**/\*ALL THE CODE for creation and maintenance of the Advertising Network database**

**\*/**

--Drop tables on Advertising Network database for repeatability

IF OBJECT\_ID('AdPayment') IS NOT NULL

DROP TABLE AdPayment;

go

IF OBJECT\_ID('ContractAdvertisement') IS NOT NULL

DROP TABLE ContractAdvertisement;

go

IF OBJECT\_ID('ContractStore') IS NOT NULL

DROP TABLE ContractStore;

go

IF OBJECT\_ID('Contract') IS NOT NULL

DROP TABLE Contract;

go

IF OBJECT\_ID('Advertisement') IS NOT NULL

DROP TABLE Advertisement;

go

IF OBJECT\_ID('AdPaymentType') IS NOT NULL

DROP TABLE AdPaymentType;

go

IF OBJECT\_ID('Advertiser') IS NOT NULL

DROP TABLE Advertiser;

go

IF OBJECT\_ID('AdvertisingType') IS NOT NULL

DROP TABLE AdvertisingType;

go

IF OBJECT\_ID('SalesRep') IS NOT NULL

DROP TABLE SalesRep;

go

IF OBJECT\_ID('Store') IS NOT NULL

DROP TABLE Store;

go

IF OBJECT\_ID('Chain') IS NOT NULL

DROP TABLE Chain;

go

IF OBJECT\_ID('Region') IS NOT NULL

DROP TABLE Region;

go

**--Creating tables for the Advertising Network database**

-- Creating the Region lookup table

CREATE TABLE Region

--Columns for the Region table

(

RegionID int Identity NOT NULL,

RegionName varchar(30) NOT NULL,

--Constraints on the Region table

CONSTRAINT RegionPK PRIMARY KEY (RegionID)

);

--End creating the Region table

-- Creating the Chain lookup table

CREATE TABLE Chain

-- Columns for the Chain table

(

ChainID int Identity NOT NULL,

ChainName varchar(30) NOT NULL,

--Constraints on the Chain table

CONSTRAINT ChainPK PRIMARY KEY (ChainID)

);

--End creating the Chain table

-- Creating the Store table

CREATE TABLE Store

-- Columns for the Store table

(

StoreID int Identity NOT NULL,

StoreName varchar(30) NOT NULL,

StoreAddressLine1 varchar(30),

StoreCity varchar(30),

StoreState varchar(30),

StorePostalcode varchar(30),

ChainID int NOT NULL,

RegionID int NOT NULL,

--Constraints on the Store table

CONSTRAINT StorePK PRIMARY KEY (StoreID),

CONSTRAINT StoreU1 UNIQUE (StoreName),

CONSTRAINT StoreFK1 FOREIGN KEY (ChainID) REFERENCES Chain(ChainID),

CONSTRAINT StoreFK2 FOREIGN KEY (RegionID) REFERENCES Region(RegionID),

);

--End creating the Store table

-- Creating the SalesRep table

CREATE TABLE SalesRep

-- Columns for the SalesRep table

(

SalesRepID int Identity NOT NULL,

SalesRepFirstName varchar(30) NOT NULL,

SalesRepLastName varchar(30) NOT NULL,

--Constraints on the SalesRep table

CONSTRAINT SalesRepPK PRIMARY KEY (SalesRepID),

);

--End creating the SalesRep table

-- Creating the AdvertisingType lookup table

CREATE TABLE AdvertisingType

-- Columns for the AdvertisingType table

(

AdvertisingTypeID int Identity NOT NULL,

AdvertisingTypeDesc varchar(20),

--Constraints on the AdvertisingType table

CONSTRAINT AdvertisingTypePK PRIMARY KEY (AdvertisingTypeID)

);

--End creating the AdvertisingType table

-- Creating the Advertiser table

CREATE TABLE Advertiser

-- Columns for the Advertiser table

(

AdvertiserID int Identity NOT NULL,

AdvertiserName varchar(30) NOT NULL,

AdvertiserAddressLine1 varchar(30),

AdvertiserAddressLine2 varchar(30),

AdvertiserCity varchar(30),

AdvertiserState varchar(30),

AdvertiserPostalcode varchar(30),

EmailAddress varchar(30),

PhoneNumber varchar(30),

WebsiteURL varchar(30),

AdvertisingTypeID int NOT NULL,

SalesRepID int NOT NULL,

--Constraints on the Advertiser table

CONSTRAINT AdvertiserPK PRIMARY KEY (AdvertiserID),

CONSTRAINT AdvertiserU1 UNIQUE (AdvertiserName),

CONSTRAINT AdvertiserFK1 FOREIGN KEY (AdvertisingTypeID) REFERENCES AdvertisingType(AdvertisingTypeID),

CONSTRAINT AdvertiserFK2 FOREIGN KEY (SalesRepID) REFERENCES SalesRep(SalesRepID),

);

--End creating the Advertiser table

-- Creating the AdPaymentType lookup table

CREATE TABLE AdPaymentType

-- Columns for the AdPaymentType table

(

AdPaymentTypeID int Identity NOT NULL,

AdPaymentTypeDesc varchar(20),

--Constraints on the AdPaymentType table

CONSTRAINT AdPaymentTypePK PRIMARY KEY (AdPaymentTypeID)

);

--End creating the AdPaymentType table

-- Creating the Advertisement table

CREATE TABLE Advertisement

-- Columns for the Advertisement table

(

AdvertisementID int Identity NOT NULL,

AdvertisementName varchar(30) NOT NULL,

AdvertisementDuration int NOT NULL,

AdvertiserID int NOT NULL,

--Constraints on the Advertisement table

CONSTRAINT AdvertisementPK PRIMARY KEY (AdvertisementID),

CONSTRAINT AdvertisementU1 UNIQUE (AdvertisementName),

CONSTRAINT AdvertisementFK1 FOREIGN KEY (AdvertiserID) REFERENCES Advertiser(AdvertiserID),

);

--End creating the Advertisement table

-- Creating the Contract table

CREATE TABLE Contract

-- Columns for the Contract table

(

ContractID int Identity NOT NULL,

ContractStartDate datetime NOT NULL,

ContractTotal int,

AdvertiserID int NOT NULL,

SalesRepID int NOT NULL,

AdvertisementID int NOT NULL,

--Constraints on the Contract table

CONSTRAINT ContractPK PRIMARY KEY (ContractID),

CONSTRAINT ContractFK1 FOREIGN KEY (AdvertiserID) REFERENCES Advertiser(AdvertiserID),

CONSTRAINT ContractFK2 FOREIGN KEY (SalesRepID) REFERENCES SalesRep(SalesRepID),

CONSTRAINT ContractFK3 FOREIGN KEY (AdvertisementID) REFERENCES Advertisement(AdvertisementID),

);

--End creating the Contract table

-- Creating the ContractStore lookup table

CREATE TABLE ContractStore

-- Columns for the ContractStore table

(

ContractStoreID int Identity NOT NULL,

ContractID int NOT NULL,

StoreID int NOT NULL,

--Constraints on the ContractStore table

CONSTRAINT ContractStorePK PRIMARY KEY (ContractStoreID),

CONSTRAINT ContractStoreFK1 FOREIGN KEY (ContractID) REFERENCES Contract(ContractID),

CONSTRAINT ContractStoreFK2 FOREIGN KEY (StoreID) REFERENCES Store(StoreID),

);

--End creating the ContractStore table

-- Creating the ContractAdvertisement table

CREATE TABLE ContractAdvertisement

-- Columns for the ContractAdvertisement table

(

ContractAdvertisementID int Identity NOT NULL,

AdStartDate datetime NOT NULL,

AdEndDate datetime NOT NULL,

AdFee int NOT NULL,

ContractStoreID int NOT NULL,

AdvertisementID int

--Constraints on the ContractAdvertisement table

CONSTRAINT ContractAdvertisementPK PRIMARY KEY (ContractAdvertisementID),

CONSTRAINT AdFee UNIQUE (AdFee),

CONSTRAINT ContractAdvertisementFK1 FOREIGN KEY (ContractStoreID) REFERENCES ContractStore(ContractStoreID),

CONSTRAINT ContractAdvertisementFK2 FOREIGN KEY (AdvertisementID) REFERENCES Advertisement(AdvertisementID),

);

--End creating the ContractAdvertisement table

-- Creating the AdPayment table

CREATE TABLE AdPayment

-- Columns for the AdPayment table

(

AdPaymentID int Identity NOT NULL,

BillingAddressLine1 varchar(30),

BillingAddressLine2 varchar(30),

BillingCity varchar(30),

BillingState varchar(30),

BillingPostalcode varchar(30),

AdPaymentAmount int,

ContractAdvertisementID int,

AdvertiserID int,

AdPaymentTypeID int

--Constraints on the AdPayment table

CONSTRAINT AdPaymentPK PRIMARY KEY (AdPaymentID),

CONSTRAINT AdPaymentFK1 FOREIGN KEY (ContractAdvertisementID) REFERENCES ContractAdvertisement(ContractAdvertisementID),

CONSTRAINT AdPaymentFK2 FOREIGN KEY (AdvertiserID) REFERENCES Advertiser(AdvertiserID),

CONSTRAINT AdPaymentFK3 FOREIGN KEY (AdPaymentTypeID) REFERENCES AdPaymentType(AdPaymentTypeID),

);

--End creating the AdPayment table

**-- Advertising Network INSERT statements**

-- Inserting data into the Advertising Type lookup table

INSERT INTO AdvertisingType(AdvertisingTypeDesc)

VALUES ('Local'), ('National'), ('Regional'), ('In-Store')

SELECT \* FROM AdvertisingType

-- Inserting data into the SalesRep lookup table

INSERT INTO SalesRep(SalesRepFirstName, SalesRepLastName)

VALUES ('Liz', 'Wilson')

, ('Josh', 'Hanas')

, ('Carlos', 'Rodriguez')

, ('Samantha', 'Jones')

, ('Bob', 'Smith')

SELECT \* FROM SalesRep

-- Inserting data into the Advertiser table

INSERT INTO Advertiser(AdvertiserName, AdvertiserAddressLine1, AdvertiserAddressLine2, AdvertiserCity,

AdvertiserState, AdvertiserPostalcode, EmailAddress, PhoneNumber, WebsiteURL, AdvertisingTypeID, SalesRepID)

VALUES ('PepsiCo', '700 AndersonHill Road', 'Suite 100', 'Purchase', 'NY', '10577', 'hello@craft.co', '914-555-7689'

, 'www.craft.co', '2', '1')

, ('White Plains Auto', '420 Tarrytown Road', 'null', 'White Plains', 'NY', '10607', 'contact@autozone.com', '914-946-4880'

, 'www.autozone.com', '1', '3')

, ('Rand Realty', '222 Martins Avenue', 'null', 'Philadelphia', 'PA', '19093', 'cedric@randrealty.com', '215-645-8883'

, 'www.randrealty.com', '3', '4')

, ('Frito Lay', '10 Cheetos Drive', 'null', 'Plano', 'TX', '75024', 'helpme@fritolay.com', '972-334-7000'

, 'www.fritolay.com', '2', '2')

, ('White Plains Chiropractor', '1111 Crack Bones Road', 'null', 'White Plains', 'NY', '10608', 'crackme@wpchiro.com', '914-834-9990'

, 'www.wpchiro.com', '1', '3')

, ('The Kroger Co', '1014 Vine Street', 'Suite 100', 'Cincinnati', 'OH', '45202', 'emailus@thekrogerco.com', '513-762-4000'

, 'www.thekrogerco.com', '4', '1')

--Taking a look at all my lovely Advertisers

SELECT \* FROM Advertiser

-- Inserting data into the Advertisement table

INSERT INTO Advertisement(AdvertisementName, AdvertisementDuration, AdvertiserID)

VALUES ('Quaker Oats Heart Health', '15', '1')

, ('White Plains Auto Store', '30', '2')

, ('Rand Realtors', '30', '3')

, ('Pepsi', '30', '1')

, ('Cheetos', '15', '4')

, ('Lays Chips', '15', '4')

, ('WP Chiropractor', '30', '5')

, ('Kroger Brand Cookies', '30', '6')

, ('Simple Truth Organic Milk', '15', '6')

, ('Private Selection Olive Oil', '30', '6')

--Taking a look at all the lovely ads

SELECT \* FROM Advertisement

-- Inserting data into the Contract table

INSERT INTO Contract(ContractStartDate, ContractTotal, AdvertiserID, SalesRepID, AdvertisementID)

VALUES ('6/1/2019', '2500', '1', '1', '1')

, ('4/15/2019', '750', '2', '3', '2')

, ('1/1/2019', '1500', '3', '4', '3')

, ('3/18/2019', '3000', '1', '1', '4')

, ('12/1/2018', '2500', '4', '2', '5')

, ('12/1/2018', '2500', '4', '2', '6')

, ('3/15/2019', '750', '5', '3', '7')

, ('5/5/2019', '3500', '6', '1', '8')

, ('5/5/2019', '3500', '6', '1', '9')

, ('6/1/2019', '3500', '6', '1', '10')

--Taking a look at all the juicy contracts

SELECT \* FROM Contract

-- Inserting data into the Chain lookup table

INSERT INTO Chain(ChainName)

VALUES ('Kroger'), ('Fresh Grocer'), ('Publix'), ('Earth Fare'), ('SRS North')

--Let's see those Chains

SELECT \* FROM Chain

-- Inserting data into the Region lookup table

INSERT INTO Region(RegionName)

VALUES ('Philadelphia'), ('Maryland'), ('New York')

-- Where are my Regions?

SELECT \* FROM Region

-- Inserting data into the Store table

INSERT INTO Store(StoreName, StoreAddressLine1, StoreCity, StoreState, StorePostalcode, ChainID, RegionID)

VALUES ('Fresh Grocer of Walnut Street', '4001 Walnut Street', 'Philadelphia', 'PA', '19104', '2', '1')

, ('SRS White Plains', '12 City Place', 'White Plains', 'NY', '10601', '5', '3')

, ('Kroger Howard Park', '5601 Liberty Heights Road', 'Baltimore', 'MD', '21207', '1', '2')

--What are our Stores?

SELECT \* FROM Store

-- Inserting data into the ContractStore lookup table

INSERT INTO ContractStore(ContractID, StoreID)

VALUES ('1', '1')

, ('1', '2')

, ('1', '3')

, ('2', '2')

, ('3', '1')

, ('4', '1')

, ('4', '2')

, ('4', '3')

, ('5', '1')

, ('5', '3')

, ('6', '1')

, ('6', '2')

, ('7', '2')

, ('8', '1')

, ('8', '3')

, ('9', '1')

, ('9', '3')

, ('10', '1')

, ('10', '3')

--Let's see which stores have contracts

SELECT \* FROM ContractStore

**-- Advertising Network UPDATE and DELETE statements**

/\*The Sales Team have decided that a more appropriate name for 'In-Store' ads

should be used as they do not use this terminology

\*/

--First take a look at the Advertising Type descriptions to find the right one

SELECT \* FROM AdvertisingType

--Now update the description 'In-Store' so that it now reads 'House'

UPDATE AdvertisingType SET AdvertisingTypeDesc = 'House' WHERE AdvertisingTypeID = '4'

SELECT \* FROM AdvertisingType WHERE AdvertisingTypeID = '4'

--We have an underperformer!

SELECT \* FROM SalesRep

DELETE SalesRep WHERE SalesRepFirstName = 'Bob' AND SalesRepLastName = 'Smith'

SELECT \* FROM SalesRep

**-- Advertising Network ANSWERING DATA QUESTIONS**

--DATA QUESTION #1

/\*

Create a view that shows which stores are running ads with a total runtime of less

than 4 minutes

\*/

--First drop the view

DROP VIEW StoresWithInventory

GO

--Now we can write it!

CREATE VIEW StoresWithInventory AS

SELECT Store.StoreName

, SUM(Advertisement.AdvertisementDuration) AS TotalInventoryDuration

FROM Advertisement

JOIN Contract ON Advertisement.AdvertisementID = Contract.AdvertisementID

JOIN ContractStore ON Contract.ContractID = ContractStore.ContractID

JOIN Store ON ContractStore.StoreID = Store.StoreID

GROUP BY Store.StoreName

HAVING SUM(Advertisement.AdvertisementDuration) < 240

GO

--Now let's see which stores have some inventory left to sell

SELECT \* FROM StoresWithInventory ORDER BY TotalInventoryDuration DESC

--DATA QUESTION #2 - Which Advertisers are spending the most on the network

/\*

Create a view that shows which advertisers have the highest number of advertising

contracts

\*/

--First drop the view

DROP VIEW AdvertiserContracts

GO

--Now we can write it!

CREATE VIEW AdvertiserContracts AS

SELECT Advertiser.AdvertiserName

, SUM(Contract.ContractTotal) AS TotalContractAmount

FROM Contract

JOIN Advertiser ON Contract.AdvertiserID = Advertiser.AdvertiserID

GROUP BY Advertiser.AdvertiserName

GO

--Now let's see which Advertiser has spent the most money!

SELECT \* FROM AdvertiserContracts ORDER BY TotalContractAmount DESC

--DATA QUESTION #3 - Which stores in the network have the most valuable inventory ?

--First drop the view

DROP VIEW StoresWithPremiumSlots

GO

--Now we can write it!

CREATE VIEW StoresWithPremiumSlots AS

SELECT Store.StoreName

, SUM(Advertisement.AdvertisementDuration) AS TotalInventoryDuration

FROM Advertisement

JOIN Contract ON Advertisement.AdvertisementID = Contract.AdvertisementID

JOIN ContractStore ON Contract.ContractID = ContractStore.ContractID

JOIN Store ON ContractStore.StoreID = Store.StoreID

GROUP BY Store.StoreName

HAVING SUM(Advertisement.AdvertisementDuration) BETWEEN 180 AND 240

GO

--Now show me the money!

SELECT \* FROM StoresWithPremiumSlots ORDER BY TotalInventoryDuration DESC

--DATA QUESTION #4 - Which Chain or Region has the most advertisements sold?

--First drop the view

DROP VIEW MostProlificChain

GO

--Now we can write it!

CREATE VIEW MostProlificChain AS

SELECT Chain.ChainName

, Region.RegionName

, COUNT(Contract.ContractID) AS TotalContracts

, SUM(Contract.ContractTotal) AS TotalContractAmount

FROM Contract

JOIN ContractStore ON Contract.ContractID = ContractStore.ContractID

JOIN Store ON ContractStore.StoreID = Store.StoreID

JOIN Chain ON Store.ChainID = Chain.ChainID

JOIN Region ON Store.RegionID = Region.RegionID

GROUP BY Chain.ChainName, Region.RegionName

GO

--Now let's see who is the most prolific!

SELECT \* FROM MostProlificChain ORDER BY TotalContracts DESC

/\*

DATA QUESTION #5 - Which sales rep is the top performer?

Create a Function to count the value of contracts sold by a given SalesRep

\*/

--First, drop the function

DROP FUNCTION dbo.ContractSold

go

--Now create the function

CREATE FUNCTION dbo.ContractSold(@SalesRepID int)

RETURNS int AS

BEGIN

DECLARE @ContractNumber int

/\* Get the count of contracts for the provided SalesRepID

and assign that value to @ContractNumber

\*/

SELECT @ContractNumber = SUM(ContractTotal) FROM Contract

WHERE Contract.SalesRepID = @SalesRepID

RETURN @ContractNumber

END

go

--Now Return the sales rep with the highest value of contracts sold in descending order

/\* Well as I used a function for this query then I guess I need to wrap my SELECT statement

for results into a VIEW to make my report easier to create :)

\*/

--Of course for repeatability I need to drop the view first

DROP VIEW MostProlificSalesRep

GO

--Now I create the view

CREATE VIEW MostProlificSalesRep AS

SELECT TOP 3

\*

, dbo.ContractSold(SalesRepID) as ContractSold

FROM SalesRep

GO

SELECT \* FROM MostProlificSalesRep ORDER BY ContractSold DESC