HNG Ride Business Analysis

(June 2021 - Dec 2024)

By

Emmanuel Adesoye

Objective

This report analyzes cleaned data from HNG Ride to identify trends in performance, driver activity, and revenue from June 2021 to December 2024. The goal is to provide actionable insights for management to understand operations and identify areas for improvement.

Data Cleaning & Preparation

To ensure data integrity, several cleaning steps were performed on the raw data:

- **Date Conversion:** All text-based date columns (`request_time`, `pickup_time`, `dropoff_time`, `signup_date`, and `paid_date`) were successfully converted to a standard `YYYY-MM-DD HH:MM:SS` timestamp format.
- **Invalid Payments:** Removed 10,582 payment records where the `amount` was 0.0, ensuring that analysis was performed only on completed, paid rides.
- **Invalid Fares:** Deleted 653 rides where the `fare` was less than or equal to zero.
- **Status Correction:** Standardized the `status` column by correcting 955 entries from "complted" to "completed".

Business Questions & Findings

Below are the 8 business questions, the SQL queries used to answer them, and the findings from the data.

1. Top 10 Longest Rides

Business Question: Find the top 10 longest rides (by distance), including driver name, rider name, pickup/dropoff cities, and payment method.

Final SQL Query (`1.sql`):

```
SQL

SELECT
    rd.distance_km,
    dr.name as driver_name,
    rdr.name as rider_name,
    rd.pickup_city,
    rd.dropoff_city,
    pym.method AS payment_method
FROM rides rd
JOIN drivers dr ON rd.driver_id = dr.driver_id
JOIN riders rdr ON rd.rider_id = rdr.rider_id
JOIN payments pym ON rd.ride_id = pym.ride_id
WHERE
    pym.amount > 0
    AND rd.request_time BETWEEN '2022-06-01 00:00:00' AND '2024-12-31 23:59:59'
ORDER BY rd.distance_km DESC
LIMIT 10;
```

Finding: The top 10 longest completed rides were all 29.99 km or 30.0 km. The longest ride (30.0 km) was by **Driver_1774** with **Rider_292**, traveling from Calgary to Los Angeles.

2. Returning Riders (2021 Signup, 2024 Rides)

Business Question: How many riders who signed up in 2021 still took rides in 2024?

Final SQL Query (`2.sql`):

```
SQL

SELECT COUNT(DISTINCT rd.rider_id) AS Active_Riders
FROM rides rd
JOIN riders rdr ON rd.rider_id = rdr.rider_id
WHERE rd.request_time BETWEEN '2024-01-01 00:00:00' AND '2024-12-31 23:59:59'
AND rdr.signup_date BETWEEN '2021-01-01 00:00:00' AND '2021-12-31 23:59:59';
```

Finding: A total of **2,051 riders** who signed up in 2021 returned to take at least one ride in 2024.

3. Quarter with Biggest YoY Growth

Business Question: Compare quarterly revenue between 2021, 2022, 2023, and 2024. Which quarter had the biggest YoY growth?

Final SQL Query (3.sql):

```
SQL

WITH QuarterlyRevenue AS (
    SELECT
        strftime('%Y', paid_date) AS SalesYear,
        (strftime('%m', paid_date) - 1) / 3 + 1 AS SalesQuarter,
        SUM(amount) AS QuarterRevenue
    FROM payments
    WHERE
        amount > 0
        AND paid_date BETWEEN '2021-06-01 00:00:00' AND '2024-12-31 23:59:59'
    GROUP BY SalesYear, SalesQuarter
),
YOYGrowth AS (
    SELECT
        SalesYear, SalesQuarter, QuarterRevenue,
        LAG(QuarterRevenue, 1, 0) OVER(
            PARTITION BY SalesQuarter
            ORDER BY SalesYear
        ) as PrevYearRevenue
    FROM QuarterlyRevenue
)
SELECT
    SalesYear, SalesQuarter, QuarterRevenue, PrevYearRevenue,
    (QuarterRevenue - PrevYearRevenue) * 100.0 / NULLIF(PrevYearRevenue, 0) AS
YOYGrowthPercentage
FROM YOYGrowth
WHERE PrevYearRevenue > 0
ORDER BY YOYGrowthPercentage DESC
LIMIT 1;
```

Finding: The quarter with the biggest Year-over-Year growth was **Q4 2024**, which saw a **4.04% increase** in revenue compared to Q4 2023.

4. Top 5 Most Consistent Drivers

Business Question: For each driver, calculate their average monthly rides since signup. Who are the top 5 drivers with the highest consistency?

Final SQL Query (4.sql):

```
SQL

WITH DriverActiveMonth AS (
    SELECT
        driver_id, name,
        (strftime('%Y', '2024-12-31') - strftime('%Y', signup_date)) * 12 +
        (strftime('%m', '2024-12-31') - strftime('%m', signup_date)) + 1 AS total_months_active
    FROM drivers
),
DriverTotalRides AS (
    SELECT
        rd.driver_id,
        COUNT(rd.ride_id) AS total_rides_completed
    FROM rides rd
    JOIN payments pym ON rd.ride_id = pym.ride_id
    WHERE pym.amount > 0
    AND rd.request_time BETWEEN '2021-06-01' AND '2024-12-31'
    GROUP BY rd.driver_id
)
SELECT
    d.name,
    dt.total_rides_completed,
    d.total_months_active,
    (CAST(dt.total_rides_completed AS REAL) / d.total_months_active) AS rides_per_active_month
FROM DriverActiveMonth d
JOIN DriverTotalRides dt ON dt.driver_id = d.driver_id
ORDER BY rides_per_active_month DESC
LIMIT 5;
```

Finding: The top 5 most consistent drivers, based on average rides per active month, are led by **Driver_219** (2.38), **Driver_1005** (2.21), and **Driver_1029** (2.15).

5. City with Highest Cancellation Rate

Business Question: Calculate the cancellation rate per city and identify which city had the highest cancellation rate.

Final SQL Query (5.sql):

```
SQL

SELECT
    pickup_city,
    COUNT(rider_id) as total_rides,
    SUM(CASE WHEN status = 'cancelled' THEN 1 ELSE 0 END) as cancelled_rides,
    CAST(SUM(CASE WHEN status = 'cancelled' THEN 1 ELSE 0 END) AS REAL) * 100.0 /
COUNT(rider_id) AS cancellation_rate
FROM rides
WHERE
    request_time BETWEEN '2021-06-01 00:00:00' AND '2024-12-31 23:59:59'
GROUP BY pickup_city
ORDER BY cancellation_rate DESC
LIMIT 1;
```

Finding: Chicago had the highest cancellation rate at **19.27%**.

6. Riders with >10 Rides, 0 Cash Payments

Business Question: Identify riders who have taken more than 10 rides but never paid with cash.

Final SQL Query (6.sql):

```
SQL

SELECT
    rd.rider_id,
    rdr.name,
    COUNT(rd.ride_id) total_ride
FROM rides rd
JOIN payments pym ON rd.ride_id = pym.ride_id
JOIN riders rdr ON rd.rider_id = rdr.rider_id
WHERE
    pym.amount > 0
    AND rd.request_time BETWEEN '2021-06-01 00:00:00' AND '2024-12-31 23:59:59'
GROUP BY rd.rider_id
HAVING
    COUNT(rd.ride_id) > 10
    AND SUM(CASE WHEN pym.method = 'cash' THEN 1 ELSE 0 END) = 0;
```

Finding: Only one rider, **Rider_7823**, took more than 10 completed rides (12 total) and never used cash as a payment method.

7. Top 3 Drivers by Revenue in Each City

Business Question: Find the top 3 drivers in each city by total revenue earned between June 2021 and Dec 2024.

Final SQL Query (7.sql):

```
SQL

WITH DriverRevenueByCity AS (
    SELECT
        rd.pickup_city,
        dr.name AS driver_name,
        SUM(p.amount) AS total_revenue
    FROM rides rd
    JOIN payments p ON rd.ride_id = p.ride_id
    JOIN drivers dr ON rd.driver_id = dr.driver_id
    WHERE
        p.amount > 0
        AND rd.request_time BETWEEN '2021-06-01 00:00:00' AND '2024-12-31 23:59:59'
    GROUP BY rd.pickup_city, dr.name
),
RankedDrivers AS (
    SELECT
        pickup_city, driver_name, total_revenue,
        ROW_NUMBER() OVER (
            PARTITION BY pickup_city
            ORDER BY total_revenue DESC
        ) AS rank
    FROM DriverRevenueByCity
)
SELECT
    pickup_city, driver_name, total_revenue, rank
FROM RankedDrivers
WHERE rank <= 3
ORDER BY pickup_city, rank;
```

Finding: The top 3 drivers by revenue in each city were identified. For example, in **Boston**, the top drivers are **Driver_1176** ($448.40), **Driver_286** ($326.58), and **Driver_1141** ($315.88).

8. Top 10 Bonus-Qualified Drivers

Business Question: Identify the top 10 drivers qualified for a bonus (at least 30 completed rides, average rating $\ge$ 4.5, and cancellation rate < 5%).

Final SQL Query (8.sql):

```
SQL

WITH DriverRideStats AS (
    SELECT
        driver_id,
        COUNT(ride_id) AS total_rides_attempted,
        SUM(CASE WHEN status = 'cancelled' THEN 1 ELSE 0 END) AS total_cancelled
    FROM rides
    WHERE
        request_time BETWEEN '2021-06-01 00:00:00' AND '2024-12-31 23:59:59'
    GROUP BY driver_id),
DriverCompletedStats AS (
    SELECT
        rd.driver_id,
        COUNT(rd.ride_id) AS total_completed_rides
    FROM rides rd
    JOIN payments p ON rd.ride_id = p.ride_id
    WHERE
        p.amount > 0
        AND rd.request_time BETWEEN '2021-06-01 00:00:00' AND '2024-12-31 23:59:59'
    GROUP BY rd.driver_id)
SELECT
    dr.name,
    dr.rating AS average_rating,
    cs.total_completed_rides,
    (CAST(rs.total_cancelled AS REAL) * 100.0 / NULLIF(rs.total_rides_attempted, 0)) AS
cancellation_rate_percent
FROM drivers dr
JOIN DriverRideStats rs ON dr.driver_id = rs.driver_id
JOIN DriverCompletedStats cs ON dr.driver_id = cs.driver_id
WHERE
    cs.total_completed_rides >= 30
    AND dr.rating >= 4.5
    AND (CAST(rs.total_cancelled AS REAL) * 100.0 / NULLIF(rs.total_rides_attempted, 0)) < 5.0
ORDER BY cs.total_completed_rides DESC
LIMIT 10;
```

Finding: Only two drivers met all criteria for the bonus: **Driver_1005** (31 rides, 4.8 rating, 0% cancellation) and **Driver_1181** (31 rides, 4.6 rating, 3.125% cancellation).