**Tyler McAfee**

**IST659**

**Database design for Murray Hill Premium Wines**

**Part 1: Design**

Narrative

Data Dictionary

Data Questions

ERD

Logical Model (Normalized)

**Part 2: Implementation**

SQL DDL

SQL DML

INSERT Statements

Delivery

mhpw\_Wine

WineDeliveryList

WineShelf

WineShelfList

mhpw\_Customer

CustomerOrder

OrderItem

Answers to Data Questions (SELECT, PROCEDURE, and VIEW statements)

What wines are at the store?

What wines were recently delivered?

Can you apply a discount to a wine? How?

Where are the stored in the store?

Who are the MHPW customers?

Which wines are associated with the largest orders?

GUI Prototype (Excel)

Reflection

**Design**

**Narrative**

Murray Hill Premium Wines is a local wine store that focuses on selling the best local and international wines to anyone and everyone. Customers come in everyday excited to select their favorite wine, or try something new on the shelves. Based on what is hot on the shelves, MHPW looks to supply the demands of their customers. Their selection includes different types of wines, from different parts of the globe and of all ages.

MHPW is looking to upgrade their inventory tracking methods to better supply their current and future customers. The main data component will, obviously, be *wine*. Associated with the wine will be where it is stored in the store (ie on *wine shelves*), the schedule of its *deliveries* and *orders*. And finally, MHPW would like to track the *customers* that place these orders. Once data is collected for some time, MHPW would like to invest some time and resources into setting up a membership program to give their regular shoppers some incentive and be able to better provide for their wine needs.

(\*Italicized terms will be the initial tables in this database. For any M-M relationships, we will need to add in bridge tables to the logical model and beyond.)

Stakeholders for this database include MHPW, the companies/people who deliver to them, and their customers.

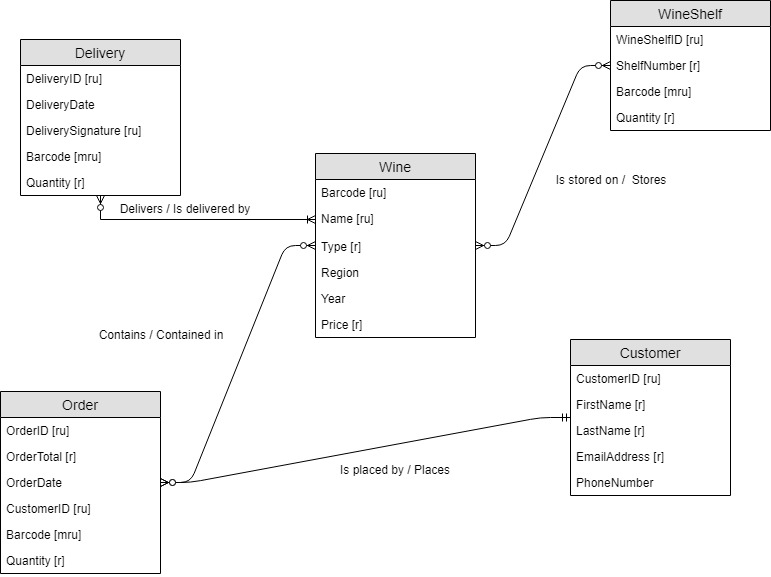
**Data Dictionary:**

|  |  |
| --- | --- |
| **Entity** | **Attributes/Properties** |
| Wine | Name ru  Type r  Region o  Year o  Price r  Barcode ru |
| Customer | Name r  Email Address r  Phone Number |
| Delivery Invoice | Barcode ru  Date  Signature ru  Quantity r |
| Order | Items rd  Quantity r  Balance rd  Date r |
| Wine Shelf | Shelf Number r  Barcode mru  Quantity r |

**Data Questions:**

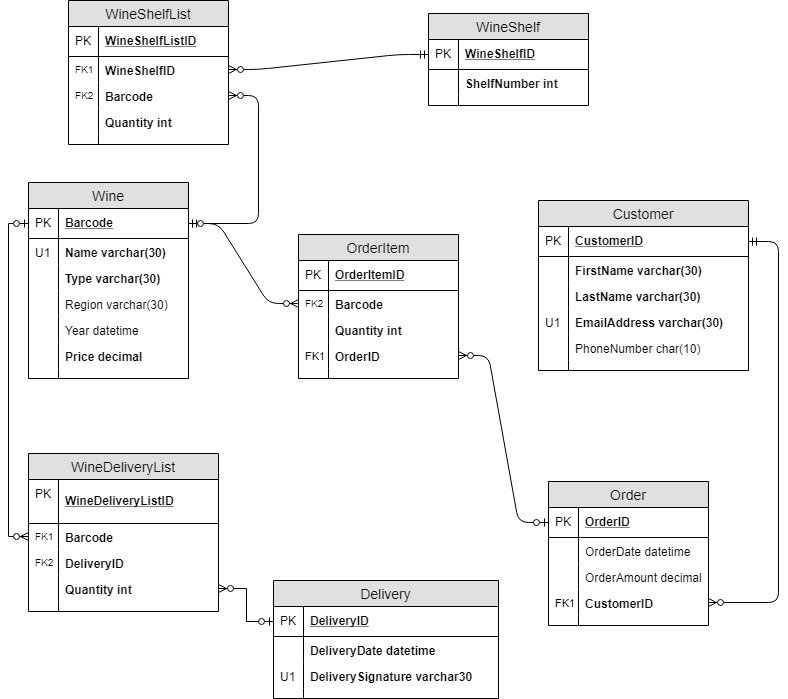
1. What wines are at the store?
2. What wines were recently delivered?
3. Can you apply a discount to a wine? How?
4. Where are the stored in the store?
5. Who are the MHPW customers?
6. Which wines are associated with the largest orders?

**Conceptual Model (ERD)**

****

**Normalized Logical Model**

* All primary keys will be treated as surrogate keys.
* I created the Logical model after learning about normalization. So many relationships, data types, etc. were used in preparation of creating a normalized model.
* This yielded a model which I believe to be in third normal form as shown below.



**Part 2: Implementation**

**SQL DDL:**

First and foremost I need to create the tables for MHPW. I will start with the tables that have no FK constraints to make sure no constraints are created which do not already have tables in the database.

/\*

Author : Tyler McAfee

Course : IST659 M400

Term : March, 2019

Project Deliverable 2

\*/

-- Before creating all my tables, I will drop the tables first, if they exist

DROP TABLE IF EXISTS WineDelivery

DROP TABLE IF EXISTS mhpw\_Customer

DROP TABLE IF EXISTS mhpw\_Wine

DROP TABLE IF EXISTS WineShelf

DROP TABLE IF EXISTS WineShelfList

DROP TABLE IF EXISTS WineDeliveryList

DROP TABLE IF EXISTS CustomerOrder

DROP TABLE IF EXISTS OrderItem

/\*

Now create each table, starting with strong tables

which do not have any FK constraints

\*/

-- Creating the Delivery table

create table Delivery (

--Columns for the Delivery table

DeliveryID int identity,

DeliveryDate datetime not null,

DeliverySignature varchar(30) not null,

--Constraints on the Delivery table

constraint PK\_Delivery primary key (DeliveryID),

constraint U1\_Delivery unique (DeliverySignature)

)

--End creating the Delivery table

-- Creating the Wine table

create table mhpw\_Wine (

--Columns for the Wine table

WineBarcode int identity,

WineName varchar(30) not null,

WineType varchar(30) not null,

WineRegion varchar(30),

WineYear datetime,

WinePrice decimal not null,

--Constraints on the User table

constraint PK\_mhpw\_Wine primary key (WineBarcode)

)

--End creating the Wine table

-- Creating the WineDeliveryList table

create table WineDeliveryList (

--Columns for the WineDeliveryList table

WineDeliveryListID int identity,

WineBarcode int not null,

DeliveryID int not null,

Quantity int not null,

--Constraints on the WineDeliveryList table

constraint PK\_WineDeliveryList primary key (WineDeliveryListID),

constraint FK1\_WineDeliveryList foreign key (WineBarcode) references mhpw\_Wine(WineBarcode),

constraint FK2\_WineDeliveryList foreign key (DeliveryID) references Delivery(DeliveryID)

)

--End creating the WineDeliveryList table

-- Creating the WineShelf table

create table WineShelf (

--Columns for the WineShelf table

WineshelfID int identity,

ShelfNumber int not null,

--Constraints on the WineShelf table

constraint PK\_WineShelf primary key (WineshelfID)

)

--End creating the WineShelf table

-- Creating the WineShelfList table

create table WineShelfList (

--Columns for the WineShelfList table

WineShelfListID int identity,

WineShelfID int not null,

WineBarcode int not null,

Quantity int not null,

--Constraints on the WineShelfList table

constraint PK\_WineShelfList primary key (WineShelfListID),

constraint FK1\_WineShelfList foreign key (WineShelfID) references WineShelf(WineShelfID),

constraint FK2\_WineShelfList foreign key (WineBarcode) references mhpw\_Wine(WineBarcode)

)

--End creating the WineShelfList table

--Creating the mhpw\_Customer table

create table mhpw\_Customer (

--Columns for the mhpw\_Customer table

CustomerID int identity,

FirstName varchar(30) not null,

LastName varchar(30) not null,

Emailaddress varchar(50) not null,

PhoneNumber char(10),

--Constraints on the mhpw\_Customer table

constraint PK\_mhpw\_Customer primary key (CustomerID),

constraint U1\_mhpw\_Customer unique (EmailAddress)

)

--End creating the mhpw\_Customer table

--Creating the CustomerOrder table

create table CustomerOrder (

--Columns for the CustomerOrder table

OrderID int identity,

OrderDate datetime,

OrderAmount decimal,

CustomerID int not null,

--Constraints on the CustomerOrder table

constraint PK\_CustomerOrder primary key (OrderID),

constraint FK1\_CustomerOrder foreign key (CustomerId) references mhpw\_Customer(CustomerID)

)

--End creating the CustomerOrder table

--Creating the OrderItem table

create table OrderItem (

--Columns for the OrderItem table

OrderItemID int identity,

WineBarcode int not null,

Quantity int not null,

OrderID int not null,

--Constraints on the OrderItem table

constraint PK\_OrderItem primary key (OrderItemID),

constraint FK1\_OrderItem foreign key (WineBarcode) references mhpw\_Wine(WineBarcode),

constraint FK2\_OrderItem foreign key (OrderID) references CustomerOrder(OrderID)

)

--End creating the OrderItem table

**SQL DML:**

**INSERT Statements**

Now that the tables are creating, I can generate some data to include in each table. Note that only a few entries are captured pertaining to each table. These tables have many more rows in the physical database.

/\*

Insert some data into our tables. Many other rows have been added to these tables,

but for simplicity, only a few entries will be outlined below

\*/

insert into Delivery (DeliveryDate, DeliverySignature) values ('2/14/2019', 'Krispin Ancliff');

insert into Delivery (DeliveryDate, DeliverySignature) values ('1/31/2019', 'Izaak Ludovici');

insert into Delivery (DeliveryDate, DeliverySignature) values ('2/27/2019', 'Cosetta Dillingham');

insert into Delivery (DeliveryDate, DeliverySignature) values ('2/3/2019', 'Eddie Henighan');

insert into Delivery (DeliveryDate, DeliverySignature) values ('2/28/2019', 'Caria Cottam');

insert into mhpw\_Wine (WineName, WineType, WineRegion, WineYear, WinePrice) values ('Raynor', 'Rose', 'California', 2011, 29.99);

insert into mhpw\_Wine (WineName, WineType, WineRegion, WineYear, WinePrice) values ('Romaguera', 'White', 'Oregon', 2013, 76.69);

insert into mhpw\_Wine (WineName, WineType, WineRegion, WineYear, WinePrice) values ('Dibbert and Schaden', 'Red', 'Italy', 2005, 72.89);

insert into mhpw\_Wine (WineName, WineType, WineRegion, WineYear, WinePrice) values ('Rosenbaum-Rosenbaum', 'Red', 'Washington', 1992, 55.50);

insert into mhpw\_Wine (WineName, WineType, WineRegion, WineYear, WinePrice) values ('Reynolds Boyle', 'White', 'Oregon', 1995, 63.50);

insert into WineDeliveryList (WineBarcode, DeliveryID, Quantity) values (24, 17, 33);

insert into WineDeliveryList (WineBarcode, DeliveryID, Quantity) values (15, 5, 84);

insert into WineDeliveryList (WineBarcode, DeliveryID, Quantity) values (11, 6, 25);

insert into WineDeliveryList (WineBarcode, DeliveryID, Quantity) values (21, 8, 69);

insert into WineDeliveryList (WineBarcode, DeliveryID, Quantity) values (23, 8, 13);

insert into WineShelf (ShelfNumber) values (4);

insert into WineShelf (ShelfNumber) values (3);

insert into WineShelf (ShelfNumber) values (1);

insert into WineShelf (ShelfNumber) values (8);

insert into WineShelf (ShelfNumber) values (4);

insert into WineShelfList (WineShelfID, WineBarcode, Quantity) values (9, 4, 43);

insert into WineShelfList (WineShelfID, WineBarcode, Quantity) values (8, 36, 23);

insert into WineShelfList (WineShelfID, WineBarcode, Quantity) values (16, 11, 18);

insert into WineShelfList (WineShelfID, WineBarcode, Quantity) values (11, 48, 44);

insert into WineShelfList (WineShelfID, WineBarcode, Quantity) values (19, 39, 45);

insert into mhpw\_Customer (FirstName, LastName, EmailAddress, PhoneNumber) values ('Pierce', 'Jessep', 'pjessep0@yandex.ru', '8863122901');

insert into mhpw\_Customer (FirstName, LastName, EmailAddress, PhoneNumber) values ('Broddie', 'Richt', 'bricht1@twitpic.com', '3165776359');

insert into mhpw\_Customer (FirstName, LastName, EmailAddress, PhoneNumber) values ('Dyan', 'Tranfield', 'dtranfield2@sina.com.cn', '5832899475');

insert into mhpw\_Customer (FirstName, LastName, EmailAddress, PhoneNumber) values ('Corrie', 'McGerraghty', 'cmcgerraghty3@nifty.com', '5827755575');

insert into mhpw\_Customer (FirstName, LastName, EmailAddress, PhoneNumber) values ('Myca', 'Moger', 'mmoger4@answers.com', '2286861360');

insert into CustomerOrder (OrderDate, OrderAmount, CustomerID) values ('2/13/2019', 18, 48);

insert into CustomerOrder (OrderDate, OrderAmount, CustomerID) values ('3/2/2019', 44, 24);

insert into CustomerOrder (OrderDate, OrderAmount, CustomerID) values ('1/2/2019', 49, 29);

insert into CustomerOrder (OrderDate, OrderAmount, CustomerID) values ('2/14/2019', 17, 13);

insert into CustomerOrder (OrderDate, OrderAmount, CustomerID) values ('1/6/2019', 26, 3);

insert into OrderItem (WineBarcode, OrderAmount, CustomerID) values (3, 1, 16);

insert into OrderItem (WineBarcode, OrderAmount, CustomerID) values (14, 42, 54);

insert into OrderItem (WineBarcode, OrderAmount, CustomerID) values (17, 27, 53);

insert into OrderItem (WineBarcode, OrderAmount, CustomerID) values (12, 48, 53);

insert into OrderItem (WineBarcode, OrderAmount, CustomerID) values (34, 34, 10);

--Done inserting data

**Answers to Data Questions:**

The following is a view which will answer the question: *What wines are at MHPW?*

/\*

Next is a VIEW which will take a basic select statement about the wine at the store and define a VIEW for it.

\*/

CREATE VIEW WinesAtStore AS

select

mhpw\_Wine.WineBarcode,

mhpw\_Wine.WineName,

mhpw\_Wine.WinePrice

FROM mhpw\_Wine

GROUP BY

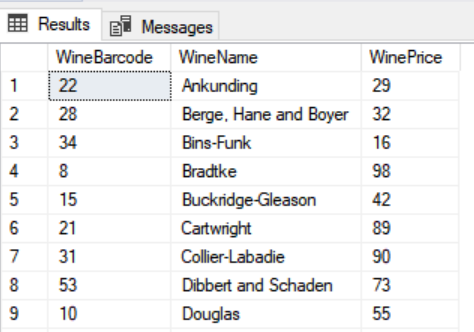
mhpw\_Wine.WineBarcode,

mhpw\_Wine.WineName,

mhpw\_Wine.WinePrice

SELECT \* FROM WinesAtStore ORDER BY WineName

Sample output:



The following is a view which will answer the question: *What wines were recently delivered?*

/\*

The next VIEW will combine data about the deliveries and wines and give the dates for when certain wines were delivered

\*/

CREATE VIEW RecentDeliveries AS

SELECT

Delivery.DeliveryDate,

mhpw\_Wine.WineName

FROM WineDeliveryList

JOIN Delivery ON WineDeliveryList.DeliveryID = Delivery.DeliveryID

JOIN mhpw\_Wine ON WineDeliveryList.WineBarcode = mhpw\_Wine.WineBarcode

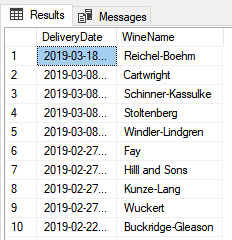
GROUP BY

mhpw\_Wine.WineName,

Delivery.DeliveryDate

SELECT \* FROM RecentDeliveries ORDER BY DeliveryDate desc

Sample output:



The following is a procedure which implement a discount on a wine. The goal here was to be able to update the prices of certain wines due to promotions, etc. Thus it answers our question of: *Can you apply a discount to a wine? How?*

/\*

Below is a PROCEDURE which has two parameters; WineName and NewWinePrice (a discounted price for that wine).

\*/

CREATE PROCEDURE WineDiscount (@wineName varchar(30), @newWinePrice int) AS

BEGIN

DECLARE @wineBarcode int

SELECT @wineBarcode = WineBarcode FROM mhpw\_Wine

WHERE WineName = @wineName

Update mhpw\_Wine SET WinePrice = @newWinePrice

RETURN @wineBarcode

END

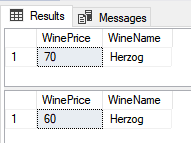
GO

SELECT WinePrice, WineName FROM mhpw\_Wine WHERE WineName = 'Herzog'

EXEC WineDiscount Herzog, 60

SELECT WinePrice, WineName FROM mhpw\_Wine WHERE WineName = 'Herzog'

Sample output. The wine Herzog had a $10 discount applied, changin the price from $70 to $60.



The following is a view which will answer the question: *Where are the wines stored in the store?*

/\*

The VIEW below will join the WineShelf table with the Wines on those shelves

\*/

CREATE VIEW StorageOfWines AS

SELECT

WineShelf.ShelfNumber,

mhpw\_Wine.WineName

FROM WineShelfList

JOIN WineShelf ON WineShelfList.WineShelfID = WineShelf.WineShelfID

JOIN mhpw\_Wine ON WineShelfList.WineBarcode = mhpw\_Wine.WineBarcode

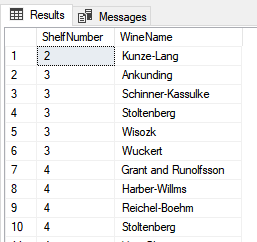
GROUP BY

mhpw\_Wine.WineName,

WineShelf.ShelfNumber

SELECT \* FROM StorageOfWines ORDER BY ShelfNumber asc

Sample output:



The following is a view which will answer the question: *Who are the customers of MHPW?*

/\*

The VIEW below will be another simple select statement, which will yield a VIEW for the customers that have registered their information at MHPWs.

\*/

CREATE VIEW ActiveCustomers AS

SELECT

mhpw\_Customer.LastName,

mhpw\_Customer.FirstName,

mhpw\_Customer.Emailaddress,

mhpw\_Customer.PhoneNumber

FROM mhpw\_Customer

GROUP BY

mhpw\_Customer.LastName,

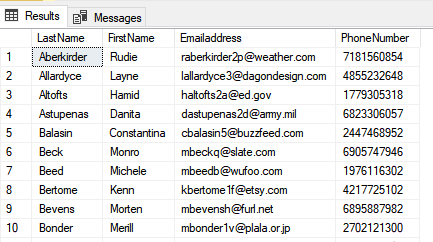
mhpw\_Customer.FirstName,

mhpw\_Customer.Emailaddress,

mhpw\_Customer.PhoneNumber

SELECT \* FROM ActiveCustomers ORDER BY LastName

Sample output:



The following is a view which will answer the question: *Which wines are associated with the largest orders?*

/\*

This next VIEW will join the Wine table with the CustomerOrder table to provide the user will information about the size of orders

\*/

CREATE VIEW LargestOrders AS

SELECT

CustomerOrder.OrderAmount,

mhpw\_Wine.WineName

FROM OrderItem

JOIN CustomerOrder ON OrderItem.OrderID = CustomerOrder.OrderID

JOIN mhpw\_Wine ON OrderItem.WineBarcode = mhpw\_Wine.WineBarcode

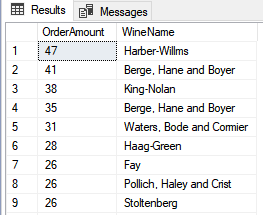
GROUP BY

mhpw\_Wine.WineName,

CustomerOrder.OrderAmount

select \* from LargestOrders order by OrderAmount desc

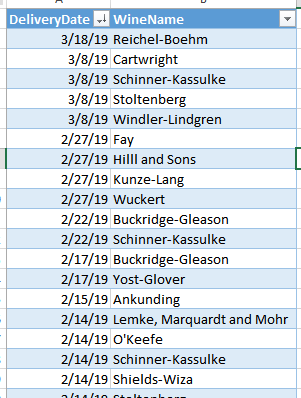
Sample output:



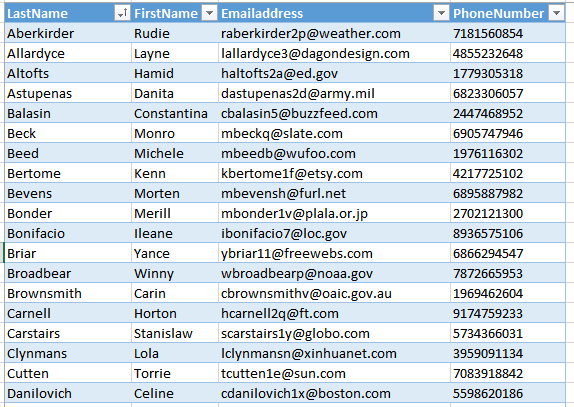
**GUI Prototype:**

I struggled with getting Access to work towards the end of the term, so Excel will be used for the interface that users can interact with.

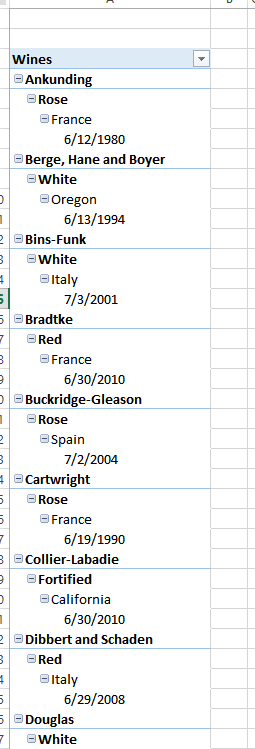
First is an Excel output of some of MHPWs recent deliveries using the VIEW RecentDeliveries:



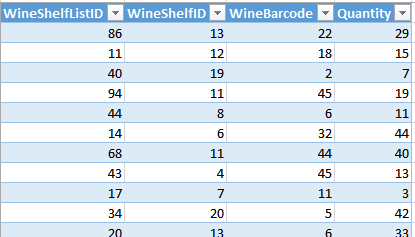
Excel was then used to show an output of customers and all the info that MHPW has using the VIEW ActiveCustomers:



Next, is a pivot table of wines, including the type, region, and year. Price was not included because this is something that can change by the WineDiscount procedure.



Lastly, a simple SELECT statement was used and the Excel output shows how many wines are on certain wine shelves:



**Reflection:**

When I first started this project, I made many assumptions about this being a simple database to construct and play with. After creating the database and working with certain questions (ie *Which wines are delivered most frequently?*), I quickly realized that this was more complex than I originally assumed. MHPW is not a static process, and so trying to understand the flow of my data and the timeline of how it was entered, tested my knowledge of the concepts covered in this course.

One thing that I glanced over in the early stages was defining tables and fields. I didn't design this in a systematic way, although at the time I figured my naming conventions were okay. But the readability of the tables was lacking. A key phrase, like the "vc\_" from the labs is a good takeaway.

One other area that I could improve is that I would put more thought into what data is stored in which tables, and perhaps define some more tables, so that the data is more concrete. I was able to join tables together with not much problem, so with better defined data, I can manipulate and present the data more easily.

Altogether, after learning about the flow of building a simple database, I have a surface level understanding about just how important basic concepts are such as relationships and normalization. I deal with data a lot with my current job, and I find myself deconstructing the data and imagining what the database looks like and how it is managed.

It was nice to take the learnings from the labs and apply them to my own idea. I struggled at times, but in the end, am proud of the results captured in this report. There is much more to learn, but this was a great foundational project to my data science path ahead.