**Namma Yatri MySQL Queries**

**1.Total Trips**

SELECT COUNT(DISTINCT tripid) AS Total\_trips

FROM trips\_details4;



**2.Total Drivers**

SELECT COUNT(DISTINCT driverid) AS Total\_drivers

FROM trips;



**3.Total Earnings**

SELECT SUM(fare) AS Total\_earnings

FROM trips;



**4.Total Completed Trips**

SELECT SUM(end\_ride) AS Total\_completed\_rides

FROM trips\_details4;



**5.Total Searches**

SELECT SUM(searches) AS total\_searches

FROM trips\_details4;



**6.Total Searches Which got Estimate**

SELECT

SUM(searches\_got\_estimate) AS total\_estimate\_searches

FROM trips\_details4;



**7.Total Searches for Quotes**

SELECT

SUM(searches\_for\_quotes) AS total\_quote\_searches

FROM trips\_details4;



**8.Total searches which got quotes**

SELECT

SUM(searches\_got\_quotes) AS total\_successful\_quotes

FROM trips\_details4;



**9.Total driver cancelled**

SELECT

SUM(driver\_not\_cancelled) AS total\_driver\_cancelled

FROM trips\_details4;



**10.Total OTP entered**

SELECT

SUM(otp\_entered) AS total\_otp\_entered

FROM trips\_details4;



**11.Total end ride**

SELECT

SUM(end\_ride) AS total\_completed\_rides

FROM trips\_details4;



**12.Cancelled bookings by driver**

SELECT

SUM(customer\_not\_cancelled) AS total\_customers\_kept,

FROM trips\_details4;



**13.Average distance per trip**

SELECT AVG(distance) AS AVG\_DISTANCE

FROM trips;



**14.Average fare per trip**

SELECT AVG(fare) AS AVG\_FARE

FROM trips;



**15.Distance travelled**

SELECT SUM(distance) AS Total\_Distance

FROM trips;



**16.Which is the most used payment method**

SELECT a.method AS Payment\_Method

FROM payment a

INNER JOIN

(

SELECT

faremethod,

COUNT(DISTINCT tripid) AS total\_trips

FROM trips

GROUP BY faremethod

ORDER BY total\_trips DESC

LIMIT 1

) b

ON a.id = b.faremethod;



**17.The highest payment was made through which instrument**

SELECT a.method AS Payment\_Method

FROM payment a

Inner join

(select

faremethod,

SUM(fare)

FROM trips

GROUP BY faremethod

ORDER BY sum(fare) desc LIMIT 1) b

WHERE a.id = b.faremethod; ;



**18.Which two locations had the most trips**

SELECT \*

FROM (

SELECT \*, DENSE\_RANK() OVER (ORDER BY total\_trips DESC) AS ranking

FROM (

SELECT

loc\_from,

loc\_to,

COUNT(DISTINCT tripid) AS total\_trips

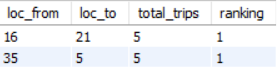
FROM trips

GROUP BY loc\_from, loc\_to

) AS a

) AS b

WHERE ranking = 1;



**19.Top 5 earning drivers**

SELECT \*

FROM (

SELECT \*, DENSE\_RANK() OVER (ORDER BY fare DESC) AS ranking

FROM (

SELECT driverid, SUM(fare) AS fare

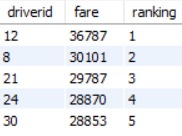
FROM trips

GROUP BY driverid

) b

) c

WHERE ranking < 6;



**20.Which duration had more trips**

SELECT \*

FROM (

SELECT \*, RANK() OVER (ORDER BY Counts DESC) AS Ranking

FROM (

SELECT duration, COUNT(DISTINCT tripid) AS Counts

FROM trips

GROUP BY duration

) b

) c

WHERE Ranking = 1;



**21.Which driver , customer pair had more orders**

SELECT \*

FROM (

SELECT \*, RANK() OVER (ORDER BY Counts DESC) AS Ranking

FROM (

SELECT driverid, custid, COUNT(DISTINCT tripid) AS Counts

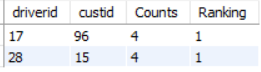
FROM trips

GROUP BY driverid, custid

) b

) c

WHERE Ranking = 1;



**22.Search to estimate rate**

SELECT

ROUND(SUM(searches\_got\_estimate) \* 100.0 / SUM(searches), 2) AS Search\_to\_estimate\_rate

FROM trips\_details4;

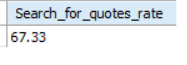


**23.Estimate to search for quote rates**

SELECT

ROUND(SUM(searches\_for\_quotes) \* 100.0 / SUM(searches), 2) AS Search\_for\_quotes\_rate

FROM trips\_details4;



**24.Which area got highest trips in which duration**

SELECT \*

FROM (

SELECT \*, RANK() OVER (PARTITION BY duration ORDER BY Counts DESC) AS Ranking

FROM (

SELECT duration, loc\_from, COUNT(DISTINCT tripid) AS Counts

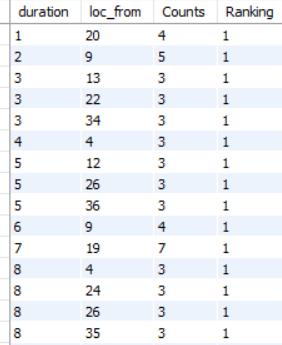
FROM trips

GROUP BY duration, loc\_from

) AS a

) AS c

WHERE Ranking = 1;



**25.Which duration got highest number of trips in each of the location present**

SELECT \*

FROM (

SELECT \*, RANK() OVER (PARTITION BY loc\_from ORDER BY Counts DESC) AS Ranking

FROM (

SELECT loc\_from ,duration, COUNT(DISTINCT tripid) AS Counts

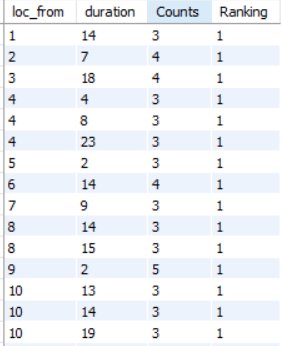
FROM trips

GROUP BY duration, loc\_from

) AS a

) AS c

WHERE Ranking = 1;



**26.Which area got the highest fares, cancellations,trips**

a. Highest fares

SELECT \*

FROM (

SELECT \*,

RANK() OVER (ORDER BY Total\_fare DESC) AS Ranking

FROM

(SELECT loc\_from,sum(fare) AS Total\_fare FROM trips

GROUP BY loc\_from) AS B

) AS a

WHERE Ranking = 1;



b. Highest cancellations

SELECT \*

FROM (

SELECT \*,

RANK() OVER (ORDER BY Cancelled DESC) AS Ranking

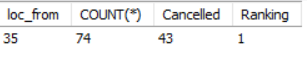
FROM

(SELECT loc\_from,COUNT(\*),sum(driver\_not\_cancelled) AS Cancelled FROM trips\_details4

GROUP BY loc\_from) AS B

) AS a

WHERE Ranking = 1;



c. Highest trips

SELECT \*

FROM (

SELECT \*,

RANK() OVER (ORDER BY Cancelled DESC) AS Ranking

FROM

(SELECT loc\_from,COUNT(\*),sum(customer\_not\_cancelled) AS Cancelled FROM trips\_details4

GROUP BY loc\_from) AS B

) AS a

WHERE Ranking = 1;



**27.Which duration got the highest trips and fares**

SELECT \*

FROM (

SELECT \*,

RANK() OVER (ORDER BY Total\_fare DESC) AS Ranking

FROM

(SELECT duration,COUNT(distinct tripid) AS Count,sum(fare) AS Total\_fare FROM trips

GROUP BY duration) AS B

) AS a

WHERE Ranking = 1;

