# Database Project Proposal

DTSC691: Applied Data Science

<Paul Walker - Used Vehicle Relational Database>

## Project Overview

### 1. **Project Goals**

A. Purpose: The purpose of this project is to design and implement a comprehensive relational database for used car sales. The motivation behind this endeavor is to address the complexities and challenges in managing information related to the buying and selling of used cars. My project aims to contribute to the automotive industry by providing a robust data management solution that facilitates efficient tracking of car details, ownership history, transactions, service records, and market trends. This database will serve as a foundation for [database integration option].

B. Project Focus: The primary areas of investigation revolve around creating a well-structured database that captures key aspects of the used car sales process. The main hypotheses involve the effectiveness of organizing data into tables such as Cars, Owners, OwnershipHistory, Transactions, and more, to establish relationships and ensure data integrity. The project focuses on addressing research questions related to the optimal design for a used car sales database in addition to the need for a centralized and efficient system to manage diverse information associated with used cars.

C. Specific Goals

a. Design and implement tables to store information about cars, owners, transactions, services, and market trends.

b. Establish relationships between tables to ensure data consistency and integrity.

c. Enable efficient search and query capabilities for users to retrieve information about cars and transactions.

d. Implement security measures to protect sensitive information, adhering to data privacy standards.

e. Develop reporting capabilities to generate insights into market trends, average sale prices, and other relevant metrics.

f. Create an audit trail to track changes made to the database.

D. Expected Outcomes

a. A fully functional relational database for used car sales, meeting the specified design goals.

b. Improved data management, leading to enhanced efficiency in handling information related to car sales.

c. Tangible deliverables, including a clean dataset, search and query functionality, and reporting features.

d. Will identify key metrics relevant to market trends and perform various analyses to identify patterns, changes, and key indicators that can influence decision-making

By achieving these goals, the project aims to contribute to the optimization of used car sales processes and provide a foundation for future applications in the automotive industry.

# 2. **Project Description**

A. Problem Domain: The problem domain for this project is the management of information related to used car sales. The domain involves the complexities of tracking and organizing data associated with cars, owners, transactions, service history, and market trends. Challenges include maintaining data integrity, facilitating efficient search and query capabilities, and preparing for [database integration option].

B. Database Design and Assumptions

   a. The planned **database design** involves a relational model with tables representing entities such as Cars, Owners, Transactions, and more (see relational schema for complete table information).

   **b. Assumptions**

      i. Each car has a unique VIN (Vehicle Identification Number).

      ii. Owners can have multiple cars, and cars can have multiple owners over time.

      iii. Each sale transaction involves one seller and one buyer, both of whom are associated with specific cars.

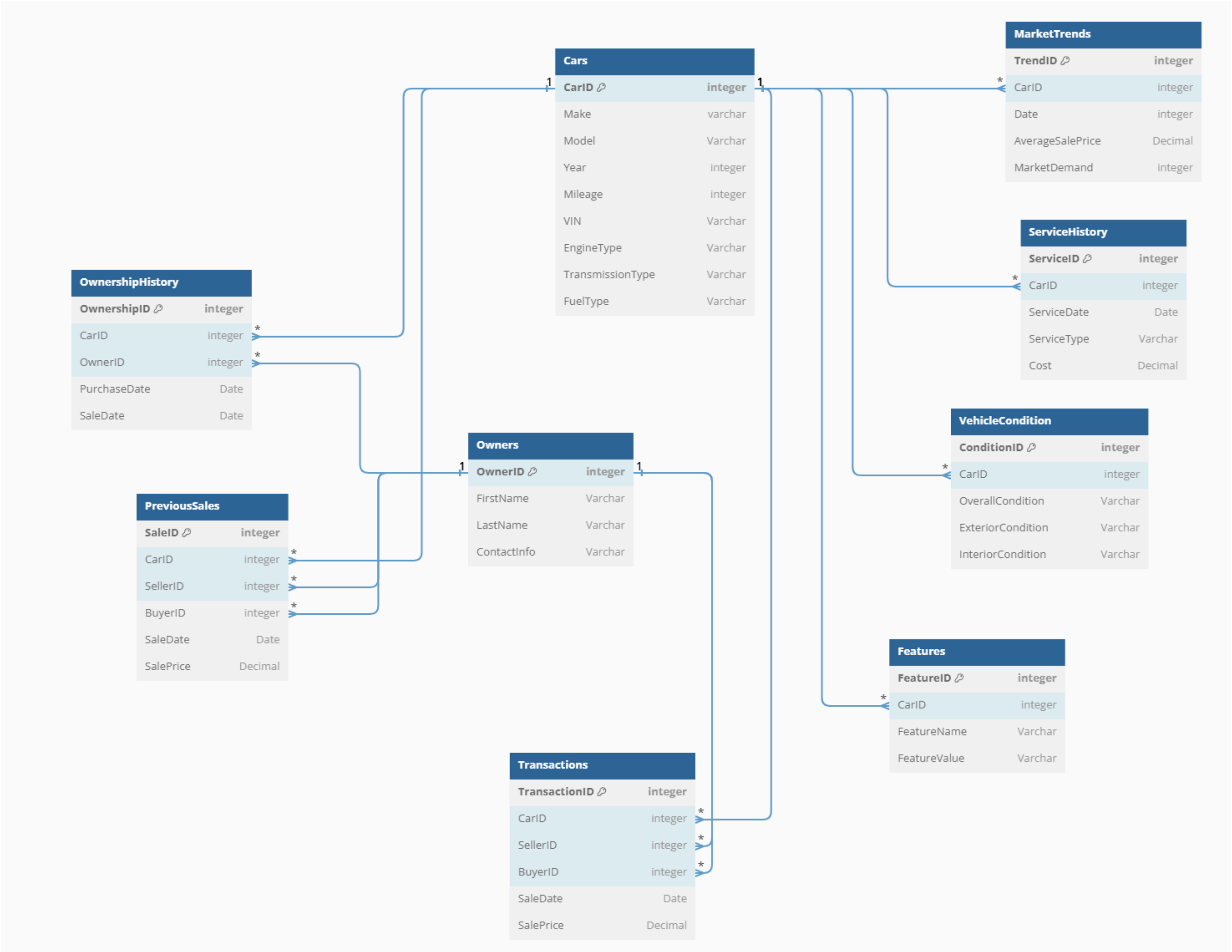
      iv. Features of a car are stored as name-value pairs in the "Features" table.

      v. Ownership history includes purchase and sale dates, allowing for tracking changes in ownership over time.

      vi. The "ServiceHistory" table captures various services performed on a car over time.

      vii. Market trends data provides insights into average sale prices and market demand for specific cars.

C. Relational Schema:

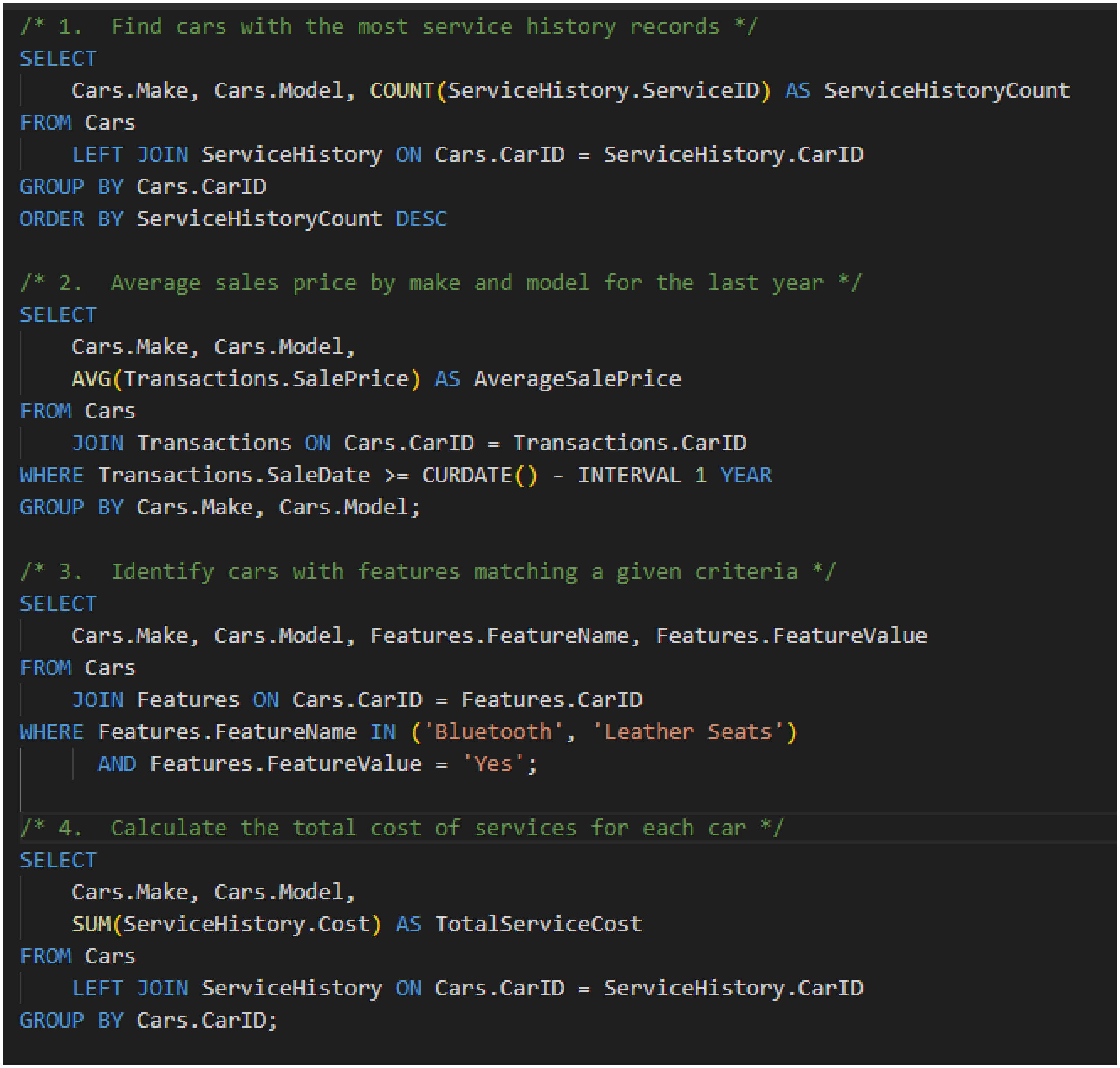

MarketTrends
TrendID integer
CarID integer
Date integer
AverageSalePrice Decimal
MarketDemand integer
Cars
CarID integer
Make varchar
Model Varchar
Year integer
Mileage integer
VIN Varchar
EngineType Varchar
TransmissionType Varchar
FuelType Varchar
ServiceHistory
ServiceID integer
CarID integer
ServiceDate Date
ServiceType Varchar
Cost Decimal
OwnershipHistory
OwnershipID integer
CarID integer
OwnerID integer
PurchaseDate Date
SaleDate Date
VehicleCondition
ConditionID integer
CarID integer
OverallCondition Varchar
ExteriorCondition Varchar
InteriorCondition Varchar
Owners
OwnerID integer
FirstName Varchar
LastName Varchar
ContactInfo Varchar
PreviousSales
SaleID integer
CarID integer
SellerID integer
BuyerID integer
SaleDate Date
SalePrice Decimal
Features
FeatureID integer
CarID integer
FeatureName Varchar
FeatureValue Varchar
Transactions
TransactionID integer
CarID integer
SellerID integer
BuyerID integer
SaleDate Date
SalePrice Decimal
1
1
1
1
*
*
*
*
*
*
*
*
*
*
*
*

D. Database Implementation: The database will be implemented using SQL, and for the graphical user interface (GUI), I will utilize MySQL as my main tool along with DBeaver to provide an intuitive interface for designing, querying, and managing the database.

E. Data Insertion: Data sources will include a combination of manual input and potentially automated processes. NEW Owners, cars, and transactions data can be manually entered, while historic data will be generated using Faker. In addition, market trends and service history may be collected from external sources and imported into the database. Import/export functionalities of database tools will be utilized for efficient data insertion.

F. Data Manipulation: Data cleaning will involve techniques such as handling missing values, ensuring data consistency, and addressing outliers. SQL queries will be employed to identify and rectify data issues. Duplicates will be removed using SQL's DISTINCT clause or appropriate aggregation methods.

G. Query Examples:

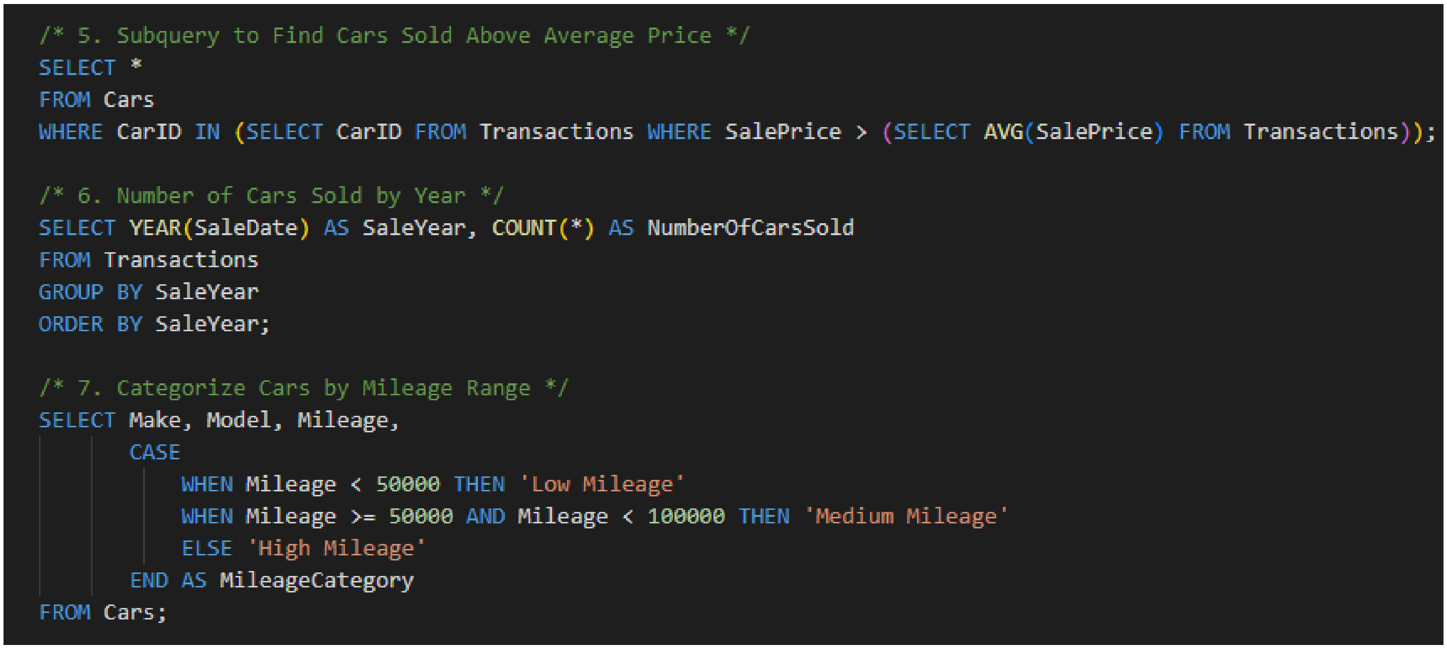


```sql
/* 1.  Find cars with the most service history records */
SELECT
    Cars.Make, Cars.Model, COUNT(ServiceHistory.ServiceID) AS ServiceHistoryCount
FROM Cars
    LEFT JOIN ServiceHistory ON Cars.CarID = ServiceHistory.CarID
GROUP BY Cars.CarID
ORDER BY ServiceHistoryCount DESC

/* 2.  Average sales price by make and model for the last year */
SELECT
    Cars.Make, Cars.Model,
    AVG(Transactions.SalePrice) AS AverageSalePrice
FROM Cars
    JOIN Transactions ON Cars.CarID = Transactions.CarID
WHERE Transactions.SaleDate >= CURDATE() - INTERVAL 1 YEAR
GROUP BY Cars.Make, Cars.Model;

/* 3.  Identify cars with features matching a given criteria */
SELECT
    Cars.Make, Cars.Model, Features.FeatureName, Features.FeatureValue
FROM Cars
    JOIN Features ON Cars.CarID = Features.CarID
WHERE Features.FeatureName IN ('Bluetooth', 'Leather Seats')
        AND Features.FeatureValue = 'Yes';

/* 4.  Calculate the total cost of services for each car */
SELECT
    Cars.Make, Cars.Model,
    SUM(ServiceHistory.Cost) AS TotalServiceCost
FROM Cars
    LEFT JOIN ServiceHistory ON Cars.CarID = ServiceHistory.CarID
GROUP BY Cars.CarID;
```



```
/* 5. Subquery to Find Cars Sold Above Average Price */
SELECT *
FROM Cars
WHERE CarID IN (SELECT CarID FROM Transactions WHERE SalePrice > (SELECT AVG(SalePrice) FROM Transactions));

/* 6. Number of Cars Sold by Year */
SELECT YEAR(SaleDate) AS SaleYear, COUNT(*) AS NumberOfCarsSold
FROM Transactions
GROUP BY SaleYear
ORDER BY SaleYear;

/* 7. Categorize Cars by Mileage Range */
SELECT Make, Model, Mileage,
       CASE
           WHEN Mileage < 50000 THEN 'Low Mileage'
           WHEN Mileage >= 50000 AND Mileage < 100000 THEN 'Medium Mileage'
           ELSE 'High Mileage'
       END AS MileageCategory
FROM Cars;
```

H. Database Integration: I plan to proceed with option 1 for database integration as outlined in the proposal guidelines. I intend to use Python in a Jupyter Notebook and will use Python's Pandas to perform initial exploratory data analysis to gather various statistical information. Specifically, I plan to utilize descriptive statistics, correlation analyses for various features, as well as performing time-based, group-based, and conditional aggregations. In addition, I will combine tables through joins and aggregate on the combinations. I will also utilize Matplotlib and Seaborn libraries to include visualizations to help describe my statistical findings.

## 3. **Capstone Complexity**

a. My project will meet the standards of master's level complexity in the following ways;

i. **Data Selection and Diversity:** I will choose a diverse and extensive dataset that includes a wide range of variables, capturing various aspects of the used car market. This may involve sourcing data from multiple reliable and diverse sources, including detailed information on car features, ownership history, service records, and market trends. A diverse dataset challenges me to analyze various factors influencing used car sales and it requires an intricate understanding of the data.

ii. **Technical**: I will implement advanced database design principles, considering factors such as normalization, indexing strategies, and query optimization. The complexity lies in designing a robust and scalable database architecture that can handle complex relationships, large volumes of data, and advanced queries. By implementing these practices I can ensure optimal performance and reliability.

iii. **Statistical**: I will conduct advanced statistical analyses to identify patterns, correlations, and trends in the data. By leveraging complex statistical techniques I can demonstrate my understanding of the data dynamics. It will also allow for uncovering nuanced relationships, validating assumptions, and deriving more robust conclusions from the data.

iv. **Visual:** I will create interactive visualizations to complement my statistical findings and showcase my ability to communicate complex findings in a succinct manner. Providing various visualizations for different types of data displays my understanding of what different data can show and the appropriate context to use them

v. **Reporting and Documentation:** I will develop a comprehensive project report that goes beyond a standard analysis and will include in-depth discussions on methodology, limitations, and recommendations for future work. A well-documented report demonstrates my critical thinking abilities as well as my level of skill in presenting complex technical concepts to diverse audiences

## 4. **Software:**

a. Database Management System (DBMS):

- Software Tool: MySQL
- Primary Function: MySQL will serve as the relational database management system (DBMS) for storing and managing the used car sales database as it is powerful, open-source, and supports complex queries and transactions.

b. Faker
    - Software Tool: Faker
    - Primary Function: Generate massive amounts of fake (but realistic) data for testing and development.

c. Python Programming Language:
    - Software Tool: Python (using Jupyter Notebooks)
    - Primary Function: Python will be the primary programming language for data analysis, manipulation, and modeling. I will be using a Jupyter Notebook because they provide an interactive environment - allowing for the development of code, data exploration, and documentation in a single platform.

d. Pandas Library:
    - Software Tool: Pandas
    - Primary Function: Pandas is a powerful data manipulation library and I will use it for reading data from the database into DataFrames, cleaning and preprocessing data, and conducting exploratory data analysis. Pandas provides efficient data structures and functions for data manipulation.

e. Matplotlib and Seaborn Libraries:
    - Software Tool: Matplotlib and Seaborn
    - Primary Function: Matplotlib and Seaborn are Python libraries for data visualization. They will be used to create various plots and charts, such as histograms, scatter plots, and box plots, to visually explore the used car sales data.

f. SQLAlchemy Library:
    - Software Tool: SQLAlchemy
    - Primary Function: SQLAlchemy is a SQL toolkit and Object-Relational Mapping (ORM) library for Python. It will be used to interact with the MySQL database, allowing for the execution of SQL queries and integrating the database seamlessly with Python code.

## 5. **Project Completion Plan**

- Week 1: Project Setup and Data Collection
    - Define Project Scope and Objectives
    - Data Source Identification and Collection

  - Database Design and Setup

- Week 2: Data Cleaning and Preprocessing
  - Data Loading and Initial Exploration
  - Data Cleaning
  - Data Integration and Feature Engineering

- Week 3: Database Integration with Python
  - Connect Python to the Database
  - SQL Queries and Data Retrieval
  - Initial Data Analysis in Python

- Week 4: Initial Data Analysis, Visualization, & Interpretation
  - Advanced Statistical Analysis
  - Data visualizations
  - Record findings, interpretations, and create documentation

- Week 5: Refinement and Optimization
  - Database Optimization
  - Documentation Review
  - Code Review

- Week 6: Final Analysis and Reporting
  - Final Data Analysis
  - Final draft of Report and Documentation

- Week 7: Final Analysis and Reporting
  - Prepare final materials and synthesize presentation
  - Practice, practice, practice
  - Record and Submit

## 6. **Presentation Plan**

*At this point in time I am currently unsure of the route I want to take for my presentation, specifically the format in which I want to present. I think I would like to get an idea of the findings and how everything looks before making a final decision on how to present. I am leaning towards a Jupyter notebook as I think that will be the best format to present my project in as much detail as possible.

## 7. **Resources**

a. MySQL
b. Python and Jupyter Notebooks
c. Pandas
d. Matplotlib & Seaborn Libraries
e. Faker
f. SQLAlchemy