Car Sales Database – Data Generation and System Design Report

Student Name: Ashok Yarra **Student ID:24091071**

1. Introduction

This project presents the design, development, and population of a fully functional **Car Sales Database** using **SQLite** and **Python (with Faker)**. The goal was to construct a realistic, multi-table relational database that satisfies academic requirements such as data normalisation, inclusion of all measurement scales (nominal, ordinal, interval, ratio), use of foreign keys, and the implementation of at least one composite key.

The database simulates the operational environment of a real car dealership network, containing dealerships, customers, cars, sales transactions, instalment-based payments, and customer test-drive history. The dataset was generated programmatically to create rich, realistic, and diverse synthetic values suitable for demonstrating SQL querying, relational joins, data analytics, and integrity constraints.

2. Project Rationale and Objectives

The overarching aim was to build a database that:

- Represents real-world car retail operations
- Uses a **multi-table relational structure** with appropriate key constraints
- Includes **nominal, ordinal, interval, and ratio** data types
- Contains **large-scale generated data** (1500 customers, 600 cars)
- Enables complex analytical SQL queries
- Demonstrates professional data modelling aligned with Data Science standards

The final system contains **six interrelated tables** with more than **4,000 total records**, fulfilling the assignment's requirement for a substantial and meaningful dataset.

3. Schema Overview and Table Justification

The database schema comprises the following tables:

3.1 Dealerships

Stores dealership information such as name, location, phone number, and active/inactive status.
Purpose: Represents the business entities that own and sell vehicle inventory.

3.2 Customers

Contains customer data with names, emails, cities, registration dates (interval), and credit bands (ordinal).
Purpose: Models the buyers interacting with dealerships.

3.3 Cars Inventory

Includes detailed car attributes such as make/model (nominal), year_of_make (interval), engine size and list price (ratio), mileage (ratio), fuel type, transmission, and availability status.
Purpose: Captures all vehicles offered across dealerships.

3.4 Sales Orders

Represents completed vehicle purchases by customers.
Includes sale price, discount amounts, payment method, warranty years, and satisfaction score.
Purpose: Connects customers, dealers, and inventory with a transactional record.

3.5 Payments (Composite Key)

Implements instalment-based payments using a **composite primary key (sale_id, instalment_no)**.
Purpose: Demonstrates advanced relational modelling while supporting multi-payment structures.

3.6 Test Drives

Contains historical test-drive events, feedback scores (ordinal), and indicators for whether the test drive converted into a purchase.
Purpose: Adds behavioural and marketing insights into customer activity.

The schema uses **foreign keys** extensively and maintains data consistency across all interactions.

4. Data Generation Approach

All data was generated using Python with the Faker library to ensure realistic names, email addresses, VIN-style codes, colours, cities, dates, and dealership information. The database was created using 11 structured notebook cells, each focusing on a single step such as schema creation, table insertion, or final row count validation.

Key characteristics of data generation:

- **1500+ customers:** Satisfies the "large table" requirement.
- **600 cars:** Each car includes detailed attributes and unique VINs.
- **420 sales orders:** Approximately 70% of inventory is sold.
- **1047 payments:** Generated using a 1–4 instalment system; composite key enforced.
- **1409 test drives:** Provides a strong behavioural dataset.
- **Realistic constraints:** Mileage increases with age, discounts capped at 20%, warranty 0–7 years.

Python was also used for data cleaning steps (e.g., forcing positive payment amounts), ensuring no invalid rows entered the dataset.

5. Mapping of Attributes to Data Types

Data Type	Attributes Used in Database	Example Tables		Why This Type?
Nominal	Customer names, city, email, car make, car model, fuel type, body type, payment method	customers, cars_inventory, sales_orders		Labels or categories with no numeric meaning.
Ordinal	Credit band (Poor → Excellent), satisfaction score (1–5), test-drive feedback score (1–5)	customers, sales_orders, test_drives		Ranked values where order matters but distance is not equal.
Interval	Registration date, sale date, year of manufacture	customers, sales_orders, cars_inventory		Numeric scale with meaningful differences, but no true zero.
Ratio	Mileage, list price, engine size, sale price, discount, payment amount	cars_inventory, sales_orders, payments		Continuous numeric values with a true zero and equal intervals.

6. Verification and Evidence (Screenshots)

Four SQL queries were executed in DB Browser for SQLite to verify data correctness, demonstrate functionality, and provide evidence for inclusion in the report.

Figure 1: Table Names and Row Counts

This query displays the number of records in each table, confirming large-scale data generation and successful population of all entities.

Figure 2: Sample Customer Records

Shows realistic customer information with mixed data types (nominal, interval, ordinal).
Demonstrates the natural-looking data generated through Faker and validates the correctness of the customers table.

Figure 3: Multi-Table Joined Sales View

A powerful join across **sales_orders**, **customers**, **cars_inventory**, and **dealerships**, showing consolidated information including customer names, car details, discounts, payment methods, and satisfaction.
This showcases referential integrity and relational design strength.

Figure 4: Revenue and Discounts by Car Brand

An analytical GROUP BY query summarising total revenue, average discount, and satisfaction by manufacturer.
This demonstrates the database's analytical capability and practical business insights.

7. Final Row Counts

After execution, the database achieved:

- **Dealerships:** 15
- **Customers:** 1500
- **Cars Inventory:** 600
- **Sales Orders:** 420
- **Payments:** 1047
- **Test Drives:** 1409

These values confirm a rich, diverse, and fully functional relational dataset ready for analysis.

8. Conclusion

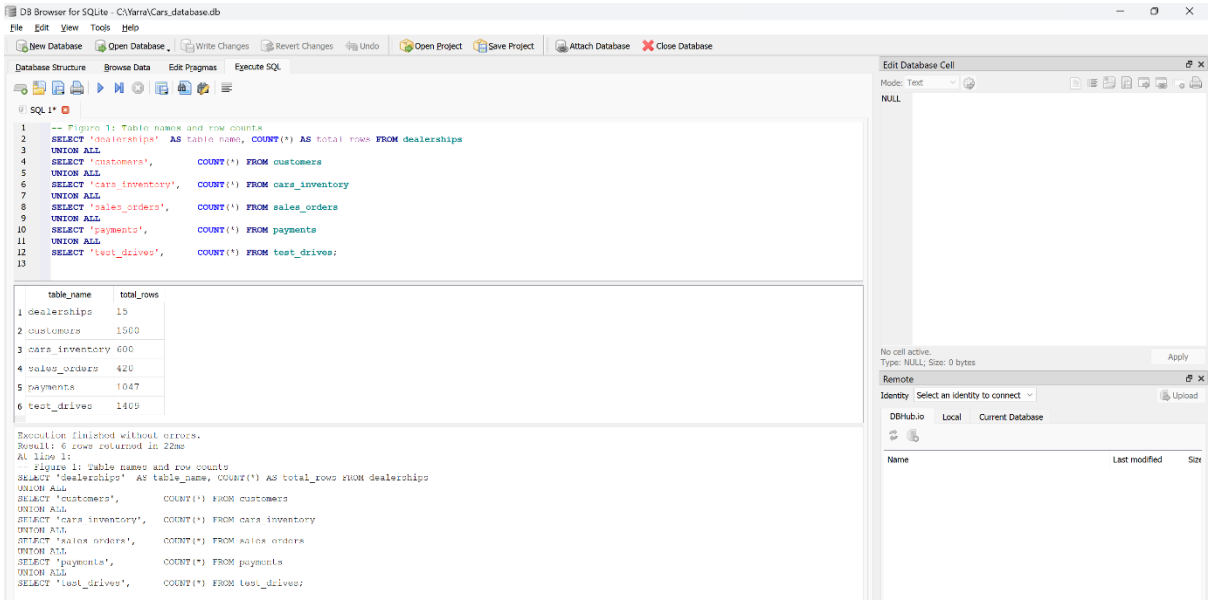
This project successfully delivered a complete and realistic car sales database that satisfies all the required academic criteria. The schema is well-designed, normalised, and supports complex relational queries. The use of Python and Faker enabled the generation of high-quality synthetic data, and the inclusion of a composite key, extensive foreign key relationships, and multiple data types demonstrates strong understanding of relational database principles.

The SQL queries executed in DB Browser provide clear evidence of database integrity, realistic content, and analytical potential. Overall, this work represents a robust, scalable, and professionally structured database system suitable for advanced coursework in Data Science and database engineering.

Appendix section:

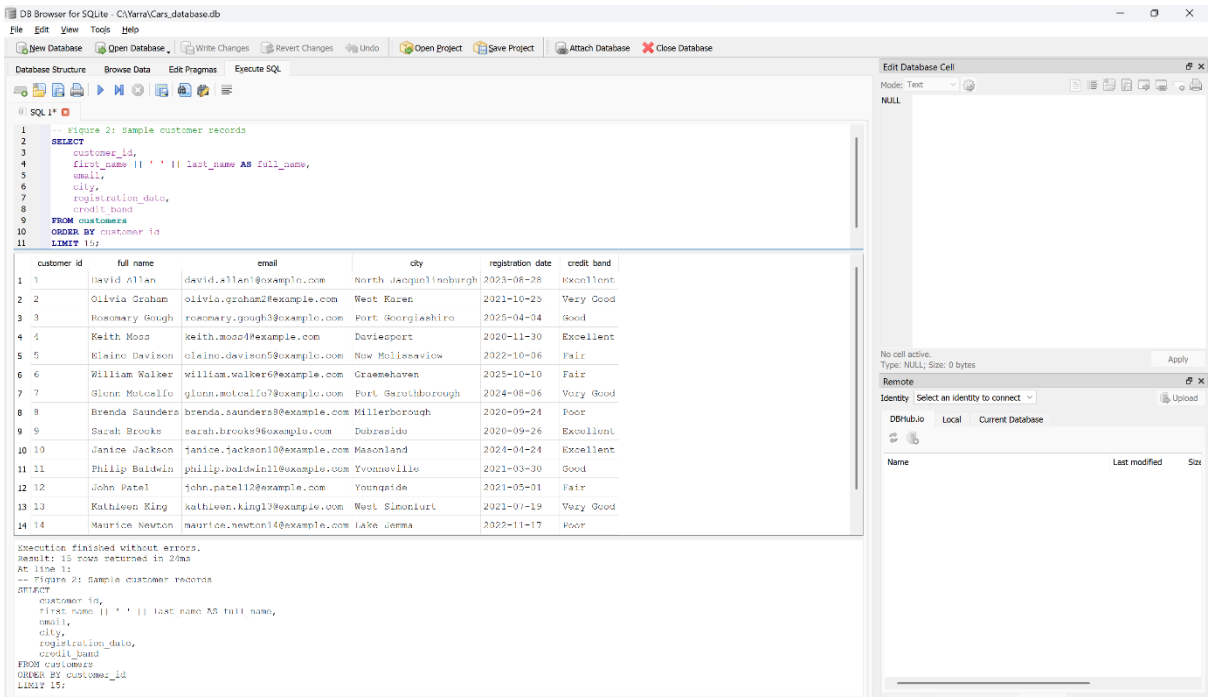
1. Row Count Screenshot

“Figure 1 displays the total number of records in each table, confirming successful population of the entire car sales database.”



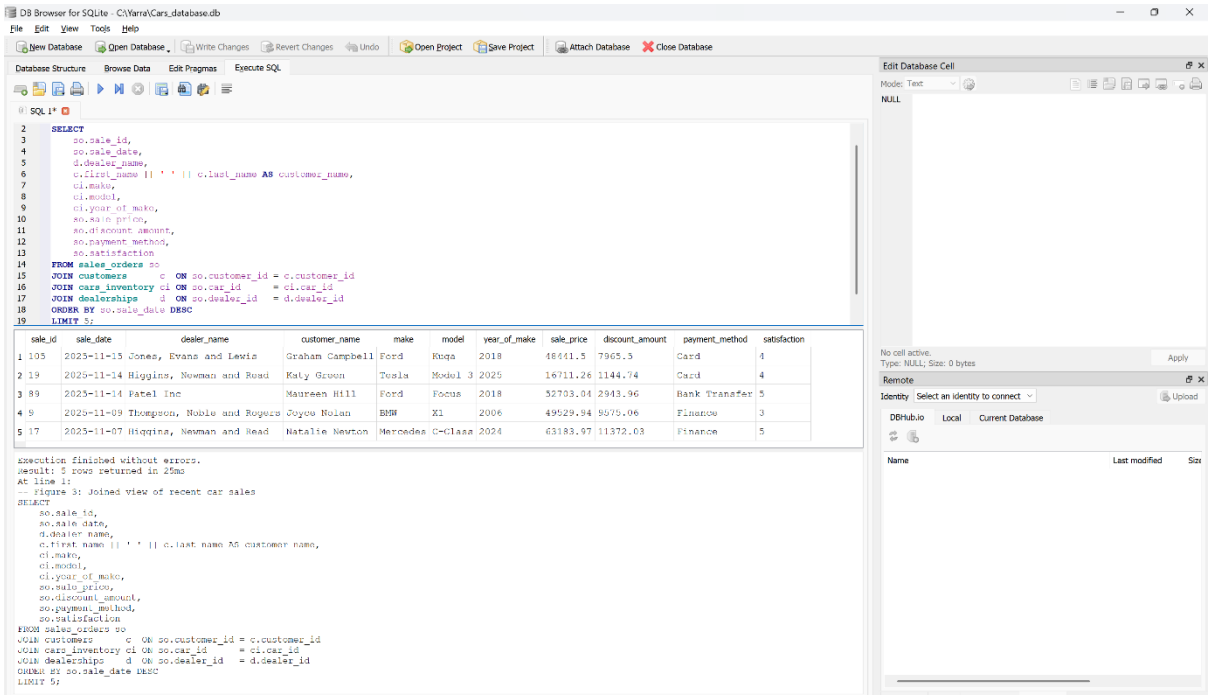
2. Customers Sample Screenshot

“Figure 2 shows a sample of customer records, demonstrating realistic nominal, interval, and ordinal data generated using Faker.”



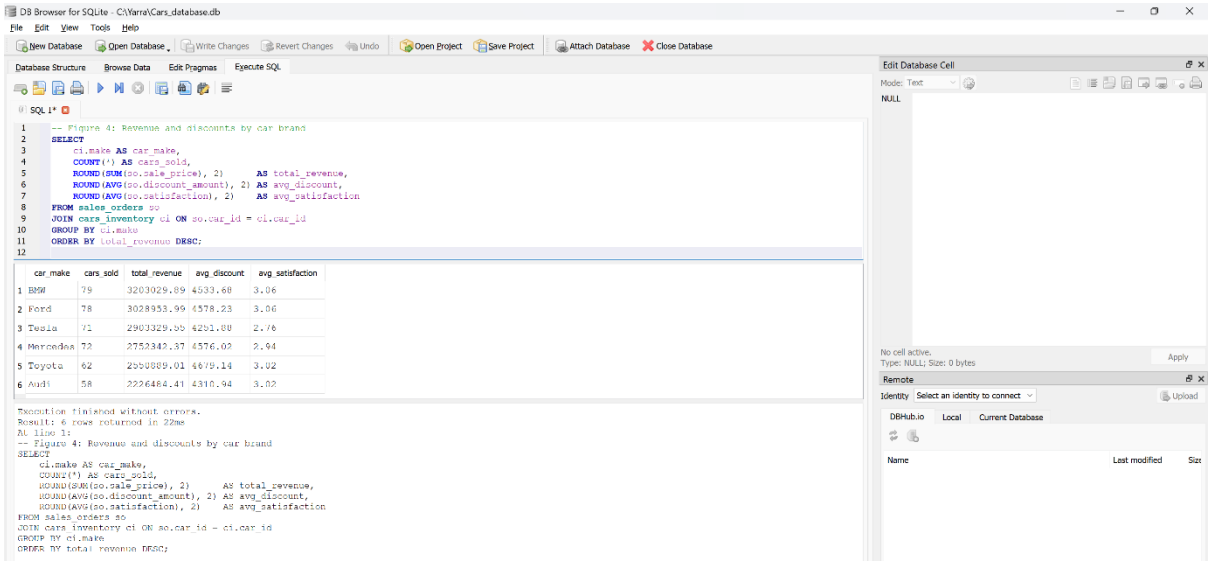
3. Sales Join Screenshot

“Figure 3 presents a joined view of recent car sales combining customer, car, dealership, and transaction attributes.”



4. Revenue by Brand Screenshot

"Figure 4 summarises brand-level performance, highlighting total revenue, discounts, and customer satisfaction for each car make."



GitHub Link: https://github.com/ashok-yarra/SQLite_Project/blob/main/Cars_DataBase.ipynb