# The Advertising Network

#### A Database from Design through Implementation

TABLE OF CONTENTS

[The Advertising Network 1](#_Toc11534918)

[Project Summary 3](#_Toc11534919)

[Stakeholder List 4](#_Toc11534920)

[THE DATA MODEL 4](#_Toc11534921)

[Data Questions 8](#_Toc11534922)

[Relationships 9](#_Toc11534923)

[Conceptual Model Diagram 10](#_Toc11534924)

[Logical Model Diagram 11](#_Toc11534925)

[PHYSICAL DATABASE DESIGN 12](#_Toc11534926)

[ANSWERING THE DATA QUESTIONS 33](#_Toc11534927)

[RAW DATA SAMPLES 41](#_Toc11534928)

[IMPLEMENTATION 45](#_Toc11534929)

[REFLECTION 51](#_Toc11534930)

## Project Summary

The Advertising Network is a digital video network that reaches consumers through the use of high-definition television screens located within grocery stores. The Advertising Network (hereafter referred to as the ‘Ad Network’) combines video content from several different sources to provide informative and engaging advertising and other programming to shoppers as they browse the aisles. The digital videos displayed deliver an advertising ‘loop’ containing advertisements (ads) of varying lengths. Each ad plays once in rotation until all ads have been delivered and then the loop starts over. Timing of the ‘loop’ varies by store location and by aisle. Ads consist of in-house product specials, local business advertising and national brand campaigns. Ads are sold across the entire network to all stores (national ads), individual stores (local ads) or sub-sections within the network (program specific by chain).

The purpose of this database project is to address the fact that as the Advertising Network grows, and includes an increasing number of stores and advertisements, it becomes more complex to account for each piece of content that is running on each store at any given time. From a sales perspective, while advertising delivery is dynamic and delivered according to the content that is available at each store, the sales team sell advertisers on a 4-minute “loop”. This allows them to provide a guarantee to each advertiser that their ad will be displayed every 4 minutes on the screens. The creation of this database helps solve the question of how to keep up with the ads that are running and the inventory space that is left available at each store, so that the sales team know both what can be upsold as “premium” advertising and also when they have reached advertising capacity. The data questions answered focus exclusively on what would have the most value to the sales team and 2 out of the 5 questions feature the role of inventory, as it would seem to have the most potential value implementing this database model. As sales would likely heavily use the system, I was conscious during Phase 2 of keeping things as user-friendly as possible. I decided to revise my original naming scheme in the logical model from using underscores to an Uppercase/Lowercase system as this has more readability. My logical diagram was therefore revised. The data questions themselves where formulated almost entirely using Views and were predominantly based on the Contracts and Advertisements tables. To implement the database into a more user-friendly format I ultimately used MS Access as it was fairly straightforward to source each report using the View that had been created. I also used Access to incorporate two examples of input forms, which would make it easier for a user of the database to create a new Advertiser, along with contract and advertisements. I created a similar form for inputting information for a new store to be added to the appropriate chain.

## Stakeholder List

The stakeholders identified for this database system are as follows:

* Advertising Operations Manager
* Sales Representatives
* Sales Managers
* Advertising Customers
* Advertising Network business owners
* Systems Operations technicians

## THE DATA MODEL

The following actors and resources are identified for this database model:

###### Store

A grocery store that contains digital display screens included in the Advertising Network. Attributes for each store include:

* **Store name**

*The name assigned to the grocery store location.*

* **Location address**

*The street address where the grocery store is located.*

* **Phone number**

*The telephone number where someone can be contacted inside the grocery store.*

* **Chain**

*Within the network, stores may be grouped together according to chain. Chain name refers to the chain assigned to the specified grocery store location.*

* **Region**

*The geographical area assigned to the store. Region may represent a city, such as New York City, or a state, such as New Jersey. The region a store is located in may determine the advertising rate that can be charged, as some regions are more valuable to advertisers than others.*

###### Advertiser

A business or individual that wants to display their product or service via digital advertisement on the network. Advertisers may represent any of the following: a local business, a national brand, or a company representing one or more consumer-packaged goods (CPG) products. Attributes for each Advertiser include:

* **Advertiser name**

*The name of the company that wishes to advertise.*

* **Mailing address**

*The street address where the advertiser can be contacted.*

* **Advertiser phone number**

*The telephone number specified as the main contact number for the advertiser.*

* **Contact Email address**

*The email address specified as the contact email address for the advertiser.*

* **Advertising Type**

*This refers to the type of advertising the advertiser represents. Type is selected from one of the following options: “Local”, “National”, “Regional”, “In-Store”\**

*\*(this has been updated from deliverable 1 as we will be changing back during an Update statement!)*

###### Advertisement

An item of digital video content that describes an Advertiser’s product or service. Advertisements are either 15 or 30 seconds in length. Advertisements may be any of the following: local business advertisements, regional product advertisements, regional business advertisements, national product advertisements, chain-specific in-store product items, network-specific in-store product items, and network-specific content programming, such as recipes or wellness tips. The Advertisement Entity has the following attributes:

* **Advertisement name**

*The name given to the piece of advertising video content.*

* **Duration**

*The length of time the ad runs provided in seconds. Duration is selected from the following options: “15 seconds”, “30 seconds”.*

* **Advertiser Name**

*The name of the advertiser that the advertisement is representing.*

* **Start Date**

*The date on which the advertisement is scheduled to start playing on the network.*

* **End Date**

*The date on which the advertisement is to stop playing on the network.*

###### Contract

A document representing the terms of the agreement for an Advertiser to deliver Advertisement(s) on the Network. Attributes for the contract are:

* **Contract Number**

*The identification number assigned to the contract document.*

* **Advertiser Name**

*The name of the Advertiser wishing to deliver advertisements on the Advertising Network.*

* **Ad Name**

*The name of the digital video advertisement(s) included on the contract.*

* **Ad Start Date**

*The date on which the advertiser would like the advertisement(s) to start playing on the Advertising Network.*

* **Ad End Date**

*The date on which the advertiser would like the advertisement(s) to start playing on the Advertising Network.*

* **Ad Fee**

*The agreed fee to be paid per Advertisement by the Advertiser for delivery on the Advertising Network per this contract.*

* **Contract Total**

*The total amount contracted for delivery of the advertisement(s) on the Advertising Network. May include setup fees and other fees, in addition to the Ad Fee.*

* **Store Names**

*The name of the store(s) that the advertisement(s) will run on for the duration of the contract.*

* **Sales Rep**

*The name of the sales person responsible for negotiating the terms of the Contract with the Advertiser.*

###### Ad Payment

Payments that are paid by Advertisers for Advertising delivery will be recorded and tracked in this database system. The Advertising Payment entity will include the following:

* **Payment Number**

*The identification number assigned to the payment.*

* **Advertiser name**

*The name of the company making the advertising payment.*

* **Billing address**

*The street address of the advertiser that is used for billing purposes.*

* **Payment Amount**

*The dollar amount of the payment for advertising. May be partial or total contract amount as specified by the Contract Number.*

* **Payment Type**

*The type of payment made by the Advertiser. Type is selected from the following: “Check”, “Credit Card.”*

* **Contract Number**

*The identification number of the contract document the payment is in reference to.*

###### Advertising Inventory

Advertising Inventory represents the advertisements that are run on each store within the framework of time duration. Advertisers are provided a guarantee that their ad will be displayed every 4 minutes, and so each store may display up to 4 minutes (240 seconds) of advertising content at any one time.

In addition, Sales wishes to calculate which stores have what is considered “premium” inventory. These are the time slots that are available once a minimum threshold of 3 minutes has been reached, but not to exceed the 4-minute loop time. These slots are considered “premium” due to their limited availability within a supply and demand model and the real (or perceived) value of advertising in that particular store. Thus 2 of my 5 data questions focus on answering questions related to inventory as the ability for the database to query this information and make it readily available to sales would prove a massive benefit for this system when implemented.

## Data Questions

***The Advertising Network database should be able to answer the following questions:***

1. Which stores have advertising inventory available to sell?
2. Original question: Which Agency is running the most ads on the network? – this data question was changed for deliverable 2 to read “Which Advertiser is spending the most advertising dollars on the network”. *When coding the SQL for this question I realized that I could choose between doing a COUNT on the number of ad contracts (which would satisfy the original question) or SUM the value of the contracts. The latter seems a more valuable business question to answer as the total number of contracts does not necessarily equate to higher/most value.*
3. Which individual stores in the network have the most valuable inventory? (i.e. ads running are considered “premium inventory” and may be sold at a higher price)
4. Which region or chain has the most advertisements sold?
5. Which sales representative has sold the most ads?

## Relationships

The relations between each **entity** in the data model and the business rules that govern them are as follows:

**Store**

A *store* may display more than one *advertisement*. An advertisement may be displayed in many stores**.**

A *store* is assigned to one *region*. Each region may contain many stores.

A *store* is assigned to one *chain*. Each chain may contain many stores.

**Advertiser**

An *Advertiser* must have at least one *Advertisement*. Each Advertiser may have many Advertisements.

A *store* is assigned to one *chain*. Each chain may contain many stores.

**Advertisement**

An *Advertisement* may be displayed in one *Store*, and may be displayed in many stores.

**Contract**

A *Sales Contract* may have only one *Advertiser*, and is dependent on that Advertiser.

A *Sales Contract* must relate to at least one *Advertisement*, and may relate to many Advertisements.

A *Sales Contract* must have one *Sales Rep* and is dependent on that Sales Rep. Each Sales Rep may have more than one Sales Contract.

The *Advertising Fee* is dependent on the contract.

**Ad Payment**

A *Sales Contract* results in one *Ad Payment*, and may result in many Ad Payments.

## Conceptual Model Diagram

A close up of text on a white background

Description automatically generated

## Logical Model Diagram

A screenshot of a cell phone

Description automatically generated

## PHYSICAL DATABASE DESIGN

The following documents how I coded the creation of my database in SSMS.

###### Tables

Before embarking upon the DDL code, I thought it might be helpful to list out each table with its column names, properties and descriptions. This did help me tremendously when putting together the code in SQL to have this to reference to and so I have decided to leave this reference here as this is the foundation upon which the DDL code will be structured. Hopefully it is not superfluous to include here.

###### Table Advertiser

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **Type** | **Properties** | **Description** |
| AdvertiserID | int identity | PK | Unique Identifier |
| AdvertisingTypeID | int identity | FK | *Links to advertising type table* |
| SalesRepId | int identity | FK | *Links to sales rep table* |
| AdvertiserName | varchar(30) | Unique, 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) |  |  |

###### Table AdvertisingType

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **Type** | **Properties** | **Description** |
| AdvertisingTypeID | int identity | PK | Unique Identifier |
| AdvertisingTypeDesc | varchar(20) |  |  |

###### Table SalesRep

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **Type** | **Properties** | **Description** |
| SalesRepID | int identity | PK | Unique Identifier |
| SalesRepFirstName | varchar(30) | NOT NULL |  |
| SalesRepLastName | varchar(30) | NOT NULL |  |

###### Table Store

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **Type** | **Properties** | **Description** |
| StoreID | int identity | PK | Unique Identifier |
| StoreName | varchar(30) | UNIQUE, NOT NULL |  |
| StoreAddressLine1 | varchar(30) |  |  |
| StoreCity | varchar(30) |  |  |
| StoreState | varchar(30) |  |  |
| StorePostalcode | varchar(30) |  |  |
| ChainID | int | FK, NOT NULL | *References chain table* |
| RegionID | int | FK, NOT NULL | *References region table* |

###### Table Chain

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **Type** | **Properties** | **Description** |
| ChainID | int identity | PK | Unique Identifier |
| ChainName | varchar(30) | NOT NULL |  |

###### Table Region

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **Type** | **Properties** | **Description** |
| RegionID | int identity | PK | Unique Identifier |
| RegionName | varchar(30) | NOT NULL |  |

###### Table Advertisement

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **Type** | **Properties** | **Description** |
| AdvertisementID | int identity | PK | Unique Identifier |
| AdvertisementName | varchar(30) | Unique, NOT NULL |  |
| AdvertisementDuration | Time (changed later to int) | NOT NULL |  |
| AdvertiserID | int identity | FK | *References Advertiser table* |

###### Table Contract

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **Type** | **Properties** | **Description** |
| ContractID | int identity | PK | Unique Identifier |
| ContractStartDate | datetime | NOT NULL |  |
| ContractTotal | int |  |  |
| AdvertiserID | int identity | FK | *References Advertiser table* |
| SalesRepID | int identity | FK | *References SalesRep table* |
| AdvertisementID | int | FK | *References Advertisement table* |

###### Table ContractStore bridge

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **Type** | **Properties** | **Description** |
| ContractStoreID | int identity | PK | Unique Identifier |
| ContractID | int | FK, NOT NULL | *References Contract table* |
| StoreID | int | FK, NOT NULL | *References Store table* |

###### Table ContractAdvertisement

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **Type** | **Properties** | **Description** |
| ContractAdvertisementID | int identity | PK | Unique Identifier |
| AdStartDate | datetime | NOT NULL |  |
| AdEndDate | datetime | NOT NULL |  |
| AdFee | int | Unique, NOT NULL |  |
| ContractStoreID | int | FK | *References ContractStore table* |
| AdvertisementID | int | FK | *References Advertisement table* |

###### Table AdPaymentType

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **Type** | **Properties** | **Description** |
| AdPaymentTypeID | int identity | PK | Unique Identifier |
| AdPaymentTypeDesc | varchar(20) |  |  |

###### Table AdPayment

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **Type** | **Properties** | **Description** |
| AdPaymentID | int identity | PK | Unique Identifier |
| BillingAddressLine1 | varchar(30) |  |  |
| BillingAddressLine2 | varchar(30) |  |  |
| BillingCity | varchar(30) |  |  |
| BillingState | varchar(30) |  |  |
| BillingPostalcode | varchar(30) |  |  |
| AdPaymentAmount | int |  |  |
| ContractAdvertisementID | int | FK | *Links to contract advertisement table* |
| AdvertiserID | int | FK | *Links to advertiser table* |
| AdPaymentTypeID | int | FK | *Links to ad payment type table* |

###### DDL: Building the tables

The following documents how I coded the creation of my database in SSMS.

**--Drop all tables on the 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

-- Create the AdPayment table

CREATE TABLE AdPayment

-- Here are the 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

###### DML – Inserting data

Here are the INSERT statements I used to populate the tables in my database that will be used to answer my data questions.

**--Insert data into the Advertising Type lookup table**

INSERT INTO AdvertisingType(AdvertisingTypeDesc)

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

**--Now look to make sure it is there**

SELECT \* FROM AdvertisingType

**-- Insert data into the SalesRep lookup table**

INSERT INTO SalesRep(SalesRepFirstName, SalesRepLastName)

VALUES ('Liz', 'Wilson')

, ('Josh', 'Hanas')

, ('Carlos', 'Rodriguez')

, ('Samantha', 'Jones')

, ('Bob', 'Smith')

**-- Look and see if all the reps are in there**

SELECT \* FROM SalesRep

**-- Insert 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')

**--Take a look at all my lovely Advertisers**

SELECT \* FROM Advertiser

**-- Insert 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 exciting ads**

SELECT \* FROM Advertisement

**-- Insert 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

**-- Insert data into the Region lookup table**

INSERT INTO Region(RegionName)

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

**-- Where are my Regions?**

SELECT \* FROM Region

**-- Insert 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

**-- Insert data into the ContractStore lookup table (We need quite a few of these so we will have enough ads ‘running’ to simulate at least 3 minutes of ad content for one of the data questions)**

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')

**--Now let's see if we have enough contracts for these stores**

SELECT \* FROM ContractStore

###### Further DML – UPDATE and DELETE

When creating the Advertising Type Descriptions for the Advertising Type lookup table, the value “In-store” was used as the final option. However, the sales team more often use the terminology “House” to describe ads that are running in a store that are advertising products and brands owned by the store chain itself (in this database, ‘Kroger Howard Park’ store and ‘The Kroger Co’ are examples of this, where the Kroger company is choosing to run advertising within its own store for some of the products they are selling in the store). While these are ‘in-store ads’, the terminology used by sales and the advertising production team is “house ads”. We will therefore perform an update statement to change the name of this column from ‘in-store’ to ‘house’.

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

SELECT \* FROM AdvertisingType

**--Now I need to update the description 'In-Store' so that it reads 'House'**

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

**--Now to make sure that worked!**

SELECT \* FROM AdvertisingType WHERE AdvertisingTypeID = '4'

Another scenario within sales is that there is an “underperformer” who decides to leave the company (hmm…did they jump or were they pushed?…). In this case, we need to remove the record related to this employee from the database completely.

**--We have an underperformer! First we have to find the appropriate record within the database**

SELECT \* FROM SalesRep

**--Now we better remove it**

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

**--And finally to confirm. Poor Bob.**

SELECT \* FROM SalesRep

## ANSWERING THE DATA QUESTIONS

**These are the data questions the database is able to provide answers for.**

1. **Which stores have advertising inventory available to sell?**

###### SQL Query

**/\***

**Create a view that shows which stores are running ads with a total run time of less than 4 minutes (*ad space availability is based on a 4-minute content loop*)**

**\*/**

**-- First drop the view so the code is repeatable**

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 inventory available to sell**

SELECT \* FROM StoresWithInventory ORDER BY TotalInventoryDuration DESC

###### Query Results

A screenshot of a cell phone

Description automatically generated

1. **Which Advertiser is spending the most advertising dollars on the network?**

###### SQL Query

**/\***

**Create a view that shows which advertisers have the highest value of advertising contracts by $ amount**

**\*/**

**--First drop the view so the code is repeatable later**

DROP VIEW AdvertiserContracts

GO

**--Now figure out which Advertisers have the highest total contract value**

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

###### Here are the results of the query within SSMS:

A screenshot of a cell phone

Description automatically generated

1. **Which individual stores in the network have the most valuable inventory?**

###### SQL Query

**--Once again we drop the view first!**

DROP VIEW StoresWithPremiumSlots

GO

**--This view is similar to the one for data question #1 but we need to figure out which stores are running more than 3 minutes of ads but no more than 4 minutes total**

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

###### Query Results

**A screenshot of a cell phone

Description automatically generated**

1. **Which region or chain has the most advertisements sold?**

###### SQL Query

**--Drop the view**

DROP VIEW MostProlificChain

GO

**--Now select the contracts that are most plentiful AND most valuable and output according to Chain Name and Region they belong to**

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

###### Query Results

A screenshot of a cell phone

Description automatically generated

1. **Which sales representative has sold the most ads?**

###### SQL Query

**--For this, my last and final data question, I decided to create a function! First however, I must drop it**

DROP FUNCTION dbo.ContractSold

Go

**--Now create a function that returns the value of contracts sold by each Sales Rep**

CREATE FUNCTION dbo.ContractSold(@SalesRepID int)

RETURNS int AS

BEGIN

DECLARE @ContractNumber int

/\* Get the value 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 let’s take a look at the top 3 sales reps in descending order**

/\* Well, originally this was a straight-up select statement, but as I have used a function for this query then I guess I should wrap it into a VIEW to make my report easier to create with Access :)

\*/

**--And, as always I need to drop the view first**

DROP VIEW MostProlificSalesRep

GO

**--Now here is that view to select the Top 3 Sales Rep**

CREATE VIEW MostProlificSalesRep AS

SELECT TOP 3

\*

, dbo.ContractSold(SalesRepID) as ContractSold

FROM SalesRep

GO

**--Now to view the results to find the top performer we will order by the value of contracts sold, with most valuable first**

SELECT \* FROM MostProlificSalesRep ORDER BY ContractSold DESC

###### Here we have the results. Way to go, Liz!

A screenshot of a cell phone

Description automatically generated

## RAW DATA SAMPLES

#### Advertising Contract sample

A close up of a newspaper

Description automatically generated

#### Advertisement Submission Form (this comes from the sales rep and is the submittal form for the ad to be placed on the network)

A screenshot of a cell phone

Description automatically generated

#### Advertising Sales Data Spreadsheet (sensitive info redacted)

A screenshot of a cell phone

Description automatically generated

#### Advertising Sales Data Spreadsheet Cont’d (this is a very wide spreadsheet)….

A screenshot of a cell phone

Description automatically generated

#### Network Store Data spreadsheet

A screenshot of a cell phone

Description automatically generated

## IMPLEMENTATION

Here are examples of how my database is implemented using MS Access and R Studio.

###### DATA ENTRY FORMS

1. **View / input new Advertisers and their associated contracts and advertisements**

A screenshot of a cell phone

Description automatically generated

1. **View / input the stores in the network organized by Chain and input a new one**

A screenshot of a social media post

Description automatically generated

***NOTE****: I couldn’t capture all the columns in the snapshot – at the end of the address information the form also pulls in the Region column from the Region lookup table so you can view the Region the Store belongs to*

###### REPORTS

**Data Question #1 - Which stores have advertising inventory available to sell?**

A screenshot of a cell phone

Description automatically generated

###### Reports

**Data Question #2 - Which Advertiser is spending the most ad dollars on the network?**

A screenshot of a cell phone

Description automatically generated

###### Reports

**Data Question #3 - Which stores are running ads in premium inventory slots?**

A screenshot of a cell phone

Description automatically generated

###### Reports

**Data Question #4 - Which region or chain has the most advertising sold?**

**A screenshot of a cell phone

Description automatically generated**

###### Reports

**Data Question #5 - Which sales representative is the top performer? (Hint – we know it isn’t Bob ;)**

A screenshot of a cell phone

Description automatically generated

## REFLECTION

“It is so simple to be happy, but so difficult to be simple”

~ Gururaj Ananda Yogi

I am reminded of this quote when reflecting upon the creation of this database. Yes, it has brought me some degree of happiness (I am actually looking forward to populating it with real data for work and using it in the future)….but more importantly, it has been a lesson in keeping things simple. When choosing a topic related directly to a work need, I was aware of the possibility that the project could expand exponentially (and Dr. Block did well to remind us of this). I therefore made every effort to “keep things simple”. In building from the logical model (which seemed relatively straightforward) to the logical diagram I realized my project was already seeing signs of spiraling and so I made a conscious effort to trim it down. Yet in the feedback from deliverable 2 it was suggested that I could (at least for the purposes of this project) consolidate a couple of tables further, to trim things back. I incorporated this feedback into the coding of my tables for deliverable 2, creating one simple Advertiser table instead of having an Agency **and** an Advertiser table which added an undue level of complexity. However later, when coding my data questions, I came to find that in the process of normalization I had still managed to create an additional table that is possibly redundant (there being a ContractStore table and a ContractAdvertisement Table). Upon further reflection it seemed that these could also be condensed into two and would make coding the data questions much more efficient. As, I became aware of it I had a desire to try to restructure some of my tables to streamline things further….alas, in the interests of time and staying on track to complete the project requirements I had to ultimately disband this line of thought. (Note: I do have a copy of the DDL code I re-worked if it is of interest! Available upon request). I found some additional learning opportunities presented themselves in following this desire to restructure, and so even if it did not ultimately affect the outcome of my project I am glad I uncovered it when I did. It also reinforced further that **simplicity** is inherently the **key**. In all things, but especially database projects!!! When next approaching a new project of this – or indeed any scale – I will be keeping this firmly in mind.

**What would I have done differently?**

Again, it is tempting to try to solve ALL the problems and so if I was to do this over again I would really focus on one very small piece of the puzzle. I did this to some extent by focusing on answering only sales-related queries in my data questions. This database could also be used to keep track of more operations-focused items, related to the stores and equipment, etc. and I had considered incorporating this also at the beginning of this project. Boy I am glad I didn’t! I would have liked to have had more time to really get familiar and comfortable with writing stored procedures, as it is the one element that I did not utilize in the creation of this project as I am still fairly new to the concept and am not a programmer by profession. If I had more time to spend (or…in the future as I implement this in real-life) I will spend some time going more deeply into all that stored procedures may have to offer this and any future projects.