Project Title: Car Dealership Management

Project Summary

This project aims to develop a web-based application that leverages a relational SQL database to manage a car dealership's inventory. The website will allow dealerships to handle several tasks related to managing the availability of cars in their dealership. The primary features include searching, creating, updating, and deleting entries through input car information, as well as seeing which cars in their inventory are the most popular according to market trends. It will provide users with an easy-to-use interface, allowing quick access to the database. They will be able to input details such as make, model, year, transmission type, drive wheel, engine specifications, and listing price to use the functionalities listed above. This allows dealerships to manage their car inventories in real-time.

Description

Nowadays, dealerships rely on outdated methods such as spreadsheets or manual records, to keep track of their car inventory. Using these methods reduces accuracy and makes it difficult to maintain up-to-date information across the dealership's members.

The project will address this issue by providing a system for users to easily search for cars, add new cars to the inventory, update existing records, or delete cars that are no longer available. Users will be able to tailor their inventory through a market analysis that can compare their inventory to in order to help determine what current cars are likely to sell and what cars they can consider showing more to improve their sales.

Creative Component

We will integrate a smart search functionality that enhances the basic search feature by allowing users to apply complex filters or combinations of multiple attributes.

For example, not only can an employee search for cars by make and model, but they can also filter results further based on price range or transmission type. This feature could be implemented with advanced query handling in SQL and a user-friendly interface to help users build and refine their searches dynamically.

We will also implement data validation and automated error-checking features to enhance input accuracy. For instance, when dealership staff enter data for new cars, the application will look for potential errors, such as prices that appear unusually high or low for the specified make and model. This will prompt users to review their entries, helping to minimize mistakes and maintain an accurate inventory.

We will also integrate a function that will sort the dealerships inventory from most popular car to least popular through market analysis done with car sales datasets. A user will be able to click a button that requests a table from the market analysis and returns the entries in the inventory database that have the same year, make, and model as the market analysis table(s).

Usefulness

Our web application will be highly useful for car dealerships by offering an efficient way to handle inventory. With features like creating, searching, updating, and deleting car records, dealership staff can easily manage vehicle details without relying on outdated methods or manual tracking. Since dealerships often need to add new cars, update records as vehicles are sold or modified, and remove cars that are no longer available, the application helps streamline these tasks, saving time on administrative work while reducing errors in data entry.

Unlike many existing systems, which are part of larger dealership management platforms that include sales tracking and other unnecessary features, this application is designed with simplicity in mind. It focuses exclusively on inventory management, offering a straightforward and user-friendly solution. The combination of intelligent search functionality and built-in error-checking makes it ideal for dealerships seeking a more efficient way to manage their car listings without added complexity.

The system provides five main functionalities: adding, searching, updating, deleting car records, as well as returning the most popular cars in their inventory or current search request.

- Adding functionality:
    - Users can input vehicle information (such as make, model, year, engine size, transmission type, and price) which is then added as a new entry in the relational database
- Searching functionality:
    - Users can retrieve entries in the database by inputting any of above information and clicking search
    - The search feature also enables users to apply filters like make, model, price range, and year to narrow down and quickly find cars of interest
    - Search results are displayed in a table for easy reference
- Updating functionality:
    - Users can update entries in the database by inputting new information after searching for them
- Deleting functionality:
    - Users can remove entries in the database when they are no longer available, keeping the database clean and up-to-date
- Most popular cars functionality:
    - Users can request for their inventory or current query to be sorted by most to least popular based on market analysis
    - The most popular cars functionality has multiple layers of function. We will use the real datasets we found from Kaggle (sourced from Carvana) to

determine what the count of sold cars in both datasets are, combine the sold count car data obtained from both sets into a single table, and use the sold count to sort the cars in the users car inventory or current query by the count generated from the Kaggle datasets.

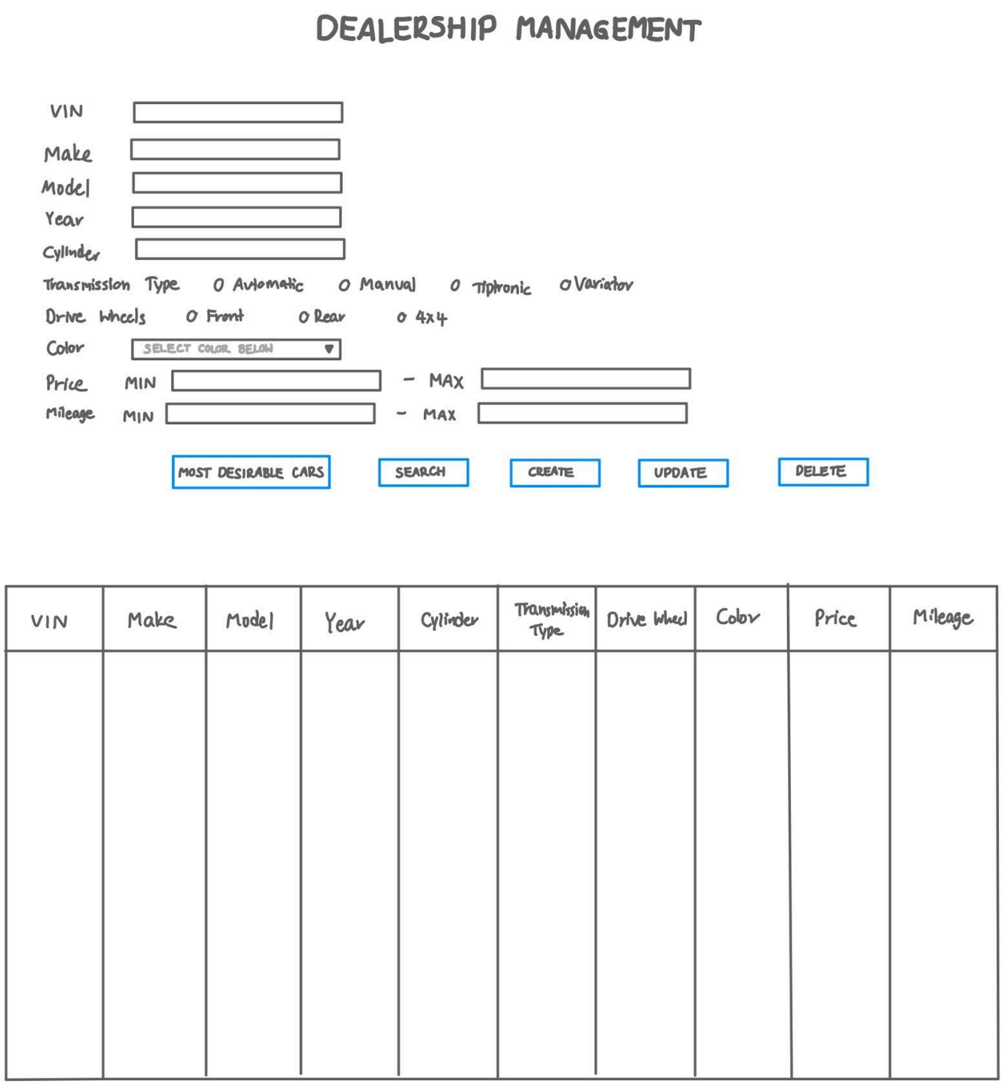
These features allow dealership employees to manage inventory efficiently, while also giving them information about cars that could sell better than the rest.

Realness

1. Craigslist vehicles Dataset :
Source: Kaggle
Format: CSV
Cardinality: 426880
Degree: 28
Information Captured: This dataset includes used car listings with details like price, make, model, mileage, year, and location. We will use this data to analyze market trends and competition, particularly for used cars, helping dealerships set appropriate pricing strategies.

2. Carvana Car Daily Sales for United States
Source: Kaggle
Format: CSV
Cardinality: 44366
Degree: 13
Information Captured: This dataset consists of details on daily Carvana cars sold in United States from Carvana. Data fields include vehicle_id, make, year, model, miles, trim, sold price, discounted sold price, partnered dealership, delivery fee, earliest delivery date, sold date.

Low-fidelity UI mockup



Project Work Distribution

There’s five main functionalities relating to the database: adding entries, searching entries, updating entries, deleting entries, and returning the most popular cars.

Each member will be tasked with developing the SQL query or procedure for either adding entries, searching entries, updating entries, deleting entries

Backend Distribution

1. Database Schema & Table Design

    Pitupoom

    - Tasks:
        - Create Tables: Set up tables for cars, users, transactions, and popularity metrics.
        - Define Relationships: Establish links between tables using primary and foreign keys.
        - Set Constraints: Apply rules to ensure data accuracy
        - Manage Schema Changes: Handle updates and modifications to the database structure as the project evolves.

2. Search Functionality & Indexes

    Matupoom

    - Tasks:
        - Create Indexes: Add indexes on important columns (e.g., make, model, year) to speed up search operations.
        - Build Search Queries: Develop queries that allow users to search by various filters like make, model, price range, and year.
        - Add Functionality: Develop the logic for adding new car records to the database, including validation and data integrity checks.

3. Database Transactions & Update/Delete Operations

    Alondra

    - Tasks:
        - Update Functionality: Implement functionality to update existing car records based on user input.
        - Delete Functionality: Handle the removal of car records from the database, ensuring accurate updates and data integrity.
        - Error Functionality: Implement mechanisms to handle and log errors during these operations.

4. Data Analytics & Popularity Metrics Storage

    Praise

    - Tasks:
        - Load Kaggle Data: Import data from Kaggle datasets into the database.
        - Combine Data: Merge sales data with inventory data to analyze popularity.

    - Generate Metrics: Generate popularity metrics based on combined sales data.