**QUERIES USED IN THIS PROJECT:**

**1. COUNT OF APP DOWNLOADS:**

SELECT COUNT(\*) FROM app\_downloads;

**2. COUNT OF USERS SIGNED UP ON THE APP:**

SELECT COUNT(\*) FROM signups;

**3. COUNT OF RIDES REQUESTED THROUGH APP:**

SELECT COUNT(\*) FROM ride\_requests;

**4. COUNT OF RIDES REQUESTED AND COMPLETED THROUGH THE APP:**

SELECT

COUNT(\*) AS total\_rides,

SUM(CASE WHEN request\_ts IS NOT NULL THEN 1 ELSE 0 END) AS requested\_rides,

SUM(CASE WHEN dropoff\_ts IS NOT NULL THEN 1 ELSE 0 END) AS completed\_rides

FROM

ride\_requests;

**5. COUNT OF RIDE REQUEST AND UNIQUE USERS REQUESTED A RIDE:**

SELECT

COUNT(\*) AS total\_requested\_rides,

COUNT(DISTINCT user\_id) AS unique\_requesting\_users

FROM

ride\_requests

WHERE

request\_ts IS NOT NULL;

**6. AVERAGE TIME OF A RIDE FROM PICKUP TO DROPOFF:**

SELECT

TO\_CHAR(

INTERVAL '1 second' \* AVG(EXTRACT(EPOCH FROM (dropoff\_ts - pickup\_ts))),

'MI:SSSS'

) AS average\_ride\_duration

FROM

ride\_requests

WHERE

request\_ts IS NOT NULL;

**7. COUNT OF RIDES ACCEPTED BY A DRIVER:**

SELECT COUNT(accept\_ts) FROM ride\_requests;

**8. COUNT OF RIDES SUCCESSFULLY COLLECTED PAYMENT AND HOW MUCH WAS COLLECTED:**

SELECT COUNT(ride\_id),SUM(purchase\_amount\_usd)

FROM transactions

WHERE charge\_status='Approved';

**9. COUNT OF RIDE REQUEST ON EACH PLATFORM:**

SELECT platform,

COUNT(ride\_id) FROM metrocar

GROUP BY platform;

**10. DROPOFF RATE FROM USER SIGNUP TO USER RIDE REQUEST:**

SELECT

(SUM(CASE WHEN rr.user\_id IS NULL THEN 1 ELSE 0 END) \* 100.0)/COUNT(DISTINCT s.user\_id)

AS droppoff\_rate\_percentage

FROM

signups s

LEFT JOIN ride\_requests rr ON s.user\_id = rr.user\_id

**BUSINESS QUESTION QUERIES:**

**1.TOTAL EXTRACTED ANALYSIS FOR FUNNEL VISUALIZATION:**

WITH total\_downloads AS (

SELECT

COUNT(app\_download\_key) AS user\_count,

download\_ts::date as download\_dt,

platform,

age\_range

FROM app\_downloads ad

left join signups si

on ad.app\_download\_key = si.session\_id

group by download\_dt, age\_range, platform

),

user\_ride\_status AS (

SELECT

user\_id

FROM ride\_requests

GROUP BY user\_id

),

sign\_up AS (

SELECT

COUNT(\*) AS user\_count,

age\_range,

platform,

download\_ts::date as download\_dt

FROM signups s

LEFT JOIN app\_downloads ad ON s.session\_id = ad.app\_download\_key

group by age\_range, platform,download\_dt

),

total\_request AS (

SELECT

COUNT(distinct rr.user\_id) AS user\_count,

age\_range,

platform,

COUNT(rr.ride\_id) AS ride\_count,

download\_ts::date as download\_dt

FROM ride\_requests rr

left join signups s ON rr.user\_id = s.user\_id

Left join app\_downloads ad on ad.app\_download\_key = s.session\_id

group by platform, age\_range, download\_dt

),

total\_accept AS (

SELECT

COUNT(rr.ride\_id) AS ride\_count,

COUNT(DISTINCT rr.user\_id) AS user\_count,

age\_range,

platform,

download\_ts::date as download\_dt

FROM ride\_requests rr

left join signups s ON rr.user\_id = s.user\_id

Left join app\_downloads ad on ad.app\_download\_key = s.session\_id

WHERE rr.accept\_ts IS NOT NULL

group by platform, age\_range, download\_dt

),

total\_complete AS (

SELECT

COUNT(rr.ride\_id) AS ride\_count,

COUNT(DISTINCT rr.user\_id) AS user\_count,

age\_range,

platform,

download\_ts::date as download\_dt

FROM ride\_requests rr

left join signups s ON rr.user\_id = s.user\_id

Left join app\_downloads ad on ad.app\_download\_key = s.session\_id

WHERE rr.dropoff\_ts IS NOT NULL

group by platform, age\_range, download\_dt

),

total\_payment AS (

SELECT

COUNT(DISTINCT rr.user\_id) AS user\_count,

COUNT(CASE WHEN t.charge\_status = 'Approved' THEN rr.user\_id ELSE NULL END) AS ride\_count,

age\_range,

platform,

download\_ts::date as download\_dt

FROM ride\_requests rr

LEFT JOIN transactions t ON rr.ride\_id = t.ride\_id

left join signups s ON rr.user\_id = s.user\_id

Left join app\_downloads ad on ad.app\_download\_key = s.session\_id

WHERE t.charge\_status = 'Approved'

group by platform, age\_range, download\_dt

),

total\_review AS (

SELECT

COUNT(DISTINCT rr.user\_id) AS user\_count,

COUNT(CASE WHEN r.review IS NOT NULL THEN rr.user\_id ELSE NULL END) AS ride\_count,

age\_range,

platform,

download\_ts::date as download\_dt

FROM ride\_requests rr

LEFT JOIN reviews r ON rr.ride\_id = r.ride\_id

left join signups s ON rr.user\_id = s.user\_id

Left join app\_downloads ad on ad.app\_download\_key = s.session\_id

WHERE r.review IS NOT NULL

group by platform, age\_range, download\_dt

),

funnel\_stages AS (

SELECT

1 AS funnel\_step,

'downloads' AS funnel\_name,

age\_range,

platform,

download\_dt,

user\_count,

0 AS ride\_count

FROM total\_downloads

UNION

SELECT

2 AS funnel\_step,

'signups' AS funnel\_name,

age\_range,

platform,

download\_dt,

user\_count,

0 AS ride\_count

FROM sign\_up

UNION

SELECT

3 AS funnel\_step,

'ride\_requested' AS funnel\_name,

age\_range,

platform,

download\_dt,

user\_count,

ride\_count

FROM total\_request

UNION

SELECT

4 AS funnel\_step,

'ride\_accepted' AS funnel\_name,

age\_range,

platform,

download\_dt,

user\_count,

ride\_count

FROM total\_accept

UNION

SELECT

5 AS funnel\_step,

'ride\_completed' AS funnel\_name,

age\_range,

platform,

download\_dt,

user\_count,

ride\_count

FROM total\_complete

UNION

SELECT

6 AS funnel\_step,

'payment' AS funnel\_name,

age\_range,

platform,

download\_dt,

user\_count,

ride\_count

FROM total\_payment

UNION

SELECT

7 AS funnel\_step,

'review' AS funnel\_name,

age\_range,

platform,

download\_dt,

user\_count,

ride\_count

FROM total\_review

)

Select \* from funnel\_stages

order by funnel\_step;

**2. WEEKLY ANALYSIS:**

SELECT '1' As step, 'Total\_signups' AS funnel\_name, age\_range, platform, request\_ts::date AS request\_dt, COUNT(si.user\_id) AS Total

FROM signups si

left join app\_downloads ad

on si.session\_id = ad.app\_download\_key

left join ride\_requests rr

on rr.user\_id=si.user\_id

group by age\_range, platform,request\_dt

UNION

SELECT '2' As step, 'Total\_Rides ' AS funnel\_name, age\_range, platform,request\_ts::date AS request\_dt, COUNT(ride\_id) AS Total FROM ride\_requests rr

left join signups si

on rr.user\_id = si.user\_id

left join app\_downloads ad

on si.session\_id = ad.app\_download\_key

group by age\_range, platform, request\_dt

UNION

SELECT '3' As step, 'Total\_Review' AS funnel\_name, age\_range, platform, request\_ts::date AS request\_dt, COUNT(review\_id) AS Total FROM reviews re

left join ride\_requests rr

on rr.ride\_id=re.ride\_id

left join signups si

on re.user\_id = si.user\_id

left join app\_downloads ad

on si.session\_id = ad.app\_download\_key

group by age\_range, platform, request\_dt

UNION

SELECT '4' As step, 'Total\_Revenue' AS funnel\_name, age\_range, platform,request\_ts::date AS request\_dt, SUM(purchase\_amount\_usd) AS Total FROM transactions tr

left join ride\_requests rr

on tr.ride\_id = rr.ride\_id

left join signups si

on rr.user\_id = si.user\_id

left join app\_downloads ad

on si.session\_id = ad.app\_download\_key

group by age\_range, platform, request\_dt

UNION

SELECT '5' As step, 'Approved\_Payments' AS funnel\_name, age\_range, platform,request\_ts::date AS request\_dt, SUM(purchase\_amount\_usd) AS Total FROM transactions tr

left join ride\_requests rr

on tr.ride\_id = rr.ride\_id

left join signups si

on rr.user\_id = si.user\_id

left join app\_downloads ad

on si.session\_id = ad.app\_download\_key

where charge\_status='Approved'

group by age\_range, platform, request\_dt

order by step;

**3. APPROVED PAYMENTS VS AVERAGE RATING:**

SELECT

ad.platform,

si.age\_range,

rr.request\_ts::date AS request\_dt,

rr.dropoff\_location,

SUM(tr.purchase\_amount\_usd) AS Approved\_Payment,

ROUND(AVG(re.rating),0) AS average\_rating

FROM

app\_downloads ad

LEFT JOIN signups si ON ad.app\_download\_key = si.session\_id

LEFT JOIN ride\_requests rr ON si.user\_id = rr.user\_id

LEFT JOIN transactions tr ON rr.ride\_id = tr.ride\_id

LEFT JOIN reviews re ON rr.ride\_id = re.ride\_id

GROUP BY

ad.platform,

si.age\_range,

rr.request\_ts,

rr.dropoff\_location;

**4. REQUEST VS ACCEPT, ACCEPT VS PICKUP:**

SELECT COUNT(ride\_id) AS Ride\_Count,

EXTRACT(HOUR FROM request\_ts) AS Hour\_Of\_Request,

EXTRACT(HOUR FROM accept\_ts) AS Hour\_Of\_Accept,

EXTRACT(HOUR FROM pickup\_ts) AS Hour\_Of\_Pickup,

EXTRACT(HOUR FROM cancel\_ts) AS Hour\_Of\_Cancel

FROM ride\_requests

GROUP BY Hour\_Of\_Request, Hour\_Of\_Accept, Hour\_Of\_Pickup,

Hour\_Of\_Cancel

ORDER BY Hour\_Of\_Cancel;

**5. AVERAGE PICKUP WAITING TIME:**

SELECT EXTRACT(HOUR FROM request\_ts) AS Hour\_Of\_Pickup,

ROUND(AVG((EXTRACT(HOUR FROM (pickup\_ts - request\_ts)) + EXTRACT(MINUTE FROM (pickup\_ts - request\_ts)))),2)

AS Avg\_Waiting\_Time\_Before\_ride

FROM ride\_requests

GROUP BY Hour\_Of\_Pickup

ORDER BY Hour\_Of\_Pickup;

**6. AVERAGE CANCEL WAITING TIME:**

SELECT EXTRACT(HOUR FROM cancel\_ts) AS Hour\_Of\_Cancel,

ROUND(AVG((EXTRACT(HOUR FROM (cancel\_ts - request\_ts)) + EXTRACT(MINUTE FROM (cancel\_ts - request\_ts)))),2)

AS Avg\_Waiting\_Time\_Before\_Cancel

FROM ride\_requests

GROUP BY Hour\_Of\_Cancel

ORDER BY Hour\_Of\_Cancel;

**7. CANCEL STATUS:**

Select 'All (Cancelled)' as Type,

'Cancelled' as cancel\_status,

COUNT(rr.ride\_id) AS total\_rides,

null AS total\_revenue\_in\_usd,

null AS approved\_payments

FROM ride\_requests rr

left join transactions tr

on rr.ride\_id = tr.ride\_id

where cancel\_ts is not null

UNION

Select 'Driver Accepted' as Type,

'Not cancelled' as cancel\_status,

COUNT(rr.ride\_id) AS total\_rides,

ROUND(SUM(purchase\_amount\_usd)) AS total\_revenue\_in\_usd,

ROUND(SUM(CASE WHEN charge\_status = 'Approved' THEN purchase\_amount\_usd ELSE 0 END)) AS approved\_payments

FROM ride\_requests rr

left join transactions tr

on rr.ride\_id = tr.ride\_id

where cancel\_ts is null

Union

Select 'Driver Accepted' as Type, 'Cancelled' as cancel\_status, COUNT(rr.ride\_id) AS total\_rides,

null AS total\_revenue\_in\_usd,

null AS approved\_payments

FROM ride\_requests rr

left join transactions tr

on rr.ride\_id = tr.ride\_id

where cancel\_ts is not null and driver\_id is null

Union

Select 'Driver not accepted' as Type, 'Cancelled' as cancel\_status, COUNT(rr.ride\_id) AS total\_rides,

null AS total\_revenue\_in\_usd,

null AS approved\_payments

FROM ride\_requests rr

left join transactions tr

on rr.ride\_id = tr.ride\_id

where cancel\_ts is not null and driver\_id is not null;

**8. REVIEW IN WORDS:**

SELECT

CASE

WHEN POSITION(' ' IN review) = 0 THEN review

ELSE LEFT(review, POSITION(' ' IN review) - 1)

END AS first\_word,

COUNT(\*) AS word\_count

FROM

reviews

GROUP BY first\_word

ORDER BY word\_count DESC;

**9. REVIEW RATINGWISE:**

SELECT platform,age\_range,pickup\_ts::DATE AS date,review,rating,COUNT(\*)

FROM ride\_requests rr

LEFT JOIN reviews r ON rr.ride\_id=r.ride\_id

LEFT JOIN signups s ON rr.user\_id=s.user\_id

LEFT JOIN app\_downloads ad ON s.session\_id=ad.app\_download\_key

GROUP BY review,platform,age\_range,pickup\_ts::DATE,rating;