Data Warehouse and Visualization Project Report: Vehicle Sales Analysis

Yaekob Beyene Yowhanns
Matricola: 269860

July 2, 2025

Project Overview

This report details the design and implementation of a data warehouse, a central repository for integrated data from one or more disparate sources, used to support reporting and data analysis, and data visualization, the graphical representation of information and data to facilitate understanding and analysis. The project focuses on analyzing vehicle sales data, following the project development checklist provided for the Data Warehouse course. The objective is to create a robust data warehouse capable of supporting various analytical queries and providing insights into vehicle sales trends, which will then be communicated through effective data visualizations.

1 Phase 1: Design

1.1 Data Source Selection

The selected data source is the Kaggle Vehicle Sales Data dataset, chosen for its comprehensive coverage. It provides rich information suitable for constructing the dimensions, measures, and hierarchies required for in-depth vehicle sales analysis.

Dataset Description:

- The "Vehicle Sales and Market Trends Dataset" provides a comprehensive collection of information pertaining to the sales transactions of various vehicles. This dataset encompasses details such as the year, make, model, trim, body type, transmission type, VIN (Vehicle Identification Number), state of registration, condition rating, odometer reading, exterior and interior colors, seller information, Manheim Market Report (MMR) values, selling prices, and sale dates.[1]
- Shape of the dataset is: (558837, 16)[1]

1.2 Data Re-engineering and Reconciled Database Schema Design

Given that the original data source is a CSV file, a re-engineering step was performed to compose a database schema (Relational Model) fitting the data. This process was crucial for understanding the data's structure and imposing a well-defined organization. The resulting schema, referred to as the "schema of the reconciled database", serves as the foundation for the data warehouse.

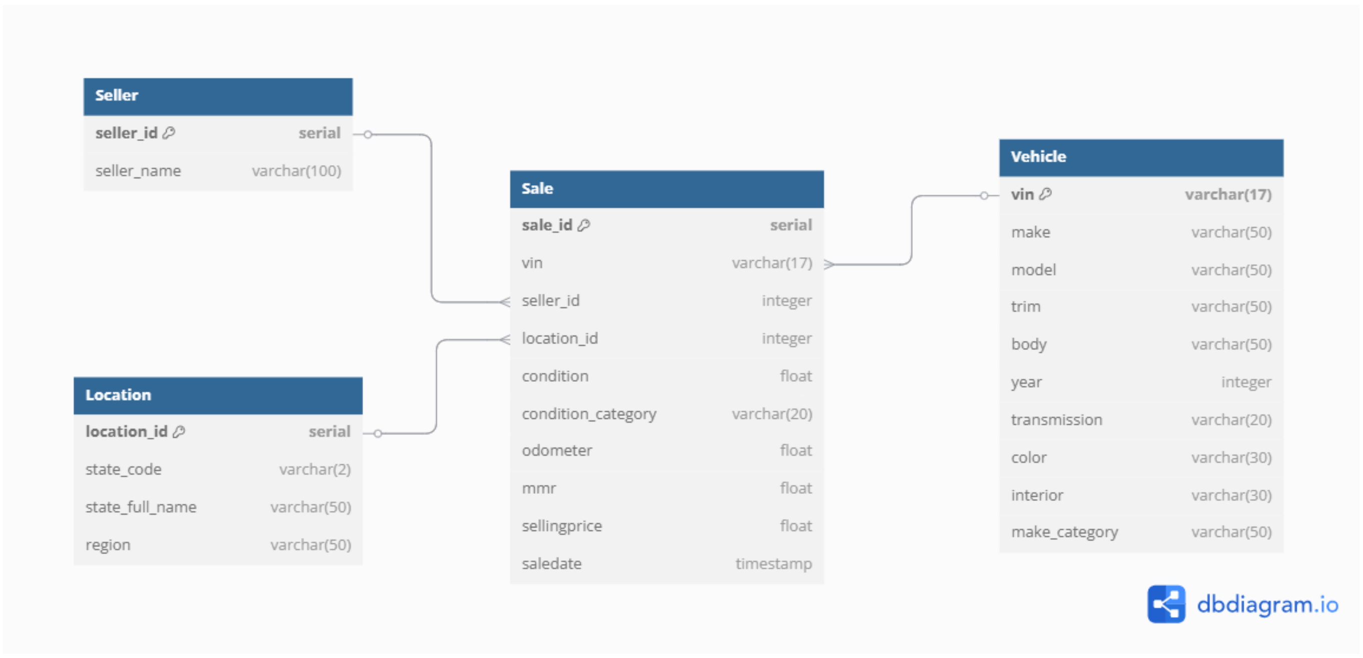


Figure 1: Schema of the reconciled database .

The schema for my dataset is shown in figure 1

1.3 Building the Reconciled Database

The schema for the reconciled database was built on a DBMS (Database Management System) of choice. Specifically, PostgreSQL was used to create the reconciled database. This instance of the database is referred to as the ”reconciled database”.

1.4 Conceptual Design of Fact Schema

The conceptual design involved the following steps:

- **Fact Selection:** The ”Sales” entity was chosen as the fact of the reconciled database.
- **Attribute Tree and DFM Schema:** The attribute tree was constructed as shown in figure 2 and all subsequent steps were followed to obtain a DFM Schema (dimensional factor model). This model conceptually represents the business process and the measures and dimensions involved.

Method Used: Grafting to transform attribute tree into Dimensional Fact Model (DFM)

To convert the initial attribute tree into a structured **Dimensional Fact Model (DFM)**, the **grafting** method was applied. Grafting involves *adding hierarchical structure* by logically grouping related attributes into dimensions and organizing them into meaningful levels of granularity.

Key Steps in the Transformation

1. **Identify Core Entities**
   Attributes were grouped based on logical entities such as *Vehicle, Location, Seller, Sale*
2. **Build Dimension Hierarchies (Grafting)**[3]
   - **Vehicle Dimension:** A hierarchy was grafted: `make_category` → `make` → `model` → `trim`, along with related attributes like `color, body, transmission, etc.`
   - **Location Dimension:** Structured as: `region` → `state_full_name` → `state_code`.

- **Date Dimension:** Derived hierarchy from `saledate`: `year` → `quarter` → `month` → `day`.

3. **Define the Fact**
   A central `Vehicle_Sale_Fact` fact was created to store measurable facts such as:

   - `sellingprice (SUM, AVG)`
   - `mmr (SUM,AVG)`
   - `odometer (SUM,AVG)`
   - `condition (AVG)`
   - `COUNT(*)`

The constructed Dimensional fact model schema is shown in Figure 4

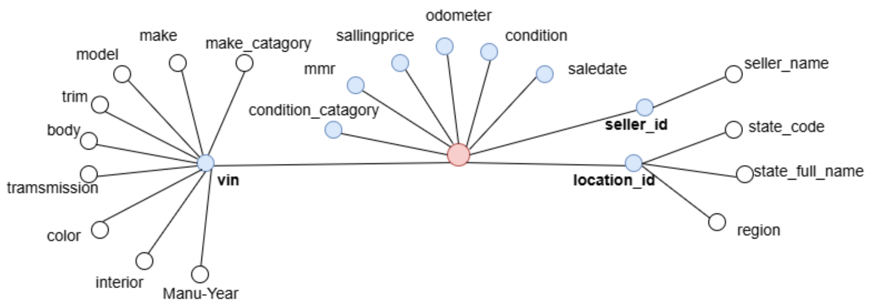


Figure 2: Atrribute Tree

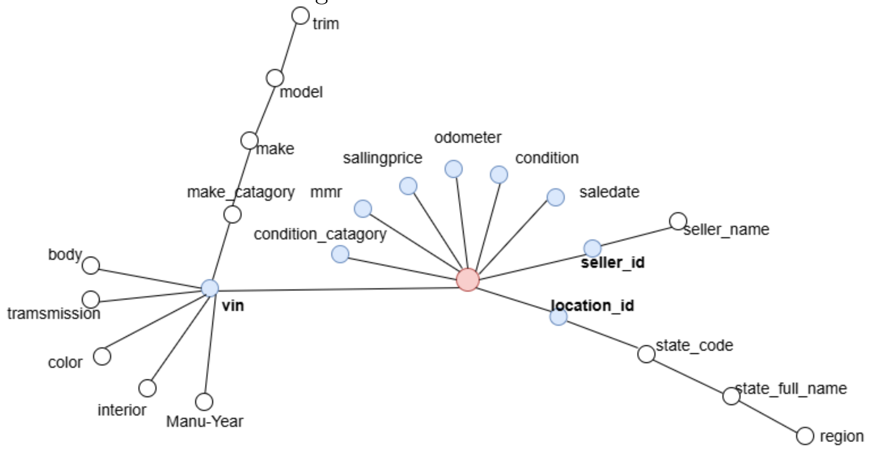


Figure 3: Edited attribute tree

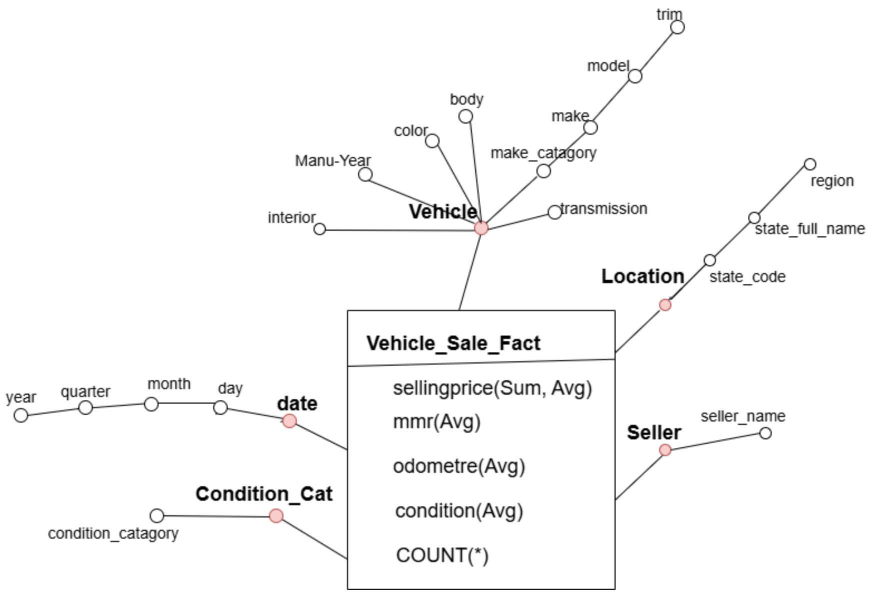


Figure 4: Dimensional Fact Model Schema

1.5 Logical Design (Star Schema)

The logical design phase involved translating the previously defined Dimensional Fact Model (DFM) into a physical **Star Schema** and implementing it using **PostgreSQL** as the database management system.

a. Translation of the DFM into a Star Schema

The DFM was mapped into a central fact table named `fact_sales`, connected to five well-structured dimension tables:

- `dim_vehicle`: Includes vehicle-related attributes such as `make`, `model`, `trim`, `body`, `year`, `color`, `interior`, `transmission`, and `make_category`.
- `dim_location`: Stores geographical data including `state`, `state_full_name`, and `region`.
- `dim_time`: Represents time-related attributes derived from `saledate`, such as `year`, `quarter`, `month`, and `day`.
- `dim_seller`: Contains `seller_id` and `seller_name`.
- `dim_condition`: Provides vehicle condition classification through `condition_category`.

The fact table `fact_sales` contains the following measurable facts:

- `sellingprice` (float)
- `mmr` (float)
- `odometer` (float)
- `condition` (float)

It also includes foreign keys referencing each dimension: `vehicle_id`, `location_id`, `time_id`, `seller_id`, and `condition_id`. The possible Star Schema design is shown in figure 5

b. Schema Implementation in PostgreSQL

The star schema was implemented in a PostgreSQL database, referred to as the *data warehouse*. Tables were created with appropriate data types, primary keys, and foreign key constraints to ensure referential integrity.

This design supports:

- Efficient OLAP queries
- Scalability for large datasets
- Drill-down, roll-up, slicing, and dicing across dimensions

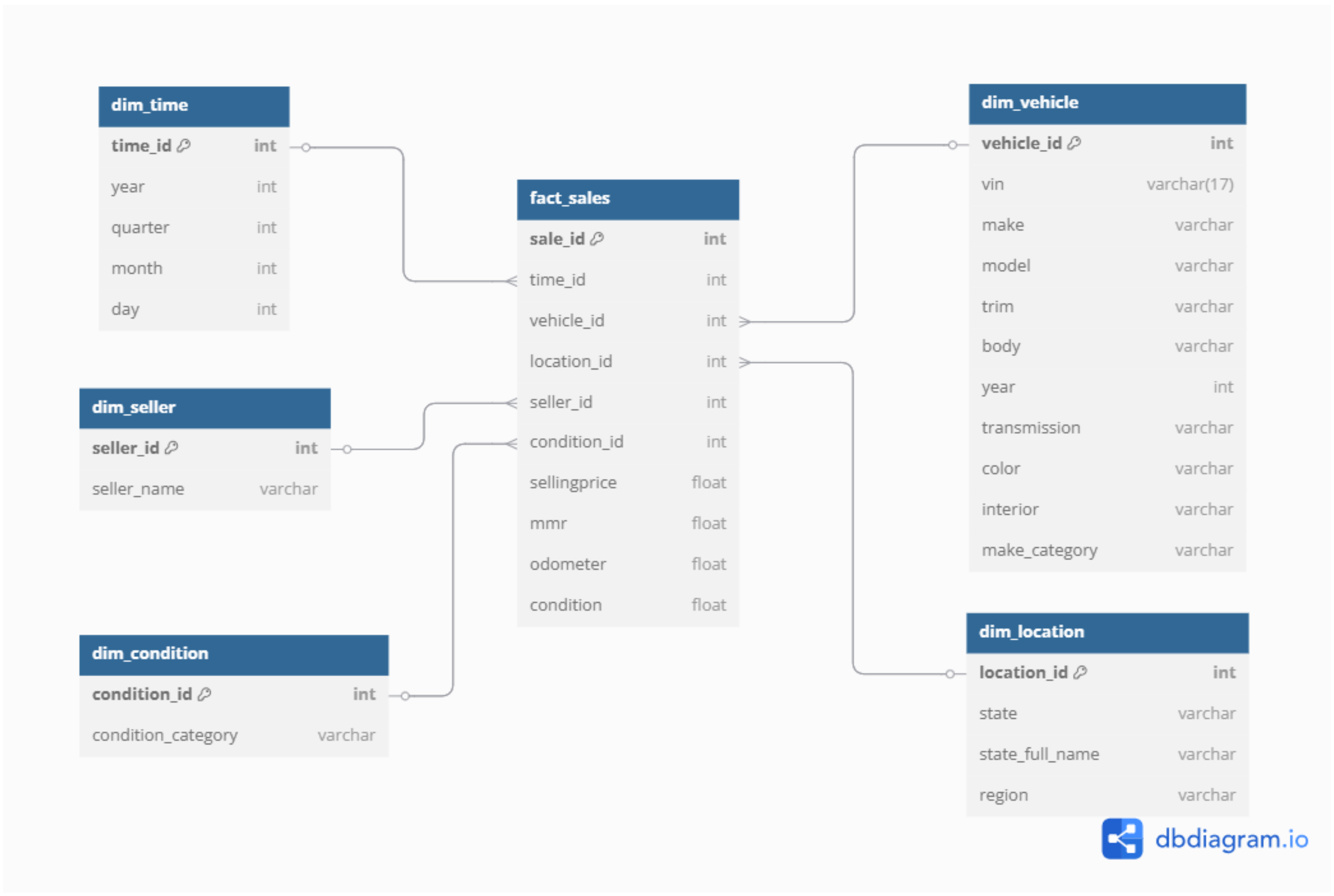


Figure 5: Star schema desingn

2 Phase 2: Data Management

2.1 Populating the Reconciled Database

The reconciled database, created during **Phase 1 – Step 3**, was populated using the cleaned dataset from **Phase 2 – Step 2**. The cleaned CSV file (`exported after data cleaning`) served as the data source.

The data was loaded into a PostgreSQL database using Python, utilizing the `pandas` and `SQLAlchemy` libraries to automate the process. Tables such as `seller`, `location`, and `vehicle` were populated by extracting unique records based on relevant attributes. The `sale` table was filled with transactional data and linked to tables through appropriate foreign key mappings.

This ensured a structured and consistent dataset within the `reconciled` schema, ready for the subsequent ETL phase.

2.2 Data Cleaning Design and Implementation

The original dataset consisted of 558,837 records and 16 attributes[1]. To ensure the quality and consistency of the data for analytical purposes, a comprehensive cleaning was performed using Python (pandas) before loading it into the reconciled database. After cleaning, the dataset was reduced to 465,248 records with 20 refined columns.

Key Cleaning Tasks Performed

1. **Handling Missing Values**
   Records with missing values in critical columns (e.g., `make`, `model`, `vin`, `sellingprice`) were

removed. Less critical fields like `transmission` were filled with the placeholder "Unknown".

2. **Filtering by Manufacturing Year**
   Vehicles produced before 2000 were excluded, as they represented only 1.39% of the data and to make an analysis on recent vehicles.

3. **VIN Validation**
   Only records with valid 17-character VINs were retained, following industry standards.

4. **Standardization and Cleanup**
   Text fields were cleaned and standardized. Invalid or corrupted entries in columns such as `model`, `trim`, `interior`, and `color` were removed.

5. **Duplicate Handling**
   Duplicate sale records (same `vin`, `saledate`, and `seller`) were eliminated. Valid multiple sales were preserved. Conflicting vehicle attributes for the same VIN were excluded.

6. **Outlier Removal**
   Unreasonable values in `odometer`, `mmr`, and `sellingprice` were filtered out using logical thresholds.

7. **Feature Engineering**
   New attributes were introduced, including:

   - `condition_category` (based on condition scores)
   - `state_full_name` and `region` (geographical enrichment)
   - `make_category` (classified by origin, market segment, and volume)

8. **Date and Year Consistency**
   Sale dates and manufacturing years were validated for logical correctness. Invalid entries were removed.

Outcome:

The cleaned dataset comprises **465,248 complete, high-quality records** across **20 well-structured columns**, providing a robust foundation for ETL processes and downstream analytical tasks.

2.3 ETL Process Design and Implementation

The ETL (Extract, Transform, Load) process was implemented using `Python` with the `pandas` and `SQLAlchemy` libraries to transfer data from the `reconciled` schema to the `dw` (data warehouse) schema in `PostgreSQL`. I Used Python (pandas, SQLAlchemy) for ETL due to its programmatic flexibility, enabling efficient transformation and error handling compared to Pentaho.

Extract

A joined dataset was retrieved from the `sale`, `vehicle`, `seller`, and `location` tables in the `reconciled` schema. This dataset included all relevant vehicle attributes, sales data, and contextual information such as location and seller.

Transform

The extracted data was processed to generate dimension tables:

- `dim_time`: derived from `saledate` with year, quarter, month, and day
- `dim_location`: unique state, region, and full state names
- `dim_vehicle`: based on unique VINs and vehicle attributes
- `dim_seller`: cleaned and deduplicated seller names

- `dim_condition`: distinct condition categories

Foreign keys were mapped by joining with the respective dimension tables.

Load

A fact table `fact_sales` was created using the mapped keys and key measures (`sellingprice`, `mmr`, `odometer`, `condition`). Each row was assigned a surrogate `sale_id`.

Outcome:

All dimension tables and the fact table were successfully populated in the `data warehouse(dw)` schema, resulting in a clean, fully structured star schema ready for analytical querying and OLAP operations.

3 Phase 3: Data Visualization[3]

The final phase of the project involved designing and building interactive dashboards using **Tableau**. The purpose of this phase was to communicate key insights discovered during analysis and support **OLAP-style dynamic exploration**, such as drill-down, slice, dice, and roll-up. The dashboards were designed to serve business decision-makers, such as dealerships, sales managers, and market analysts.

3.1 Dashboard Overview

Three dashboards were created to cover different analytical perspectives, arranged to tell a logical story from macro to micro-level analysis.

3.1.1 Dashboard 1: National Vehicle Sales Trends

This dashboard presents a **macro-level overview** of vehicle sales over time and geography.

- KPI cards show **total vehicles sold**, **total sales amount**, and **number of sellers**.
- A **time series** visualizes sales trends, with drill-down from **year** → **quarter** → **month** → **day**.
- A **geographic map** displays vehicle sales by state, helping identify high-performing regions.
- Additional charts show the distribution of **model years** and **vehicle body types**.

OLAP used: Roll-up, drill-down, slice by region/year/model.
Business question: Where, when, and how are vehicles being sold across the U.S.?

3.1.2 Dashboard 2: Vehicle Pricing and Market Value

This dashboard analyzes the relationship between **odometer reading**, **vehicle condition**, **market value (MMR)**, and **actual selling price**.

- A **bar chart** compares the average price of newer and older vehicles
- A **scatter plot with trend line** shows a statistically significant **negative correlation** between mileage and selling price.
- A **dual-axis chart** compares actual prices against the MMR benchmark.
- Users can toggle between **area and line charts** to explore price trends by condition.
- Drill-downs and filters allow analysis by **model year**, **condition**, and **make**.

OLAP used: Drill-down, toggle (pivot), slice by condition.
Business question: What affects the price of used vehicles?

3.1.3 Dashboard 3: Market Players and Buyer Preferences

This dashboard focuses on the market itself—**who is selling** and **what buyers prefer**.

- A **Top Sellers** bar chart shows leading dealerships with color indicating average selling prices.
- A **100% stacked bar chart** visualizes popular vehicle colors in sales.
- A **donut chart** illustrates overall transmission preferences.
- A **hierarchical bar chart** allows drill-down from **Make_category** → **Make** → **Model**.

OLAP used: Drill-down, slice by brand/year, dice with multiple filters.
Business question: Which sellers dominate the market? What do buyers prefer?

3.2 OLAP Operations in Practice

The dashboards implement the following key OLAP operations:

Operation	Example
Drill-down	Year → Quarter → Month → Day; Make_category → Make → Model
Roll-up	Aggregating sales from State → Region , Day → Year
Slice	Filtering by Year, Make_Category, Region
Dice	Filtering combinations like 2015 + SUVs + Asian Luxury
Pivot	Toggle between chart types (Line vs. Area)

Table 1: Summary of OLAP operations and examples

4 LLM-Assisted Insight Generation for Data Visualization

As part of a practical lab on the use of large language models (LLMs) in data analytics, I applied the LLaMA 3 model (accessed via Groq API)[2] to assist in the exploration and dashboard design process for the vehicle sales dataset.

Using a sample of 50 records from the cleaned dataset, I converted the data into JSON format and fed it into the LLM alongside the given checklist for Phase 3. The prompt asked the model to analyze the data and propose visualizations, identify trends, and suggest how to present insights interactively.

Prompt Strategy

A structured prompt was used with the following goals:

- Determine key stories and trends from the sample
- Recommend Tableau charts and dashboard structure
- Support OLAP operations (drill-down, roll-up, slice, dice)
- Guide the outline of the final report and essay

LLM Insights

The LLM generated recommendations that aligned well with Tableau best practices and our expectations, such as:

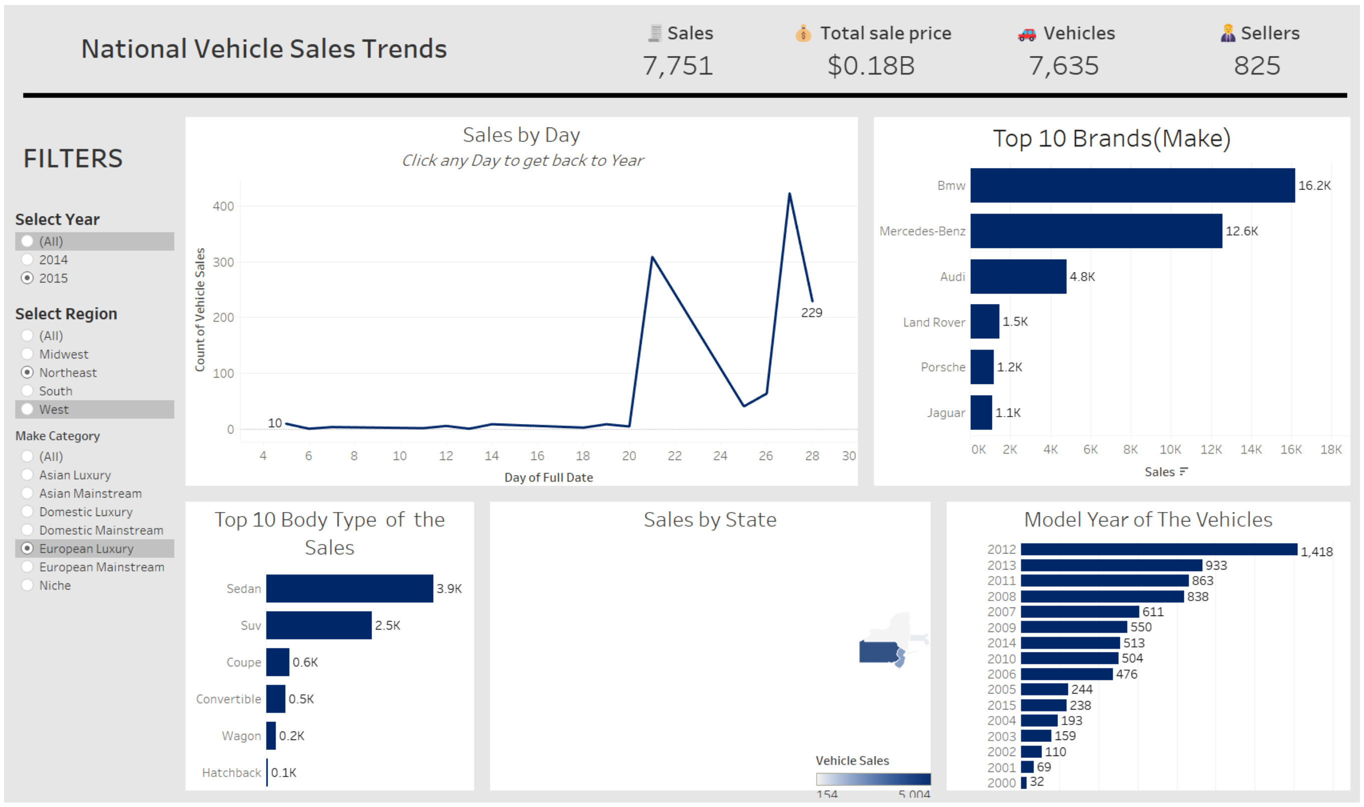
- A sales trends dashboard using time series and map visualizations
- A pricing insights dashboard using scatter plots and trend lines
- A buyer behavior dashboard focusing on color and transmission preferences
- The use of dynamic filters to enable OLAP-style interaction

Reflection

This exercise demonstrated how LLMs can assist in data warehousing tasks such as:

- Automating initial data profiling and storytelling
- Proposing meaningful visualizations
- Generating structured outlines and summaries

LLM tools are increasingly valuable in data-driven projects, especially when paired with visualization platforms like Tableau.



National Vehicle Sales Trends

Sales	Total sale price	Vehicles	Sellers
7,751	$0.18B	7,635	825

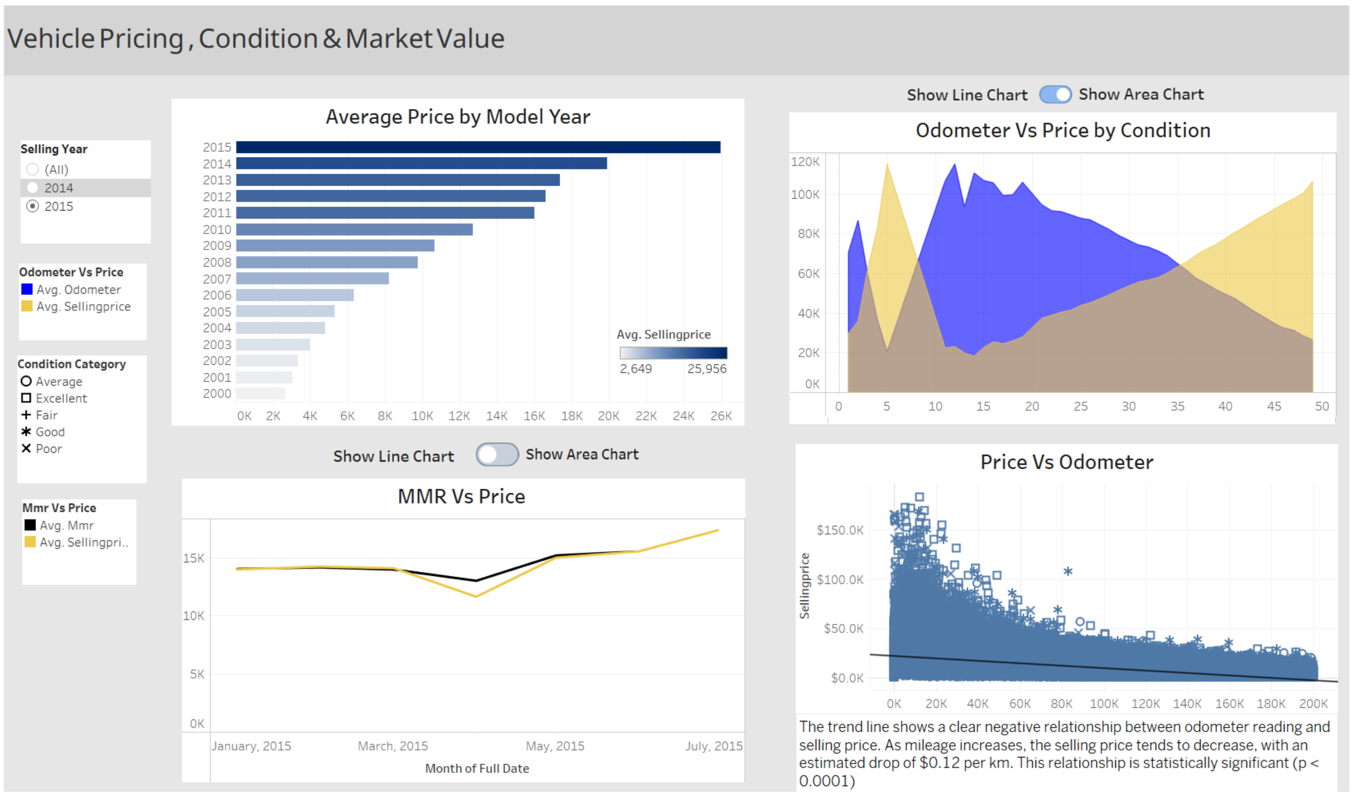
FILTERS

Select Year: (All) / 2014 / [✓ 2015]

Select Region: (All) / Midwest / [✓ Northeast] / South / West

Make Category: (All) / Asian Luxury / Asian Mainstream / Domestic Luxury / Domestic Mainstream / [✓ European Luxury] / European Mainstream / Niche

Figure 6: Dashboard 1



The trend line shows a clear negative relationship between odometer reading and selling price. As mileage increases, the selling price tends to decrease, with an estimated drop of $0.12 per km. This relationship is statistically significant (p < 0.0001)

Figure 7: Dashboard 2

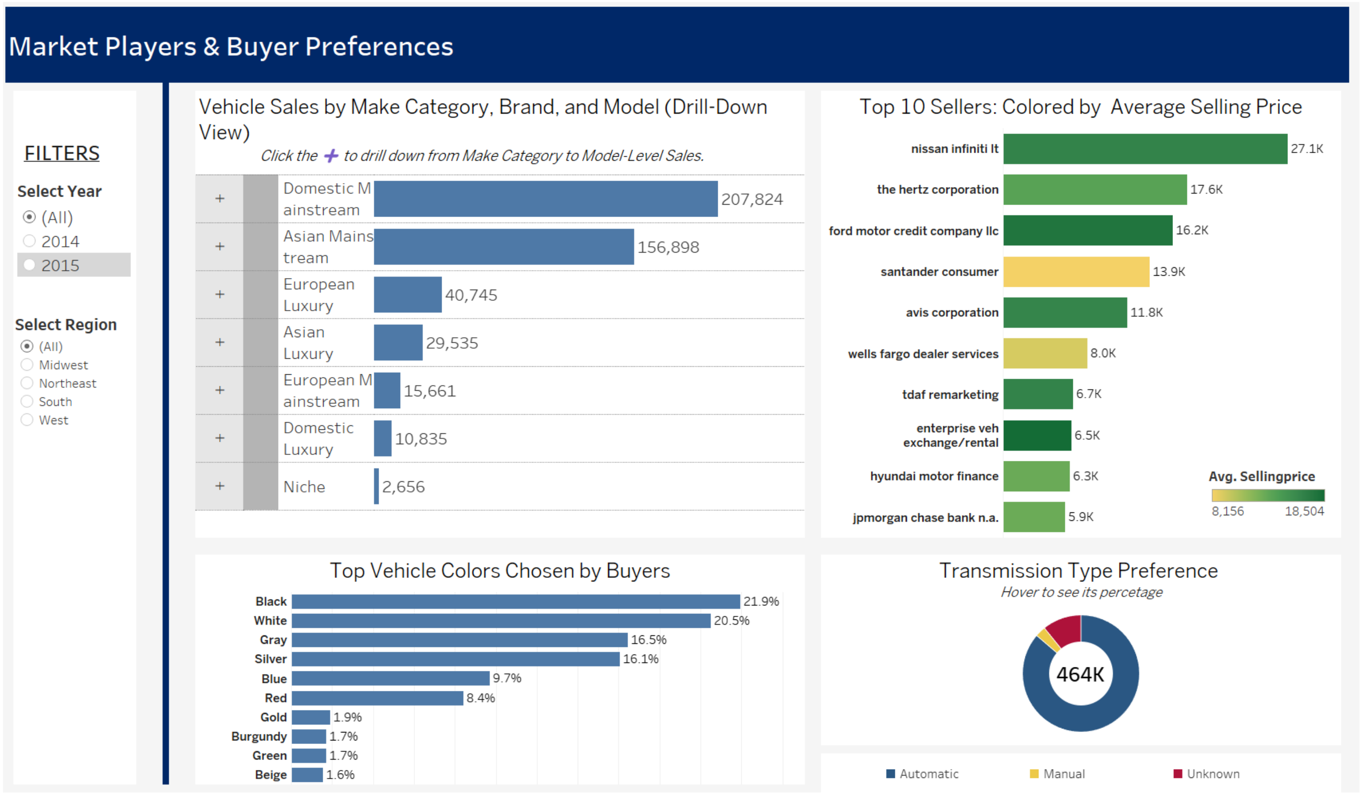


Figure 8: Dashboard 3

5 Conclusion

This project successfully covered the full lifecycle of a Data Warehouse and Visualization solution, from raw data to interactive insights. Beginning with the selection and re-engineering of a large real-world dataset, the work progressed through relational modeling, star schema construction, and full ETL implementation in PostgreSQL.

The final dashboards, built using Tableau, added a powerful, interactive layer of analysis. Users can explore sales trends by time and region, understand price influencers like mileage and market value, and examine buyer and seller behaviors. These visualizations integrate OLAP operations such as drill-down, slice, dice, and roll-up, enabling flexible exploration for business decision-makers.

Overall, the project demonstrates how data warehousing, when combined with well-designed dashboards, can transform raw sales data into actionable insights. A live demonstration will accompany this report during the exam presentation.

References

[1] Syed Anwar Afridi. Vehicle sales data. https://www.kaggle.com/datasets/syedanwarafridi/vehicle-sales-data, 2022. Accessed: 2025-05-20.

[2] Groq, Inc. Groq — high-performance ai compute platform. https://groq.com/, 2025. Accessed: 2025-06-14.

[3] Tableau Software. Tableau how-to videos. https://public.tableau.com/app/learn/how-to-videos, 2025. Accessed: 2025-05-20.