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




*/
/*
Note: This SQL Script shows how to creates two sample data sets to Find Bike Buyers.
    There are three ways to do this (vTargetMail one of them), although this script
          that I created is considered the preferred way.

Purpose: To create sample bike buyer data sets 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
```

```
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
*/
--==============================================================================================--
--========== 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
```

```
-- I think this is what you intended to see.
SELECT c.CustomerKey
    ,c.MaritalStatus
    ,c.Gender
    ,c.YearlyIncome
    ,c.NumberChildrenAtHome
    ,c.EnglishEducation
    ,c.NumberCarsOwned
    ,(CASE
        WHEN b.CustomerKey IS NULL
        THEN 'N'
        ELSE 'Y'
     END) AS BikeBuyer INTO #TEMP1
FROM DimCustomer AS c
    LEFT JOIN (SELECT DISTINCT fis.CustomerKey
            FROM FactInternetSales AS fis
                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
            WHERE pc.ProductCategoryKey = 1) AS b
                  ON c.CustomerKey = b.CustomerKey
GO
```

```sql
USE FindBikeBuyersDB
GO
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)
       )

INSERT INTO TrainingData1
       (CustomerKey
       ,MaritalStatus
       ,Gender
       ,YearlyIncome
       ,NumberChildrenAtHome
       ,Education
       ,NumberCarsOwned
       ,BikeBuyer
       )

SELECT CustomerKey
      ,MaritalStatus
      ,Gender
      ,YearlyIncome
      ,NumberChildrenAtHome
      ,EnglishEducation
      ,NumberCarsOwned
      ,BikeBuyer
FROM #TEMP1
```

```
--=========================================================================--
--======= 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
```

```sql
--=============================================================--
--========== Create the Training Data for DataMining Model 2 ==============--
--=============================================================--
USE [AdventureWorksDW2012]; GO
SELECT c.CustomerKey
     ,c.FirstName
     ,c.LastName
     ,c.Gender
    ,c.TotalChildren
    ,c.AddressLine1
    ,c.NumberCarsOwned
    ,c.BirthDate
    ,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 b.CustomerKey IS NULL
    THEN 'N'
    ELSE 'Y'
   END) AS BikeBuyer INTO #TEMP2
FROM DimCustomer AS c
  LEFT JOIN (SELECT DISTINCT fis.CustomerKey
      FROM FactInternetSales AS fis
        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
      WHERE pc.ProductCategoryKey = 1) AS b
          ON c.CustomerKey = b.CustomerKey
```

```
GO
/*
SELECT * FROM #TEMP2
*/
USE FindBikeBuyersDB;
GO
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)
    )
```

```
INSERT INTO TrainingData2
	(CustomerKey
	,FirstName
	,LastName
	,Gender
	,Age
	,TotalChildren
	,Region
	,NumberCarsOwned
	,BikeBuyer
	)

SELECT 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
```

```sql
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 All 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].[ProspectiveBuyer2]' , [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---
```