```
/*
Name: Christopher Singleton          05/29/2017




*/
/*
Note: This SQL Script shows how to create two sample Find Bike Buyer data sets.
     There are three ways to do this (vTargetMail included), although this is
          considered another way.

Purpose: To create sample bike buyer data for finding the likely bike buyer.
*/

--=============================================================================--
--============ Begin with Dropping Temp Tables/Dropping FindBikeBuyersDB ==========--
--=============================================================================--

USE [master];
GO

IF OBJECT_ID('tempdb..#TEMP1') IS NOT NULL
  DROP TABLE #TEMP1
GO
IF OBJECT_ID('tempdb..#TEMP2') IS NOT NULL
  DROP TABLE #TEMP2
GO
IF OBJECT_ID('tempdb..#DUP1') IS NOT NULL
  DROP TABLE #DUP1
GO
IF OBJECT_ID('tempdb..#DUP2') IS NOT NULL
  DROP TABLE #DUP2
GO
```

```
If EXISTS (Select * from Sysdatabases Where Name = 'FindBikeBuyersDB')
  BEGIN
    ALTER DATABASE [FindBikeBuyersDB] SET SINGLE_USER WITH ROLLBACK IMMEDIATE
    DROP DATABASE [FindBikeBuyersDB]
  END
GO

--============================================================================--
--========================== Create BikeBuyersDB ==============================--
--============================================================================--

CREATE DATABASE [FindBikeBuyersDB] ON PRIMARY
 (NAME = N'FindBikeBuyers'
 , FILENAME = N'C:\_BISolutions\Module08\FindBikeBuyersDB.mdf' --Store Database Here
 , SIZE = 10MB
 , MAXSIZE = 1GB
 , FILEGROWTH = 10MB)
 LOG ON
 (NAME = N'FindBikeBuyers_log'
 , FILENAME = N'C:\_BISolutions\Module08\FindBikeBuyersDB.LDF'--Store Log File Here
 , SIZE = 1MB
 , MAXSIZE = 1GB
 , FILEGROWTH = 10MB)
GO

--=========== Allow System Admin's to use FindBikeBuyersDB ================--

EXEC [FindBikeBuyersDB].dbo.sp_changedbowner @loginame = N'SA', @map=false
GO
USE [FindBikeBuyersDB];
GO
```

```
--=========================================================================--
--======= Migrate Data into FindBikeBuyersDB ProspectiveBuyer1 Table ======--
--=========================================================================--

--Create the Table ProspectiveBuyer1 in FindBikeBuyersDB.
--Note: I also set up the Data Types for the Columns at the same time.
CREATE TABLE ProspectiveBuyer1
      (ProspectiveBuyerKey INT PRIMARY KEY
      ,MaritalStatus NCHAR(1)
      ,Gender NVARCHAR(1)
      ,YearlyIncome MONEY
      ,NumberChildrenAtHome TINYINT
      ,Education NVARCHAR(40)
      ,NumberCarsOwned TINYINT
      )
GO
--Insert into these Columns.
INSERT INTO ProspectiveBuyer1
      (ProspectiveBuyerKey
      ,MaritalStatus
      ,Gender
      ,YearlyIncome
      ,NumberChildrenAtHome
      ,Education
      ,NumberCarsOwned
      )

--Take the Data from these Columns in the AventureWorksDW2012 Database Table ProspectiveBuyer.
SELECT ProspectiveBuyerKey
     ,MaritalStatus
     ,Gender
     ,YearlyIncome
     ,NumberChildrenAtHome
     ,Education
     ,NumberCarsOwned
FROM AdventureWorksDW2012.dbo.ProspectiveBuyer
```

```
GO
--Test our ProspectiveBuyer1 Table.
/*
SELECT *
FROM ProspectiveBuyer1
Find-Bike-Buyers
*/
--============================================================================--
--========== Create The Training Data For DataMining Model 1 ==============--
--============================================================================--
/*
Get the Derived Column for BikeBuyer (Yes or No) From the AdventureWorksDW2012 ProductCategory Table.
This includes duplicate records for each customer(CustomerKey) that has bought a Bike (BikeBuyer column:'Y')
and have bought something else and/or Not even Purchased a Bike (BikeBuyer column:'N').
Note: The Purpose of Temporary Table(s) are serve as a repository to make those changes
and then I can check the Table for what is needed without Disturbing the Database and
continue to modify the Data as needed and/or until I am satisfied with the result(s).
I can then migrate the data into the TrainingData Table(s) from the Temp Table(s).
I will Use the #Temp1 to modify, Change ProductCategoryKey 1 to Yes and 0 to No.
Note: I used INNER JOIN's to get our ProductCategoryKey and an argument to define the ProductCategoryKey Data
1 = Y and 0 = N.
*/
USE AdventureWorksDW2012;
GO
```

```
SELECT DISTINCT c.CustomerKey
          ,c.MaritalStatus
          ,c.Gender
          ,c.YearlyIncome
          ,c.NumberChildrenAtHome
          ,c.EnglishEducation
          ,c.NumberCarsOwned
          ,(CASE
              WHEN pc.ProductCategoryKey = 1
              THEN 'Y'
              ELSE 'N'
            END) AS BikeBuyer into #TEMP1
FROM DimCustomer AS c
   INNER JOIN FactInternetSales AS fis
            ON c.CustomerKey = fis.CustomerKey
   INNER JOIN DimProduct AS p
            ON fis.ProductKey = p.ProductKey
   INNER JOIN DimProductSubcategory AS ps
            ON p.ProductSubcategoryKey = ps.ProductSubCategoryKey
   INNER JOIN DimProductCategory AS pc
            ON ps.ProductCategoryKey = pc.ProductCategoryKey

--Defining the duplicate records for CustomerKey inside the #TEMP1 Table.
--Note: I placed a filter to place duplicates into the #DUP1 Table.
GO

SELECT CustomerKey, COUNT(*) AS BikeBuyer INTO #DUP1
FROM #TEMP1
GROUP BY CustomerKey
HAVING COUNT(*) > 1

GO
```

```sql
--Checking #Temp1 and #Dup1 Tables Data.
/*
SELECT *
FROM #TEMP1
SELECT *
FROM #DUP1
*/
/*
Delete Duplicates in the #TEMP1 Table Based on BikeBuyer
and the Same CustomerKey exists in the #DUP1 Table.
*/
DELETE
FROM #TEMP1
WHERE BikeBuyer = 'N'
AND CustomerKey IN (SELECT CustomerKey FROM #DUP1)
GO

/*
TrainingData1 - From DimCustomer
Note: I will use the #Temp1 table where I modified the data,
then insert the data into the TrainingData1 Table.
*/
USE FindBikeBuyersDB;
GO

--Create the TrainngData1 Table
CREATE TABLE TrainingData1
    (CustomerKey INT PRIMARY KEY
    ,MaritalStatus NCHAR(1)
    ,Gender NVARCHAR(1)
    ,YearlyIncome MONEY
    ,NumberChildrenAtHome TINYINT
    ,Education NVARCHAR(40)
    ,NumberCarsOwned TINYINT
    ,BikeBuyer NVARCHAR(255)
    )
```

```
GO

-- Insert into these Columns of the TrainingData1 Table.
INSERT INTO TrainingData1
      (CustomerKey
      ,MaritalStatus
      ,Gender
      ,YearlyIncome
      ,NumberChildrenAtHome
      ,Education
      ,NumberCarsOwned
      ,BikeBuyer
      )

--Take the Data from the Columns in the #Temp1 Table.
SELECT CustomerKey
     ,MaritalStatus
     ,Gender
     ,YearlyIncome
     ,NumberChildrenAtHome
     ,EnglishEducation
     ,NumberCarsOwned
     ,BikeBuyer
FROM #TEMP1
GO

--Test our TrainingData1 Table.
/*
Select *
From TrainingData1
*/
```

```
--=========================================================================--
--======= Migrate Data into FindBikeBuyersDB ProspectiveBuyer2 Table ======--
--=========================================================================--

/*
Please Note: This is basically the same process as Data Mining Model 1 Procedures.
Create the ProspectiveBuyer2 Table and Insert the Data into the Table From
AdvenureWorksDW2012 ProspectiveBuyer Table.
*/
USE FindBikeBuyersDB;
GO
--Create the Table ProspectiveBuyer2 with Data Types for each Column.
CREATE TABLE ProspectiveBuyer2
      (ProspectiveBuyerKey INT PRIMARY KEY
      ,FirstName NVARCHAR(50)
      ,LastName NVARCHAR(50)
      ,Gender NVARCHAR(1)
      ,TotalChildren TINYINT
      ,Region NVARCHAR(120)
      ,NumberCarsOwned TINYINT
      )
GO
```

```
--Insert into these Columns of the ProspectiveBuyer2 Table.
INSERT INTO ProspectiveBuyer2
      (ProspectiveBuyerKey
      ,FirstName
      ,LastName
      ,Gender
      ,TotalChildren
      ,Region
      ,NumberCarsOwned
      )

--Take the Data From these Columns in the Database
--AdventureWorksDW2012 ProspectiveBuyer Table.

SELECT ProspectiveBuyerKey
     ,FirstName
     ,LastName
     ,Gender
     ,TotalChildren
     ,AddressLine1
     ,NumberCarsOwned
FROM AdventureWorksDW2012.dbo.ProspectiveBuyer
GO
/* Checking ProspectiveBuyer2 Table.
SELECT *
FROM ProspectiveBuyer2
*/
```

```
--=====================================================================--
--========== Create The Training Data For DataMining Model 2 ==========--
--=====================================================================--
--Once again, I will Use the #TEMP2 Table to Modify. Change ProductCategoryKey 1 to Yes and 0 to No.
USE AdventureWorksDW2012;
GO

SELECT DISTINCT c.CustomerKey
          ,c.FirstName
          ,c.LastName
          ,c.Gender
          ,c.TotalChildren
          ,c.AddressLine1
          ,c.NumberCarsOwned
          ,DATEDIFF(yy, Birthdate, GETDATE()) -
          CASE
               WHEN (MONTH(Birthdate) > MONTH(GETDATE()))
                         OR (MONTH(Birthdate) = MONTH(GETDATE())
                         AND DAY(Birthdate) > DAY(GETDATE()))
               THEN 1
               ELSE 0
          END AS Age
          ,(CASE
               WHEN pc.ProductCategoryKey = 1
               THEN 'Y'
               ELSE 'N'
           END) AS BikeBuyer into #TEMP2
FROM DimCustomer AS c
   INNER JOIN FactInternetSales AS fis
      ON c.CustomerKey = fis.CustomerKey
   INNER JOIN DimProduct AS p
            ON fis.ProductKey = p.ProductKey
   INNER JOIN DimProductSubcategory AS ps
            ON p.ProductSubcategoryKey = ps.ProductSubCategoryKey
   INNER JOIN DimProductCategory AS pc
            ON ps.ProductCategoryKey = pc.ProductCategoryKey
```

```
--Defining the duplicate records for CustomerKey. I will put the duplicate information into #DUP2.
--Note: I will use our #DUP2 Table to define which ones to delete in our #TEMP2 Table.
GO
SELECT CustomerKey, COUNT(*) AS BikeBuyer into #DUP2
FROM #TEMP2
GROUP BY CustomerKey
HAVING COUNT(*) > 1

--Delete matching duplicates From the #TEMP2 that match in #DUP2.
GO
DELETE
FROM #TEMP2
WHERE BikeBuyer = 'N' AND CustomerKey IN (SELECT CustomerKey FROM #DUP2)
GO

--Checking #TEMP2 and #DUP2 Tables Data.
/*
SELECT *
FROM #TEMP2
SELECT *
FROM #DUP2
*/
```

```
--=============================================================================--
--================= Create the Training Data For TrainingData2 ==============--
--=============================================================================--

USE FindBikeBuyersDB;
GO

--Create TrainingData2 Table, then migrate the Data From the #TEMP2 Table.
CREATE TABLE TrainingData2
      (CustomerKey INT PRIMARY KEY
      ,FirstName NVARCHAR(50)
      ,LastName NVARCHAR(50)
      ,Gender NVARCHAR(1)
      ,Age INTEGER
      ,TotalChildren TINYINT
      ,Region NVARCHAR(120)
      ,NumberCarsOwned TINYINT
      ,BikeBuyer NVARCHAR(255)
      )
GO

--Insert into these Columns of the TrainingData2 Table.
INSERT INTO TrainingData2
      (CustomerKey
      ,FirstName
      ,LastName
      ,Gender
      ,Age
      ,TotalChildren
      ,Region
      ,NumberCarsOwned
      ,BikeBuyer
      )
```

```
--Select the Columns to Insert From the #TEMP2 Table.
SELECT DISTINCT CustomerKey
		,FirstName
		,LastName
		,Gender
		,Age
		,TotalChildren
		,AddressLine1
		,NumberCarsOwned
		,BikeBuyer
FROM #TEMP2
--Check TrainingData2 (Rows 18484) matches the Customer Table.
/*
SELECT *
FROM TrainingData2
*/
GO

USE FindBikeBuyersDB;
GO
```

```
IF EXISTS(SELECT 1 FROM sys.views WHERE NAME='vAllTrainingData' and TYPE='v')
DROP VIEW vAllTrainingData;
GO
CREATE VIEW vAllTrainingData
AS
/*
Created By: Chris Singleton
Purpose: View to use as Training Data.
Date: 05/29/2017
*/
SELECT t1.[CustomerKey]
     ,t1.[MaritalStatus]
     ,t1.[Gender]
     ,t2.[Age]
     ,t1.[YearlyIncome]
     ,t1.[NumberChildrenAtHome]
     ,t1.[Education]
     ,t1.[NumberCarsOwned]
     ,t1.[BikeBuyer]
     ,t2.[TotalChildren]
     ,t2.[Region]
     ,t2.[FirstName]
     ,t2.[LastName]

FROM [FindBikeBuyersDB].[dbo].[TrainingData1] AS t1
    INNER JOIN [FindBikeBuyersDB].[dbo].[TrainingData2] AS t2
          ON t1.CustomerKey = t2.CustomerKey
GO
```

```
--=========================================================================--
--=================== Review the results of this script ====================--
--=========================================================================--

SELECT 'FindBikeBuyersDB Database Filled' AS FindBikeBuyersDB
SELECT [TableName] = '[dbo].[ProspectiveBuyer1]' , [RowCount] = Count(*) from [dbo].[ProspectiveBuyer1]
SELECT [TableName] = '[dbo].[ProspectiveBuyer1]' , [RowCount] = Count(*) from [dbo].[ProspectiveBuyer2]
SELECT [TableName] = '[dbo].[TrainingData1]' , [RowCount] = Count(*) from [dbo].[TrainingData1]
SELECT [TableName] = '[dbo].[TrainingData2]' , [RowCount] = Count(*) from [dbo].[TrainingData2]

/*

USE FindBikeBuyersDB;
SELECT * FROM [FindBikeBuyersDB].[dbo].[ProspectiveBuyer1]
SELECT * FROM [FindBikeBuyersDB].[dbo].[ProspectiveBuyer2]
SELECT * FROM [FindBikeBuyersDB].[dbo].[TrainingData1]
SELECT * FROM [FindBikeBuyersDB].[dbo].[TrainingData2]

*/

----This Concludes the SQL Code I Created to Complete the Database with FindBikeBuyersDB Training Data---
```