + TaxAmt + Freight.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [TotalDue];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Sales representative comments.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [Comment];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Cross-reference table mapping sales orders to sales reason codes.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeaderSalesReason], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Foreign key to SalesOrderHeader.SalesOrderID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeaderSalesReason], N'COLUMN', [SalesOrderID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Foreign key to SalesReason.SalesReasonID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeaderSalesReason], N'COLUMN', [SalesReasonID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeaderSalesReason], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Sales representative current information.', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for SalesPerson records. Foreign key to Employee.BusinessEntityID', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'COLUMN', [BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Territory currently assigned to. Foreign key to SalesTerritory.SalesTerritoryID.', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'COLUMN', [TerritoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Projected yearly sales.', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'COLUMN', [SalesQuota];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Bonus due if quota is met.', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'COLUMN', [Bonus];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Commision percent received per sale.', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'COLUMN', [CommissionPct];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Sales total year to date.', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'COLUMN', [SalesYTD];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Sales total of previous year.', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'COLUMN', [SalesLastYear];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Sales performance tracking.', N'SCHEMA', [Sales], N'TABLE', [SalesPersonQuotaHistory], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Sales person identification number. Foreign key to SalesPerson.BusinessEntityID.', N'SCHEMA', [Sales], N'TABLE', [SalesPersonQuotaHistory], N'COLUMN', [BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Sales quota date.', N'SCHEMA', [Sales], N'TABLE', [SalesPersonQuotaHistory], N'COLUMN', [QuotaDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Sales quota amount.', N'SCHEMA', [Sales], N'TABLE', [SalesPersonQuotaHistory], N'COLUMN', [SalesQuota];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Sales], N'TABLE', [SalesPersonQuotaHistory], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Sales], N'TABLE', [SalesPersonQuotaHistory], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Lookup table of customer purchase reasons.', N'SCHEMA', [Sales], N'TABLE', [SalesReason], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for SalesReason records.', N'SCHEMA', [Sales], N'TABLE', [SalesReason], N'COLUMN', [SalesReasonID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Sales reason description.', N'SCHEMA', [Sales], N'TABLE', [SalesReason], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Category the sales reason belongs to.', N'SCHEMA', [Sales], N'TABLE', [SalesReason], N'COLUMN', [ReasonType];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Sales], N'TABLE', [SalesReason], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Tax rate lookup table.', N'SCHEMA', [Sales], N'TABLE', [SalesTaxRate], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for SalesTaxRate records.', N'SCHEMA', [Sales], N'TABLE', [SalesTaxRate], N'COLUMN', [SalesTaxRateID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'State, province, or country/region the sales tax applies to.', N'SCHEMA', [Sales], N'TABLE', [SalesTaxRate], N'COLUMN', [StateProvinceID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'1 = Tax applied to retail transactions, 2 = Tax applied to wholesale transactions, 3 = Tax applied to all sales (retail and wholesale) transactions.', N'SCHEMA', [Sales], N'TABLE', [SalesTaxRate], N'COLUMN', [TaxType];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Tax rate amount.', N'SCHEMA', [Sales], N'TABLE', [SalesTaxRate], N'COLUMN', [TaxRate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Tax rate description.', N'SCHEMA', [Sales], N'TABLE', [SalesTaxRate], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Sales], N'TABLE', [SalesTaxRate], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Sales], N'TABLE', [SalesTaxRate], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Sales territory lookup table.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for SalesTerritory records.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'COLUMN', [TerritoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Sales territory description', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ISO standard country or region code. Foreign key to CountryRegion.CountryRegionCode. ', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'COLUMN', [CountryRegionCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Geographic area to which the sales territory belong.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'COLUMN', [Group];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Sales in the territory year to date.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'COLUMN', [SalesYTD];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Sales in the territory the previous year.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'COLUMN', [SalesLastYear];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Business costs in the territory year to date.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'COLUMN', [CostYTD];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Business costs in the territory the previous year.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'COLUMN', [CostLastYear];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Sales representative transfers to other sales territories.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritoryHistory], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. The sales rep. Foreign key to SalesPerson.BusinessEntityID.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritoryHistory], N'COLUMN', [BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Territory identification number. Foreign key to SalesTerritory.SalesTerritoryID.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritoryHistory], N'COLUMN', [TerritoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Date the sales representive started work in the territory.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritoryHistory], N'COLUMN', [StartDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date the sales representative left work in the territory.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritoryHistory], N'COLUMN', [EndDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritoryHistory], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritoryHistory], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Manufacturing failure reasons lookup table.', N'SCHEMA', [Production], N'TABLE', [ScrapReason], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for ScrapReason records.', N'SCHEMA', [Production], N'TABLE', [ScrapReason], N'COLUMN', [ScrapReasonID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Failure description.', N'SCHEMA', [Production], N'TABLE', [ScrapReason], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [ScrapReason], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Work shift lookup table.', N'SCHEMA', [HumanResources], N'TABLE', [Shift], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for Shift records.', N'SCHEMA', [HumanResources], N'TABLE', [Shift], N'COLUMN', [ShiftID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Shift description.', N'SCHEMA', [HumanResources], N'TABLE', [Shift], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Shift start time.', N'SCHEMA', [HumanResources], N'TABLE', [Shift], N'COLUMN', [StartTime];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Shift end time.', N'SCHEMA', [HumanResources], N'TABLE', [Shift], N'COLUMN', [EndTime];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [HumanResources], N'TABLE', [Shift], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Shipping company lookup table.', N'SCHEMA', [Purchasing], N'TABLE', [ShipMethod], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for ShipMethod records.', N'SCHEMA', [Purchasing], N'TABLE', [ShipMethod], N'COLUMN', [ShipMethodID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Shipping company name.', N'SCHEMA', [Purchasing], N'TABLE', [ShipMethod], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Minimum shipping charge.', N'SCHEMA', [Purchasing], N'TABLE', [ShipMethod], N'COLUMN', [ShipBase];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Shipping charge per pound.', N'SCHEMA', [Purchasing], N'TABLE', [ShipMethod], N'COLUMN', [ShipRate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Purchasing], N'TABLE', [ShipMethod], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Purchasing], N'TABLE', [ShipMethod], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Contains online customer orders until the order is submitted or cancelled.', N'SCHEMA', [Sales], N'TABLE', [ShoppingCartItem], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for ShoppingCartItem records.', N'SCHEMA', [Sales], N'TABLE', [ShoppingCartItem], N'COLUMN', [ShoppingCartItemID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Shopping cart identification number.', N'SCHEMA', [Sales], N'TABLE', [ShoppingCartItem], N'COLUMN', [ShoppingCartID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product quantity ordered.', N'SCHEMA', [Sales], N'TABLE', [ShoppingCartItem], N'COLUMN', [Quantity];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product ordered. Foreign key to Product.ProductID.', N'SCHEMA', [Sales], N'TABLE', [ShoppingCartItem], N'COLUMN', [ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date the time the record was created.', N'SCHEMA', [Sales], N'TABLE', [ShoppingCartItem], N'COLUMN', [DateCreated];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Sales], N'TABLE', [ShoppingCartItem], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Sale discounts lookup table.', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for SpecialOffer records.', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'COLUMN', [SpecialOfferID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Discount description.', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'COLUMN', [Description];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Discount precentage.', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'COLUMN', [DiscountPct];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Discount type category.', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'COLUMN', [Type];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Group the discount applies to such as Reseller or Customer.', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'COLUMN', [Category];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Discount start date.', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'COLUMN', [StartDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Discount end date.', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'COLUMN', [EndDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Minimum discount percent allowed.', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'COLUMN', [MinQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Maximum discount percent allowed.', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'COLUMN', [MaxQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Cross-reference table mapping products to special offer discounts.', N'SCHEMA', [Sales], N'TABLE', [SpecialOfferProduct], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for SpecialOfferProduct records.', N'SCHEMA', [Sales], N'TABLE', [SpecialOfferProduct], N'COLUMN', [SpecialOfferID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product identification number. Foreign key to Product.ProductID.', N'SCHEMA', [Sales], N'TABLE', [SpecialOfferProduct], N'COLUMN', [ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Sales], N'TABLE', [SpecialOfferProduct], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Sales], N'TABLE', [SpecialOfferProduct], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'State and province lookup table.', N'SCHEMA', [Person], N'TABLE', [StateProvince], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for StateProvince records.', N'SCHEMA', [Person], N'TABLE', [StateProvince], N'COLUMN', [StateProvinceID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ISO standard state or province code.', N'SCHEMA', [Person], N'TABLE', [StateProvince], N'COLUMN', [StateProvinceCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ISO standard country or region code. Foreign key to CountryRegion.CountryRegionCode. ', N'SCHEMA', [Person], N'TABLE', [StateProvince], N'COLUMN', [CountryRegionCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'0 = StateProvinceCode exists. 1 = StateProvinceCode unavailable, using CountryRegionCode.', N'SCHEMA', [Person], N'TABLE', [StateProvince], N'COLUMN', [IsOnlyStateProvinceFlag];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'State or province description.', N'SCHEMA', [Person], N'TABLE', [StateProvince], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ID of the territory in which the state or province is located. Foreign key to SalesTerritory.SalesTerritoryID.', N'SCHEMA', [Person], N'TABLE', [StateProvince], N'COLUMN', [TerritoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Person], N'TABLE', [StateProvince], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Person], N'TABLE', [StateProvince], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Customers (resellers) of Adventure Works products.', N'SCHEMA', [Sales], N'TABLE', [Store], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Foreign key to Customer.BusinessEntityID.', N'SCHEMA', [Sales], N'TABLE', [Store], N'COLUMN', [BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Name of the store.', N'SCHEMA', [Sales], N'TABLE', [Store], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ID of the sales person assigned to the customer. Foreign key to SalesPerson.BusinessEntityID.', N'SCHEMA', [Sales], N'TABLE', [Store], N'COLUMN', [SalesPersonID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Demographic informationg about the store such as the number of employees, annual sales and store type.', N'SCHEMA', [Sales], N'TABLE', [Store], N'COLUMN', [Demographics];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'ROWGUIDCOL number uniquely identifying the record. Used to support a merge replication sample.', N'SCHEMA', [Sales], N'TABLE', [Store], N'COLUMN', [rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Sales], N'TABLE', [Store], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Record of each purchase order, sales order, or work order transaction year to date.', N'SCHEMA', [Production], N'TABLE', [TransactionHistory], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for TransactionHistory records.', N'SCHEMA', [Production], N'TABLE', [TransactionHistory], N'COLUMN', [TransactionID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product identification number. Foreign key to Product.ProductID.', N'SCHEMA', [Production], N'TABLE', [TransactionHistory], N'COLUMN', [ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Purchase order, sales order, or work order identification number.', N'SCHEMA', [Production], N'TABLE', [TransactionHistory], N'COLUMN', [ReferenceOrderID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Line number associated with the purchase order, sales order, or work order.', N'SCHEMA', [Production], N'TABLE', [TransactionHistory], N'COLUMN', [ReferenceOrderLineID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time of the transaction.', N'SCHEMA', [Production], N'TABLE', [TransactionHistory], N'COLUMN', [TransactionDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'W = WorkOrder, S = SalesOrder, P = PurchaseOrder', N'SCHEMA', [Production], N'TABLE', [TransactionHistory], N'COLUMN', [TransactionType];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product quantity.', N'SCHEMA', [Production], N'TABLE', [TransactionHistory], N'COLUMN', [Quantity];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product cost.', N'SCHEMA', [Production], N'TABLE', [TransactionHistory], N'COLUMN', [ActualCost];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [TransactionHistory], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Transactions for previous years.', N'SCHEMA', [Production], N'TABLE', [TransactionHistoryArchive], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for TransactionHistoryArchive records.', N'SCHEMA', [Production], N'TABLE', [TransactionHistoryArchive], N'COLUMN', [TransactionID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product identification number. Foreign key to Product.ProductID.', N'SCHEMA', [Production], N'TABLE', [TransactionHistoryArchive], N'COLUMN', [ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Purchase order, sales order, or work order identification number.', N'SCHEMA', [Production], N'TABLE', [TransactionHistoryArchive], N'COLUMN', [ReferenceOrderID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Line number associated with the purchase order, sales order, or work order.', N'SCHEMA', [Production], N'TABLE', [TransactionHistoryArchive], N'COLUMN', [ReferenceOrderLineID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time of the transaction.', N'SCHEMA', [Production], N'TABLE', [TransactionHistoryArchive], N'COLUMN', [TransactionDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'W = Work Order, S = Sales Order, P = Purchase Order', N'SCHEMA', [Production], N'TABLE', [TransactionHistoryArchive], N'COLUMN', [TransactionType];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product quantity.', N'SCHEMA', [Production], N'TABLE', [TransactionHistoryArchive], N'COLUMN', [Quantity];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product cost.', N'SCHEMA', [Production], N'TABLE', [TransactionHistoryArchive], N'COLUMN', [ActualCost];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [TransactionHistoryArchive], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unit of measure lookup table.', N'SCHEMA', [Production], N'TABLE', [UnitMeasure], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key.', N'SCHEMA', [Production], N'TABLE', [UnitMeasure], N'COLUMN', [UnitMeasureCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unit of measure description.', N'SCHEMA', [Production], N'TABLE', [UnitMeasure], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [UnitMeasure], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Companies from whom Adventure Works Cycles purchases parts or other goods.', N'SCHEMA', [Purchasing], N'TABLE', [Vendor], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for Vendor records. Foreign key to BusinessEntity.BusinessEntityID', N'SCHEMA', [Purchasing], N'TABLE', [Vendor], N'COLUMN', [BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Vendor account (identification) number.', N'SCHEMA', [Purchasing], N'TABLE', [Vendor], N'COLUMN', [AccountNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Company name.', N'SCHEMA', [Purchasing], N'TABLE', [Vendor], N'COLUMN', [Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'1 = Superior, 2 = Excellent, 3 = Above average, 4 = Average, 5 = Below average', N'SCHEMA', [Purchasing], N'TABLE', [Vendor], N'COLUMN', [CreditRating];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'0 = Do not use if another vendor is available. 1 = Preferred over other vendors supplying the same product.', N'SCHEMA', [Purchasing], N'TABLE', [Vendor], N'COLUMN', [PreferredVendorStatus];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'0 = Vendor no longer used. 1 = Vendor is actively used.', N'SCHEMA', [Purchasing], N'TABLE', [Vendor], N'COLUMN', [ActiveFlag];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Vendor URL.', N'SCHEMA', [Purchasing], N'TABLE', [Vendor], N'COLUMN', [PurchasingWebServiceURL];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Purchasing], N'TABLE', [Vendor], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Manufacturing work orders.', N'SCHEMA', [Production], N'TABLE', [WorkOrder], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key for WorkOrder records.', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'COLUMN', [WorkOrderID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product identification number. Foreign key to Product.ProductID.', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'COLUMN', [ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product quantity to build.', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'COLUMN', [OrderQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Quantity built and put in inventory.', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'COLUMN', [StockedQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Quantity that failed inspection.', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'COLUMN', [ScrappedQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Work order start date.', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'COLUMN', [StartDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Work order end date.', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'COLUMN', [EndDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Work order due date.', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'COLUMN', [DueDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Reason for inspection failure.', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'COLUMN', [ScrapReasonID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'COLUMN', [ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Work order details.', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Foreign key to WorkOrder.WorkOrderID.', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'COLUMN', [WorkOrderID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Foreign key to Product.ProductID.', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'COLUMN', [ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key. Indicates the manufacturing process sequence.', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'COLUMN', [OperationSequence];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Manufacturing location where the part is processed. Foreign key to Location.LocationID.', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'COLUMN', [LocationID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Planned manufacturing start date.', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'COLUMN', [ScheduledStartDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Planned manufacturing end date.', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'COLUMN', [ScheduledEndDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Actual start date.', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'COLUMN', [ActualStartDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Actual end date.', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'COLUMN', [ActualEndDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Number of manufacturing hours used.', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'COLUMN', [ActualResourceHrs];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Estimated manufacturing cost.', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'COLUMN', [PlannedCost];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Actual manufacturing cost.', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'COLUMN', [ActualCost];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Date and time the record was last updated.', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'COLUMN', [ModifiedDate];

GO

PRINT ' Triggers';

GO

-- Triggers

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'INSTEAD OF DELETE trigger which keeps Employees from being deleted.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'TRIGGER', [dEmployee];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'AFTER INSERT, UPDATE trigger inserting Individual only if the Customer does not exist in the Store table and setting the ModifiedDate column in the Person table to the current date.', N'SCHEMA', [Person], N'TABLE', [Person], N'TRIGGER', [iuPerson];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'AFTER INSERT trigger that inserts a row in the TransactionHistory table and updates the PurchaseOrderHeader.SubTotal column.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'TRIGGER', [iPurchaseOrderDetail];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'AFTER UPDATE trigger that inserts a row in the TransactionHistory table, updates ModifiedDate in PurchaseOrderDetail and updates the PurchaseOrderHeader.SubTotal column.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'TRIGGER', [uPurchaseOrderDetail];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'AFTER UPDATE trigger that updates the RevisionNumber and ModifiedDate columns in the PurchaseOrderHeader table.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'TRIGGER', [uPurchaseOrderHeader];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'AFTER INSERT, DELETE, UPDATE trigger that inserts a row in the TransactionHistory table, updates ModifiedDate in SalesOrderDetail and updates the SalesOrderHeader.SubTotal column.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'TRIGGER', [iduSalesOrderDetail];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'AFTER UPDATE trigger that updates the RevisionNumber and ModifiedDate columns in the SalesOrderHeader table.Updates the SalesYTD column in the SalesPerson and SalesTerritory tables.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'TRIGGER', [uSalesOrderHeader];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'INSTEAD OF DELETE trigger which keeps Vendors from being deleted.', N'SCHEMA', [Purchasing], N'TABLE', [Vendor], N'TRIGGER', [dVendor];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'AFTER INSERT trigger that inserts a row in the TransactionHistory table.', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'TRIGGER', [iWorkOrder];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'AFTER UPDATE trigger that inserts a row in the TransactionHistory table, updates ModifiedDate in the WorkOrder table.', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'TRIGGER', [uWorkOrder];

GO

PRINT ' Views';

GO

-- Views

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Displays the contact name and content from each element in the xml column AdditionalContactInfo for that person.', N'SCHEMA', [Person], N'VIEW', [vAdditionalContactInfo], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Employee names and addresses.', N'SCHEMA', [HumanResources], N'VIEW', [vEmployee], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Returns employee name, title, and current department.', N'SCHEMA', [HumanResources], N'VIEW', [vEmployeeDepartment], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Returns employee name and current and previous departments.', N'SCHEMA', [HumanResources], N'VIEW', [vEmployeeDepartmentHistory], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Individual customers (names and addresses) that purchase Adventure Works Cycles products online.', N'SCHEMA', [Sales], N'VIEW', [vIndividualCustomer], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Displays the content from each element in the xml column Demographics for each customer in the Person.Person table.', N'SCHEMA', [Sales], N'VIEW', [vPersonDemographics], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Job candidate names and resumes.', N'SCHEMA', [HumanResources], N'VIEW', [vJobCandidate], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Displays the content from each employement history related element in the xml column Resume in the HumanResources.JobCandidate table. The content has been localized into French, Simplified Chinese and Thai. Some data may not display correctly unless supplemental language support is installed.', N'SCHEMA', [HumanResources], N'VIEW', [vJobCandidateEmployment], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Displays the content from each education related element in the xml column Resume in the HumanResources.JobCandidate table. The content has been localized into French, Simplified Chinese and Thai. Some data may not display correctly unless supplemental language support is installed.', N'SCHEMA', [HumanResources], N'VIEW', [vJobCandidateEducation], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Product names and descriptions. Product descriptions are provided in multiple languages.', N'SCHEMA', [Production], N'VIEW', [vProductAndDescription], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Displays the content from each element in the xml column CatalogDescription for each product in the Production.ProductModel table that has catalog data.', N'SCHEMA', [Production], N'VIEW', [vProductModelCatalogDescription], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Displays the content from each element in the xml column Instructions for each product in the Production.ProductModel table that has manufacturing instructions.', N'SCHEMA', [Production], N'VIEW', [vProductModelInstructions], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Sales representiatives (names and addresses) and their sales-related information.', N'SCHEMA', [Sales], N'VIEW', [vSalesPerson], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Uses PIVOT to return aggregated sales information for each sales representative.', N'SCHEMA', [Sales], N'VIEW', [vSalesPersonSalesByFiscalYears], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Joins StateProvince table with CountryRegion table.', N'SCHEMA', [Person], N'VIEW', [vStateProvinceCountryRegion], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Stores (including demographics) that sell Adventure Works Cycles products to consumers.', N'SCHEMA', [Sales], N'VIEW', [vStoreWithDemographics], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Stores (including store contacts) that sell Adventure Works Cycles products to consumers.', N'SCHEMA', [Sales], N'VIEW', [vStoreWithContacts], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Stores (including store addresses) that sell Adventure Works Cycles products to consumers.', N'SCHEMA', [Sales], N'VIEW', [vStoreWithAddresses], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Vendor (company) names and the names of vendor employees to contact.', N'SCHEMA', [Purchasing], N'VIEW', [vVendorWithContacts], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Vendor (company) names and addresses .', N'SCHEMA', [Purchasing], N'VIEW', [vVendorWithAddresses], NULL, NULL;

GO

PRINT ' Indexes';

GO

-- Indexes

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Person], N'TABLE', [Address], N'INDEX', [AK\_Address\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Person], N'TABLE', [Address], N'INDEX', [IX\_Address\_AddressLine1\_AddressLine2\_City\_StateProvinceID\_PostalCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Person], N'TABLE', [Address], N'INDEX', [IX\_Address\_StateProvinceID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Person], N'TABLE', [Address], N'INDEX', [PK\_Address\_AddressID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Person], N'TABLE', [AddressType], N'INDEX', [AK\_AddressType\_Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Person], N'TABLE', [AddressType], N'INDEX', [AK\_AddressType\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Person], N'TABLE', [AddressType], N'INDEX', [PK\_AddressType\_AddressTypeID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [dbo], N'TABLE', [AWBuildVersion], N'INDEX', [PK\_AWBuildVersion\_SystemInformationID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index.', N'SCHEMA', [Production], N'TABLE', [BillOfMaterials], N'INDEX', [AK\_BillOfMaterials\_ProductAssemblyID\_ComponentID\_StartDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Production], N'TABLE', [BillOfMaterials], N'INDEX', [IX\_BillOfMaterials\_UnitMeasureCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [BillOfMaterials], N'INDEX', [PK\_BillOfMaterials\_BillOfMaterialsID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Person], N'TABLE', [BusinessEntity], N'INDEX', [AK\_BusinessEntity\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Person], N'TABLE', [BusinessEntity], N'INDEX', [PK\_BusinessEntity\_BusinessEntityID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityAddress], N'INDEX', [AK\_BusinessEntityAddress\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityAddress], N'INDEX', [IX\_BusinessEntityAddress\_AddressID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityAddress], N'INDEX', [IX\_BusinessEntityAddress\_AddressTypeID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityAddress], N'INDEX', [PK\_BusinessEntityAddress\_BusinessEntityID\_AddressID\_AddressTypeID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityContact], N'INDEX', [AK\_BusinessEntityContact\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityContact], N'INDEX', [IX\_BusinessEntityContact\_PersonID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityContact], N'INDEX', [IX\_BusinessEntityContact\_ContactTypeID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityContact], N'INDEX', [PK\_BusinessEntityContact\_BusinessEntityID\_PersonID\_ContactTypeID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Person], N'TABLE', [ContactType], N'INDEX', [AK\_ContactType\_Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Person], N'TABLE', [ContactType], N'INDEX', [PK\_ContactType\_ContactTypeID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Person], N'TABLE', [CountryRegion], N'INDEX', [AK\_CountryRegion\_Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Person], N'TABLE', [CountryRegion], N'INDEX', [PK\_CountryRegion\_CountryRegionCode];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Sales], N'TABLE', [CountryRegionCurrency], N'INDEX', [IX\_CountryRegionCurrency\_CurrencyCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Sales], N'TABLE', [CountryRegionCurrency], N'INDEX', [PK\_CountryRegionCurrency\_CountryRegionCode\_CurrencyCode];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Sales], N'TABLE', [CreditCard], N'INDEX', [AK\_CreditCard\_CardNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Sales], N'TABLE', [CreditCard], N'INDEX', [PK\_CreditCard\_CreditCardID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Production], N'TABLE', [Culture], N'INDEX', [AK\_Culture\_Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [Culture], N'INDEX', [PK\_Culture\_CultureID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Sales], N'TABLE', [Currency], N'INDEX', [AK\_Currency\_Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Sales], N'TABLE', [Currency], N'INDEX', [PK\_Currency\_CurrencyCode];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Sales], N'TABLE', [CurrencyRate], N'INDEX', [AK\_CurrencyRate\_CurrencyRateDate\_FromCurrencyCode\_ToCurrencyCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Sales], N'TABLE', [CurrencyRate], N'INDEX', [PK\_CurrencyRate\_CurrencyRateID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Sales], N'TABLE', [Customer], N'INDEX', [AK\_Customer\_AccountNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Sales], N'TABLE', [Customer], N'INDEX', [AK\_Customer\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Sales], N'TABLE', [Customer], N'INDEX', [IX\_Customer\_TerritoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Sales], N'TABLE', [Customer], N'INDEX', [PK\_Customer\_CustomerID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index created by a primary key constraint.', N'SCHEMA', [dbo], N'TABLE', [DatabaseLog], N'INDEX', [PK\_DatabaseLog\_DatabaseLogID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [HumanResources], N'TABLE', [Department], N'INDEX', [AK\_Department\_Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [HumanResources], N'TABLE', [Department], N'INDEX', [PK\_Department\_DepartmentID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Production], N'TABLE', [Document], N'INDEX', [AK\_Document\_DocumentLevel\_DocumentNode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Production], N'TABLE', [Document], N'INDEX', [IX\_Document\_FileName\_Revision];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [Document], N'INDEX', [PK\_Document\_DocumentNode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support FileStream.', N'SCHEMA', [Production], N'TABLE', [Document], N'INDEX', [AK\_Document\_rowguid];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Person], N'TABLE', [EmailAddress], N'INDEX', [IX\_EmailAddress\_EmailAddress];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Person], N'TABLE', [EmailAddress], N'INDEX', [PK\_EmailAddress\_BusinessEntityID\_EmailAddressID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'INDEX', [AK\_Employee\_LoginID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'INDEX', [AK\_Employee\_NationalIDNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'INDEX', [AK\_Employee\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'INDEX', [IX\_Employee\_OrganizationNode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'INDEX', [IX\_Employee\_OrganizationLevel\_OrganizationNode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'INDEX', [PK\_Employee\_BusinessEntityID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [HumanResources], N'TABLE', [EmployeeDepartmentHistory], N'INDEX', [IX\_EmployeeDepartmentHistory\_DepartmentID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [HumanResources], N'TABLE', [EmployeeDepartmentHistory], N'INDEX', [IX\_EmployeeDepartmentHistory\_ShiftID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [HumanResources], N'TABLE', [EmployeeDepartmentHistory], N'INDEX', [PK\_EmployeeDepartmentHistory\_BusinessEntityID\_StartDate\_DepartmentID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [HumanResources], N'TABLE', [EmployeePayHistory], N'INDEX', [PK\_EmployeePayHistory\_BusinessEntityID\_RateChangeDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [dbo], N'TABLE', [ErrorLog], N'INDEX', [PK\_ErrorLog\_ErrorLogID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [Illustration], N'INDEX', [PK\_Illustration\_IllustrationID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [HumanResources], N'TABLE', [JobCandidate], N'INDEX', [IX\_JobCandidate\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [HumanResources], N'TABLE', [JobCandidate], N'INDEX', [PK\_JobCandidate\_JobCandidateID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Production], N'TABLE', [Location], N'INDEX', [AK\_Location\_Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [Location], N'INDEX', [PK\_Location\_LocationID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Person], N'TABLE', [Password], N'INDEX', [PK\_Password\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Person], N'TABLE', [Person], N'INDEX', [AK\_Person\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Person], N'TABLE', [Person], N'INDEX', [PK\_Person\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary XML index.', N'SCHEMA', [Person], N'TABLE', [Person], N'INDEX', [PXML\_Person\_Demographics];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Secondary XML index for path.', N'SCHEMA', [Person], N'TABLE', [Person], N'INDEX', [XMLPATH\_Person\_Demographics];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Secondary XML index for property.', N'SCHEMA', [Person], N'TABLE', [Person], N'INDEX', [XMLPROPERTY\_Person\_Demographics];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Secondary XML index for value.', N'SCHEMA', [Person], N'TABLE', [Person], N'INDEX', [XMLVALUE\_Person\_Demographics];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary XML index.', N'SCHEMA', [Person], N'TABLE', [Person], N'INDEX', [PXML\_Person\_AddContact];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Sales], N'TABLE', [PersonCreditCard], N'INDEX', [PK\_PersonCreditCard\_BusinessEntityID\_CreditCardID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Person], N'TABLE', [PersonPhone], N'INDEX', [PK\_PersonPhone\_BusinessEntityID\_PhoneNumber\_PhoneNumberTypeID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Person], N'TABLE', [PersonPhone], N'INDEX', [IX\_PersonPhone\_PhoneNumber];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Person], N'TABLE', [PhoneNumberType], N'INDEX', [PK\_PhoneNumberType\_PhoneNumberTypeID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Production], N'TABLE', [Product], N'INDEX', [AK\_Product\_ProductNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [Product], N'INDEX', [PK\_Product\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Production], N'TABLE', [Product], N'INDEX', [AK\_Product\_Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Production], N'TABLE', [Product], N'INDEX', [AK\_Product\_rowguid];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Production], N'TABLE', [ProductCategory], N'INDEX', [AK\_ProductCategory\_Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Production], N'TABLE', [ProductCategory], N'INDEX', [AK\_ProductCategory\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [ProductCategory], N'INDEX', [PK\_ProductCategory\_ProductCategoryID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [ProductCostHistory], N'INDEX', [PK\_ProductCostHistory\_ProductID\_StartDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Production], N'TABLE', [ProductDescription], N'INDEX', [AK\_ProductDescription\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [ProductDescription], N'INDEX', [PK\_ProductDescription\_ProductDescriptionID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [ProductDocument], N'INDEX', [PK\_ProductDocument\_ProductID\_DocumentNode];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [ProductInventory], N'INDEX', [PK\_ProductInventory\_ProductID\_LocationID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [ProductListPriceHistory], N'INDEX', [PK\_ProductListPriceHistory\_ProductID\_StartDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Production], N'TABLE', [ProductModel], N'INDEX', [AK\_ProductModel\_Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Production], N'TABLE', [ProductModel], N'INDEX', [AK\_ProductModel\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [ProductModel], N'INDEX', [PK\_ProductModel\_ProductModelID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary XML index.', N'SCHEMA', [Production], N'TABLE', [ProductModel], N'INDEX', [PXML\_ProductModel\_CatalogDescription];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary XML index.', N'SCHEMA', [Production], N'TABLE', [ProductModel], N'INDEX', [PXML\_ProductModel\_Instructions];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [ProductModelIllustration], N'INDEX', [PK\_ProductModelIllustration\_ProductModelID\_IllustrationID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [ProductModelProductDescriptionCulture], N'INDEX', [PK\_ProductModelProductDescriptionCulture\_ProductModelID\_ProductDescriptionID\_CultureID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [ProductPhoto], N'INDEX', [PK\_ProductPhoto\_ProductPhotoID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [ProductProductPhoto], N'INDEX', [PK\_ProductProductPhoto\_ProductID\_ProductPhotoID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Production], N'TABLE', [ProductReview], N'INDEX', [IX\_ProductReview\_ProductID\_Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [ProductReview], N'INDEX', [PK\_ProductReview\_ProductReviewID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Production], N'TABLE', [ProductSubcategory], N'INDEX', [AK\_ProductSubcategory\_Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Production], N'TABLE', [ProductSubcategory], N'INDEX', [AK\_ProductSubcategory\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [ProductSubcategory], N'INDEX', [PK\_ProductSubcategory\_ProductSubcategoryID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'INDEX', [IX\_ProductVendor\_UnitMeasureCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'INDEX', [IX\_ProductVendor\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'INDEX', [PK\_ProductVendor\_ProductID\_BusinessEntityID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'INDEX', [IX\_PurchaseOrderDetail\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'INDEX', [PK\_PurchaseOrderDetail\_PurchaseOrderID\_PurchaseOrderDetailID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'INDEX', [IX\_PurchaseOrderHeader\_EmployeeID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'INDEX', [IX\_PurchaseOrderHeader\_VendorID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'INDEX', [PK\_PurchaseOrderHeader\_PurchaseOrderID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'INDEX', [AK\_SalesOrderDetail\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'INDEX', [IX\_SalesOrderDetail\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'INDEX', [PK\_SalesOrderDetail\_SalesOrderID\_SalesOrderDetailID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'INDEX', [AK\_SalesOrderHeader\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'INDEX', [AK\_SalesOrderHeader\_SalesOrderNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'INDEX', [IX\_SalesOrderHeader\_CustomerID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'INDEX', [IX\_SalesOrderHeader\_SalesPersonID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'INDEX', [PK\_SalesOrderHeader\_SalesOrderID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeaderSalesReason], N'INDEX', [PK\_SalesOrderHeaderSalesReason\_SalesOrderID\_SalesReasonID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'INDEX', [AK\_SalesPerson\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'INDEX', [PK\_SalesPerson\_BusinessEntityID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Sales], N'TABLE', [SalesPersonQuotaHistory], N'INDEX', [AK\_SalesPersonQuotaHistory\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Sales], N'TABLE', [SalesPersonQuotaHistory], N'INDEX', [PK\_SalesPersonQuotaHistory\_BusinessEntityID\_QuotaDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Sales], N'TABLE', [SalesReason], N'INDEX', [PK\_SalesReason\_SalesReasonID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Sales], N'TABLE', [SalesTaxRate], N'INDEX', [AK\_SalesTaxRate\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Sales], N'TABLE', [SalesTaxRate], N'INDEX', [AK\_SalesTaxRate\_StateProvinceID\_TaxType];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Sales], N'TABLE', [SalesTaxRate], N'INDEX', [PK\_SalesTaxRate\_SalesTaxRateID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'INDEX', [AK\_SalesTerritory\_Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'INDEX', [AK\_SalesTerritory\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'INDEX', [PK\_SalesTerritory\_TerritoryID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritoryHistory], N'INDEX', [AK\_SalesTerritoryHistory\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritoryHistory], N'INDEX', [PK\_SalesTerritoryHistory\_BusinessEntityID\_StartDate\_TerritoryID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Production], N'TABLE', [ScrapReason], N'INDEX', [AK\_ScrapReason\_Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [ScrapReason], N'INDEX', [PK\_ScrapReason\_ScrapReasonID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [HumanResources], N'TABLE', [Shift], N'INDEX', [AK\_Shift\_Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [HumanResources], N'TABLE', [Shift], N'INDEX', [AK\_Shift\_StartTime\_EndTime];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [HumanResources], N'TABLE', [Shift], N'INDEX', [PK\_Shift\_ShiftID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Purchasing], N'TABLE', [ShipMethod], N'INDEX', [AK\_ShipMethod\_Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Purchasing], N'TABLE', [ShipMethod], N'INDEX', [AK\_ShipMethod\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Purchasing], N'TABLE', [ShipMethod], N'INDEX', [PK\_ShipMethod\_ShipMethodID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Sales], N'TABLE', [ShoppingCartItem], N'INDEX', [IX\_ShoppingCartItem\_ShoppingCartID\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Sales], N'TABLE', [ShoppingCartItem], N'INDEX', [PK\_ShoppingCartItem\_ShoppingCartItemID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'INDEX', [AK\_SpecialOffer\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'INDEX', [PK\_SpecialOffer\_SpecialOfferID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Sales], N'TABLE', [SpecialOfferProduct], N'INDEX', [AK\_SpecialOfferProduct\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Sales], N'TABLE', [SpecialOfferProduct], N'INDEX', [IX\_SpecialOfferProduct\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Sales], N'TABLE', [SpecialOfferProduct], N'INDEX', [PK\_SpecialOfferProduct\_SpecialOfferID\_ProductID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Person], N'TABLE', [StateProvince], N'INDEX', [AK\_StateProvince\_Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Person], N'TABLE', [StateProvince], N'INDEX', [AK\_StateProvince\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Person], N'TABLE', [StateProvince], N'INDEX', [AK\_StateProvince\_StateProvinceCode\_CountryRegionCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Person], N'TABLE', [StateProvince], N'INDEX', [PK\_StateProvince\_StateProvinceID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index. Used to support replication samples.', N'SCHEMA', [Sales], N'TABLE', [Store], N'INDEX', [AK\_Store\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Sales], N'TABLE', [Store], N'INDEX', [PK\_Store\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Sales], N'TABLE', [Store], N'INDEX', [IX\_Store\_SalesPersonID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary XML index.', N'SCHEMA', [Sales], N'TABLE', [Store], N'INDEX', [PXML\_Store\_Demographics];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Production], N'TABLE', [TransactionHistory], N'INDEX', [IX\_TransactionHistory\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Production], N'TABLE', [TransactionHistory], N'INDEX', [IX\_TransactionHistory\_ReferenceOrderID\_ReferenceOrderLineID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [TransactionHistory], N'INDEX', [PK\_TransactionHistory\_TransactionID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Production], N'TABLE', [TransactionHistoryArchive], N'INDEX', [IX\_TransactionHistoryArchive\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Production], N'TABLE', [TransactionHistoryArchive], N'INDEX', [IX\_TransactionHistoryArchive\_ReferenceOrderID\_ReferenceOrderLineID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [TransactionHistoryArchive], N'INDEX', [PK\_TransactionHistoryArchive\_TransactionID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Production], N'TABLE', [UnitMeasure], N'INDEX', [AK\_UnitMeasure\_Name];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [UnitMeasure], N'INDEX', [PK\_UnitMeasure\_UnitMeasureCode];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Unique nonclustered index.', N'SCHEMA', [Purchasing], N'TABLE', [Vendor], N'INDEX', [AK\_Vendor\_AccountNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Purchasing], N'TABLE', [Vendor], N'INDEX', [PK\_Vendor\_BusinessEntityID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'INDEX', [IX\_WorkOrder\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'INDEX', [IX\_WorkOrder\_ScrapReasonID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'INDEX', [PK\_WorkOrder\_WorkOrderID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Nonclustered index.', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'INDEX', [IX\_WorkOrderRouting\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index created by a primary key constraint.', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'INDEX', [PK\_WorkOrderRouting\_WorkOrderID\_ProductID\_OperationSequence];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index on the view vProductAndDescription.', N'SCHEMA', [Production], N'VIEW', [vProductAndDescription], N'INDEX', [IX\_vProductAndDescription];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Clustered index on the view vStateProvinceCountryRegion.', N'SCHEMA', [Person], N'VIEW', [vStateProvinceCountryRegion], N'INDEX', [IX\_vStateProvinceCountryRegion];

GO

PRINT ' Constraints - PK, FK, DF, CK';

GO

-- Constraints - PK, FK, DF, CK

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Person], N'TABLE', [Address], N'CONSTRAINT', [PK\_Address\_AddressID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing StateProvince.StateProvinceID.', N'SCHEMA', [Person], N'TABLE', [Address], N'CONSTRAINT', [FK\_Address\_StateProvince\_StateProvinceID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Person], N'TABLE', [Address], N'CONSTRAINT', [DF\_Address\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Person], N'TABLE', [Address], N'CONSTRAINT', [DF\_Address\_rowguid];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Person], N'TABLE', [AddressType], N'CONSTRAINT', [PK\_AddressType\_AddressTypeID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Person], N'TABLE', [AddressType], N'CONSTRAINT', [DF\_AddressType\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Person], N'TABLE', [AddressType], N'CONSTRAINT', [DF\_AddressType\_rowguid];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [dbo], N'TABLE', [AWBuildVersion], N'CONSTRAINT', [PK\_AWBuildVersion\_SystemInformationID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [dbo], N'TABLE', [AWBuildVersion], N'CONSTRAINT', [DF\_AWBuildVersion\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [BillOfMaterials], N'CONSTRAINT', [PK\_BillOfMaterials\_BillOfMaterialsID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Product.ComponentID.', N'SCHEMA', [Production], N'TABLE', [BillOfMaterials], N'CONSTRAINT', [FK\_BillOfMaterials\_Product\_ComponentID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Product.ProductAssemblyID.', N'SCHEMA', [Production], N'TABLE', [BillOfMaterials], N'CONSTRAINT', [FK\_BillOfMaterials\_Product\_ProductAssemblyID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing UnitMeasure.UnitMeasureCode.', N'SCHEMA', [Production], N'TABLE', [BillOfMaterials], N'CONSTRAINT', [FK\_BillOfMaterials\_UnitMeasure\_UnitMeasureCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 1.0', N'SCHEMA', [Production], N'TABLE', [BillOfMaterials], N'CONSTRAINT', [DF\_BillOfMaterials\_PerAssemblyQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [BillOfMaterials], N'CONSTRAINT', [DF\_BillOfMaterials\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [BillOfMaterials], N'CONSTRAINT', [DF\_BillOfMaterials\_StartDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [PerAssemblyQty] >= (1.00)', N'SCHEMA', [Production], N'TABLE', [BillOfMaterials], N'CONSTRAINT', [CK\_BillOfMaterials\_PerAssemblyQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [ProductAssemblyID] <> [ComponentID]', N'SCHEMA', [Production], N'TABLE', [BillOfMaterials], N'CONSTRAINT', [CK\_BillOfMaterials\_ProductAssemblyID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint EndDate] > [StartDate] OR [EndDate] IS NULL', N'SCHEMA', [Production], N'TABLE', [BillOfMaterials], N'CONSTRAINT', [CK\_BillOfMaterials\_EndDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [ProductAssemblyID] IS NULL AND [BOMLevel] = (0) AND [PerAssemblyQty] = (1) OR [ProductAssemblyID] IS NOT NULL AND [BOMLevel] >= (1)', N'SCHEMA', [Production], N'TABLE', [BillOfMaterials], N'CONSTRAINT', [CK\_BillOfMaterials\_BOMLevel];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Person], N'TABLE', [BusinessEntity], N'CONSTRAINT', [PK\_BusinessEntity\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Person], N'TABLE', [BusinessEntity], N'CONSTRAINT', [DF\_BusinessEntity\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Person], N'TABLE', [BusinessEntity], N'CONSTRAINT', [DF\_BusinessEntity\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Person], N'TABLE', [BusinessEntityAddress], N'CONSTRAINT', [PK\_BusinessEntityAddress\_BusinessEntityID\_AddressID\_AddressTypeID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Address.AddressID.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityAddress], N'CONSTRAINT', [FK\_BusinessEntityAddress\_Address\_AddressID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing AddressType.AddressTypeID.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityAddress], N'CONSTRAINT', [FK\_BusinessEntityAddress\_AddressType\_AddressTypeID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing BusinessEntity.BusinessEntityID.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityAddress], N'CONSTRAINT', [FK\_BusinessEntityAddress\_BusinessEntity\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Person], N'TABLE', [BusinessEntityAddress], N'CONSTRAINT', [DF\_BusinessEntityAddress\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Person], N'TABLE', [BusinessEntityAddress], N'CONSTRAINT', [DF\_BusinessEntityAddress\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Person], N'TABLE', [BusinessEntityContact], N'CONSTRAINT', [PK\_BusinessEntityContact\_BusinessEntityID\_PersonID\_ContactTypeID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Person.BusinessEntityID.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityContact], N'CONSTRAINT', [FK\_BusinessEntityContact\_Person\_PersonID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing ContactType.ContactTypeID.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityContact], N'CONSTRAINT', [FK\_BusinessEntityContact\_ContactType\_ContactTypeID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing BusinessEntity.BusinessEntityID.', N'SCHEMA', [Person], N'TABLE', [BusinessEntityContact], N'CONSTRAINT', [FK\_BusinessEntityContact\_BusinessEntity\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Person], N'TABLE', [BusinessEntityContact], N'CONSTRAINT', [DF\_BusinessEntityContact\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Person], N'TABLE', [BusinessEntityContact], N'CONSTRAINT', [DF\_BusinessEntityContact\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Person], N'TABLE', [ContactType], N'CONSTRAINT', [PK\_ContactType\_ContactTypeID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Person], N'TABLE', [ContactType], N'CONSTRAINT', [DF\_ContactType\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Person], N'TABLE', [CountryRegion], N'CONSTRAINT', [PK\_CountryRegion\_CountryRegionCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Person], N'TABLE', [CountryRegion], N'CONSTRAINT', [DF\_CountryRegion\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Sales], N'TABLE', [CountryRegionCurrency], N'CONSTRAINT', [PK\_CountryRegionCurrency\_CountryRegionCode\_CurrencyCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing CountryRegion.CountryRegionCode.', N'SCHEMA', [Sales], N'TABLE', [CountryRegionCurrency], N'CONSTRAINT', [FK\_CountryRegionCurrency\_CountryRegion\_CountryRegionCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Currency.CurrencyCode.', N'SCHEMA', [Sales], N'TABLE', [CountryRegionCurrency], N'CONSTRAINT', [FK\_CountryRegionCurrency\_Currency\_CurrencyCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Sales], N'TABLE', [CountryRegionCurrency], N'CONSTRAINT', [DF\_CountryRegionCurrency\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Sales], N'TABLE', [CreditCard], N'CONSTRAINT', [PK\_CreditCard\_CreditCardID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Sales], N'TABLE', [CreditCard], N'CONSTRAINT', [DF\_CreditCard\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [Culture], N'CONSTRAINT', [PK\_Culture\_CultureID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [Culture], N'CONSTRAINT', [DF\_Culture\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Sales], N'TABLE', [Currency], N'CONSTRAINT', [PK\_Currency\_CurrencyCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Sales], N'TABLE', [Currency], N'CONSTRAINT', [DF\_Currency\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Sales], N'TABLE', [CurrencyRate], N'CONSTRAINT', [PK\_CurrencyRate\_CurrencyRateID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Currency.FromCurrencyCode.', N'SCHEMA', [Sales], N'TABLE', [CurrencyRate], N'CONSTRAINT', [FK\_CurrencyRate\_Currency\_FromCurrencyCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Currency.ToCurrencyCode.', N'SCHEMA', [Sales], N'TABLE', [CurrencyRate], N'CONSTRAINT', [FK\_CurrencyRate\_Currency\_ToCurrencyCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Sales], N'TABLE', [CurrencyRate], N'CONSTRAINT', [DF\_CurrencyRate\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Sales], N'TABLE', [Customer], N'CONSTRAINT', [PK\_Customer\_CustomerID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Person.BusinessEntityID.', N'SCHEMA', [Sales], N'TABLE', [Customer], N'CONSTRAINT', [FK\_Customer\_Person\_PersonID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Store.BusinessEntityID.', N'SCHEMA', [Sales], N'TABLE', [Customer], N'CONSTRAINT', [FK\_Customer\_Store\_StoreID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing SalesTerritory.TerritoryID.', N'SCHEMA', [Sales], N'TABLE', [Customer], N'CONSTRAINT', [FK\_Customer\_SalesTerritory\_TerritoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Sales], N'TABLE', [Customer], N'CONSTRAINT', [DF\_Customer\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Sales], N'TABLE', [Customer], N'CONSTRAINT', [DF\_Customer\_rowguid];

GO

EXEC [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (nonclustered) constraint', N'SCHEMA', [dbo], N'TABLE', [DatabaseLog], N'CONSTRAINT', [PK\_DatabaseLog\_DatabaseLogID];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [HumanResources], N'TABLE', [Department], N'CONSTRAINT', [PK\_Department\_DepartmentID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [HumanResources], N'TABLE', [Department], N'CONSTRAINT', [DF\_Department\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [Document], N'CONSTRAINT', [PK\_Document\_DocumentNode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Employee.BusinessEntityID.', N'SCHEMA', [Production], N'TABLE', [Document], N'CONSTRAINT', [FK\_Document\_Employee\_Owner];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0', N'SCHEMA', [Production], N'TABLE', [Document], N'CONSTRAINT', [DF\_Document\_ChangeNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [Document], N'CONSTRAINT', [DF\_Document\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [Status] BETWEEN (1) AND (3)', N'SCHEMA', [Production], N'TABLE', [Document], N'CONSTRAINT', [CK\_Document\_Status];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Production], N'TABLE', [Document], N'CONSTRAINT', [DF\_Document\_rowguid];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Person], N'TABLE', [EmailAddress], N'CONSTRAINT', [PK\_EmailAddress\_BusinessEntityID\_EmailAddressID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Person.BusinessEntityID.', N'SCHEMA', [Person], N'TABLE', [EmailAddress], N'CONSTRAINT', [FK\_EmailAddress\_Person\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Person], N'TABLE', [EmailAddress], N'CONSTRAINT', [DF\_EmailAddress\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Person], N'TABLE', [EmailAddress], N'CONSTRAINT', [DF\_EmailAddress\_rowguid];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'CONSTRAINT', [PK\_Employee\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Person.BusinessEntityID.', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'CONSTRAINT', [FK\_Employee\_Person\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'CONSTRAINT', [DF\_Employee\_SickLeaveHours];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'CONSTRAINT', [DF\_Employee\_VacationHours];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 1 (TRUE)', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'CONSTRAINT', [DF\_Employee\_SalariedFlag];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 1', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'CONSTRAINT', [DF\_Employee\_CurrentFlag];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'CONSTRAINT', [DF\_Employee\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'CONSTRAINT', [DF\_Employee\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [BirthDate] >= ''1930-01-01'' AND [BirthDate] <= dateadd(year,(-18),GETDATE())', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'CONSTRAINT', [CK\_Employee\_BirthDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [HireDate] >= ''1996-07-01'' AND [HireDate] <= dateadd(day,(1),GETDATE())', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'CONSTRAINT', [CK\_Employee\_HireDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [SickLeaveHours] >= (0) AND [SickLeaveHours] <= (120)', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'CONSTRAINT', [CK\_Employee\_SickLeaveHours];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [VacationHours] >= (-40) AND [VacationHours] <= (240)', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'CONSTRAINT', [CK\_Employee\_VacationHours];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [Gender]=''f'' OR [Gender]=''m'' OR [Gender]=''F'' OR [Gender]=''M''', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'CONSTRAINT', [CK\_Employee\_Gender];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [MaritalStatus]=''s'' OR [MaritalStatus]=''m'' OR [MaritalStatus]=''S'' OR [MaritalStatus]=''M''', N'SCHEMA', [HumanResources], N'TABLE', [Employee], N'CONSTRAINT', [CK\_Employee\_MaritalStatus];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [HumanResources], N'TABLE', [EmployeeDepartmentHistory], N'CONSTRAINT', [PK\_EmployeeDepartmentHistory\_BusinessEntityID\_StartDate\_DepartmentID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Shift.ShiftID', N'SCHEMA', [HumanResources], N'TABLE', [EmployeeDepartmentHistory], N'CONSTRAINT', [FK\_EmployeeDepartmentHistory\_Shift\_ShiftID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Department.DepartmentID.', N'SCHEMA', [HumanResources], N'TABLE', [EmployeeDepartmentHistory], N'CONSTRAINT', [FK\_EmployeeDepartmentHistory\_Department\_DepartmentID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Employee.EmployeeID.', N'SCHEMA', [HumanResources], N'TABLE', [EmployeeDepartmentHistory], N'CONSTRAINT', [FK\_EmployeeDepartmentHistory\_Employee\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [HumanResources], N'TABLE', [EmployeeDepartmentHistory], N'CONSTRAINT', [DF\_EmployeeDepartmentHistory\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [EndDate] >= [StartDate] OR [EndDate] IS NUL', N'SCHEMA', [HumanResources], N'TABLE', [EmployeeDepartmentHistory], N'CONSTRAINT', [CK\_EmployeeDepartmentHistory\_EndDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [HumanResources], N'TABLE', [EmployeePayHistory], N'CONSTRAINT', [PK\_EmployeePayHistory\_BusinessEntityID\_RateChangeDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Employee.EmployeeID.', N'SCHEMA', [HumanResources], N'TABLE', [EmployeePayHistory], N'CONSTRAINT', [FK\_EmployeePayHistory\_Employee\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [HumanResources], N'TABLE', [EmployeePayHistory], N'CONSTRAINT', [DF\_EmployeePayHistory\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [Rate] >= (6.50) AND [Rate] <= (200.00)', N'SCHEMA', [HumanResources], N'TABLE', [EmployeePayHistory], N'CONSTRAINT', [CK\_EmployeePayHistory\_Rate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [PayFrequency]=(3) OR [PayFrequency]=(2) OR [PayFrequency]=(1)', N'SCHEMA', [HumanResources], N'TABLE', [EmployeePayHistory], N'CONSTRAINT', [CK\_EmployeePayHistory\_PayFrequency];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [dbo], N'TABLE', [ErrorLog], N'CONSTRAINT', [PK\_ErrorLog\_ErrorLogID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [dbo], N'TABLE', [ErrorLog], N'CONSTRAINT', [DF\_ErrorLog\_ErrorTime];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [Illustration], N'CONSTRAINT', [PK\_Illustration\_IllustrationID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [Illustration], N'CONSTRAINT', [DF\_Illustration\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [HumanResources], N'TABLE', [JobCandidate], N'CONSTRAINT', [PK\_JobCandidate\_JobCandidateID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Employee.EmployeeID.', N'SCHEMA', [HumanResources], N'TABLE', [JobCandidate], N'CONSTRAINT', [FK\_JobCandidate\_Employee\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [HumanResources], N'TABLE', [JobCandidate], N'CONSTRAINT', [DF\_JobCandidate\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [Location], N'CONSTRAINT', [PK\_Location\_LocationID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.00', N'SCHEMA', [Production], N'TABLE', [Location], N'CONSTRAINT', [DF\_Location\_Availability];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.0', N'SCHEMA', [Production], N'TABLE', [Location], N'CONSTRAINT', [DF\_Location\_CostRate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [Location], N'CONSTRAINT', [DF\_Location\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [Availability] >= (0.00)', N'SCHEMA', [Production], N'TABLE', [Location], N'CONSTRAINT', [CK\_Location\_Availability];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [CostRate] >= (0.00)', N'SCHEMA', [Production], N'TABLE', [Location], N'CONSTRAINT', [CK\_Location\_CostRate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Person], N'TABLE', [Password], N'CONSTRAINT', [PK\_Password\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Person.BusinessEntityID.', N'SCHEMA', [Person], N'TABLE', [Password], N'CONSTRAINT', [FK\_Password\_Person\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Person], N'TABLE', [Password], N'CONSTRAINT', [DF\_Password\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Person], N'TABLE', [Password], N'CONSTRAINT', [DF\_Password\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Person], N'TABLE', [Person], N'CONSTRAINT', [PK\_Person\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing BusinessEntity.BusinessEntityID.', N'SCHEMA', [Person], N'TABLE', [Person], N'CONSTRAINT', [FK\_Person\_BusinessEntity\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0', N'SCHEMA', [Person], N'TABLE', [Person], N'CONSTRAINT', [DF\_Person\_EmailPromotion];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0', N'SCHEMA', [Person], N'TABLE', [Person], N'CONSTRAINT', [DF\_Person\_NameStyle];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Person], N'TABLE', [Person], N'CONSTRAINT', [DF\_Person\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Person], N'TABLE', [Person], N'CONSTRAINT', [DF\_Person\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [EmailPromotion] >= (0) AND [EmailPromotion] <= (2)', N'SCHEMA', [Person], N'TABLE', [Person], N'CONSTRAINT', [CK\_Person\_EmailPromotion];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [PersonType] is one of SC, VC, IN, EM or SP.', N'SCHEMA', [Person], N'TABLE', [Person], N'CONSTRAINT', [CK\_Person\_PersonType];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Sales], N'TABLE', [PersonCreditCard], N'CONSTRAINT', [PK\_PersonCreditCard\_BusinessEntityID\_CreditCardID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Person.BusinessEntityID.', N'SCHEMA', [Sales], N'TABLE', [PersonCreditCard], N'CONSTRAINT', [FK\_PersonCreditCard\_Person\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing CreditCard.CreditCardID.', N'SCHEMA', [Sales], N'TABLE', [PersonCreditCard], N'CONSTRAINT', [FK\_PersonCreditCard\_CreditCard\_CreditCardID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Sales], N'TABLE', [PersonCreditCard], N'CONSTRAINT', [DF\_PersonCreditCard\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Person], N'TABLE', [PersonPhone], N'CONSTRAINT', [PK\_PersonPhone\_BusinessEntityID\_PhoneNumber\_PhoneNumberTypeID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Person.BusinessEntityID.', N'SCHEMA', [Person], N'TABLE', [PersonPhone], N'CONSTRAINT', [FK\_PersonPhone\_Person\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing PhoneNumberType.PhoneNumberTypeID.', N'SCHEMA', [Person], N'TABLE', [PersonPhone], N'CONSTRAINT', [FK\_PersonPhone\_PhoneNumberType\_PhoneNumberTypeID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Person], N'TABLE', [PersonPhone], N'CONSTRAINT', [DF\_PersonPhone\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Person], N'TABLE', [PhoneNumberType], N'CONSTRAINT', [PK\_PhoneNumberType\_PhoneNumberTypeID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Person], N'TABLE', [PhoneNumberType], N'CONSTRAINT', [DF\_PhoneNumberType\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [Product], N'CONSTRAINT', [PK\_Product\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing ProductModel.ProductModelID.', N'SCHEMA', [Production], N'TABLE', [Product], N'CONSTRAINT', [FK\_Product\_ProductModel\_ProductModelID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing ProductSubcategory.ProductSubcategoryID.', N'SCHEMA', [Production], N'TABLE', [Product], N'CONSTRAINT', [FK\_Product\_ProductSubcategory\_ProductSubcategoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing UnitMeasure.UnitMeasureCode.', N'SCHEMA', [Production], N'TABLE', [Product], N'CONSTRAINT', [FK\_Product\_UnitMeasure\_SizeUnitMeasureCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing UnitMeasure.UnitMeasureCode.', N'SCHEMA', [Production], N'TABLE', [Product], N'CONSTRAINT', [FK\_Product\_UnitMeasure\_WeightUnitMeasureCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 1', N'SCHEMA', [Production], N'TABLE', [Product], N'CONSTRAINT', [DF\_Product\_FinishedGoodsFlag];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 1', N'SCHEMA', [Production], N'TABLE', [Product], N'CONSTRAINT', [DF\_Product\_MakeFlag];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [Product], N'CONSTRAINT', [DF\_Product\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Production], N'TABLE', [Product], N'CONSTRAINT', [DF\_Product\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [DaysToManufacture] >= (0)', N'SCHEMA', [Production], N'TABLE', [Product], N'CONSTRAINT', [CK\_Product\_DaysToManufacture];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [ListPrice] >= (0.00)', N'SCHEMA', [Production], N'TABLE', [Product], N'CONSTRAINT', [CK\_Product\_ListPrice];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [ReorderPoint] > (0)', N'SCHEMA', [Production], N'TABLE', [Product], N'CONSTRAINT', [CK\_Product\_ReorderPoint];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [SafetyStockLevel] > (0)', N'SCHEMA', [Production], N'TABLE', [Product], N'CONSTRAINT', [CK\_Product\_SafetyStockLevel];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [SafetyStockLevel] > (0)', N'SCHEMA', [Production], N'TABLE', [Product], N'CONSTRAINT', [CK\_Product\_StandardCost];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [Weight] > (0.00)', N'SCHEMA', [Production], N'TABLE', [Product], N'CONSTRAINT', [CK\_Product\_Weight];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [Class]=''h'' OR [Class]=''m'' OR [Class]=''l'' OR [Class]=''H'' OR [Class]=''M'' OR [Class]=''L'' OR [Class] IS NULL', N'SCHEMA', [Production], N'TABLE', [Product], N'CONSTRAINT', [CK\_Product\_Class];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [ProductLine]=''r'' OR [ProductLine]=''m'' OR [ProductLine]=''t'' OR [ProductLine]=''s'' OR [ProductLine]=''R'' OR [ProductLine]=''M'' OR [ProductLine]=''T'' OR [ProductLine]=''S'' OR [ProductLine] IS NULL', N'SCHEMA', [Production], N'TABLE', [Product], N'CONSTRAINT', [CK\_Product\_ProductLine];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [SellEndDate] >= [SellStartDate] OR [SellEndDate] IS NULL', N'SCHEMA', [Production], N'TABLE', [Product], N'CONSTRAINT', [CK\_Product\_SellEndDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [Style]=''u'' OR [Style]=''m'' OR [Style]=''w'' OR [Style]=''U'' OR [Style]=''M'' OR [Style]=''W'' OR [Style] IS NULL', N'SCHEMA', [Production], N'TABLE', [Product], N'CONSTRAINT', [CK\_Product\_Style];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [ProductCategory], N'CONSTRAINT', [PK\_ProductCategory\_ProductCategoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [ProductCategory], N'CONSTRAINT', [DF\_ProductCategory\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()()', N'SCHEMA', [Production], N'TABLE', [ProductCategory], N'CONSTRAINT', [DF\_ProductCategory\_rowguid];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [ProductCostHistory], N'CONSTRAINT', [PK\_ProductCostHistory\_ProductID\_StartDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Product.ProductID.', N'SCHEMA', [Production], N'TABLE', [ProductCostHistory], N'CONSTRAINT', [FK\_ProductCostHistory\_Product\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [ProductCostHistory], N'CONSTRAINT', [DF\_ProductCostHistory\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [StandardCost] >= (0.00)', N'SCHEMA', [Production], N'TABLE', [ProductCostHistory], N'CONSTRAINT', [CK\_ProductCostHistory\_StandardCost];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [EndDate] >= [StartDate] OR [EndDate] IS NULL', N'SCHEMA', [Production], N'TABLE', [ProductCostHistory], N'CONSTRAINT', [CK\_ProductCostHistory\_EndDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [ProductDescription], N'CONSTRAINT', [PK\_ProductDescription\_ProductDescriptionID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [ProductDescription], N'CONSTRAINT', [DF\_ProductDescription\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Production], N'TABLE', [ProductDescription], N'CONSTRAINT', [DF\_ProductDescription\_rowguid];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [ProductDocument], N'CONSTRAINT', [PK\_ProductDocument\_ProductID\_DocumentNode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Document.DocumentNode.', N'SCHEMA', [Production], N'TABLE', [ProductDocument], N'CONSTRAINT', [FK\_ProductDocument\_Document\_DocumentNode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Product.ProductID.', N'SCHEMA', [Production], N'TABLE', [ProductDocument], N'CONSTRAINT', [FK\_ProductDocument\_Product\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [ProductDocument], N'CONSTRAINT', [DF\_ProductDocument\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [ProductInventory], N'CONSTRAINT', [PK\_ProductInventory\_ProductID\_LocationID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Location.LocationID.', N'SCHEMA', [Production], N'TABLE', [ProductInventory], N'CONSTRAINT', [FK\_ProductInventory\_Location\_LocationID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Product.ProductID.', N'SCHEMA', [Production], N'TABLE', [ProductInventory], N'CONSTRAINT', [FK\_ProductInventory\_Product\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0', N'SCHEMA', [Production], N'TABLE', [ProductInventory], N'CONSTRAINT', [DF\_ProductInventory\_Quantity];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [ProductInventory], N'CONSTRAINT', [DF\_ProductInventory\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Production], N'TABLE', [ProductInventory], N'CONSTRAINT', [DF\_ProductInventory\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [Bin] BETWEEN (0) AND (100)', N'SCHEMA', [Production], N'TABLE', [ProductInventory], N'CONSTRAINT', [CK\_ProductInventory\_Bin];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [Shelf] like ''[A-Za-z]'' OR [Shelf]=''N/A''', N'SCHEMA', [Production], N'TABLE', [ProductInventory], N'CONSTRAINT', [CK\_ProductInventory\_Shelf];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [ProductListPriceHistory], N'CONSTRAINT', [PK\_ProductListPriceHistory\_ProductID\_StartDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Product.ProductID.', N'SCHEMA', [Production], N'TABLE', [ProductListPriceHistory], N'CONSTRAINT', [FK\_ProductListPriceHistory\_Product\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [ProductListPriceHistory], N'CONSTRAINT', [DF\_ProductListPriceHistory\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [ListPrice] > (0.00)', N'SCHEMA', [Production], N'TABLE', [ProductListPriceHistory], N'CONSTRAINT', [CK\_ProductListPriceHistory\_ListPrice];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [EndDate] >= [StartDate] OR [EndDate] IS NULL', N'SCHEMA', [Production], N'TABLE', [ProductListPriceHistory], N'CONSTRAINT', [CK\_ProductListPriceHistory\_EndDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [ProductModel], N'CONSTRAINT', [PK\_ProductModel\_ProductModelID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [ProductModel], N'CONSTRAINT', [DF\_ProductModel\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Production], N'TABLE', [ProductModel], N'CONSTRAINT', [DF\_ProductModel\_rowguid];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [ProductModelIllustration], N'CONSTRAINT', [PK\_ProductModelIllustration\_ProductModelID\_IllustrationID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Illustration.IllustrationID.', N'SCHEMA', [Production], N'TABLE', [ProductModelIllustration], N'CONSTRAINT', [FK\_ProductModelIllustration\_Illustration\_IllustrationID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing ProductModel.ProductModelID.', N'SCHEMA', [Production], N'TABLE', [ProductModelIllustration], N'CONSTRAINT', [FK\_ProductModelIllustration\_ProductModel\_ProductModelID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [ProductModelIllustration], N'CONSTRAINT', [DF\_ProductModelIllustration\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [ProductModelProductDescriptionCulture], N'CONSTRAINT', [PK\_ProductModelProductDescriptionCulture\_ProductModelID\_ProductDescriptionID\_CultureID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Culture.CultureID.', N'SCHEMA', [Production], N'TABLE', [ProductModelProductDescriptionCulture], N'CONSTRAINT', [FK\_ProductModelProductDescriptionCulture\_Culture\_CultureID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing ProductDescription.ProductDescriptionID.', N'SCHEMA', [Production], N'TABLE', [ProductModelProductDescriptionCulture], N'CONSTRAINT', [FK\_ProductModelProductDescriptionCulture\_ProductDescription\_ProductDescriptionID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing ProductModel.ProductModelID.', N'SCHEMA', [Production], N'TABLE', [ProductModelProductDescriptionCulture], N'CONSTRAINT', [FK\_ProductModelProductDescriptionCulture\_ProductModel\_ProductModelID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [ProductModelProductDescriptionCulture], N'CONSTRAINT', [DF\_ProductModelProductDescriptionCulture\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [ProductPhoto], N'CONSTRAINT', [PK\_ProductPhoto\_ProductPhotoID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [ProductPhoto], N'CONSTRAINT', [DF\_ProductPhoto\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [ProductProductPhoto], N'CONSTRAINT', [PK\_ProductProductPhoto\_ProductID\_ProductPhotoID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Product.ProductID.', N'SCHEMA', [Production], N'TABLE', [ProductProductPhoto], N'CONSTRAINT', [FK\_ProductProductPhoto\_Product\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing ProductPhoto.ProductPhotoID.', N'SCHEMA', [Production], N'TABLE', [ProductProductPhoto], N'CONSTRAINT', [FK\_ProductProductPhoto\_ProductPhoto\_ProductPhotoID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0 (FALSE)', N'SCHEMA', [Production], N'TABLE', [ProductProductPhoto], N'CONSTRAINT', [DF\_ProductProductPhoto\_Primary];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [ProductProductPhoto], N'CONSTRAINT', [DF\_ProductProductPhoto\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [ProductReview], N'CONSTRAINT', [PK\_ProductReview\_ProductReviewID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Product.ProductID.', N'SCHEMA', [Production], N'TABLE', [ProductReview], N'CONSTRAINT', [FK\_ProductReview\_Product\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [ProductReview], N'CONSTRAINT', [DF\_ProductReview\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [ProductReview], N'CONSTRAINT', [DF\_ProductReview\_ReviewDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [Rating] BETWEEN (1) AND (5)', N'SCHEMA', [Production], N'TABLE', [ProductReview], N'CONSTRAINT', [CK\_ProductReview\_Rating];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [ProductSubcategory], N'CONSTRAINT', [PK\_ProductSubcategory\_ProductSubcategoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing ProductCategory.ProductCategoryID.', N'SCHEMA', [Production], N'TABLE', [ProductSubcategory], N'CONSTRAINT', [FK\_ProductSubcategory\_ProductCategory\_ProductCategoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [ProductSubcategory], N'CONSTRAINT', [DF\_ProductSubcategory\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Production], N'TABLE', [ProductSubcategory], N'CONSTRAINT', [DF\_ProductSubcategory\_rowguid];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'CONSTRAINT', [PK\_ProductVendor\_ProductID\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Product.ProductID.', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'CONSTRAINT', [FK\_ProductVendor\_Product\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing UnitMeasure.UnitMeasureCode.', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'CONSTRAINT', [FK\_ProductVendor\_UnitMeasure\_UnitMeasureCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Vendor.BusinessEntityID.', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'CONSTRAINT', [FK\_ProductVendor\_Vendor\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'CONSTRAINT', [DF\_ProductVendor\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [AverageLeadTime] >= (1)', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'CONSTRAINT', [CK\_ProductVendor\_AverageLeadTime];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [LastReceiptCost] > (0.00)', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'CONSTRAINT', [CK\_ProductVendor\_LastReceiptCost];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [MaxOrderQty] >= (1)', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'CONSTRAINT', [CK\_ProductVendor\_MaxOrderQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [MinOrderQty] >= (1)', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'CONSTRAINT', [CK\_ProductVendor\_MinOrderQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [OnOrderQty] >= (0)', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'CONSTRAINT', [CK\_ProductVendor\_OnOrderQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [StandardPrice] > (0.00)', N'SCHEMA', [Purchasing], N'TABLE', [ProductVendor], N'CONSTRAINT', [CK\_ProductVendor\_StandardPrice];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'CONSTRAINT', [PK\_PurchaseOrderDetail\_PurchaseOrderID\_PurchaseOrderDetailID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Product.ProductID.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'CONSTRAINT', [FK\_PurchaseOrderDetail\_Product\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing PurchaseOrderHeader.PurchaseOrderID.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'CONSTRAINT', [FK\_PurchaseOrderDetail\_PurchaseOrderHeader\_PurchaseOrderID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'CONSTRAINT', [DF\_PurchaseOrderDetail\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [OrderQty] > (0)', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'CONSTRAINT', [CK\_PurchaseOrderDetail\_OrderQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [ReceivedQty] >= (0.00)', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'CONSTRAINT', [CK\_PurchaseOrderDetail\_ReceivedQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [RejectedQty] >= (0.00)', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'CONSTRAINT', [CK\_PurchaseOrderDetail\_RejectedQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [UnitPrice] >= (0.00)', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderDetail], N'CONSTRAINT', [CK\_PurchaseOrderDetail\_UnitPrice];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'CONSTRAINT', [PK\_PurchaseOrderHeader\_PurchaseOrderID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Employee.EmployeeID.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'CONSTRAINT', [FK\_PurchaseOrderHeader\_Employee\_EmployeeID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing ShipMethod.ShipMethodID.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'CONSTRAINT', [FK\_PurchaseOrderHeader\_ShipMethod\_ShipMethodID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Vendor.VendorID.', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'CONSTRAINT', [FK\_PurchaseOrderHeader\_Vendor\_VendorID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'CONSTRAINT', [DF\_PurchaseOrderHeader\_RevisionNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.0', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'CONSTRAINT', [DF\_PurchaseOrderHeader\_Freight];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.0', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'CONSTRAINT', [DF\_PurchaseOrderHeader\_SubTotal];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.0', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'CONSTRAINT', [DF\_PurchaseOrderHeader\_TaxAmt];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 1', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'CONSTRAINT', [DF\_PurchaseOrderHeader\_Status];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'CONSTRAINT', [DF\_PurchaseOrderHeader\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'CONSTRAINT', [DF\_PurchaseOrderHeader\_OrderDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [Freight] >= (0.00)', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'CONSTRAINT', [CK\_PurchaseOrderHeader\_Freight];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [SubTotal] >= (0.00)', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'CONSTRAINT', [CK\_PurchaseOrderHeader\_SubTotal];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [TaxAmt] >= (0.00)', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'CONSTRAINT', [CK\_PurchaseOrderHeader\_TaxAmt];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [ShipDate] >= [OrderDate] OR [ShipDate] IS NULL', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'CONSTRAINT', [CK\_PurchaseOrderHeader\_ShipDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [Status] BETWEEN (1) AND (4)', N'SCHEMA', [Purchasing], N'TABLE', [PurchaseOrderHeader], N'CONSTRAINT', [CK\_PurchaseOrderHeader\_Status];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'CONSTRAINT', [PK\_SalesOrderDetail\_SalesOrderID\_SalesOrderDetailID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing SalesOrderHeader.PurchaseOrderID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'CONSTRAINT', [FK\_SalesOrderDetail\_SalesOrderHeader\_SalesOrderID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing SpecialOfferProduct.SpecialOfferIDProductID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'CONSTRAINT', [FK\_SalesOrderDetail\_SpecialOfferProduct\_SpecialOfferIDProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.0', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'CONSTRAINT', [DF\_SalesOrderDetail\_UnitPriceDiscount];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'CONSTRAINT', [DF\_SalesOrderDetail\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'CONSTRAINT', [DF\_SalesOrderDetail\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [OrderQty] > (0)', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'CONSTRAINT', [CK\_SalesOrderDetail\_OrderQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [UnitPrice] >= (0.00)', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'CONSTRAINT', [CK\_SalesOrderDetail\_UnitPrice];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [UnitPriceDiscount] >= (0.00)', N'SCHEMA', [Sales], N'TABLE', [SalesOrderDetail], N'CONSTRAINT', [CK\_SalesOrderDetail\_UnitPriceDiscount];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [PK\_SalesOrderHeader\_SalesOrderID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Address.AddressID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [FK\_SalesOrderHeader\_Address\_BillToAddressID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Address.AddressID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [FK\_SalesOrderHeader\_Address\_ShipToAddressID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing CreditCard.CreditCardID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [FK\_SalesOrderHeader\_CreditCard\_CreditCardID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing CurrencyRate.CurrencyRateID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [FK\_SalesOrderHeader\_CurrencyRate\_CurrencyRateID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Customer.CustomerID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [FK\_SalesOrderHeader\_Customer\_CustomerID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing SalesPerson.SalesPersonID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [FK\_SalesOrderHeader\_SalesPerson\_SalesPersonID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing SalesTerritory.TerritoryID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [FK\_SalesOrderHeader\_SalesTerritory\_TerritoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing ShipMethod.ShipMethodID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [FK\_SalesOrderHeader\_ShipMethod\_ShipMethodID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [DF\_SalesOrderHeader\_RevisionNumber];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.0', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [DF\_SalesOrderHeader\_Freight];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.0', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [DF\_SalesOrderHeader\_SubTotal];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.0', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [DF\_SalesOrderHeader\_TaxAmt];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 1 (TRUE)', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [DF\_SalesOrderHeader\_OnlineOrderFlag];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 1', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [DF\_SalesOrderHeader\_Status];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [DF\_SalesOrderHeader\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [DF\_SalesOrderHeader\_OrderDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [DF\_SalesOrderHeader\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [Freight] >= (0.00)', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [CK\_SalesOrderHeader\_Freight];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [SubTotal] >= (0.00)', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [CK\_SalesOrderHeader\_SubTotal];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [TaxAmt] >= (0.00)', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [CK\_SalesOrderHeader\_TaxAmt];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [DueDate] >= [OrderDate]', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [CK\_SalesOrderHeader\_DueDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [ShipDate] >= [OrderDate] OR [ShipDate] IS NULL', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [CK\_SalesOrderHeader\_ShipDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [Status] BETWEEN (0) AND (8)', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeader], N'CONSTRAINT', [CK\_SalesOrderHeader\_Status];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeaderSalesReason], N'CONSTRAINT', [PK\_SalesOrderHeaderSalesReason\_SalesOrderID\_SalesReasonID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing SalesOrderHeader.SalesOrderID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeaderSalesReason], N'CONSTRAINT', [FK\_SalesOrderHeaderSalesReason\_SalesOrderHeader\_SalesOrderID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing SalesReason.SalesReasonID.', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeaderSalesReason], N'CONSTRAINT', [FK\_SalesOrderHeaderSalesReason\_SalesReason\_SalesReasonID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Sales], N'TABLE', [SalesOrderHeaderSalesReason], N'CONSTRAINT', [DF\_SalesOrderHeaderSalesReason\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'CONSTRAINT', [PK\_SalesPerson\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Employee.EmployeeID.', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'CONSTRAINT', [FK\_SalesPerson\_Employee\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing SalesTerritory.TerritoryID.', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'CONSTRAINT', [FK\_SalesPerson\_SalesTerritory\_TerritoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.0', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'CONSTRAINT', [DF\_SalesPerson\_Bonus];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.0', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'CONSTRAINT', [DF\_SalesPerson\_CommissionPct];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.0', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'CONSTRAINT', [DF\_SalesPerson\_SalesLastYear];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.0', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'CONSTRAINT', [DF\_SalesPerson\_SalesYTD];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'CONSTRAINT', [DF\_SalesPerson\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'CONSTRAINT', [DF\_SalesPerson\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [Bonus] >= (0.00)', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'CONSTRAINT', [CK\_SalesPerson\_Bonus];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [CommissionPct] >= (0.00)', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'CONSTRAINT', [CK\_SalesPerson\_CommissionPct];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [SalesLastYear] >= (0.00)', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'CONSTRAINT', [CK\_SalesPerson\_SalesLastYear];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [SalesQuota] > (0.00)', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'CONSTRAINT', [CK\_SalesPerson\_SalesQuota];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [SalesYTD] >= (0.00)', N'SCHEMA', [Sales], N'TABLE', [SalesPerson], N'CONSTRAINT', [CK\_SalesPerson\_SalesYTD];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Sales], N'TABLE', [SalesPersonQuotaHistory], N'CONSTRAINT', [PK\_SalesPersonQuotaHistory\_BusinessEntityID\_QuotaDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing SalesPerson.SalesPersonID.', N'SCHEMA', [Sales], N'TABLE', [SalesPersonQuotaHistory], N'CONSTRAINT', [FK\_SalesPersonQuotaHistory\_SalesPerson\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Sales], N'TABLE', [SalesPersonQuotaHistory], N'CONSTRAINT', [DF\_SalesPersonQuotaHistory\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Sales], N'TABLE', [SalesPersonQuotaHistory], N'CONSTRAINT', [DF\_SalesPersonQuotaHistory\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [SalesQuota] > (0.00)', N'SCHEMA', [Sales], N'TABLE', [SalesPersonQuotaHistory], N'CONSTRAINT', [CK\_SalesPersonQuotaHistory\_SalesQuota];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Sales], N'TABLE', [SalesReason], N'CONSTRAINT', [PK\_SalesReason\_SalesReasonID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Sales], N'TABLE', [SalesReason], N'CONSTRAINT', [DF\_SalesReason\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Sales], N'TABLE', [SalesTaxRate], N'CONSTRAINT', [PK\_SalesTaxRate\_SalesTaxRateID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing StateProvince.StateProvinceID.', N'SCHEMA', [Sales], N'TABLE', [SalesTaxRate], N'CONSTRAINT', [FK\_SalesTaxRate\_StateProvince\_StateProvinceID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.0', N'SCHEMA', [Sales], N'TABLE', [SalesTaxRate], N'CONSTRAINT', [DF\_SalesTaxRate\_TaxRate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Sales], N'TABLE', [SalesTaxRate], N'CONSTRAINT', [DF\_SalesTaxRate\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Sales], N'TABLE', [SalesTaxRate], N'CONSTRAINT', [DF\_SalesTaxRate\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [TaxType] BETWEEN (1) AND (3)', N'SCHEMA', [Sales], N'TABLE', [SalesTaxRate], N'CONSTRAINT', [CK\_SalesTaxRate\_TaxType];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'CONSTRAINT', [PK\_SalesTerritory\_TerritoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.0', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'CONSTRAINT', [DF\_SalesTerritory\_CostLastYear];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.0', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'CONSTRAINT', [DF\_SalesTerritory\_CostYTD];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.0', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'CONSTRAINT', [DF\_SalesTerritory\_SalesLastYear];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.0', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'CONSTRAINT', [DF\_SalesTerritory\_SalesYTD];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'CONSTRAINT', [DF\_SalesTerritory\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'CONSTRAINT', [DF\_SalesTerritory\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [CostLastYear] >= (0.00)', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'CONSTRAINT', [CK\_SalesTerritory\_CostLastYear];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [CostYTD] >= (0.00)', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'CONSTRAINT', [CK\_SalesTerritory\_CostYTD];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [SalesLastYear] >= (0.00)', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'CONSTRAINT', [CK\_SalesTerritory\_SalesLastYear];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [SalesYTD] >= (0.00)', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'CONSTRAINT', [CK\_SalesTerritory\_SalesYTD];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing CountryRegion.CountryRegionCode.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritory], N'CONSTRAINT', [FK\_SalesTerritory\_CountryRegion\_CountryRegionCode];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Sales], N'TABLE', [SalesTerritoryHistory], N'CONSTRAINT', [PK\_SalesTerritoryHistory\_BusinessEntityID\_StartDate\_TerritoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing SalesPerson.SalesPersonID.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritoryHistory], N'CONSTRAINT', [FK\_SalesTerritoryHistory\_SalesPerson\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing SalesTerritory.TerritoryID.', N'SCHEMA', [Sales], N'TABLE', [SalesTerritoryHistory], N'CONSTRAINT', [FK\_SalesTerritoryHistory\_SalesTerritory\_TerritoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Sales], N'TABLE', [SalesTerritoryHistory], N'CONSTRAINT', [DF\_SalesTerritoryHistory\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Sales], N'TABLE', [SalesTerritoryHistory], N'CONSTRAINT', [DF\_SalesTerritoryHistory\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [EndDate] >= [StartDate] OR [EndDate] IS NULL', N'SCHEMA', [Sales], N'TABLE', [SalesTerritoryHistory], N'CONSTRAINT', [CK\_SalesTerritoryHistory\_EndDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [ScrapReason], N'CONSTRAINT', [PK\_ScrapReason\_ScrapReasonID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [ScrapReason], N'CONSTRAINT', [DF\_ScrapReason\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [HumanResources], N'TABLE', [Shift], N'CONSTRAINT', [PK\_Shift\_ShiftID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [HumanResources], N'TABLE', [Shift], N'CONSTRAINT', [DF\_Shift\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Purchasing], N'TABLE', [ShipMethod], N'CONSTRAINT', [PK\_ShipMethod\_ShipMethodID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.0', N'SCHEMA', [Purchasing], N'TABLE', [ShipMethod], N'CONSTRAINT', [DF\_ShipMethod\_ShipBase];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.0', N'SCHEMA', [Purchasing], N'TABLE', [ShipMethod], N'CONSTRAINT', [DF\_ShipMethod\_ShipRate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Purchasing], N'TABLE', [ShipMethod], N'CONSTRAINT', [DF\_ShipMethod\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Purchasing], N'TABLE', [ShipMethod], N'CONSTRAINT', [DF\_ShipMethod\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [ShipBase] > (0.00)', N'SCHEMA', [Purchasing], N'TABLE', [ShipMethod], N'CONSTRAINT', [CK\_ShipMethod\_ShipBase];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [ShipRate] > (0.00)', N'SCHEMA', [Purchasing], N'TABLE', [ShipMethod], N'CONSTRAINT', [CK\_ShipMethod\_ShipRate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Sales], N'TABLE', [ShoppingCartItem], N'CONSTRAINT', [PK\_ShoppingCartItem\_ShoppingCartItemID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Product.ProductID.', N'SCHEMA', [Sales], N'TABLE', [ShoppingCartItem], N'CONSTRAINT', [FK\_ShoppingCartItem\_Product\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 1', N'SCHEMA', [Sales], N'TABLE', [ShoppingCartItem], N'CONSTRAINT', [DF\_ShoppingCartItem\_Quantity];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Sales], N'TABLE', [ShoppingCartItem], N'CONSTRAINT', [DF\_ShoppingCartItem\_DateCreated];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Sales], N'TABLE', [ShoppingCartItem], N'CONSTRAINT', [DF\_ShoppingCartItem\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [Quantity] >= (1)', N'SCHEMA', [Sales], N'TABLE', [ShoppingCartItem], N'CONSTRAINT', [CK\_ShoppingCartItem\_Quantity];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'CONSTRAINT', [PK\_SpecialOffer\_SpecialOfferID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.0', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'CONSTRAINT', [DF\_SpecialOffer\_DiscountPct];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0.0', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'CONSTRAINT', [DF\_SpecialOffer\_MinQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'CONSTRAINT', [DF\_SpecialOffer\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'CONSTRAINT', [DF\_SpecialOffer\_rowguid];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [DiscountPct] >= (0.00)', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'CONSTRAINT', [CK\_SpecialOffer\_DiscountPct];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [MaxQty] >= (0)', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'CONSTRAINT', [CK\_SpecialOffer\_MaxQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [MinQty] >= (0)', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'CONSTRAINT', [CK\_SpecialOffer\_MinQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [EndDate] >= [StartDate]', N'SCHEMA', [Sales], N'TABLE', [SpecialOffer], N'CONSTRAINT', [CK\_SpecialOffer\_EndDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Sales], N'TABLE', [SpecialOfferProduct], N'CONSTRAINT', [PK\_SpecialOfferProduct\_SpecialOfferID\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Product.ProductID.', N'SCHEMA', [Sales], N'TABLE', [SpecialOfferProduct], N'CONSTRAINT', [FK\_SpecialOfferProduct\_Product\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing SpecialOffer.SpecialOfferID.', N'SCHEMA', [Sales], N'TABLE', [SpecialOfferProduct], N'CONSTRAINT', [FK\_SpecialOfferProduct\_SpecialOffer\_SpecialOfferID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Sales], N'TABLE', [SpecialOfferProduct], N'CONSTRAINT', [DF\_SpecialOfferProduct\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Sales], N'TABLE', [SpecialOfferProduct], N'CONSTRAINT', [DF\_SpecialOfferProduct\_rowguid];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Person], N'TABLE', [StateProvince], N'CONSTRAINT', [PK\_StateProvince\_StateProvinceID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing CountryRegion.CountryRegionCode.', N'SCHEMA', [Person], N'TABLE', [StateProvince], N'CONSTRAINT', [FK\_StateProvince\_CountryRegion\_CountryRegionCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing SalesTerritory.TerritoryID.', N'SCHEMA', [Person], N'TABLE', [StateProvince], N'CONSTRAINT', [FK\_StateProvince\_SalesTerritory\_TerritoryID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 1 (TRUE)', N'SCHEMA', [Person], N'TABLE', [StateProvince], N'CONSTRAINT', [DF\_StateProvince\_IsOnlyStateProvinceFlag];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Person], N'TABLE', [StateProvince], N'CONSTRAINT', [DF\_StateProvince\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Person], N'TABLE', [StateProvince], N'CONSTRAINT', [DF\_StateProvince\_rowguid];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Sales], N'TABLE', [Store], N'CONSTRAINT', [PK\_Store\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing BusinessEntity.BusinessEntityID', N'SCHEMA', [Sales], N'TABLE', [Store], N'CONSTRAINT', [FK\_Store\_BusinessEntity\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing SalesPerson.SalesPersonID', N'SCHEMA', [Sales], N'TABLE', [Store], N'CONSTRAINT', [FK\_Store\_SalesPerson\_SalesPersonID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Sales], N'TABLE', [Store], N'CONSTRAINT', [DF\_Store\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of NEWID()', N'SCHEMA', [Sales], N'TABLE', [Store], N'CONSTRAINT', [DF\_Store\_rowguid];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [TransactionHistory], N'CONSTRAINT', [PK\_TransactionHistory\_TransactionID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Product.ProductID.', N'SCHEMA', [Production], N'TABLE', [TransactionHistory], N'CONSTRAINT', [FK\_TransactionHistory\_Product\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0', N'SCHEMA', [Production], N'TABLE', [TransactionHistory], N'CONSTRAINT', [DF\_TransactionHistory\_ReferenceOrderLineID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [TransactionHistory], N'CONSTRAINT', [DF\_TransactionHistory\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [TransactionHistory], N'CONSTRAINT', [DF\_TransactionHistory\_TransactionDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [TransactionType]=''p'' OR [TransactionType]=''s'' OR [TransactionType]=''w'' OR [TransactionType]=''P'' OR [TransactionType]=''S'' OR [TransactionType]=''W'')', N'SCHEMA', [Production], N'TABLE', [TransactionHistory], N'CONSTRAINT', [CK\_TransactionHistory\_TransactionType];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [TransactionHistoryArchive], N'CONSTRAINT', [PK\_TransactionHistoryArchive\_TransactionID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 0', N'SCHEMA', [Production], N'TABLE', [TransactionHistoryArchive], N'CONSTRAINT', [DF\_TransactionHistoryArchive\_ReferenceOrderLineID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [TransactionHistoryArchive], N'CONSTRAINT', [DF\_TransactionHistoryArchive\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [TransactionHistoryArchive], N'CONSTRAINT', [DF\_TransactionHistoryArchive\_TransactionDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [TransactionType]=''p'' OR [TransactionType]=''s'' OR [TransactionType]=''w'' OR [TransactionType]=''P'' OR [TransactionType]=''S'' OR [TransactionType]=''W''', N'SCHEMA', [Production], N'TABLE', [TransactionHistoryArchive], N'CONSTRAINT', [CK\_TransactionHistoryArchive\_TransactionType];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [UnitMeasure], N'CONSTRAINT', [PK\_UnitMeasure\_UnitMeasureCode];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [UnitMeasure], N'CONSTRAINT', [DF\_UnitMeasure\_ModifiedDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Purchasing], N'TABLE', [Vendor], N'CONSTRAINT', [PK\_Vendor\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing BusinessEntity.BusinessEntityID', N'SCHEMA', [Purchasing], N'TABLE', [Vendor], N'CONSTRAINT', [FK\_Vendor\_BusinessEntity\_BusinessEntityID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 1 (TRUE)', N'SCHEMA', [Purchasing], N'TABLE', [Vendor], N'CONSTRAINT', [DF\_Vendor\_ActiveFlag];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of 1 (TRUE)', N'SCHEMA', [Purchasing], N'TABLE', [Vendor], N'CONSTRAINT', [DF\_Vendor\_PreferredVendorStatus];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Purchasing], N'TABLE', [Vendor], N'CONSTRAINT', [DF\_Vendor\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [CreditRating] BETWEEN (1) AND (5)', N'SCHEMA', [Purchasing], N'TABLE', [Vendor], N'CONSTRAINT', [CK\_Vendor\_CreditRating];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'CONSTRAINT', [PK\_WorkOrder\_WorkOrderID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Product.ProductID.', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'CONSTRAINT', [FK\_WorkOrder\_Product\_ProductID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing ScrapReason.ScrapReasonID.', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'CONSTRAINT', [FK\_WorkOrder\_ScrapReason\_ScrapReasonID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'CONSTRAINT', [DF\_WorkOrder\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [OrderQty] > (0)', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'CONSTRAINT', [CK\_WorkOrder\_OrderQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [ScrappedQty] >= (0)', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'CONSTRAINT', [CK\_WorkOrder\_ScrappedQty];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [EndDate] >= [StartDate] OR [EndDate] IS NULL', N'SCHEMA', [Production], N'TABLE', [WorkOrder], N'CONSTRAINT', [CK\_WorkOrder\_EndDate];

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Primary key (clustered) constraint', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'CONSTRAINT', [PK\_WorkOrderRouting\_WorkOrderID\_ProductID\_OperationSequence];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing Location.LocationID.', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'CONSTRAINT', [FK\_WorkOrderRouting\_Location\_LocationID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Foreign key constraint referencing WorkOrder.WorkOrderID.', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'CONSTRAINT', [FK\_WorkOrderRouting\_WorkOrder\_WorkOrderID];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Default constraint value of GETDATE()', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'CONSTRAINT', [DF\_WorkOrderRouting\_ModifiedDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [ActualCost] > (0.00)', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'CONSTRAINT', [CK\_WorkOrderRouting\_ActualCost];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [ActualResourceHrs] >= (0.0000)', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'CONSTRAINT', [CK\_WorkOrderRouting\_ActualResourceHrs];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [PlannedCost] > (0.00)', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'CONSTRAINT', [CK\_WorkOrderRouting\_PlannedCost];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [ActualEndDate] >= [ActualStartDate] OR [ActualEndDate] IS NULL OR [ActualStartDate] IS NULL', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'CONSTRAINT', [CK\_WorkOrderRouting\_ActualEndDate];

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Check constraint [ScheduledEndDate] >= [ScheduledStartDate]', N'SCHEMA', [Production], N'TABLE', [WorkOrderRouting], N'CONSTRAINT', [CK\_WorkOrderRouting\_ScheduledEndDate];

GO

PRINT ' Functions';

GO

-- Functions

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Scalar function used in the uSalesOrderHeader trigger to set the starting account date.', N'SCHEMA', [dbo], N'FUNCTION', [ufnGetAccountingEndDate], NULL, NULL;

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Scalar function used in the uSalesOrderHeader trigger to set the ending account date.', N'SCHEMA', [dbo], N'FUNCTION', [ufnGetAccountingStartDate], NULL, NULL;

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Table value function returning the first name, last name, job title and contact type for a given contact.', N'SCHEMA', [dbo], N'FUNCTION', [ufnGetContactInformation], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the table value function ufnGetContactInformation. Enter a valid PersonID from the Person.Contact table.', N'SCHEMA', [dbo], N'FUNCTION', [ufnGetContactInformation], N'PARAMETER', '@PersonID';

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Scalar function returning the text representation of the Status column in the Document table.', N'SCHEMA', [dbo], N'FUNCTION', [ufnGetDocumentStatusText], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the scalar function ufnGetDocumentStatusText. Enter a valid integer.', N'SCHEMA', [dbo], N'FUNCTION', [ufnGetDocumentStatusText], N'PARAMETER', '@Status';

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Scalar function returning the dealer price for a given product on a particular order date.', N'SCHEMA', [dbo], N'FUNCTION', [ufnGetProductDealerPrice], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the scalar function ufnGetProductDealerPrice. Enter a valid ProductID from the Production.Product table.', N'SCHEMA', [dbo], N'FUNCTION', [ufnGetProductDealerPrice], N'PARAMETER', '@ProductID';

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the scalar function ufnGetProductDealerPrice. Enter a valid order date.', N'SCHEMA', [dbo], N'FUNCTION', [ufnGetProductDealerPrice], N'PARAMETER', '@OrderDate';

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Scalar function returning the list price for a given product on a particular order date.', N'SCHEMA', [dbo], N'FUNCTION', [ufnGetProductListPrice], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the scalar function ufnGetProductListPrice. Enter a valid ProductID from the Production.Product table.', N'SCHEMA', [dbo], N'FUNCTION', [ufnGetProductListPrice], N'PARAMETER', '@ProductID';

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the scalar function ufnGetProductListPrice. Enter a valid order date.', N'SCHEMA', [dbo], N'FUNCTION', [ufnGetProductListPrice], N'PARAMETER', '@OrderDate';

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Scalar function returning the standard cost for a given product on a particular order date.', N'SCHEMA', [dbo], N'FUNCTION', [ufnGetProductStandardCost], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the scalar function ufnGetProductStandardCost. Enter a valid ProductID from the Production.Product table.', N'SCHEMA', [dbo], N'FUNCTION', [ufnGetProductStandardCost], N'PARAMETER', '@ProductID';

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the scalar function ufnGetProductStandardCost. Enter a valid order date.', N'SCHEMA', [dbo], N'FUNCTION', [ufnGetProductStandardCost], N'PARAMETER', '@OrderDate';

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Scalar function returning the text representation of the Status column in the PurchaseOrderHeader table.', N'SCHEMA', [dbo], N'FUNCTION', [ufnGetPurchaseOrderStatusText], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the scalar function ufnGetPurchaseOrdertStatusText. Enter a valid integer.', N'SCHEMA', [dbo], N'FUNCTION', [ufnGetPurchaseOrderStatusText], N'PARAMETER', '@Status';

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Scalar function returning the text representation of the Status column in the SalesOrderHeader table.', N'SCHEMA', [dbo], N'FUNCTION', [ufnGetSalesOrderStatusText], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the scalar function ufnGetSalesOrderStatusText. Enter a valid integer.', N'SCHEMA', [dbo], N'FUNCTION', [ufnGetSalesOrderStatusText], N'PARAMETER', '@Status';

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Scalar function returning the quantity of inventory in LocationID 6 (Miscellaneous Storage)for a specified ProductID.', N'SCHEMA', [dbo], N'FUNCTION', [ufnGetStock], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the scalar function ufnGetStock. Enter a valid ProductID from the Production.ProductInventory table.', N'SCHEMA', [dbo], N'FUNCTION', [ufnGetStock], N'PARAMETER', '@ProductID';

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Scalar function used by the Sales.Customer table to help set the account number.', N'SCHEMA', [dbo], N'FUNCTION', [ufnLeadingZeros], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the scalar function ufnLeadingZeros. Enter a valid integer.', N'SCHEMA', [dbo], N'FUNCTION', [ufnLeadingZeros], N'PARAMETER', '@Value';

GO

PRINT ' Stored Procedures';

GO

-- Stored Procedures

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Stored procedure using a recursive query to return a multi-level bill of material for the specified ProductID.', N'SCHEMA', [dbo], N'PROCEDURE', [uspGetBillOfMaterials], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspGetBillOfMaterials. Enter a valid ProductID from the Production.Product table.', N'SCHEMA', [dbo], N'PROCEDURE', [uspGetBillOfMaterials], N'PARAMETER', '@StartProductID';

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspGetBillOfMaterials used to eliminate components not used after that date. Enter a valid date.', N'SCHEMA', [dbo], N'PROCEDURE', [uspGetBillOfMaterials], N'PARAMETER', '@CheckDate';

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Stored procedure using a recursive query to return the direct and indirect managers of the specified employee.', N'SCHEMA', [dbo], N'PROCEDURE', [uspGetEmployeeManagers], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspGetEmployeeManagers. Enter a valid BusinessEntityID from the HumanResources.Employee table.', N'SCHEMA', [dbo], N'PROCEDURE', [uspGetEmployeeManagers], N'PARAMETER', '@BusinessEntityID';

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Stored procedure using a recursive query to return the direct and indirect employees of the specified manager.', N'SCHEMA', [dbo], N'PROCEDURE', [uspGetManagerEmployees], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspGetManagerEmployees. Enter a valid BusinessEntityID of the manager from the HumanResources.Employee table.', N'SCHEMA', [dbo], N'PROCEDURE', [uspGetManagerEmployees], N'PARAMETER', '@BusinessEntityID';

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Stored procedure using a recursive query to return all components or assemblies that directly or indirectly use the specified ProductID.', N'SCHEMA', [dbo], N'PROCEDURE', [uspGetWhereUsedProductID], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspGetWhereUsedProductID. Enter a valid ProductID from the Production.Product table.', N'SCHEMA', [dbo], N'PROCEDURE', [uspGetWhereUsedProductID], N'PARAMETER', '@StartProductID';

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspGetWhereUsedProductID used to eliminate components not used after that date. Enter a valid date.', N'SCHEMA', [dbo], N'PROCEDURE', [uspGetWhereUsedProductID], N'PARAMETER', '@CheckDate';

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Logs error information in the ErrorLog table about the error that caused execution to jump to the CATCH block of a TRY...CATCH construct. Should be executed from within the scope of a CATCH block otherwise it will return without inserting error information.', N'SCHEMA', [dbo], N'PROCEDURE', [uspLogError], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Output parameter for the stored procedure uspLogError. Contains the ErrorLogID value corresponding to the row inserted by uspLogError in the ErrorLog table.', N'SCHEMA', [dbo], N'PROCEDURE', [uspLogError], N'PARAMETER', '@ErrorLogID';

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Prints error information about the error that caused execution to jump to the CATCH block of a TRY...CATCH construct. Should be executed from within the scope of a CATCH block otherwise it will return without printing any error information.', N'SCHEMA', [dbo], N'PROCEDURE', [uspPrintError], NULL, NULL;

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Updates the Employee table and inserts a new row in the EmployeePayHistory table with the values specified in the input parameters.', N'SCHEMA', [HumanResources], N'PROCEDURE', [uspUpdateEmployeeHireInfo], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a valid BusinessEntityID from the Employee table.', N'SCHEMA', [HumanResources], N'PROCEDURE', [uspUpdateEmployeeHireInfo], N'PARAMETER', '@BusinessEntityID';

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a title for the employee.', N'SCHEMA', [HumanResources], N'PROCEDURE', [uspUpdateEmployeeHireInfo], N'PARAMETER', '@JobTitle';

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a hire date for the employee.', N'SCHEMA', [HumanResources], N'PROCEDURE', [uspUpdateEmployeeHireInfo], N'PARAMETER', '@HireDate';

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the date the rate changed for the employee.', N'SCHEMA', [HumanResources], N'PROCEDURE', [uspUpdateEmployeeHireInfo], N'PARAMETER', '@RateChangeDate';

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the new rate for the employee.', N'SCHEMA', [HumanResources], N'PROCEDURE', [uspUpdateEmployeeHireInfo], N'PARAMETER', '@Rate';

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the pay frequency for the employee.', N'SCHEMA', [HumanResources], N'PROCEDURE', [uspUpdateEmployeeHireInfo], N'PARAMETER', '@PayFrequency';

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the current flag for the employee.', N'SCHEMA', [HumanResources], N'PROCEDURE', [uspUpdateEmployeeHireInfo], N'PARAMETER', '@CurrentFlag';

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Updates the Employee table with the values specified in the input parameters for the given BusinessEntityID.', N'SCHEMA', [HumanResources], N'PROCEDURE', [uspUpdateEmployeeLogin], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspUpdateEmployeeLogin. Enter a valid EmployeeID from the Employee table.', N'SCHEMA', [HumanResources], N'PROCEDURE', [uspUpdateEmployeeLogin], N'PARAMETER', '@BusinessEntityID';

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a valid ManagerID for the employee.', N'SCHEMA', [HumanResources], N'PROCEDURE', [uspUpdateEmployeeLogin], N'PARAMETER', '@OrganizationNode';

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a valid login for the employee.', N'SCHEMA', [HumanResources], N'PROCEDURE', [uspUpdateEmployeeLogin], N'PARAMETER', '@LoginID';

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a title for the employee.', N'SCHEMA', [HumanResources], N'PROCEDURE', [uspUpdateEmployeeLogin], N'PARAMETER', '@JobTitle';

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a hire date for the employee.', N'SCHEMA', [HumanResources], N'PROCEDURE', [uspUpdateEmployeeLogin], N'PARAMETER', '@HireDate';

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter the current flag for the employee.', N'SCHEMA', [HumanResources], N'PROCEDURE', [uspUpdateEmployeeLogin], N'PARAMETER', '@CurrentFlag';

GO

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Updates the Employee table with the values specified in the input parameters for the given EmployeeID.', N'SCHEMA', [HumanResources], N'PROCEDURE', [uspUpdateEmployeePersonalInfo], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspUpdateEmployeePersonalInfo. Enter a valid BusinessEntityID from the HumanResources.Employee table.', N'SCHEMA', [HumanResources], N'PROCEDURE', [uspUpdateEmployeePersonalInfo], N'PARAMETER', '@BusinessEntityID';

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a national ID for the employee.', N'SCHEMA', [HumanResources], N'PROCEDURE', [uspUpdateEmployeePersonalInfo], N'PARAMETER', '@NationalIDNumber';

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a birth date for the employee.', N'SCHEMA', [HumanResources], N'PROCEDURE', [uspUpdateEmployeePersonalInfo], N'PARAMETER', '@BirthDate';

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a marital status for the employee.', N'SCHEMA', [HumanResources], N'PROCEDURE', [uspUpdateEmployeePersonalInfo], N'PARAMETER', '@MaritalStatus';

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Input parameter for the stored procedure uspUpdateEmployeeHireInfo. Enter a gender for the employee.', N'SCHEMA', [HumanResources], N'PROCEDURE', [uspUpdateEmployeePersonalInfo], N'PARAMETER', '@Gender';

GO

[TOP](#Top)

PRINT ' XML Schemas';

GO

-- XML Schemas

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Collection of XML schemas for the AdditionalContactInfo column in the Person.Contact table.', N'SCHEMA', [Person], N'XML SCHEMA COLLECTION', [AdditionalContactInfoSchemaCollection], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Collection of XML schemas for the Resume column in the HumanResources.JobCandidate table.', N'SCHEMA', [HumanResources], N'XML SCHEMA COLLECTION', [HRResumeSchemaCollection], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Collection of XML schemas for the Demographics column in the Person.Person table.', N'SCHEMA', [Person], N'XML SCHEMA COLLECTION', [IndividualSurveySchemaCollection], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Collection of XML schemas for the Instructions column in the Production.ProductModel table.', N'SCHEMA', [Production], N'XML SCHEMA COLLECTION', [ManuInstructionsSchemaCollection], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Collection of XML schemas for the CatalogDescription column in the Production.ProductModel table.', N'SCHEMA', [Production], N'XML SCHEMA COLLECTION', [ProductDescriptionSchemaCollection], NULL, NULL;

EXECUTE [sys].[sp\_addextendedproperty] N'MS\_Description', N'Collection of XML schemas for the Demographics column in the Sales.Store table.', N'SCHEMA', [Sales], N'XML SCHEMA COLLECTION', [StoreSurveySchemaCollection], NULL, NULL;

GO

SET NOCOUNT OFF;

GO

[TOP](#Top)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Drop DDL Trigger for Database

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PRINT '';

PRINT '\*\*\* Disabling DDL Trigger for Database';

GO

DISABLE TRIGGER [ddlDatabaseTriggerLog]

ON DATABASE;

GO

[TOP](#Top)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Output database object creation messages

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT [PostTime], [DatabaseUser], [Event], [Schema], [Object], [TSQL], [XmlEvent]

FROM [dbo].[DatabaseLog];

GO

USE [master];

GO

PRINT 'Finished - ' + CONVERT(varchar, GETDATE(), 121);

GO

SET NOEXEC OFF

[TOP](#Top)