

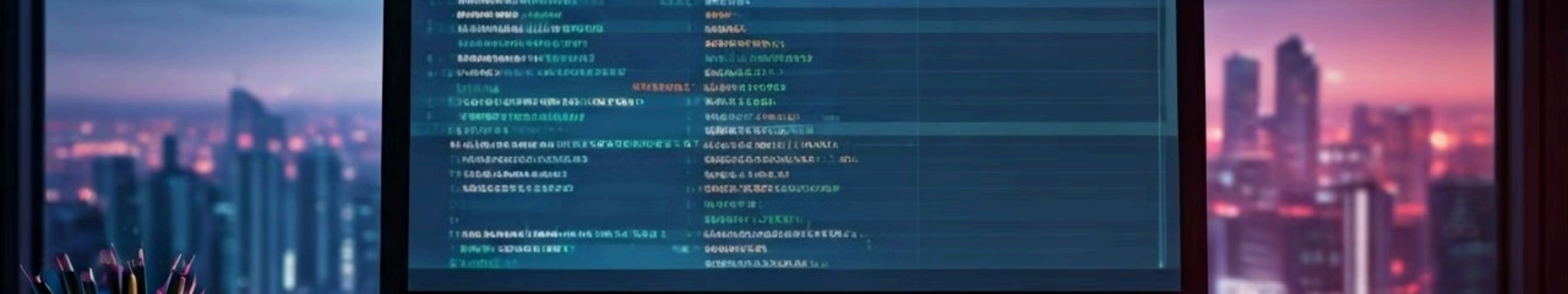
Analyzing Ride Data in SQLite: - Insights for Improved Service

This presentation explores key findings from our ride-sharing data analysis. Will examine monthly ride trends and identify our top-performing driver, providing valuable insights to enhance our service quality and customer satisfaction.



by Aparna Srivastava





Monthly Ride Analysis: Methodology

To determine the months with the highest and lowest number of rides, we utilized SQLite's "**strftime function**". This allowed us to extract month and year data from the rides table, grouping and counting total rides for each month.

Our SQL query grouped the data by month and year, ordering the results by ride count in descending order. This approach provided a clear view of ride volume fluctuations over time.

1

Data Import

Imported CSV files for **rides**, **drivers**, and **customers**.

2

Date Extraction

Used **strftime** to extract month and year from the date field.

3

Data Grouping

Grouped data by month and year.

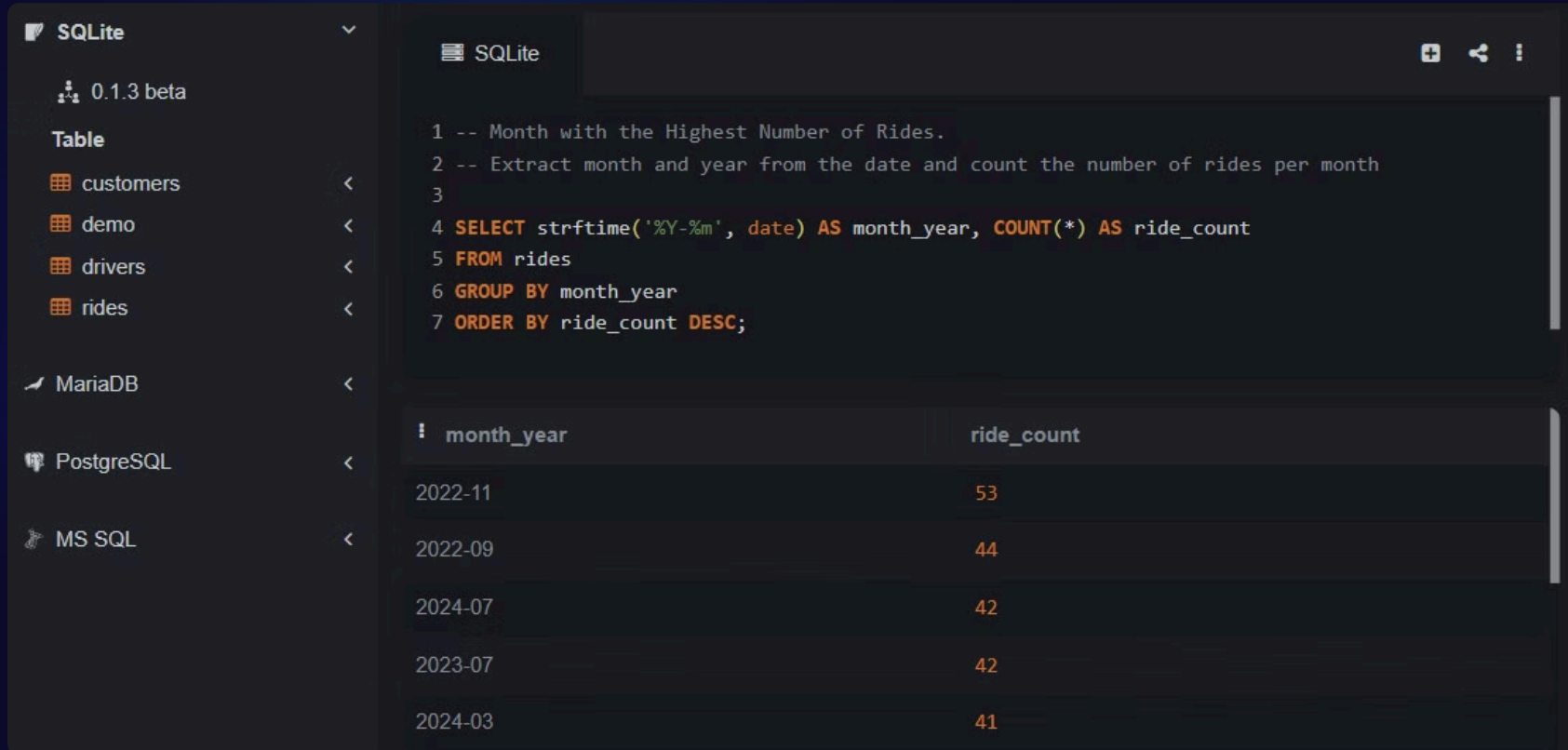
4

Ride Counting

Counted total rides for each month-year combination.

Task 1 : Monthly Ride Analysis: Results

i). The Month with the Highest Number of Rides :



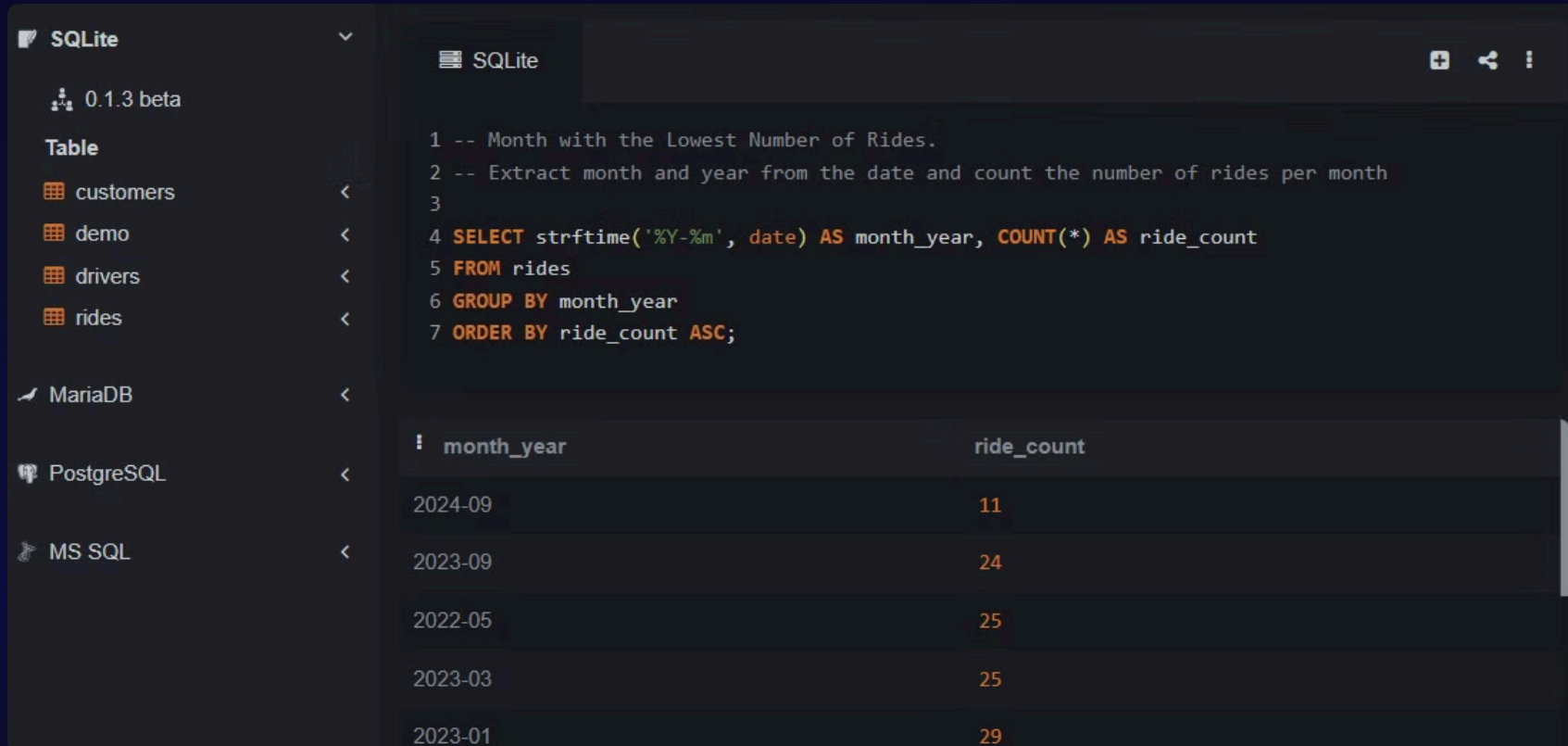
The screenshot shows a database management tool interface. On the left, a sidebar lists databases: SQLite (selected), MariaDB, PostgreSQL, and MS SQL. Under SQLite, there's a version '0.1.3 beta' and a 'Table' section with 'customers', 'demo', 'drivers', and 'rides'. The main area displays a SQL query and its results. The query is:
1 -- Month with the Highest Number of Rides.
2 -- Extract month and year from the date and count the number of rides per month
3
4 SELECT strftime('%Y-%m', date) AS month_year, COUNT(*) AS ride_count
5 FROM rides
6 GROUP BY month_year
7 ORDER BY ride_count DESC;
The results are shown in a table with two columns: 'month_year' and 'ride_count'. The data is as follows:

month_year	ride_count
2022-11	53
2022-09	44
2024-07	42
2023-07	42
2024-03	41

- The analysis shows:

The month with the **highest number of rides** is **November 2022**, emerged as the busiest month with "**53 rides**".

ii). The Month with the Lowest Number of Rides :



The screenshot shows a database interface with a sidebar on the left listing databases (SQLite, MariaDB, PostgreSQL, MS SQL) and tables (customers, demo, drivers, rides). The main area displays SQL code for finding the month with the lowest number of rides, followed by the query results.

```
1 -- Month with the Lowest Number of Rides.
2 -- Extract month and year from the date and count the number of rides per month
3
4 SELECT strftime('%Y-%m', date) AS month_year, COUNT(*) AS ride_count
5 FROM rides
6 GROUP BY month_year
7 ORDER BY ride_count ASC;
```

month_year	ride_count
2024-09	11
2023-09	24
2022-05	25
2023-03	25
2023-01	29

- The analysis shows:

The month with the **lowest number of rides** is **September 2024**, had the lowest activity with only "**11 rides**".

Implications

These findings can inform decisions on driver availability, promotions, and resource planning.

Task 2: Identifying the Best-Performing Driver(s)

To define the "best-performing driver," I used a combination of the following criteria:

1. Average Rating

A consistently high rating indicates superior customer satisfaction based on a 1-5 scale and service quality.

2. Punctuality

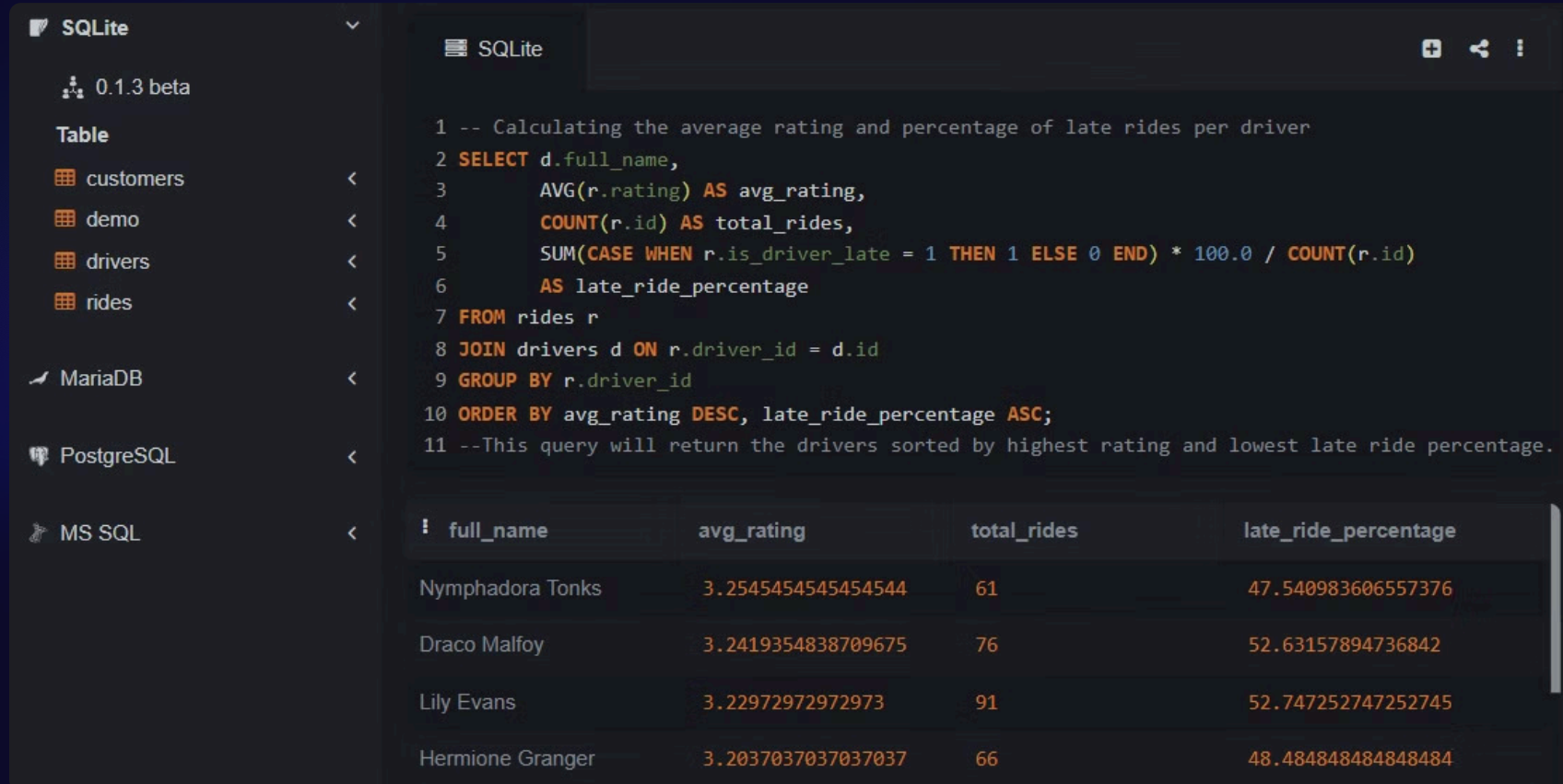
Measured by the percentage of rides where the driver was late, derived from the `is_driver_late` flag.

reflecting reliability and professionalism.

3. Total Rides Completed

Gives an idea of driver experience and consistency, though I primarily focused on ratings and punctuality.

I'll identify the best-performing driver based on a set of criteria such as the **highest average rating** and **punctuality** (i.e., lowest number of late rides). Let's proceed with that analysis.



The screenshot shows a SQLite database interface. On the left, a sidebar lists the database 'SQLite' with version '0.1.3 beta'. Under the 'Table' section, four tables are listed: 'customers', 'demo', 'drivers', and 'rides'. Below this, other database options like 'MariaDB', 'PostgreSQL', and 'MS SQL' are visible. The main area displays a SQL query in a text editor. The query calculates the average rating and percentage of late rides per driver. Below the query, the results are shown in a table with four columns: 'full_name', 'avg_rating', 'total_rides', and 'late_ride_percentage'. The results list four drivers: Nymphadora Tonks, Draco Malfoy, Lily Evans, and Hermione Granger, with their respective average ratings, total rides, and late ride percentages.

```
1 -- Calculating the average rating and percentage of late rides per driver
2 SELECT d.full_name,
3        AVG(r.rating) AS avg_rating,
4        COUNT(r.id) AS total_rides,
5        SUM(CASE WHEN r.is_driver_late = 1 THEN 1 ELSE 0 END) * 100.0 / COUNT(r.id)
6        AS late_ride_percentage
7 FROM rides r
8 JOIN drivers d ON r.driver_id = d.id
9 GROUP BY r.driver_id
10 ORDER BY avg_rating DESC, late_ride_percentage ASC;
11 --This query will return the drivers sorted by highest rating and lowest late ride percentage.
```

full_name	avg_rating	total_rides	late_ride_percentage
Nymphadora Tonks	3.2545454545454544	61	47.540983606557376
Draco Malfoy	3.2419354838709675	76	52.63157894736842
Lily Evans	3.22972972972973	91	52.747252747252745
Hermione Granger	3.2037037037037037	66	48.484848484848484

A driver with a **high average rating** and a **low percentage** of **late rides** was considered the best performer. I calculated the average rating and percentage of late rides per driver using the above query.

Best Performing Driver: Results

Our analysis identified **Nymphadora Tonks** as the top-performing driver based on our criteria:

Average Rating	3.25 (out of 5)
Late Ride Percentage	47.54%
Total Rides	61

Despite some late rides (29 out of 61 rides were late), Nymphadora's high average rating indicates consistently high customer satisfaction with her service.





Conclusion and Recommendations

Our analysis has provided valuable insights into ride patterns and driver performance. The identification of "Nymphadora Tonks" as the top driver, based on a structured approach combining customer feedback and reliability metrics, offers a clear view of exemplary performance.

These findings can be applied to:

Data-Driven Performance Optimization:

Use metrics to evaluate and enhance driver performance, ensuring high-quality service through targeted feedback and reviews.

Tailored Marketing Strategies:

Analyze monthly ride trends to adjust services and marketing efforts, improving customer engagement and satisfaction.

Incentive Programs and Resource Allocation:

Develop targeted incentives for drivers and optimize scheduling based on ride patterns to enhance operational efficiency and overall service quality.