USE [master]

GO

/\*\*\*\*\*\* Object: Database [BalloonShop] \*\*\*\*\*\*/

CREATE DATABASE [BalloonShop]

CONTAINMENT = NONE

ALTER DATABASE [BalloonShop] SET COMPATIBILITY\_LEVEL = 100

GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))

begin

EXEC [BalloonShop].[dbo].[sp\_fulltext\_database] @action = 'enable'

end

GO

ALTER DATABASE [BalloonShop] SET ANSI\_NULL\_DEFAULT OFF

GO

ALTER DATABASE [BalloonShop] SET ANSI\_NULLS OFF

GO

ALTER DATABASE [BalloonShop] SET ANSI\_PADDING OFF

GO

ALTER DATABASE [BalloonShop] SET ANSI\_WARNINGS OFF

GO

ALTER DATABASE [BalloonShop] SET ARITHABORT OFF

GO

ALTER DATABASE [BalloonShop] SET AUTO\_CLOSE OFF

GO

ALTER DATABASE [BalloonShop] SET AUTO\_SHRINK OFF

GO

ALTER DATABASE [BalloonShop] SET AUTO\_UPDATE\_STATISTICS ON

GO

ALTER DATABASE [BalloonShop] SET CURSOR\_CLOSE\_ON\_COMMIT OFF

GO

ALTER DATABASE [BalloonShop] SET CURSOR\_DEFAULT GLOBAL

GO

ALTER DATABASE [BalloonShop] SET CONCAT\_NULL\_YIELDS\_NULL OFF

GO

ALTER DATABASE [BalloonShop] SET NUMERIC\_ROUNDABORT OFF

GO

ALTER DATABASE [BalloonShop] SET QUOTED\_IDENTIFIER OFF

GO

ALTER DATABASE [BalloonShop] SET RECURSIVE\_TRIGGERS OFF

GO

ALTER DATABASE [BalloonShop] SET ENABLE\_BROKER

GO

ALTER DATABASE [BalloonShop] SET AUTO\_UPDATE\_STATISTICS\_ASYNC OFF

GO

ALTER DATABASE [BalloonShop] SET DATE\_CORRELATION\_OPTIMIZATION OFF

GO

ALTER DATABASE [BalloonShop] SET TRUSTWORTHY OFF

GO

ALTER DATABASE [BalloonShop] SET ALLOW\_SNAPSHOT\_ISOLATION OFF

GO

ALTER DATABASE [BalloonShop] SET PARAMETERIZATION SIMPLE

GO

ALTER DATABASE [BalloonShop] SET READ\_COMMITTED\_SNAPSHOT OFF

GO

ALTER DATABASE [BalloonShop] SET HONOR\_BROKER\_PRIORITY OFF

GO

ALTER DATABASE [BalloonShop] SET RECOVERY FULL

GO

ALTER DATABASE [BalloonShop] SET MULTI\_USER

GO

ALTER DATABASE [BalloonShop] SET PAGE\_VERIFY CHECKSUM

GO

ALTER DATABASE [BalloonShop] SET DB\_CHAINING OFF

GO

ALTER DATABASE [BalloonShop] SET FILESTREAM( NON\_TRANSACTED\_ACCESS = OFF )

GO

ALTER DATABASE [BalloonShop] SET TARGET\_RECOVERY\_TIME = 0 SECONDS

GO

ALTER DATABASE [BalloonShop] SET DELAYED\_DURABILITY = DISABLED

GO

EXEC sys.sp\_db\_vardecimal\_storage\_format N'BalloonShop', N'ON'

GO

USE [BalloonShop]

GO

/\*\*\*\*\*\* Object: FullTextCatalog [BalloonShopFullText] \*\*\*\*\*\*/

CREATE FULLTEXT CATALOG [BalloonShopFullText]WITH ACCENT\_SENSITIVITY = ON

AS DEFAULT

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [dbo].[addDelivery] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create function [dbo].[addDelivery]

( @CartID char(36) )

returns INT

AS

BEGIN

declare @delivery int;

IF (SELECT SUM(Product.Price \* ShoppingCart.Quantity)

FROM ShoppingCart INNER JOIN Product

ON ShoppingCart.ProductID = Product.ProductID

WHERE ShoppingCart.CartID = @CartID ) > 100

set @delivery = 20;

ELSE

set @delivery = 0;

return @delivery;

END

GO

/\*\*\*\*\*\* Object: Table [dbo].[Attribute] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Attribute](

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

[Name] [nvarchar](100) NOT NULL,

PRIMARY KEY CLUSTERED

(

[AttributeID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[AttributeValue] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[AttributeValue](

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

[AttributeID] [int] NOT NULL,

[Value] [nvarchar](100) NOT NULL,

PRIMARY KEY CLUSTERED

(

[AttributeValueID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Category] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Category](

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

[DepartmentID] [int] NOT NULL,

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

[Description] [nvarchar](1000) NULL,

CONSTRAINT [PK\_Category\_1] PRIMARY KEY CLUSTERED

(

[CategoryID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Department] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Department](

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

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

[Description] [nvarchar](1000) NULL,

CONSTRAINT [PK\_Department] PRIMARY KEY CLUSTERED

(

[DepartmentID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[OrderDetail] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[OrderDetail](

[OrderID] [int] NOT NULL,

[ProductID] [int] NOT NULL,

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

[Quantity] [int] NOT NULL,

[UnitCost] [money] NOT NULL,

[Subtotal] AS ([Quantity]\*[UnitCost]),

CONSTRAINT [PK\_OrderDetail] PRIMARY KEY CLUSTERED

(

[OrderID] ASC,

[ProductID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Orders] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Orders](

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

[DateCreated] [smalldatetime] NOT NULL CONSTRAINT [DF\_Orders\_DateCreated] DEFAULT (getdate()),

[DateShipped] [smalldatetime] NULL,

[Verified] [bit] NOT NULL CONSTRAINT [DF\_Orders\_Verified] DEFAULT ((0)),

[Completed] [bit] NOT NULL CONSTRAINT [DF\_Orders\_Completed] DEFAULT ((0)),

[Canceled] [bit] NOT NULL CONSTRAINT [DF\_Orders\_Canceled] DEFAULT ((0)),

[Comments] [nvarchar](1000) NULL,

[CustomerName] [nvarchar](50) NULL,

[CustomerEmail] [nvarchar](50) NULL,

[ShippingAddress] [nvarchar](500) NULL,

CONSTRAINT [PK\_Orders] PRIMARY KEY CLUSTERED

(

[OrderID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Product] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Product](

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

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

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

[Price] [money] NOT NULL,

[Thumbnail] [nvarchar](50) NULL,

[Image] [nvarchar](50) NULL,

[PromoFront] [bit] NOT NULL,

[PromoDept] [bit] NOT NULL,

CONSTRAINT [PK\_Product] PRIMARY KEY CLUSTERED

(

[ProductID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[ProductAttributeValue] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ProductAttributeValue](

[ProductID] [int] NOT NULL,

[AttributeValueID] [int] NOT NULL,

PRIMARY KEY CLUSTERED

(

[ProductID] ASC,

[AttributeValueID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[ProductCategory] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ProductCategory](

[ProductID] [int] NOT NULL,

[CategoryID] [int] NOT NULL,

CONSTRAINT [PK\_ProductCategory] PRIMARY KEY CLUSTERED

(

[ProductID] ASC,

[CategoryID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Reviews] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Reviews](

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

[ProductID] [int] NOT NULL,

[Name] [nvarchar](200) NOT NULL,

[Description] [nvarchar](1000) NULL,

[DateAdded] [smalldatetime] NOT NULL,

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

CONSTRAINT [PK\_Reviews] PRIMARY KEY CLUSTERED

(

[ReviewID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[ShoppingCart] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[ShoppingCart](

[CartID] [char](36) NOT NULL,

[ProductID] [int] NOT NULL,

[Quantity] [int] NOT NULL,

[Attributes] [nvarchar](1000) NULL,

[DateAdded] [smalldatetime] NOT NULL,

CONSTRAINT [PK\_ShoppingCart] PRIMARY KEY CLUSTERED

(

[CartID] ASC,

[ProductID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

/\*\*\*\*\*\* Object: View [dbo].[ProdsInCats] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE VIEW [dbo].[ProdsInCats]

AS

SELECT dbo.Product.ProductID, dbo.Product.Name, dbo.Product.Description, dbo.Product.Price, dbo.Product.Thumbnail, dbo.ProductCategory.CategoryID

FROM dbo.Product INNER JOIN

dbo.ProductCategory ON dbo.Product.ProductID = dbo.ProductCategory.ProductID

GO

SET IDENTITY\_INSERT [dbo].[Attribute] ON

INSERT [dbo].[Attribute] ([AttributeID], [Name]) VALUES (1, N'Color')

SET IDENTITY\_INSERT [dbo].[Attribute] OFF

SET IDENTITY\_INSERT [dbo].[AttributeValue] ON

INSERT [dbo].[AttributeValue] ([AttributeValueID], [AttributeID], [Value]) VALUES (1, 1, N'White')

INSERT [dbo].[AttributeValue] ([AttributeValueID], [AttributeID], [Value]) VALUES (2, 1, N'Black')

INSERT [dbo].[AttributeValue] ([AttributeValueID], [AttributeID], [Value]) VALUES (3, 1, N'Red')

INSERT [dbo].[AttributeValue] ([AttributeValueID], [AttributeID], [Value]) VALUES (4, 1, N'Orange')

INSERT [dbo].[AttributeValue] ([AttributeValueID], [AttributeID], [Value]) VALUES (5, 1, N'Yellow')

INSERT [dbo].[AttributeValue] ([AttributeValueID], [AttributeID], [Value]) VALUES (6, 1, N'Green')

INSERT [dbo].[AttributeValue] ([AttributeValueID], [AttributeID], [Value]) VALUES (7, 1, N'Blue')

INSERT [dbo].[AttributeValue] ([AttributeValueID], [AttributeID], [Value]) VALUES (8, 1, N'Indigo')

INSERT [dbo].[AttributeValue] ([AttributeValueID], [AttributeID], [Value]) VALUES (9, 1, N'Purple')

SET IDENTITY\_INSERT [dbo].[AttributeValue] OFF

SET IDENTITY\_INSERT [dbo].[Category] ON

INSERT [dbo].[Category] ([CategoryID], [DepartmentID], [Name], [Description]) VALUES (1, 1, N'Car', N'List Of Car')

INSERT [dbo].[Category] ([CategoryID], [DepartmentID], [Name], [Description]) VALUES (2, 1, N'Sport Car', N'List Of Sport Car')

INSERT [dbo].[Category] ([CategoryID], [DepartmentID], [Name], [Description]) VALUES (3, 1, N'Truck', N'List Of Truck')

INSERT [dbo].[Category] ([CategoryID], [DepartmentID], [Name], [Description]) VALUES (4, 2, N'Construction Set', N'List Of Construction Set')

INSERT [dbo].[Category] ([CategoryID], [DepartmentID], [Name], [Description]) VALUES (5, 2, N'Puzzle Game', N'List Of Education Toy')

SET IDENTITY\_INSERT [dbo].[Category] OFF

SET IDENTITY\_INSERT [dbo].[Department] ON

INSERT [dbo].[Department] ([DepartmentID], [Name], [Description]) VALUES (1, N'RC Car', N'Hey! Do You Want To Drive?')

INSERT [dbo].[Department] ([DepartmentID], [Name], [Description]) VALUES (2, N'Other Toy', N'You Are Welcome To Come In And Browse')

SET IDENTITY\_INSERT [dbo].[Department] OFF

INSERT [dbo].[OrderDetail] ([OrderID], [ProductID], [ProductName], [Quantity], [UnitCost]) VALUES (1, 4, N'Braha Porsche 911 GT3 1:24 R/C Car White', 1, 49.9000)

INSERT [dbo].[OrderDetail] ([OrderID], [ProductID], [ProductName], [Quantity], [UnitCost]) VALUES (2, 1, N'New Bright RC Full Function Sport Car Camaro', 2, 25.4800)

INSERT [dbo].[OrderDetail] ([OrderID], [ProductID], [ProductName], [Quantity], [UnitCost]) VALUES (2, 5, N'Braha Chevrolet Camaro 1:24 R/C Car Red', 5, 52.0000)

SET IDENTITY\_INSERT [dbo].[Orders] ON

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (1, CAST(N'2010-11-25 16:48:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (2, CAST(N'2010-11-26 07:05:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (3, CAST(N'2010-11-26 08:38:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (4, CAST(N'2010-11-26 11:59:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (5, CAST(N'2010-11-30 07:15:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (6, CAST(N'2010-11-30 07:17:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (7, CAST(N'2010-11-30 07:21:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (8, CAST(N'2010-11-30 11:18:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (9, CAST(N'2010-11-30 11:20:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (10, CAST(N'2010-11-30 11:39:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (11, CAST(N'2011-03-18 16:21:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (12, CAST(N'2011-03-18 16:23:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (13, CAST(N'2012-03-15 16:38:00' AS SmallDateTime), CAST(N'2011-05-15 08:34:00' AS SmallDateTime), 0, 0, 1, N'', N'', N'', N'')

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (14, CAST(N'2012-03-23 12:39:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (15, CAST(N'2012-03-23 12:39:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (16, CAST(N'2012-03-23 12:40:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (17, CAST(N'2012-03-23 12:43:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (18, CAST(N'2015-12-16 17:35:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (19, CAST(N'2015-12-16 17:56:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (20, CAST(N'2015-12-16 18:01:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (21, CAST(N'2015-12-16 20:37:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (22, CAST(N'2015-12-16 21:00:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (23, CAST(N'2015-12-16 21:02:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (24, CAST(N'2015-12-16 21:02:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (25, CAST(N'2015-12-17 04:02:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (26, CAST(N'2015-12-17 04:02:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (27, CAST(N'2015-12-17 04:03:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (28, CAST(N'2015-12-17 04:03:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (29, CAST(N'2015-12-17 12:09:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (30, CAST(N'2015-12-17 14:31:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (31, CAST(N'2015-12-17 14:48:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

INSERT [dbo].[Orders] ([OrderID], [DateCreated], [DateShipped], [Verified], [Completed], [Canceled], [Comments], [CustomerName], [CustomerEmail], [ShippingAddress]) VALUES (32, CAST(N'2015-12-17 21:02:00' AS SmallDateTime), NULL, 0, 0, 0, NULL, NULL, NULL, NULL)

SET IDENTITY\_INSERT [dbo].[Orders] OFF

SET IDENTITY\_INSERT [dbo].[Product] ON

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (1, N'New Bright RC Full Function Sport Car Camaro', N'The New Bright RC Full Function Sports Car Camaro will have your kiddo zoomin’ with joy. This mini Camaro remote control car is just like the real thing. Your child will love testing speed and agility of the vehicle using a remote control to go forward, reverse, right and left. The mini Camaro is ready to rev its engine once some batteries are added. The remote control has speed control and is the perfect size for little hands. Batteries included. Ages 10 and up. Limited manufacturer warranty.', 25.4800, N'Camaro1.jpg', N'Camaro1.jpg', 0, 1)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (2, N'Braha Lexus LFA 1:24 R/C Car Orange', N'Lexus LFA orange remote control car. Now you can get a feeling of what it is like to be in control of your very own Lexus LFA . This high class sports car designed to manuver in and out of the tightest turns. Driving this car will give you a legitimate feeling for the road and putting yourself in the drivers seat. Having Lexus top sports car in the tip of your fingers can turn any average playroom into a driving experience.', 69.9500, N'BrahaLexus.jpg', N'BrahaLexus.jpg', 0, 0)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (3, N'Braha Mercedes Benz SLR McLaren 1:24 R/C Car Red', N'Mercedes Benz SLR McLaren red remote control car. Now you can get a feeling of what it is like to be in control of your very own Mercedes Benz . This high class sports car designed to maneuver in and out of the tightest turns. Driving this car will give you a legitimate feeling for the road and putting yourself in the driver’s seat. Having Mercedes top sports car in the tip of your fingers can turn any average playroom into a driving experience.', 69.9500, N'BrahaMercedesBenz.jpg', N'BrahaMercedesBenz.jpg', 1, 0)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (4, N'Braha Porsche 911 GT3 1:24 R/C Car White', N'\*Includes: Remote Controlled Cars Vehicle Features: Lights, Indoor/Outdoor Use, Collectible \*Controller Features: Antenna, Throttle, Joystick \*Scale: 1:23 \*Dimensions: 4.150H x 5.200W x 10.500L Product \*Weight: 0.900 \*Material: Plastic \*Suggested Age: 8~13 \*Power Source: \*Battery Battery: required, \*not included, Batteries', 49.9000, N'BrahaPorsche911GT3.jpg', N'BrahaPorsche911GT3.jpg', 1, 1)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (5, N'Braha Chevrolet Camaro 1:24 R/C Car Red', N'Chevrolet Camaro Red remote control car. Now you can get a feeling of what it is like to be in control of your very own Camaro. This high class sports car designed to maneuver in and out of the tightest turns. Driving this car will give you a legitimate feeling for the road and putting yourself in the driver’s seat. Having Chevrolets top sports car in the tip of your fingers can turn any average playroom into a driving experience.', 52.0000, N'BrahaChevroletCamaroRed.jpg', N'BrahaChevroletCamaroRed.jpg', 1, 0)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (6, N'Melissa & Doug® Deluxe USA Map Sound Puzzle', N'As each piece is placed correctly in the puzzle board, your child will be delighted to hear the names of the states and capitals! Forty wooden pieces help develop map and memory skills. Pictures on pieces show images associated with each state. State borders shown under pieces. A captivating way to learn about the USA, this classical peg puzzle says the name of each state and its capital when the pieces are properly placed on the board Quality wooden sound puzzle is painted with bright animals and graphics Realistic sounds make for engaging play; the learning puzzle reinforces cause and effect, early vocabulary and matching skills. This high class sports car designed to maneuver in and out of the tightest turns. Driving this car will give you a legitimate feeling for the road and putting yourself in the driver’s seat. Having Mercedes top sports car in the tip of your fingers can turn any average playroom into a driving experience.', 12.9900, N'USA Map Sound Puzzle.jpg', N'USA Map Sound Puzzle.jpg', 0, 0)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (7, N'Ideal Tripoley Deluxe Mat Edition Card Game', N'Object of this best selling game is to collect chips through a series of Michigan Rummy, Hearts and Poker •Includes a 2-sided Las Vegas style foldable 24-inch by 24-inch deluxe playing mat •Tripoleys mat keeps cards and chips in place and lets you concentrate on winning •Designed for 2-9-players •Recommended for people 8-years of age and older', 13.4500, N'Tripoley Deluxe Mat Edition Card Game.jpg', N'Tripoley Deluxe Mat Edition Card Game.jpg', 1, 1)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (8, N'John N. Hansen Educa North American Map 1000 piece Puzzle', N'Have some educational fun with the John N. Hansen Educa North American Map 1000 piece Puzzle. This puzzle features a beautiful map of North America. Educa puzzles are made in Barcelona, Spain from the best quality boards to ensure a perfect fit between puzzle pieces without puzzle dust. Once completed, this map puzzle measures out to 26.75" x 18.75". For ages 12 and up.', 15.0000, N'North American Map 1000 piece.jpg', N'North American Map 1000 piece.jpg', 0, 0)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (9, N'Rubiks Cube Safe', N'Keep your valuables safe in the Rubiks cube safe •Hidden secret compartment inside •Turn the middle three layers in a specific combination to access your cash •Perfect gift for the Rubiks Cube enthusiast in your life.', 10.0000, N'Rubiks Cube Safe.jpg', N'Rubiks Cube Safe.jpg', 0, 0)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (10, N'Moves Rubik s Customize Speed Cube', N'NEW Winning Moves Rubik s Customize Speed Cube Pro-Pack/Acc 5029 NIB.', 16.9900, N'Moves Rubik s Customize Speed Cube.jpg', N'Moves Rubik s Customize Speed Cube.jpg', 1, 1)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (11, N'BMW X6 1:24 Remote Control Car', N'BMW X6 white remote control car. Now you can get a feeling of what it is like to be in control of your very own BMW X6. This high class sports car designed to maneuver in and out of the tightest turns. Driving this car will give you a legitimate feeling for the road and putting yourself in the driver’s seat. Having BMW’s top car in the tip of your fingers can turn any average playroom into a driving experience.', 109.9500, N'BrahaBMWM3Black.jpg', N'BrahaBMWM3Black.jpg', 0, 0)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (12, N'Braha BMW X6 1:24 R/C Car White', N'\*Includes: Remote Controlled Cars Vehicle Features: Lights, Indoor/Outdoor Use, Collectible \*Controller Features: Antenna, Throttle, Joystick \*Scale: 1:24 \*Dimensions: 4.250H x 5.500W x 11.500L Product \*Weight: 1.000 \*Material: Plastic \*Suggested Age: 8~13 \*Power Source: \*Battery Battery: required, \*not included, Batteries', 109.9500, N'BrahaBMWX6White.jpg', N'BrahaBMWX6White.jpg', 0, 0)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (13, N'Tamiya Electric RC 1:10 Datsun 240Z', N'The Datsun 240Z, which is famously known in Japan as the Fairlady Z, was developed for the international market with sporty performance, a reasonable price, and superb comfort in mind. The 240Z still remains a greatly popular machine even today.', 94.9500, N'Datsun240Z.jpg', N'Datsun240Z.jpg', 0, 0)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (14, N'Tamiya TT-01D 1/10 RC NISSAN SKYLINE GTR', N'\*TT-01D DRIFT CHASSIS, BLACK MOTOR, FULL BEARINGS \*STEALTH MOUNTS INSTALLED, NOT DESIGNED FOR HIGH SPEEDS \*2.4ghz Radio 100% Ready To Run When You Receive \*BODY HAS FRONT + REAR LED LIGHTS, ROLL CAGE, TINTED WINDOWS, MIRRORS, EXHAUST \*7.2 BATTERY + CHARGER+RADIO ALL INCLUDED.', 74.9500, N'Tamiya NISSAN SKYLINE GTR.jpg', N'Tamiya NISSAN SKYLINE GTR.jpg', 1, 1)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (15, N'Braha BMW M3 1:24 R/C Car White', N'Includes: Remote Controlled Cars Vehicle Features: Lights, Indoor/Outdoor Use, Collectible \*Controller Features: Antenna, Throttle, Joystick \*Scale: 1:25 \*Dimensions: 4.250H x 5.400W x 12.000L Product \*Weight: .930 \*Material: Plastic \*Suggested Age: 8~13 \*Power Source: \*Battery Battery: required, \*not included, Batteries.', 89.9500, N'Braha BMW White.jpg', N'Braha BMW White.jpg', 0, 0)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (16, N'Bentley Continental GT V8 1:24 Remote Control Car', N'\*This fully functional Bentley Continental GT radio control sports car allows you to give him the car of his dreams. \*PRODUCT FEATURES (•1: 24 scale •Functional head & tail lights •Hard plastic ultra-durable body •Quick speed & agile handling \*PRODUCT DETAILS(•Includes: car & transmitter •4.25"H x 11.5"W x 5.5"D (packaged) •Age: 8 years & up •Requires 5 AA batteries (not included).', 100.5000, N'Bentley Continental GT.jpg', N'Bentley Continental GT.jpg', 0, 0)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (17, N'Audi R8 AWD Stealth Black Brushless Electric', N'\*Fully Assembled Chassis •SYNCRO KT-200 2ch Radio System Installed. •Pre-painted assembled body with decals applied •Sealed differential and transmission delivers the reliability for rough road racing performance •Fixed length suspension arms eliminate the need for camber, caster and toe-in (front) adjustment •Official factory licensed body for the cleanest, sharpest looking car on the street; Factory painted, trimmed and ready to mount •Factory installed 2.4 GHz radio with a strong signal that endures beyond practical limits •High traction tires are pre-mounted to the wheels for instant action and another full set of drift-ready wheels and tires are also included •Oil-filled shocks smooth out the ride and improve handing of the Fazer VE; Driving is much easier any more predictable •Fazer VE Audi RB is an officially licensed product of AUDI AG.', 109.9500, N'Audi R8.jpg', N'Audi R8.jpg', 1, 0)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (18, N'Bentley GT Black Model', N'\*Limited Eddition\* Bentley GT speed convertible 49900 Black Model.', 170.9500, N'Bentley GT Black Model.jpg', N'Bentley GT Black Model.jpg', 1, 1)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (19, N'Tamiya 1/14 Knight Hauler Tractor Truck', N'The leader in hobby products around the world • Made using the highest quality components and materials • Test for durability and safety.', 60.0000, N'Knight Hauler Tractor Truck.jpg', N'Knight Hauler Tractor Truck.jpg', 0, 0)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (20, N'Axial AX90028 SCS10 Jeep Wrangler RTR RC Truck', N'FEATURES: Chassis: Scale steel C-channel with cross bracing for reinforcement and assembled with hex hardware Drive: 4 wheel shaft with WB8 wild boar • INCLUDES: RTR SCX10 Jeep Wrangler, 2-ch 2.4GHz radio, ESC, 27T Motor, Instruction Manual, sheet of decals • REQUIRES: Transmitter batteries: Four AA size battery: 7.2V stick pack with Standard Connector battery Charger: Compatible with selected battery Track', 50.0000, N'Jeep Wrangler.jpg', N'Jeep Wrangler.jpg', 0, 0)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (21, N'Engine Remote Control Mining Truck', N'\*Limited Eddition\* Multi-Function Electric Powered •Works like the real thing •Realistic Working Lights.', 163.2500, N'Mining Truck.jpg', N'Mining Truck.jpg', 1, 0)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (22, N'LEGO Star Wars Super Star Destroyer 10221 ', N'Includes Darth Vader, Admiral Piett, Dengar and Bossk minifigures and also includes IG-88 figure •Center section lifts off to reveal command center •Includes display stand and data sheet label •Measures nearly 50 inches (124.5 cm) long and weighs nearly 8 pounds (3.5kg) •3152 pieces.', 113.9000, N'LEGO Star Wars Super Star Destroyer.jpg', N'LEGO Star Wars Super Star Destroyer.jpg', 1, 0)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (23, N'LEGO Creator Modular Building Palace Cinema 10232 ', N'A gorgeous Lego-style tribute to the grand moviehouses of old, the LEGO Creator Modular Building Palace Cinema 10232 features fun details including spotlights, a black limousine, a star-studded sidewalk, and an incredibly detailed interior with a lobby, concession stand and ticket area. A grand staircase leads into a large-screen Lego theater, movie projector and reclining chairs for 6 minifigures. The 2-story Palace Cinema corner building includes a sidewalk of the stars, brick-built entrance doors, posters, sign frontage, a tower with spires and rooftop decorations. Also comes with 6 minifigures: a child actress, chauffeur, female guest, male guest, photographer and cinema worker.', 149.9900, N'LEGO Creator Modular Building Palace Cinema.jpg', N'LEGO Creator Modular Building Palace Cinema', 1, 0)

INSERT [dbo].[Product] ([ProductID], [Name], [Description], [Price], [Thumbnail], [Image], [PromoFront], [PromoDept]) VALUES (24, N'LEGO Technic Motorized Bulldozerk', N'Product Dimensions 25.7 x 18.9 x 3.8 inches Item Weight: 6.4 pounds Shipping Weight: 6.3 pounds (View shipping rates and policies) ASIN: B000MZHT0C Item model number:158606Manufacturer recommended age: 10 years and up Batteries:6 AA batteries required.', 99.9900, N'Lego Technic Motorized Bulldozer.jpg', N'Lego Technic Motorized Bulldozer.jpg', 1, 0)

SET IDENTITY\_INSERT [dbo].[Product] OFF

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (1, 1)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (1, 2)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (1, 3)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (1, 4)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (1, 5)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (1, 6)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (1, 7)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (1, 8)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (1, 9)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (2, 1)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (2, 2)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (2, 3)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (2, 4)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (2, 5)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (2, 6)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (2, 7)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (2, 8)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (2, 9)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (3, 1)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (3, 2)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (3, 3)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (3, 4)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (3, 5)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (3, 6)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (3, 7)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (3, 8)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (3, 9)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (4, 1)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (4, 2)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (4, 3)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (4, 4)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (4, 5)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (4, 6)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (4, 7)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (4, 8)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (4, 9)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (5, 1)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (5, 2)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (5, 3)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (5, 4)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (5, 5)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (5, 6)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (5, 7)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (5, 8)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (5, 9)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (6, 1)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (6, 2)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (6, 3)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (6, 4)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (6, 5)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (6, 6)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (6, 7)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (6, 8)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (6, 9)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (7, 1)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (7, 2)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (7, 3)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (7, 4)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (7, 5)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (7, 6)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (7, 7)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (7, 8)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (7, 9)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (8, 1)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (8, 2)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (8, 3)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (8, 4)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (8, 5)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (8, 6)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (8, 7)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (8, 8)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (8, 9)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (9, 1)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (9, 2)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (9, 3)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (9, 4)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (9, 5)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (9, 6)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (9, 7)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (9, 8)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (9, 9)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (10, 1)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (10, 2)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (10, 3)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (10, 4)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (10, 5)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (10, 6)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (10, 7)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (10, 8)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (10, 9)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (11, 1)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (11, 2)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (11, 3)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (11, 4)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (11, 5)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (11, 6)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (11, 7)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (11, 8)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (11, 9)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (12, 1)

GO

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (12, 2)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (12, 3)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (12, 4)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (12, 5)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (12, 6)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (12, 7)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (12, 8)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (12, 9)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (13, 1)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (13, 2)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (13, 3)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (13, 4)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (13, 5)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (13, 6)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (13, 7)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (13, 8)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (13, 9)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (14, 1)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (14, 2)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (14, 3)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (14, 4)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (14, 5)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (14, 6)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (14, 7)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (14, 8)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (14, 9)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (15, 1)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (15, 2)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (15, 3)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (15, 4)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (15, 5)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (15, 6)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (15, 7)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (15, 8)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (15, 9)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (16, 1)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (16, 2)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (16, 3)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (16, 4)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (16, 5)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (16, 6)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (16, 7)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (16, 8)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (16, 9)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (17, 1)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (17, 2)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (17, 3)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (17, 4)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (17, 5)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (17, 6)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (17, 7)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (17, 8)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (17, 9)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (18, 1)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (18, 2)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (18, 3)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (18, 4)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (18, 5)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (18, 6)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (18, 7)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (18, 8)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (18, 9)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (19, 1)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (19, 2)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (19, 3)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (19, 4)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (19, 5)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (19, 6)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (19, 7)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (19, 8)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (19, 9)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (20, 1)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (20, 2)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (20, 3)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (20, 4)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (20, 5)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (20, 6)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (20, 7)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (20, 8)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (20, 9)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (21, 1)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (21, 2)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (21, 3)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (21, 4)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (21, 5)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (21, 6)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (21, 7)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (21, 8)

INSERT [dbo].[ProductAttributeValue] ([ProductID], [AttributeValueID]) VALUES (21, 9)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (1, 2)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (2, 2)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (3, 2)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (4, 2)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (5, 2)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (6, 5)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (7, 5)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (8, 5)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (9, 5)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (10, 5)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (11, 1)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (12, 1)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (13, 1)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (14, 1)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (15, 1)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (16, 1)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (17, 1)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (18, 1)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (19, 3)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (20, 3)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (21, 3)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (22, 4)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (23, 4)

INSERT [dbo].[ProductCategory] ([ProductID], [CategoryID]) VALUES (24, 4)

INSERT [dbo].[ShoppingCart] ([CartID], [ProductID], [Quantity], [Attributes], [DateAdded]) VALUES (N'1c8578b4-f7c3-45fe-81f6-b7613917a899', 1, 1, N'attributes', CAST(N'2015-12-17 21:08:00' AS SmallDateTime))

INSERT [dbo].[ShoppingCart] ([CartID], [ProductID], [Quantity], [Attributes], [DateAdded]) VALUES (N'a6ac31fb-ef2a-4e38-b3c1-7904d5d6a7c9', 5, 1, N'attributes', CAST(N'2012-03-15 16:39:00' AS SmallDateTime))

ALTER TABLE [dbo].[AttributeValue] WITH CHECK ADD CONSTRAINT [FK\_AttributeValue\_Attribute] FOREIGN KEY([AttributeID])

REFERENCES [dbo].[Attribute] ([AttributeID])

GO

ALTER TABLE [dbo].[AttributeValue] CHECK CONSTRAINT [FK\_AttributeValue\_Attribute]

GO

ALTER TABLE [dbo].[Category] WITH CHECK ADD CONSTRAINT [FK\_Category\_Department] FOREIGN KEY([DepartmentID])

REFERENCES [dbo].[Department] ([DepartmentID])

GO

ALTER TABLE [dbo].[Category] CHECK CONSTRAINT [FK\_Category\_Department]

GO

ALTER TABLE [dbo].[OrderDetail] WITH CHECK ADD CONSTRAINT [FK\_OrderDetail\_Orders] FOREIGN KEY([OrderID])

REFERENCES [dbo].[Orders] ([OrderID])

GO

ALTER TABLE [dbo].[OrderDetail] CHECK CONSTRAINT [FK\_OrderDetail\_Orders]

GO

ALTER TABLE [dbo].[ProductAttributeValue] WITH CHECK ADD CONSTRAINT [FK\_ProductAttributeValue\_AttributeValue] FOREIGN KEY([AttributeValueID])

REFERENCES [dbo].[AttributeValue] ([AttributeValueID])

GO

ALTER TABLE [dbo].[ProductAttributeValue] CHECK CONSTRAINT [FK\_ProductAttributeValue\_AttributeValue]

GO

ALTER TABLE [dbo].[ProductCategory] WITH CHECK ADD CONSTRAINT [FK\_ProductCategory\_Category] FOREIGN KEY([CategoryID])

REFERENCES [dbo].[Category] ([CategoryID])

GO

ALTER TABLE [dbo].[ProductCategory] CHECK CONSTRAINT [FK\_ProductCategory\_Category]

GO

ALTER TABLE [dbo].[ShoppingCart] WITH CHECK ADD CONSTRAINT [FK\_ShoppingCart\_Product] FOREIGN KEY([ProductID])

REFERENCES [dbo].[Product] ([ProductID])

GO

ALTER TABLE [dbo].[ShoppingCart] CHECK CONSTRAINT [FK\_ShoppingCart\_Product]

GO

ALTER TABLE [dbo].[Reviews] ADD CONSTRAINT [DF\_Reviews\_DateAdded] DEFAULT (getdate()) FOR [DateAdded]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogAddDepartment] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogAddDepartment]

(@DepartmentName nvarchar(50),

@DepartmentDescription nvarchar(1000))

AS

INSERT INTO Department (Name, Description)

VALUES (@DepartmentName, @DepartmentDescription)

GO

/\*\*\*\*\*\* Object: Table [dbo].[Users] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[Users](

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

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

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

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

[CellPhone] [char](12) NOT NULL,

[Email] [nvarchar](50) NULL,

[City] [nvarchar](20) NOT NULL,

[Address] [nvarchar](100) NOT NULL,

[DateJoined] [smalldatetime] NOT NULL,

[Authority] [bit] NOT NULL,

[Province] [nchar](20) NOT NULL,

[PostalCode] [nchar](6) NULL,

CONSTRAINT [PK\_Users] PRIMARY KEY CLUSTERED

(

[UserID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[UserCreate] Script Date: 2014-12-08 오후 9:50:28 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[UserCreate]

(@UserID NVARCHAR(30),

@FirstName NVARCHAR(50),

@LastName NVARCHAR(50),

@Password NVARCHAR(50),

@CellPhone CHAR(12),

@Email NVARCHAR(50),

@City NVARCHAR(20),

@Address NVARCHAR(100),

@Province NCHAR(20),

@PostalCode NCHAR(6))

AS

INSERT INTO Users

(UserID,

FirstName,

LastName,

Password,

CellPhone,

Email,

City,

Address,

Province,

PostalCode,

DateJoined,

Authority)

VALUES

(@UserID,

@FirstName,

@LastName,

@Password,

@CellPhone,

@Email,

@City,

@Address,

@Province,

@PostalCode,

GETDATE(),

0)

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[UserDetails] Script Date: 2014-12-08 오후 9:50:28 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[UserDetails]

(@userID nvarchar(30))

AS

SELECT FirstName,

LastName,

Password,

CellPhone,

Email,

City,

Address,

Province,

PostalCode

FROM Users

WHERE UserID = @userID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[UserEdit] Script Date: 2014-12-08 오후 9:50:28 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[UserEdit]

(@UserID NVARCHAR(30),

@FirstName NVARCHAR(50),

@LastName NVARCHAR(50),

@Password NVARCHAR(50),

@CellPhone CHAR(12),

@Email NVARCHAR(50),

@City NVARCHAR(20),

@Address NVARCHAR(100),

@Province NCHAR(20),

@PostalCode NCHAR(6))

AS

Update Users

Set

FirstName = @FirstName,

LastName = @LastName,

Password = @Password,

CellPhone = @CellPhone,

Email = @Email,

City = @City,

Address = @Address,

Province = @Province,

PostalCode = @PostalCode

Where

UserID = @UserID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[UserIDCheck] Script Date: 2014-12-08 오후 9:50:28 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[UserIDCheck]

(@UserID NVARCHAR(30))

AS

Set NoCount On

Begin

SELECT top 1 @UserID FROM Users WHERE UserID = @UserID

End

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[UserLogin] Script Date: 2014-12-08 오후 9:50:28 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[UserLogin]

@UserID NVARCHAR(30),

@pwd NVARCHAR(50)

AS

--DECLARE @mem\_Nick varchar(20)

--SET @mem\_Nick = '-1'

BEGIN

IF EXISTS(SELECT Top 1 FirstName FROM Users WHERE UserID = @UserID) -- when existing ID

IF EXISTS(SELECT Top 1 FirstName FROM Users WHERE UserID = @UserID AND Password=@pwd) -- check pwd

SELECT COALESCE(FirstName,' ') + COALESCE(LastName, '') FROM Users WHERE UserID = @UserID AND Password=@pwd

--RETURN @mem\_Nick -- correct ID and pwd

ELSE

SELECT '-1'

--RETURN @mem\_Nick -- wrong pwd

ELSE -- not exist ID

SELECT '-1'

--RETURN @mem\_Nick

END

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[ReviewCreate] 8 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[ReviewCreate]

(@UserID NVARCHAR(30),

@ProductID INT,

@Name NVARCHAR(200),

@Description NVARCHAR(1000))

AS

BEGIN

INSERT INTO Reviews(ProductID, Name, Description, UserID) VALUES(@ProductID, @Name, @Description, @UserID)

END

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[ReviewDelete] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[ReviewDelete]

(@UserID NVARCHAR(30),

@ProductID INT)

AS

BEGIN

IF EXISTS(SELECT ProductID FROM Reviews WHERE ProductID = @ProductID and UserID = @UserID)

DELETE Reviews WHERE ProductID = @ProductID and UserID = @UserID

END

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[ReviewList] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[ReviewList]

(@ProductID INT)

AS

-- declare a new TABLE variable

DECLARE @Review TABLE

(RowNumber INT,

ReviewID INT,

ProductID INT,

Name NVARCHAR(200),

Description NVARCHAR(1000),

DateAdded datetime,

UserID NVARCHAR(30));

-- populate the table variable with the complete list of products

INSERT INTO @Review

SELECT ROW\_NUMBER() OVER (ORDER BY Reviews.ReviewID),

Reviews.ReviewID, Reviews.ProductID, Name,Description,

DateAdded,

Users.UserID

FROM Reviews INNER JOIN Users

ON Reviews.UserID = Users.UserID

WHERE Reviews.ProductID = @ProductID

-- extract the requested page of products

SELECT ReviewID, ProductID, Name, Description, DateAdded, UserID

FROM @Review

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogAssignProductToCategory] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogAssignProductToCategory]

(@ProductID int, @CategoryID int)

AS

INSERT INTO ProductCategory (ProductID, CategoryID)

VALUES (@ProductID, @CategoryID)

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogCreateCategory] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogCreateCategory]

(@DepartmentID int,

@CategoryName nvarchar(50),

@CategoryDescription nvarchar(50))

AS

INSERT INTO Category (DepartmentID, Name, Description)

VALUES (@DepartmentID, @CategoryName, @CategoryDescription)

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogCreateProduct] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogCreateProduct]

(@CategoryID INT,

@ProductName NVARCHAR(50),

@ProductDescription NVARCHAR(MAX),

@Price MONEY,

@Thumbnail NVARCHAR(50),

@Image NVARCHAR(50),

@PromoFront BIT,

@PromoDept BIT)

AS

-- Declare a variable to hold the generated product ID

DECLARE @ProductID int

-- Create the new product entry

INSERT INTO Product

(Name,

Description,

Price,

Thumbnail,

Image,

PromoFront,

PromoDept)

VALUES

(@ProductName,

@ProductDescription,

@Price,

@Thumbnail,

@Image,

@PromoFront,

@PromoDept)

-- Save the generated product ID to a variable

SELECT @ProductID = @@Identity

-- Associate the product with a category

INSERT INTO ProductCategory (ProductID, CategoryID)

VALUES (@ProductID, @CategoryID)

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogDeleteCategory] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogDeleteCategory]

(@CategoryID int)

AS

DELETE FROM Category

WHERE CategoryID = @CategoryID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogDeleteDepartment] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogDeleteDepartment]

(@DepartmentID int)

AS

DELETE FROM Department

WHERE DepartmentID = @DepartmentID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogDeleteProduct] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogDeleteProduct]

(@ProductID int)

AS

DELETE FROM ShoppingCart WHERE ProductID=@ProductID

DELETE FROM ProductCategory WHERE ProductID=@ProductID

DELETE FROM Product where ProductID=@ProductID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogGetAllProductsInCategory] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogGetAllProductsInCategory]

(@CategoryID INT)

AS

SELECT Product.ProductID, Name, Description, Price, Thumbnail,

Image, PromoDept, PromoFront

FROM Product INNER JOIN ProductCategory

ON Product.ProductID = ProductCategory.ProductID

WHERE ProductCategory.CategoryID = @CategoryID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogGetCategoriesInDepartment] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

--missing procedures from 05-07

CREATE PROCEDURE [dbo].[CatalogGetCategoriesInDepartment]

(@DepartmentID INT)

AS

SELECT CategoryID, Name, Description

FROM Category

WHERE DepartmentID = @DepartmentID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogGetCategoriesWithoutProduct] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogGetCategoriesWithoutProduct]

(@ProductID int)

AS

SELECT CategoryID, Name

FROM Category

WHERE CategoryID NOT IN

(SELECT Category.CategoryID

FROM Category INNER JOIN ProductCategory

ON Category.CategoryID = ProductCategory.CategoryID

WHERE ProductCategory.ProductID = @ProductID)

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogGetCategoriesWithProduct] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogGetCategoriesWithProduct]

(@ProductID int)

AS

SELECT Category.CategoryID, Name

FROM Category INNER JOIN ProductCategory

ON Category.CategoryID = ProductCategory.CategoryID

WHERE ProductCategory.ProductID = @ProductID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogGetCategoryDetails] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogGetCategoryDetails]

(@CategoryID INT)

AS

SELECT DepartmentID, Name, Description

FROM Category

WHERE CategoryID = @CategoryID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogGetDepartmentDetails] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogGetDepartmentDetails]

(@DepartmentID INT)

AS

SELECT Name, Description

FROM Department

WHERE DepartmentID = @DepartmentID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogGetDepartments] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogGetDepartments] AS

SELECT DepartmentID, Name, Description

FROM Department

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogGetProductAttributeValues] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- Create CatalogGetProductAttributeValues stored procedure

CREATE PROCEDURE [dbo].[CatalogGetProductAttributeValues]

(@ProductId INT)

AS

SELECT a.Name AS AttributeName,

av.AttributeValueID,

av.Value AS AttributeValue

FROM AttributeValue av

INNER JOIN attribute a ON av.AttributeID = a.AttributeID

WHERE av.AttributeValueID IN

(SELECT AttributeValueID

FROM ProductAttributeValue

WHERE ProductID = @ProductID)

ORDER BY a.Name;

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogGetProductDetails] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogGetProductDetails]

(@ProductID INT)

AS

SELECT Name, Description, Price, Thumbnail, Image, PromoFront, PromoDept

FROM Product

WHERE ProductID = @ProductID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogGetProductRecommendations] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogGetProductRecommendations]

(@ProductID INT,

@DescriptionLength INT)

AS

SELECT ProductID,

Name,

CASE WHEN LEN(Description) <= @DescriptionLength THEN Description

ELSE SUBSTRING(Description, 1, @DescriptionLength) + '...' END

AS Description

FROM Product

WHERE ProductID IN

(

SELECT TOP 5 od2.ProductID

FROM OrderDetail od1

JOIN OrderDetail od2 ON od1.OrderID = od2.OrderID

WHERE od1.ProductID = @ProductID AND od2.ProductID != @ProductID

GROUP BY od2.ProductID

ORDER BY COUNT(od2.ProductID) DESC

)

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogGetProductsInCategory] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogGetProductsInCategory]

(@CategoryID INT,

@DescriptionLength INT,

@PageNumber INT,

@ProductsPerPage INT,

@HowManyProducts INT OUTPUT)

AS

-- declare a new TABLE variable

DECLARE @Products TABLE

(RowNumber INT,

ProductID INT,

Name NVARCHAR(50),

Description NVARCHAR(MAX),

Price MONEY,

Thumbnail NVARCHAR(50),

Image NVARCHAR(50),

PromoFront bit,

PromoDept bit)

-- populate the table variable with the complete list of products

INSERT INTO @Products

SELECT ROW\_NUMBER() OVER (ORDER BY Product.ProductID),

Product.ProductID, Name,

CASE WHEN LEN(Description) <= @DescriptionLength THEN Description

ELSE SUBSTRING(Description, 1, @DescriptionLength) + '...' END

AS Description, Price, Thumbnail, Image, PromoFront, PromoDept

FROM Product INNER JOIN ProductCategory

ON Product.ProductID = ProductCategory.ProductID

WHERE ProductCategory.CategoryID = @CategoryID

-- return the total number of products using an OUTPUT variable

SELECT @HowManyProducts = COUNT(ProductID) FROM @Products

-- extract the requested page of products

SELECT ProductID, Name, Description, Price, Thumbnail,

Image, PromoFront, PromoDept

FROM @Products

WHERE RowNumber > (@PageNumber - 1) \* @ProductsPerPage

AND RowNumber <= @PageNumber \* @ProductsPerPage

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogGetProductsOnDeptPromo] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogGetProductsOnDeptPromo]

(@DepartmentID INT,

@DescriptionLength INT,

@PageNumber INT,

@ProductsPerPage INT,

@HowManyProducts INT OUTPUT)

AS

-- declare a new TABLE variable

DECLARE @Products TABLE

(RowNumber INT,

ProductID INT,

Name NVARCHAR(50),

Description NVARCHAR(MAX),

Price MONEY,

Thumbnail NVARCHAR(50),

Image NVARCHAR(50),

PromoFront bit,

PromoDept bit)

-- populate the table variable with the complete list of products

INSERT INTO @Products

SELECT ROW\_NUMBER() OVER (ORDER BY ProductID) AS Row,

ProductID, Name, SUBSTRING(Description, 1, @DescriptionLength)

+ '...' AS Description,

Price, Thumbnail, Image, PromoFront, PromoDept

FROM

(SELECT DISTINCT Product.ProductID, Product.Name,

CASE WHEN LEN(Product.Description) <= @DescriptionLength

THEN Product.Description

ELSE SUBSTRING(Product.Description, 1, @DescriptionLength) + '...' END

AS Description, Price, Thumbnail, Image, PromoFront, PromoDept

FROM Product INNER JOIN ProductCategory

ON Product.ProductID = ProductCategory.ProductID

INNER JOIN Category

ON ProductCategory.CategoryID = Category.CategoryID

WHERE Product.PromoDept = 1

AND Category.DepartmentID = @DepartmentID

) AS ProductOnDepPr

-- return the total number of products using an OUTPUT variable

SELECT @HowManyProducts = COUNT(ProductID) FROM @Products

-- extract the requested page of products

SELECT ProductID, Name, Description, Price, Thumbnail,

Image, PromoFront, PromoDept

FROM @Products

WHERE RowNumber > (@PageNumber - 1) \* @ProductsPerPage

AND RowNumber <= @PageNumber \* @ProductsPerPage

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogGetProductsOnFrontPromo] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogGetProductsOnFrontPromo]

(@DescriptionLength INT,

@PageNumber INT,

@ProductsPerPage INT,

@HowManyProducts INT OUTPUT)

AS

-- declare a new TABLE variable

DECLARE @Products TABLE

(RowNumber INT,

ProductID INT,

Name NVARCHAR(50),

Description NVARCHAR(MAX),

Price MONEY,

Thumbnail NVARCHAR(50),

Image NVARCHAR(50),

PromoFront bit,

PromoDept bit)

-- populate the table variable with the complete list of products

INSERT INTO @Products

SELECT ROW\_NUMBER() OVER (ORDER BY Product.ProductID),

ProductID, Name,

CASE WHEN LEN(Description) <= @DescriptionLength THEN Description

ELSE SUBSTRING(Description, 1, @DescriptionLength) + '...' END

AS Description, Price, Thumbnail, Image, PromoFront, PromoDept

FROM Product

WHERE PromoFront = 1

-- return the total number of products using an OUTPUT variable

SELECT @HowManyProducts = COUNT(ProductID) FROM @Products

-- extract the requested page of products

SELECT ProductID, Name, Description, Price, Thumbnail,

Image, PromoFront, PromoDept

FROM @Products

WHERE RowNumber > (@PageNumber - 1) \* @ProductsPerPage

AND RowNumber <= @PageNumber \* @ProductsPerPage

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogMoveProductToCategory] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogMoveProductToCategory]

(@ProductID int, @OldCategoryID int, @NewCategoryID int)

AS

UPDATE ProductCategory

SET CategoryID = @NewCategoryID

WHERE CategoryID = @OldCategoryID

AND ProductID = @ProductID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogRemoveProductFromCategory] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogRemoveProductFromCategory]

(@ProductID int, @CategoryID int)

AS

DELETE FROM ProductCategory

WHERE CategoryID = @CategoryID AND ProductID = @ProductID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogUpdateCategory] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogUpdateCategory]

(@CategoryID int,

@CategoryName nvarchar(50),

@CategoryDescription nvarchar(1000))

AS

UPDATE Category

SET Name = @CategoryName, Description = @CategoryDescription

WHERE CategoryID = @CategoryID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogUpdateDepartment] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogUpdateDepartment]

(@DepartmentID int,

@DepartmentName nvarchar(50),

@DepartmentDescription nvarchar(1000))

AS

UPDATE Department

SET Name = @DepartmentName, Description = @DepartmentDescription

WHERE DepartmentID = @DepartmentID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CatalogUpdateProduct] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CatalogUpdateProduct]

(@ProductID INT,

@ProductName VARCHAR(50),

@ProductDescription VARCHAR(5000),

@Price MONEY,

@Thumbnail VARCHAR(50),

@Image VARCHAR(50),

@PromoFront BIT,

@PromoDept BIT)

AS

UPDATE Product

SET Name = @ProductName,

Description = @ProductDescription,

Price = @Price,

Thumbnail = @Thumbnail,

Image = @Image,

PromoFront = @PromoFront,

PromoDept = @PromoDept

WHERE ProductID = @ProductID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[CreateOrder] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[CreateOrder]

(@CartID char(36))

AS

/\* Insert a new record INTo Orders \*/

DECLARE @OrderID INT

INSERT INTO Orders DEFAULT VALUES

/\* Save the new Order ID \*/

SET @OrderID = @@IDENTITY

/\* Add the order details to OrderDetail \*/

INSERT INTO OrderDetail

(OrderID, ProductID, ProductName, Quantity, UnitCost)

SELECT

@OrderID, Product.ProductID, Product.Name,

ShoppingCart.Quantity, Product.Price

FROM Product JOIN ShoppingCart

ON Product.ProductID = ShoppingCart.ProductID

WHERE ShoppingCart.CartID = @CartID

/\* Clear the shopping cart \*/

DELETE FROM ShoppingCart

WHERE CartID = @CartID

/\* Return the Order ID \*/

SELECT @OrderID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[GetShoppingCartRecommendations] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <DooHee Choi>

-- Create date: <Mar 30, 2016>

-- Description: <Online Toy Car Shop>

-- =============================================

CREATE PROCEDURE [dbo].[GetShoppingCartRecommendations]

(@CartID CHAR(36),

@DescriptionLength INT)

AS

--- Returns the product recommendations

SELECT ProductID,

Name,

SUBSTRING(Description, 1, @DescriptionLength) + '...' AS Description

FROM Product

WHERE ProductID IN

(

-- Returns the products that exist in a list of orders

SELECT TOP 5 od1.ProductID AS Rank

FROM OrderDetail od1

JOIN OrderDetail od2

ON od1.OrderID=od2.OrderID

JOIN ShoppingCart sp

ON od2.ProductID = sp.ProductID

WHERE sp.CartID = @CartID

-- Must not include products that already exist in the visitor''s cart

AND od1.ProductID NOT IN

(

-- Returns the products in the specified shopping cart

SELECT ProductID

FROM ShoppingCart

WHERE CartID = @CartID

)

-- Group the ProductID so we can calculate the rank

GROUP BY od1.ProductID

-- Order descending by rank

ORDER BY COUNT(od1.ProductID) DESC

)

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[OrderGetDetails] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[OrderGetDetails]

(@OrderID INT)

AS

SELECT Orders.OrderID,

ProductID,

ProductName,

Quantity,

UnitCost,

Subtotal

FROM OrderDetail JOIN Orders

ON Orders.OrderID = OrderDetail.OrderID

WHERE Orders.OrderID = @OrderID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[OrderGetInfo] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[OrderGetInfo]

(@OrderID INT)

AS

SELECT OrderID,

(SELECT ISNULL(SUM(Subtotal), 0) FROM OrderDetail WHERE OrderID = @OrderID)

AS TotalAmount,

DateCreated,

DateShipped,

Verified,

Completed,

Canceled,

Comments,

CustomerName,

ShippingAddress,

CustomerEmail

FROM Orders

WHERE OrderID = @OrderID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[OrderMarkCanceled] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[OrderMarkCanceled]

(@OrderID INT)

AS

UPDATE Orders

SET Canceled = 1

WHERE OrderID = @OrderID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[OrderMarkCompleted] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[OrderMarkCompleted]

(@OrderID INT)

AS

UPDATE Orders

SET Completed = 1,

DateShipped = GETDATE()

WHERE OrderID = @OrderID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[OrderMarkVerified] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[OrderMarkVerified]

(@OrderID INT)

AS

UPDATE Orders

SET Verified = 1

WHERE OrderID = @OrderID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[OrdersGetByDate] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[OrdersGetByDate]

(@StartDate SMALLDATETIME,

@EndDate SMALLDATETIME)

AS

SELECT OrderID, DateCreated, DateShipped,

Verified, Completed, Canceled, CustomerName

FROM Orders

WHERE DateCreated BETWEEN @StartDate AND @EndDate

ORDER BY DateCreated DESC

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[OrdersGetByRecent] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[OrdersGetByRecent]

(@Count smallINT)

AS

-- Set the number of rows to be returned

SET ROWCOUNT @Count

-- Get list of orders

SELECT OrderID, DateCreated, DateShipped,

Verified, Completed, Canceled, CustomerName

FROM Orders

ORDER BY DateCreated DESC

-- Reset rowcount value

SET ROWCOUNT 0

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[OrdersGetUnverifiedUncanceled] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[OrdersGetUnverifiedUncanceled]

AS

SELECT OrderID, DateCreated, DateShipped,

Verified, Completed, Canceled, CustomerName

FROM Orders

WHERE Verified=0 AND Canceled=0

ORDER BY DateCreated DESC

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[OrdersGetVerifiedUncompleted] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[OrdersGetVerifiedUncompleted]

AS

SELECT OrderID, DateCreated, DateShipped,

Verified, Completed, Canceled, CustomerName

FROM Orders

WHERE Verified=1 AND Completed=0

ORDER BY DateCreated DESC

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[OrderUpdate] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[OrderUpdate]

(@OrderID INT,

@DateCreated SMALLDATETIME,

@DateShipped SMALLDATETIME = NULL,

@Verified BIT,

@Completed BIT,

@Canceled BIT,

@Comments VARCHAR(200),

@CustomerName VARCHAR(50),

@ShippingAddress VARCHAR(200),

@CustomerEmail VARCHAR(50))

AS

UPDATE Orders

SET DateCreated=@DateCreated,

DateShipped=@DateShipped,

Verified=@Verified,

Completed=@Completed,

Canceled=@Canceled,

Comments=@Comments,

CustomerName=@CustomerName,

ShippingAddress=@ShippingAddress,

CustomerEmail=@CustomerEmail

WHERE OrderID = @OrderID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[SearchCatalog] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[SearchCatalog]

(@DescriptionLength INT,

@PageNumber TINYINT,

@ProductsPerPage TINYINT,

@HowManyResults INT OUTPUT,

@AllWords BIT,

@Word1 NVARCHAR(15) = NULL,

@Word2 NVARCHAR(15) = NULL,

@Word3 NVARCHAR(15) = NULL,

@Word4 NVARCHAR(15) = NULL,

@Word5 NVARCHAR(15) = NULL)

AS

/\* @NecessaryMatches needs to be 1 for any-word searches and

the number of words for all-words searches \*/

DECLARE @NecessaryMatches INT

SET @NecessaryMatches = 1

IF @AllWords = 1

SET @NecessaryMatches =

CASE WHEN @Word1 IS NULL THEN 0 ELSE 1 END +

CASE WHEN @Word2 IS NULL THEN 0 ELSE 1 END +

CASE WHEN @Word3 IS NULL THEN 0 ELSE 1 END +

CASE WHEN @Word4 IS NULL THEN 0 ELSE 1 END +

CASE WHEN @Word5 IS NULL THEN 0 ELSE 1 END;

/\* Create the table variable that will contain the search results \*/

DECLARE @Matches TABLE

([Key] INT NOT NULL,

Rank INT NOT NULL)

-- Save matches for the first word

IF @Word1 IS NOT NULL

INSERT INTO @Matches

EXEC SearchWord @Word1

-- Save the matches for the second word

IF @Word2 IS NOT NULL

INSERT INTO @Matches

EXEC SearchWord @Word2

-- Save the matches for the third word

IF @Word3 IS NOT NULL

INSERT INTO @Matches

EXEC SearchWord @Word3

-- Save the matches for the fourth word

IF @Word4 IS NOT NULL

INSERT INTO @Matches

EXEC SearchWord @Word4

-- Save the matches for the fifth word

IF @Word5 IS NOT NULL

INSERT INTO @Matches

EXEC SearchWord @Word5

-- Calculate the IDs of the matching products

DECLARE @Results TABLE

(RowNumber INT,

[KEY] INT NOT NULL,

Rank INT NOT NULL)

-- Obtain the matching products

INSERT INTO @Results

SELECT ROW\_NUMBER() OVER (ORDER BY COUNT(M.Rank) DESC),

M.[KEY], SUM(M.Rank) AS TotalRank

FROM @Matches M

GROUP BY M.[KEY]

HAVING COUNT(M.Rank) >= @NecessaryMatches

-- return the total number of results using an OUTPUT variable

SELECT @HowManyResults = COUNT(\*) FROM @Results

-- populate the table variable with the complete list of products

SELECT Product.ProductID, Name,

CASE WHEN LEN(Description) <= @DescriptionLength THEN Description

ELSE SUBSTRING(Description, 1, @DescriptionLength) + '...' END

AS Description, Price, Thumbnail, Image, PromoFront, PromoDept

FROM Product

INNER JOIN @Results R

ON Product.ProductID = R.[KEY]

WHERE R.RowNumber > (@PageNumber - 1) \* @ProductsPerPage

AND R.RowNumber <= @PageNumber \* @ProductsPerPage

ORDER BY R.Rank DESC

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[SearchWord] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[SearchWord] (@Word NVARCHAR(50))

AS

SET @Word = 'FORMSOF(INFLECTIONAL, "' + @Word + '")'

SELECT COALESCE(NameResults.[KEY], DescriptionResults.[KEY]) AS [KEY],

ISNULL(NameResults.Rank, 0) \* 3 +

ISNULL(DescriptionResults.Rank, 0) AS Rank

FROM

CONTAINSTABLE(Product, Name, @Word,

LANGUAGE 'English') AS NameResults

FULL OUTER JOIN

CONTAINSTABLE(Product, Description, @Word,

LANGUAGE 'English') AS DescriptionResults

ON NameResults.[KEY] = DescriptionResults.[KEY]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[ShoppingCartAddItem] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE Procedure [dbo].[ShoppingCartAddItem]

(@CartID char(36),

@ProductID int,

@Attributes nvarchar(1000))

AS

IF EXISTS

(SELECT CartID

FROM ShoppingCart

WHERE ProductID = @ProductID AND CartID = @CartID)

UPDATE ShoppingCart

SET Quantity = Quantity + 1

WHERE ProductID = @ProductID AND CartID = @CartID

ELSE

IF EXISTS (SELECT Name FROM Product WHERE ProductID=@ProductID)

INSERT INTO ShoppingCart (CartID, ProductID, Attributes, Quantity, DateAdded)

VALUES (@CartID, @ProductID, @Attributes, 1, GETDATE())

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[ShoppingCartCountOldCarts] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[ShoppingCartCountOldCarts]

(@Days smallint)

AS

SELECT COUNT(CartID)

FROM ShoppingCart

WHERE CartID IN

(SELECT CartID

FROM ShoppingCart

GROUP BY CartID

HAVING MIN(DATEDIFF(dd,DateAdded,GETDATE())) >= @Days)

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[ShoppingCartDeleteOldCarts] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[ShoppingCartDeleteOldCarts]

(@Days smallint)

AS

DELETE FROM ShoppingCart

WHERE CartID IN

(SELECT CartID

FROM ShoppingCart

GROUP BY CartID

HAVING MIN(DATEDIFF(dd,DateAdded,GETDATE())) >= @Days)

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[ShoppingCartGetItems] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[ShoppingCartGetItems]

(@CartID char(36))

AS

SELECT Product.ProductID, Product.Name, ShoppingCart.Attributes, Product.Price, ShoppingCart.Quantity,Product.Price \* ShoppingCart.Quantity AS Subtotal

FROM ShoppingCart INNER JOIN Product

ON ShoppingCart.ProductID = Product.ProductID

WHERE ShoppingCart.CartID = @CartID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[ShoppingCartGetTotalAmount] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[ShoppingCartGetTotalAmount]

(@CartID char(36))

AS

SELECT ISNULL(SUM(Product.Price \* ShoppingCart.Quantity), 0)

FROM ShoppingCart INNER JOIN Product

ON ShoppingCart.ProductID = Product.ProductID

WHERE ShoppingCart.CartID = @CartID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[ShoppingCartGetTotalAmountDiscount] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[ShoppingCartGetTotalAmountDiscount]

(@CartID char(36))

AS

IF (SELECT ISNULL(SUM(Product.Price \* ShoppingCart.Quantity), 0)

FROM ShoppingCart INNER JOIN Product

ON ShoppingCart.ProductID = Product.ProductID

WHERE ShoppingCart.CartID = @CartID ) > 200

SELECT ISNULL(SUM(Product.Price \* ShoppingCart.Quantity) \* 0.10, 0)

FROM ShoppingCart INNER JOIN Product

ON ShoppingCart.ProductID = Product.ProductID

WHERE ShoppingCart.CartID = @CartID

ELSE

SELECT ISNULL(SUM(Product.Price \* ShoppingCart.Quantity), 0)

FROM ShoppingCart INNER JOIN Product

ON ShoppingCart.ProductID = Product.ProductID

WHERE ShoppingCart.CartID = @CartID

SET ANSI\_NULLS ON

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[ShoppingCartRemoveItem] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[ShoppingCartRemoveItem]

(@CartID char(36),

@ProductID int)

AS

DELETE FROM ShoppingCart

WHERE CartID = @CartID and ProductID = @ProductID

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[ShoppingCartUpdateItem] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE Procedure [dbo].[ShoppingCartUpdateItem]

(@CartID char(36),

@ProductID int,

@Quantity int)

AS

IF @Quantity <= 0

EXEC ShoppingCartRemoveItem @CartID, @ProductID

ELSE

UPDATE ShoppingCart

SET Quantity = @Quantity, DateAdded = GETDATE()

WHERE ProductID = @ProductID AND CartID = @CartID

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_DiagramPane1', @value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]

Begin DesignProperties =

Begin PaneConfigurations =

Begin PaneConfiguration = 0

NumPanes = 4

Configuration = "(H (1[40] 4[20] 2[20] 3) )"

End

Begin PaneConfiguration = 1

NumPanes = 3

Configuration = "(H (1 [50] 4 [25] 3))"

End

Begin PaneConfiguration = 2

NumPanes = 3

Configuration = "(H (1 [50] 2 [25] 3))"

End

Begin PaneConfiguration = 3

NumPanes = 3

Configuration = "(H (4 [30] 2 [40] 3))"

End

Begin PaneConfiguration = 4

NumPanes = 2

Configuration = "(H (1 [56] 3))"

End

Begin PaneConfiguration = 5

NumPanes = 2

Configuration = "(H (2 [66] 3))"

End

Begin PaneConfiguration = 6

NumPanes = 2

Configuration = "(H (4 [50] 3))"

End

Begin PaneConfiguration = 7

NumPanes = 1

Configuration = "(V (3))"

End

Begin PaneConfiguration = 8

NumPanes = 3

Configuration = "(H (1[56] 4[18] 2) )"

End

Begin PaneConfiguration = 9

NumPanes = 2

Configuration = "(H (1 [75] 4))"

End

Begin PaneConfiguration = 10

NumPanes = 2

Configuration = "(H (1[66] 2) )"

End

Begin PaneConfiguration = 11

NumPanes = 2

Configuration = "(H (4 [60] 2))"

End

Begin PaneConfiguration = 12

NumPanes = 1

Configuration = "(H (1) )"

End

Begin PaneConfiguration = 13

NumPanes = 1

Configuration = "(V (4))"

End

Begin PaneConfiguration = 14

NumPanes = 1

Configuration = "(V (2))"

End

ActivePaneConfig = 0

End

Begin DiagramPane =

Begin Origin =

Top = 0

Left = 0

End

Begin Tables =

Begin Table = "Product"

Begin Extent =

Top = 9

Left = 57

Bottom = 156

Right = 250

End

DisplayFlags = 280

TopColumn = 4

End

Begin Table = "ProductCategory"

Begin Extent =

Top = 9

Left = 307

Bottom = 114

Right = 500

End

DisplayFlags = 280

TopColumn = 0

End

End

End

Begin SQLPane =

End

Begin DataPane =

Begin ParameterDefaults = ""

End

Begin ColumnWidths = 9

Width = 284

Width = 1000

Width = 1000

Width = 1000

Width = 1000

Width = 1000

Width = 1000

Width = 1000

Width = 1000

End

End

Begin CriteriaPane =

Begin ColumnWidths = 11

Column = 1440

Alias = 900

Table = 1170

Output = 720

Append = 1400

NewValue = 1170

SortType = 1350

SortOrder = 1410

GroupBy = 1350

Filter = 1350

Or = 1350

Or = 1350

Or = 1350

End

End

End

' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'VIEW',@level1name=N'ProdsInCats'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_DiagramPaneCount', @value=1 , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'VIEW',@level1name=N'ProdsInCats'

GO

USE [master]

GO

ALTER DATABASE [BalloonShop] SET READ\_WRITE

GO