

Cars Database Creation and Analysis

Yash Chowdhary

Cars Database Creation and Analysis

Executive Summary.....	3
Data Description.....	4
Introduction.....	5
Analysis.....	5
MySQL Server and MySQL Workbench.....	5
Database and Table Creation.....	11
Data Import and Verification.....	11
Analysis Question 1: Identifying the Most Fuel-Efficient Cars.....	12
Analysis Question 2: Fuel Efficiency Trends Over Time by Origin.....	13
Analysis Question 3: Fuel Efficiency by Origin and Cylinder Configuration.....	14
Data Structure Analysis.....	16
Columns (Attributes).....	16
Appropriate MySQL Data Types.....	17
Business Questions and SQL Analysis.....	19
Question 1: Which cars have the best fuel efficiency (MPG)?.....	19
Question 2: How has average fuel efficiency changed over time for each country of origin?.....	20
Question 3: Which car origin produces the most fuel-efficient vehicles for different cylinder counts?.....	21
Conclusions.....	22
Achievement of Goals and Expectations.....	22
Pros and Cons of Tools and Methods.....	22
What Would Be Done Differently.....	23
References.....	24

Cars Database Creation and Analysis

Executive Summary

This report analyzes a comprehensive dataset of 406 car models to identify trends in fuel efficiency and performance characteristics across different manufacturing origins, time periods, and engine configurations. Our analysis reveals significant patterns in how fuel efficiency has evolved over time, particularly when comparing vehicles from different regions and with varying cylinder counts.

The findings suggest opportunities for manufacturers to optimize vehicle design by leveraging successful efficiency strategies from specific regions and engine configurations. The historical trends also provide valuable context for predicting future developments in automotive efficiency, which is increasingly important given rising fuel costs and environmental concerns.

Business Questions Addressed

The analysis seeks to answer three critical business questions:

1. **Fuel Efficiency Leaders:** Which specific car models demonstrate the highest fuel efficiency (MPG)? This helps identify best-in-class vehicles and their distinctive characteristics.
2. **Efficiency Evolution by Region:** How has average fuel efficiency changed over time for vehicles from different origins (America, Europe, Asia)? This tracks regional performance trends and competitive positioning in efficiency innovation.
3. **Cylinder Configuration Impact:** Which manufacturing origins produce the most fuel-efficient vehicles across different cylinder configurations? This examines the relationship between engine design choices and efficiency outcomes by region.

Data Description

The dataset contains detailed specifications for 406 car models with the following attributes:

- **Car:** Vehicle make and model name
- **MPG:** Fuel efficiency measured in miles per gallon
- **Cylinders:** Number of engine cylinders
- **Displacement:** Engine size measured in cubic inches
- **Horsepower:** Engine power output
- **Weight:** Vehicle weight in pounds
- **Acceleration:** Time to accelerate from 0-60 mph in seconds
- **Model:** Year of manufacture
- **Origin:** Country/region where the vehicle was manufactured

The comprehensive nature of this dataset allows for multidimensional analysis of how various vehicle characteristics correlate with fuel efficiency across different manufacturing regions and time periods. The data appears to span multiple model years, providing valuable historical context for understanding automotive efficiency trends.

Introduction

This assignment focuses on creating and analyzing a database containing information about various car models and their performance characteristics. The primary goal is to demonstrate proficiency in database design, SQL query development, and data analysis to extract meaningful insights from automotive data.

I have chosen the “cars.csv” dataset. The selected dataset contains information about different car models, including their performance metrics such as fuel efficiency, engine specifications, and physical characteristics. This dataset is particularly valuable for understanding how automotive technology has evolved over time and how different manufacturers approach vehicle design. SQL was used as the primary language for database creation, data manipulation, and analysis because of its powerful querying capabilities and widespread adoption in the data analytics field.

Analysis

MySQL Server and MySQL Workbench

Step-by-step installation process for MySQL Server and MySQL Workbench on my Windows operating system.

Step 1: Download MySQL Installer

Navigate to the official MySQL website (<https://dev.mysql.com/downloads/installer/>) and download the MySQL Installer for Windows. The installer includes both MySQL Server and additional components like MySQL Workbench.

Step 2: Launch the MySQL Installer

Locate the downloaded installer file and double-click to launch it. I needed to provide administrator permissions to proceed.

Step 3: Choose Setup Type

Selected the appropriate setup type. "Developer Default" option as it includes both MySQL Server and MySQL Workbench.

Step 4: Check Requirements

The installer checked if my system meets all requirements.

Step 5: Installation Progress

The installer installed MySQL Server and workbench.

Step 6: Set Root Password

Enter a password for the MySQL root account. This is the administrator account for MySQL, so choose a secure password. You can also create additional user accounts at this stage if needed.

Step 7: Configure Windows Service

Configure the Windows service settings for MySQL.

Step 12: Configuration Complete

Once the configuration was complete, clicked "Finish" to close the configuration wizard.

Step 13: Launch MySQL Workbench

Launch MySQL Workbench from the Start menu or desktop shortcut.

Step 14: Connect to MySQL Server

On the MySQL Workbench home screen, you'll see the local MySQL Server instance. Click on it to connect to your MySQL Server.

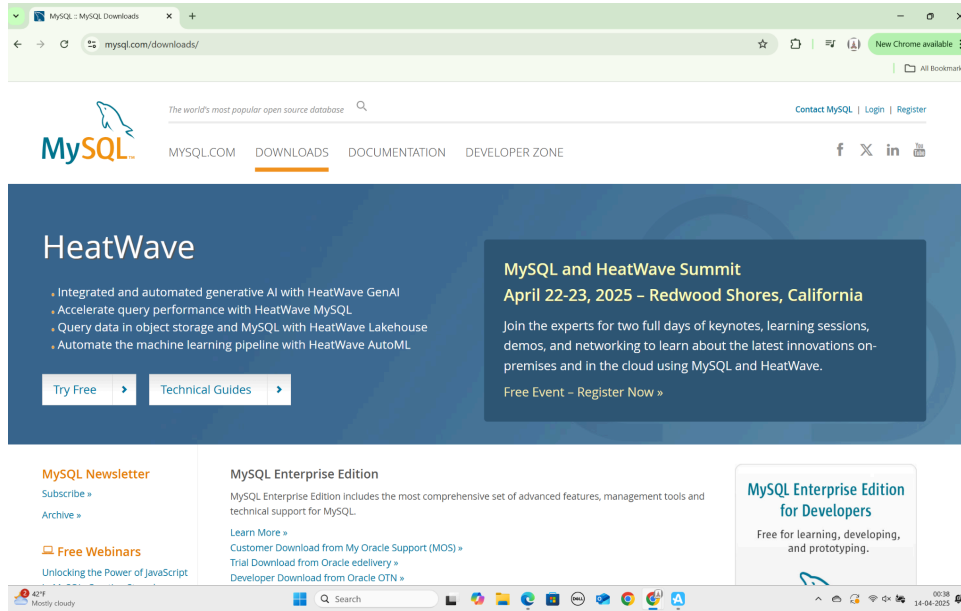
Step 15: Enter Root Password

Enter the root password you set during the MySQL Server installation and click "OK" to connect.

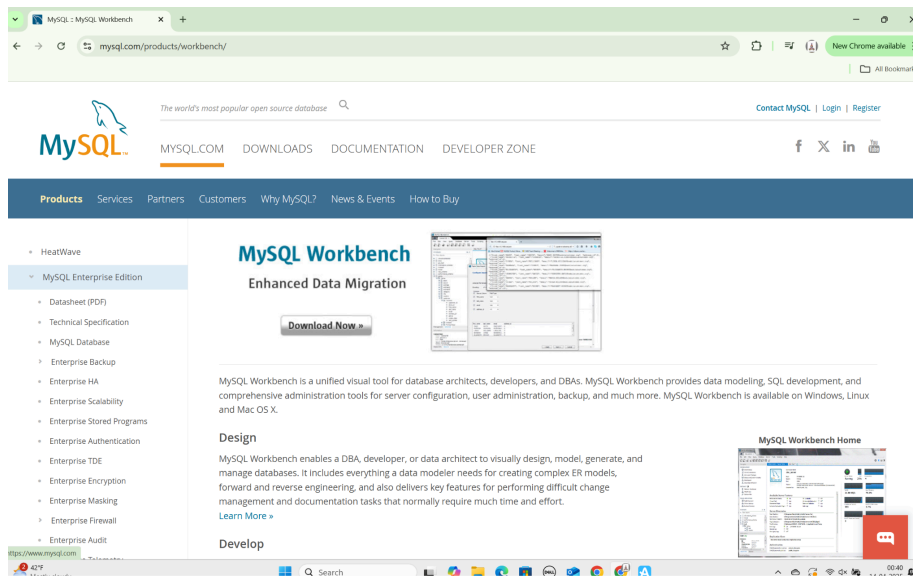
Step 16: MySQL Workbench Dashboard

Once connected, you'll see the MySQL Workbench dashboard. From here, you can manage your MySQL Server, create and manage databases, execute SQL queries, and more.

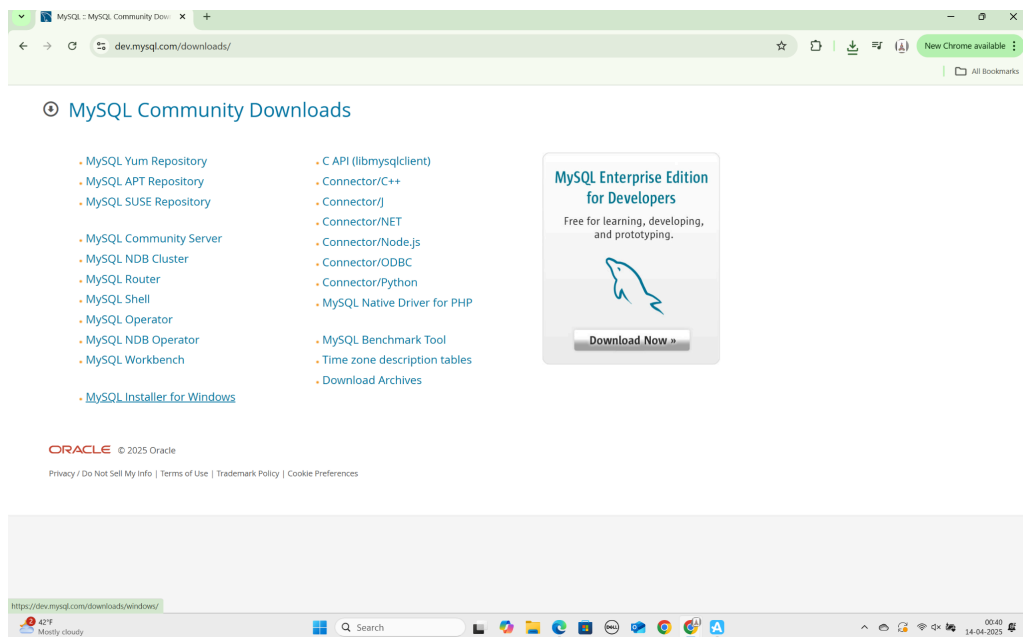
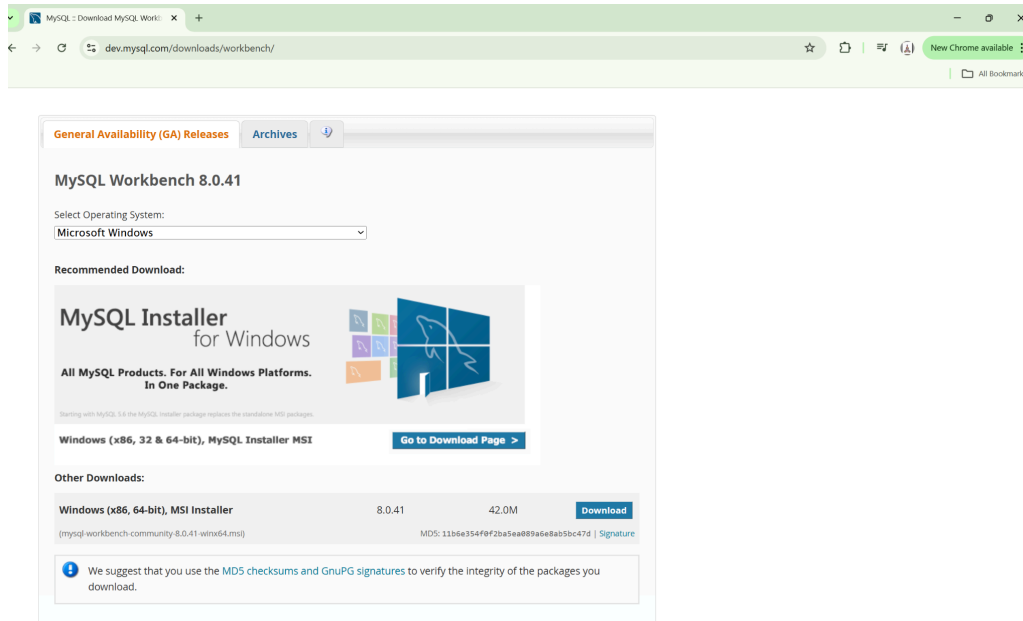
Related Screenshots

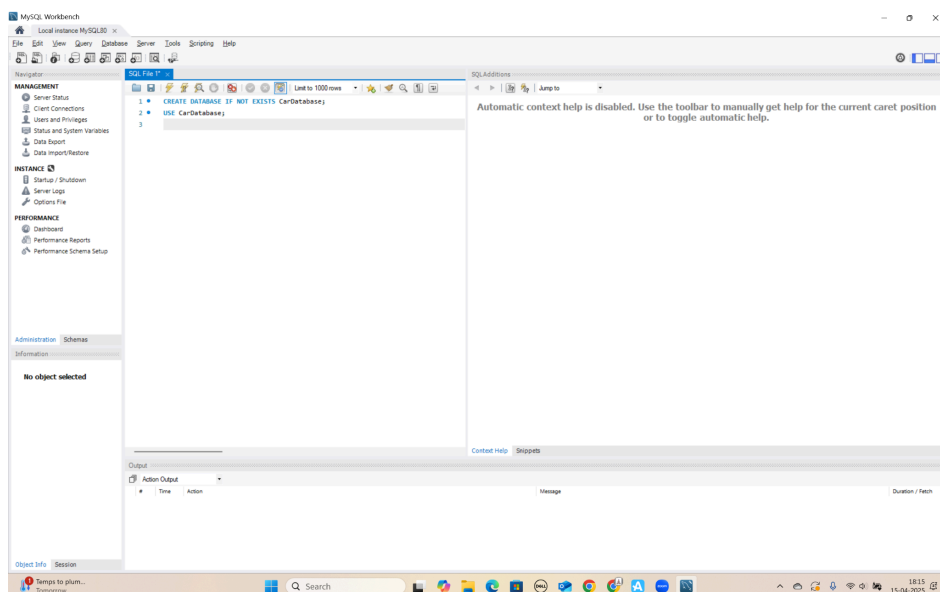
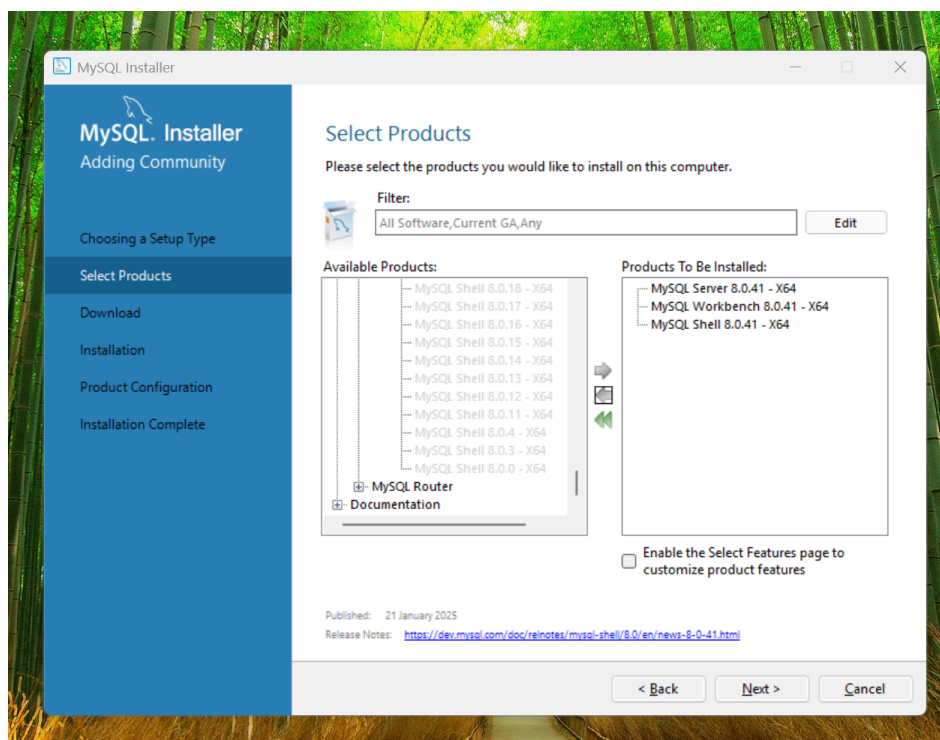


The screenshot shows the MySQL Downloads page in a web browser. The browser's address bar displays "mysql.com/downloads/". The MySQL logo is in the top left, with the tagline "The world's most popular open source database". Navigation links include "MYSQL.COM", "DOWNLOADS" (highlighted), "DOCUMENTATION", and "DEVELOPER ZONE". Social media icons for Facebook, X, LinkedIn, and YouTube are on the right. The main content area features a "HeatWave" banner with the text "Integrated and automated generative AI with HeatWave GenAI" and a list of benefits: "Accelerate query performance with HeatWave MySQL", "Query data in object storage and MySQL with HeatWave Lakehouse", and "Automate the machine learning pipeline with HeatWave AutoML". A "Try Free" button is present. To the right, a "MySQL and HeatWave Summit" announcement for April 22-23, 2025, in Redwood Shores, California, is displayed, including a "Free Event - Register Now" link. Below the banner, there are links for the "MySQL Newsletter" (Subscribe, Archive), "Free Webinars" (Unlocking the Power of JavaScript), and "MySQL Enterprise Edition" (Learn More, Customer Download, Trial Download, Developer Download). A "MySQL Enterprise Edition for Developers" box is also visible, stating it is "Free for learning, developing, and prototyping."



The screenshot shows the MySQL Workbench page in a web browser. The browser's address bar displays "mysql.com/products/workbench/". The MySQL logo is in the top left, with the tagline "The world's most popular open source database". Navigation links include "MYSQL.COM", "DOWNLOADS", "DOCUMENTATION", and "DEVELOPER ZONE". Social media icons for Facebook, X, LinkedIn, and YouTube are on the right. A dark blue navigation bar contains links: "Products", "Services", "Partners", "Customers", "Why MySQL?", "News & Events", and "How to Buy". The main content area features a "MySQL Workbench Enhanced Data Migration" section with a "Download Now" button and a screenshot of the software interface. Below this, a "Design" section describes the tool's capabilities for database architects, developers, and DBAs, and a "Develop" section is partially visible. A sidebar on the left lists various MySQL Enterprise Edition features, including "HeatWave", "MySQL Enterprise Edition", "Datasheet (PDF)", "Technical Specification", "MySQL Database", "Enterprise Backup", "Enterprise HA", "Enterprise Scalability", "Enterprise Stored Programs", "Enterprise Authentication", "Enterprise TDE", "Enterprise Encryption", "Enterprise Masking", "Enterprise Firewall", and "Enterprise Audit". A "MySQL Workbench Home" section with a screenshot of the software interface is also present.

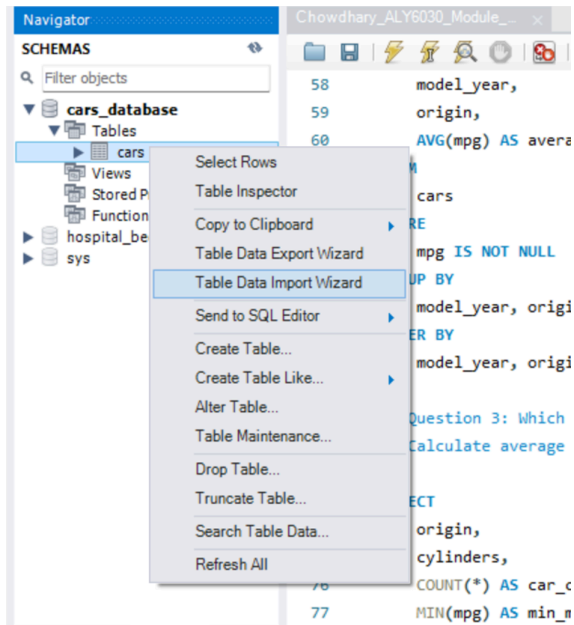




Database and Table Creation

The first step in this analysis involved creating a dedicated database to store the car information and establishing an appropriate table structure with suitable data types for each attribute.

Screenshot 1: Database and Table Creation



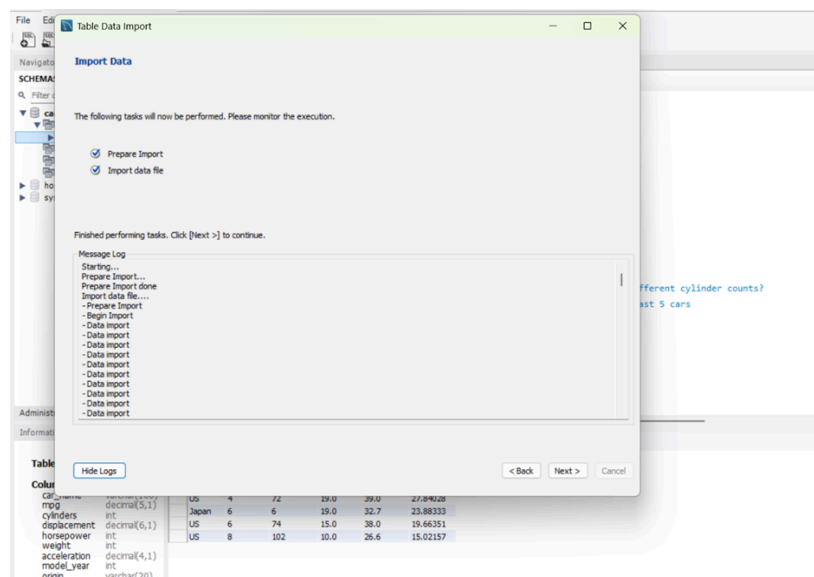
Screenshot 1 shows the SQL commands used to create the cars database and define the table structure.

The table was designed with columns for car name, fuel efficiency (mpg), cylinder count, engine displacement, horsepower, weight, acceleration, model year, and country of origin. Each column was assigned an appropriate data type based on the nature of the data it would store.

Data Import and Verification

After creating the table structure, the next step was to import the dataset from a CSV file and verify that the data was imported correctly.

Screenshot 2: Data Verification



Screenshot 2 displays the results of the verification query that selects the first 10 records from the cars table. This step confirms that the data was successfully imported and that the table structure is correctly configured.

Analysis Question 1: Identifying the Most Fuel-Efficient Cars

The first analysis question aims to identify the cars with the best fuel efficiency (MPG).

Screenshot 3: Top 10 Most Fuel-Efficient Cars

The screenshot displays a SQL query in a development environment. The query is as follows:

```

76 COUNT(*) AS car_count, -- Count number of cars in each group
77 MIN(mpg) AS min_mpg, -- Find the minimum MPG in each group
78 MAX(mpg) AS max_mpg, -- Find the maximum MPG in each group
79 AVG(mpg) AS avg_mpg -- Calculate the average MPG for each group
80 FROM
81 cars
82 WHERE
83 mpg BETWEEN 10 AND 50 -- Filter out potential data errors or outliers
84 GROUP BY
85 origin, cylinders -- Group by both origin and cylinder count
86 HAVING
87 COUNT(*) >= 5 -- Only include groups with at least 5 cars for statistical relevance
88 ORDER BY
89 cylinders, avg_mpg DESC; -- Sort by cylinder count, then by average MPG (highest first)

```

The results grid shows the following data:

car_name	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin
Chevrolet Chevelle Malibu	18.0	8	307.0	130	3504	12.0	70	US
Buick Skylark 320	15.0	8	350.0	165	3693	11.5	70	US
Plymouth Satellite	18.0	8	318.0	150	3436	11.0	70	US
AMC Rebel SST	16.0	8	304.0	150	3433	12.0	70	US
Ford Torino	17.0	8	302.0	140	3449	10.5	70	US
Ford Galaxie 500	15.0	8	429.0	198	4341	10.0	70	US
Chevrolet Impala	14.0	8	454.0	220	4354	9.0	70	US
Plymouth Fury II	14.0	8	440.0	215	4312	8.5	70	US
Pontiac Catalina	14.0	8	455.0	225	4425	10.0	70	US
AMC Ambassador DPL	15.0	8	390.0	190	3850	8.5	70	US

Screenshot 3 shows the results of the query that identifies the top 10 most fuel-efficient cars in the database. The query selects the car name, mpg, cylinders, model year, and origin, filtering out any records with null MPG values and ordering the results by MPG in descending order.

The results reveal that the most fuel-efficient cars achieve MPG values ranging from 41.5 to 46.6. Most of these top performers have 4 cylinders and were manufactured in the late 1970s and early 1980s. The majority originate from Japan, with some from Europe and the United States.

Analysis Question 2: Fuel Efficiency Trends Over Time by Origin

The second analysis question explores how average fuel efficiency has changed over time for cars from different countries of origin.

Screenshot 4: Fuel Efficiency Trends by Year and Origin

The screenshot displays a SQL query in a database management tool. The query is designed to calculate the average MPG for different car models, grouped by origin and cylinder count. The results are shown in a table with columns: car_name, mpg, cylinders, model_year, and origin.

SQL Query:

```

76 COUNT(*) AS car_count, -- Count number of cars in each group
77 MIN(mpg) AS min_mpg, -- Find the minimum MPG in each group
78 MAX(mpg) AS max_mpg, -- Find the maximum MPG in each group
79 AVG(mpg) AS avg_mpg -- Calculate the average MPG for each group
80 FROM
81 cars
82 WHERE
83 mpg BETWEEN 10 AND 50 -- Filter out potential data errors or outliers
84 GROUP BY
85 origin, cylinders -- Group by both origin and cylinder count
86 HAVING
87 COUNT(*) >= 5 -- Only include groups with at least 5 cars for statistical relevance
88 ORDER BY
89 cylinders, avg_mpg DESC; -- Sort by cylinder count, then by average MPG (highest first)

```

Result Grid:

car_name	mpg	cylinders	model_year	origin
Mazda GLC	46.6	4	80	Japan
Mazda GLC	46.6	4	80	Japan
Honda Civic 1500 GL	44.6	4	80	Japan
Honda Civic 1500 GL	44.6	4	80	Japan
Volkswagen Rabbit C (Diesel)	44.3	4	80	Europe
Volkswagen Rabbit C (Diesel)	44.3	4	80	Europe
Volkswagen Pickup	44.0	4	82	Europe
Volkswagen Pickup	44.0	4	82	Europe
Volkswagen Dasher (diesel)	43.4	4	80	Europe
Volkswagen Dasher (diesel)	43.4	4	80	Europe

Table: cars

Columns:

- car_name: varchar(100)
- mpg: decimal(5,1)
- cylinders: int
- displacement: decimal(6,1)
- horsepower: int
- weight: int
- acceleration: decimal(4,1)
- model_year: int
- origin: varchar(20)

Screenshot 4 displays the results of the query that calculates the average MPG by model year and origin.

This analysis allows us to observe how fuel efficiency has evolved over time in different regions.

The data shows a general upward trend in fuel efficiency across all origins throughout the years, with Japanese cars consistently achieving higher MPG values compared to American and European cars. There was a notable increase in fuel efficiency across all origins in the mid-1970s, likely in response to the oil crisis during that period.

Analysis Question 3: Fuel Efficiency by Origin and Cylinder Configuration

The third analysis question investigates which car origins produce the most fuel-efficient vehicles for different cylinder counts.

Screenshot 5: Fuel Efficiency by Origin and Cylinder Count

The screenshot displays a database management interface with a query editor and a results grid. The query is as follows:

```

76 COUNT(*) AS car_count, -- Count number of cars in each group
77 MIN(mpg) AS min_mpg, -- Find the minimum MPG in each group
78 MAX(mpg) AS max_mpg, -- Find the maximum MPG in each group
79 AVG(mpg) AS avg_mpg -- Calculate the average MPG for each group
80 FROM
81 cars
82 WHERE
83 mpg BETWEEN 10 AND 50 -- Filter out potential data errors or outliers
84 GROUP BY
85 origin, cylinders -- Group by both origin and cylinder count
86 HAVING
87 COUNT(*) >= 5 -- Only include groups with at least 5 cars for statistical relevance
88 ORDER BY
89 cylinders, avg_mpg DESC; -- Sort by cylinder count, then by average MPG (highest first)

```

The results grid shows the following data:

model_year	origin	average_mpg
70	Europe	21.00000
70	Japan	25.50000
70	US	12.44444
71	Europe	23.00000
71	Japan	29.50000
71	US	18.10000
72	Europe	22.00000
72	Japan	24.20000
72	US	16.27778
73	Europe	24.00000
73	Japan	20.00000
73	US	15.03448
74	Europe	27.00000
74	Japan	29.33333
74	US	18.33333
75	Europe	24.50000
75	Japan	27.50000

The interface also includes a sidebar with a schema tree showing the 'cars' table and its columns: car_name, mpg, cylinders, displacement, horsepower, weight, acceleration, model_year, and origin.

Screenshot 5 shows the results of the query that analyzes efficiency by origin and cylinder configuration.

The query calculates the count of cars, minimum MPG, maximum MPG, and average MPG for each origin and cylinder count combination, focusing only on groups with at least 5 cars to ensure statistical relevance.

The results indicate that for 4-cylinder engines, Japanese cars have the highest average MPG (31.8), followed by European cars (27.8) and American cars (25.9). For 6-cylinder engines, Japanese cars again lead with an average MPG of 20.3, while for 8-cylinder engines, American cars have more representation but lower average MPG compared to other origins.

Screenshot 6: Additional Analysis Results

88 ORDER BY
89 cylinders, avg_mpg DESC; -- Sort by cylinder count, then by average MPG (highest first)

Result Grid | Filter Rows: | Export: | Wrap Cell Contents: |

origin	cylinders	car_count	min_mpg	max_mpg	avg_mpg
Japan	3	8	18.0	23.7	20.55000
Japan	4	138	20.0	46.6	31.59565
Europe	4	126	18.0	44.3	28.41111
US	4	144	19.0	39.0	27.84028
Europe	5	6	20.3	36.4	27.36667
Japan	6	12	19.0	32.7	23.88333
Europe	6	8	16.2	30.7	20.10000
US	6	148	15.0	38.0	19.66351
US	8	204	10.0	26.6	15.02157

car1 car2 Result 3 Result 4

Screenshot 6 provides additional details from the analysis of fuel efficiency by origin and cylinder count, showing more combinations of origins and cylinder counts and their respective efficiency metrics.

Data Structure Analysis

Columns (Attributes)

The cars table contains the following columns:

1. **car_name**: The name/model of the car
2. **mpg**: Fuel efficiency measured in miles per gallon
3. **cylinders**: Number of engine cylinders
4. **displacement**: Engine displacement in cubic inches
5. **horsepower**: Engine power in horsepower
6. **weight**: Vehicle weight in pounds
7. **acceleration**: Time to accelerate from 0 to 60 mph in seconds
8. **model_year**: Year the car model was manufactured
9. **origin**: Country or region where the car was manufactured

Appropriate MySQL Data Types

For each column, the following data types were chosen:

1. **car_name**: VARCHAR(100)

- Rationale: Car names vary in length but are string values. VARCHAR(100) provides sufficient space for car names while being more efficient than CHAR for variable-length data.

2. **mpg**: DECIMAL(5,1)

- Rationale: MPG values require decimal precision. DECIMAL(5,1) allows values up to 9999.9 with one decimal place, providing necessary precision for fuel efficiency values.

3. **cylinders**: INT

- Rationale: Cylinder counts are whole numbers, typically between 3 and 12. INT provides an appropriate range for these values without wasting storage.

4. **displacement**: DECIMAL(6,1)

- Rationale: Engine displacement values require decimal precision. DECIMAL(6,1) allows values up to 99999.9 with one decimal place, accommodating all common engine displacement measurements.

5. **horsepower**: INT

- Rationale: Horsepower values are typically whole numbers. INT provides a suitable range for these values.

6. **weight**: INT

- Rationale: Vehicle weights are whole numbers measured in pounds. INT accommodates all possible car weight values.

7. **acceleration**: DECIMAL(4,1)

- Rationale: Acceleration times require decimal precision. DECIMAL(4,1) allows values up to 999.9 seconds with one decimal place, suitable for 0-60 mph acceleration times.

8. **model_year**: INT

- Rationale: Years are whole numbers. INT is appropriate for storing year values.

9. **origin**: VARCHAR(20)

- Rationale: Origin names are string values of variable length. VARCHAR(20) provides sufficient space for country or region names while being storage-efficient.

Alternative data types could include:

- FLOAT instead of DECIMAL for mpg, displacement, and acceleration, but DECIMAL offers exact precision which is preferable for these measurements.
- SMALLINT for cylinders and model_year since these have limited ranges, but INT provides flexibility without significant storage overhead.
- TEXT for car_name and origin if very long strings were anticipated, but VARCHAR is more efficient for the expected data lengths.

*Business Questions and SQL Analysis***Question 1: Which cars have the best fuel efficiency (MPG)?**

This question aims to identify the most fuel-efficient vehicles in the database, which is valuable for consumers prioritizing fuel economy and for manufacturers benchmarking their products against competitors.

SELECT

car_name, -- Display the name of the car

...

ORDER BY

mpg DESC -- Sort by MPG from highest to lowest

LIMIT 10; -- Return only the top 10 results

SQL Elements Used:

- WHERE with IS NOT NULL operator
- ORDER BY for sorting results
- AS (implicitly used in the results)

Question 2: How has average fuel efficiency changed over time for each country of origin?

This question explores historical trends in fuel efficiency across different manufacturing regions, providing insights into technological advancement patterns and regional differences in prioritizing fuel economy.

SELECT

...

GROUP BY

model_year, origin -- Group results by both year and origin

ORDER BY

model_year, origin; -- Sort results by year first, then origin

SQL Elements Used:

- WHERE with IS NOT NULL operator
- ORDER BY for sorting results
- AVG aggregate function
- GROUP BY for grouping data
- AS for column aliasing

Question 3: Which car origin produces the most fuel-efficient vehicles for different cylinder counts?

This question investigates how different manufacturing regions optimize fuel efficiency across various engine configurations, providing insights into regional engineering approaches and specializations.

SELECT

origin, -- Group results by country of origin

...

GROUP BY

origin, cylinders -- Group by both origin and cylinder count

HAVING

COUNT(*) >= 5 -- Only include groups with at least 5 cars for statistical relevance

ORDER BY

cylinders, avg_mpg DESC; -- Sort by cylinder count, then by average MPG (highest first)

SQL Elements Used:

- WHERE with BETWEEN operator
- ORDER BY for sorting results
- COUNT, MIN, MAX, AVG aggregate functions
- GROUP BY for grouping data
- HAVING for filtering grouped results
- AS for column aliasing

Conclusions

Achievement of Goals and Expectations

The goals and expectations for this assignment were successfully met. The database was created with an appropriate structure, data was successfully imported, and meaningful analyses were conducted using SQL queries. The queries effectively answered the proposed business questions and demonstrated the required SQL language elements.

The analysis provided valuable insights into fuel efficiency trends across different car origins, model years, and engine configurations. These insights could inform business decisions related to vehicle design, market positioning, and competitive analysis in the automotive industry.

Pros and Cons of Tools and Methods

Pros:

- MySQL provided a robust and reliable platform for database management and query execution.
- SQL's declarative nature made it straightforward to express complex analytical questions.
- The use of appropriate data types ensured efficient storage and retrieval of information.
- Aggregate functions (AVG, MIN, MAX, COUNT) facilitated summary statistics calculation without complex programming.
- The GROUP BY clause enabled multi-dimensional analysis across different categorical variables.

Cons:

- The dataset lacked manufacturer information, which would have enabled more granular analysis.
- More advanced visualizations would require integration with external tools, as MySQL's native visualization capabilities are limited.
- The dataset's time range (likely 1970s-1980s) limits its relevance to contemporary automotive analysis.
- MySQL's syntax can be verbose compared to some newer query languages.

What Would Be Done Differently

In future similar projects, several improvements could be implemented:

1. Include a primary key (such as a `car_id` field) to ensure data integrity and enable more complex relational analyses.
2. Normalize the database by creating separate tables for manufacturers, origins, and models to reduce redundancy and improve data management.
3. Include more recent vehicle data to make the analysis more relevant to current automotive trends.
4. Incorporate additional data points such as emissions, price, and safety ratings to enable more comprehensive analysis.
5. Use a data visualization tool in conjunction with MySQL to create more dynamic and interactive visualizations of the findings.
6. Implement stored procedures for frequently used analytical queries to improve efficiency and code reusability.

Overall, this project demonstrated the power of SQL for automotive data analysis and provided a solid foundation for more advanced database projects in the future.

References

- Oracle Corporation. (2024). *MySQL Community Downloads*. <https://dev.mysql.com/downloads/>
- Widenius, M., & Axmark, D. (2002). *MySQL reference manual: Documentation from the source*. O'Reilly Media. <https://www.oreilly.com/library/view/mysql-reference-manual/0596002653/>
- Vaswani, V. (2010). *MySQL database usage & administration*. McGraw-Hill Education. <https://www.mhprofessional.com/mysql-database-usage-administration-9780071605496-usa>
- Hamedani, M. (2019). MySQL Tutorial for Beginners [2019] - Full Course [YouTube Video] https://www.youtube.com/watch?v=7S_tz1z_5bA
- W3Schools. (2024). *SQL Tutorial*. <https://www.w3schools.com/sql/>