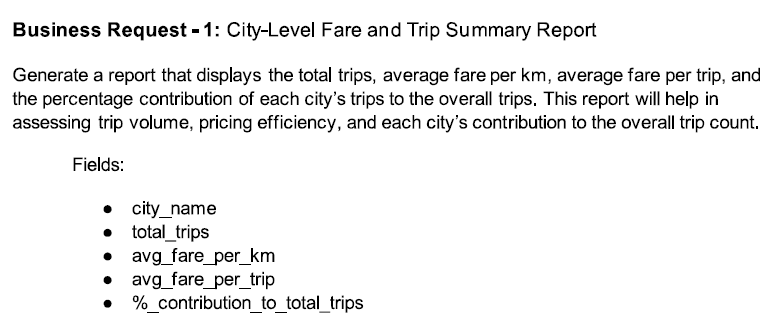
SQL QUERY FOR GOODCABS PROJECT

1. ****

Ans SELECT

dc.city\_name,

COUNT(ft.trip\_id) AS Total\_trip,

ROUND(SUM(ft.fare\_amount) / SUM(ft.distance\_travelled\_km), 2) AS Avg\_fare\_per\_km,

ROUND(SUM(ft.fare\_amount) / COUNT(ft.trip\_id), 2) AS Avg\_fare\_per\_trip,

ROUND((COUNT(ft.trip\_id) \* 100.0) / (SELECT COUNT(trip\_id) FROM fact\_trips), 2) AS Total\_trip\_percentage

FROM

fact\_trips ft

JOIN

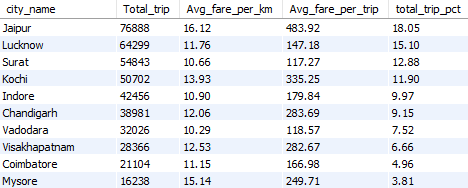
dim\_city dc

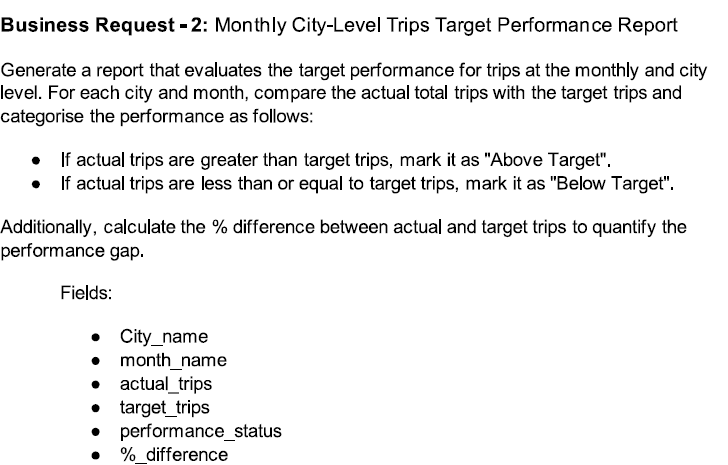
ON

dc.city\_id = ft.city\_id

GROUP BY

dc.city\_name;



1. 

Ans -- Step 1: Actual trip counts per city and month

WITH actual AS (

SELECT

dc.city\_id,

dc.city\_name,

MONTHNAME(ft.date) AS monthy,

COUNT(ft.trip\_id) AS total\_trip

FROM

fact\_trips ft

JOIN

dim\_city dc

ON

ft.city\_id = dc.city\_id

GROUP BY

dc.city\_id, dc.city\_name, MONTHNAME(ft.date)

),

-- Step 2: Target trip counts per city and month

target AS (

SELECT

dc.city\_id,

dc.city\_name,

MONTHNAME(tt.month) AS monthy,

tt.total\_target\_trips AS target\_trip

FROM

monthly\_target\_trips tt

JOIN

dim\_city dc

ON

tt.city\_id = dc.city\_id

GROUP BY

dc.city\_id, dc.city\_name, MONTHNAME(tt.month), tt.total\_target\_trips

ORDER BY

dc.city\_id ASC

)

-- Step 3: Comparing actual and target trips

SELECT

actual.city\_name,

actual.monthy,

actual.total\_trip,

target.target\_trip,

CASE

WHEN actual.total\_trip < target.target\_trip THEN 'Below Target'

ELSE 'Above Target'

END AS performance,

CONCAT(

ROUND((actual.total\_trip - target.target\_trip) \* 100 / target.target\_trip, 2), '%'

) AS pct\_difference

FROM

actual

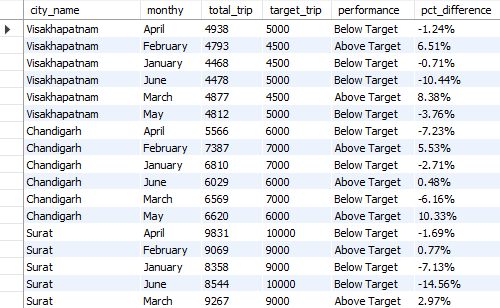
JOIN

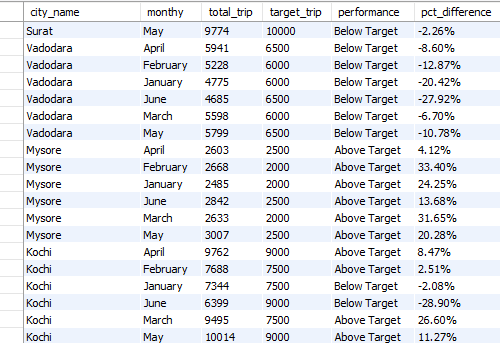
target

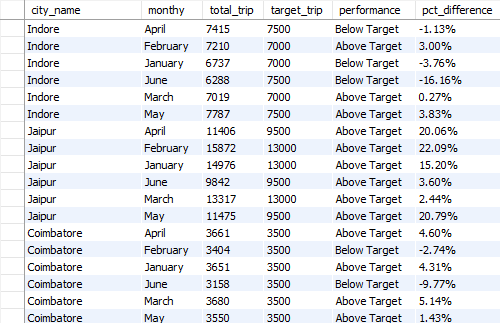
ON

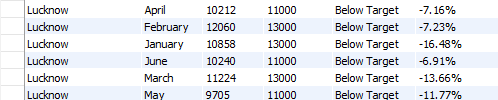
target.city\_id = actual.city\_id

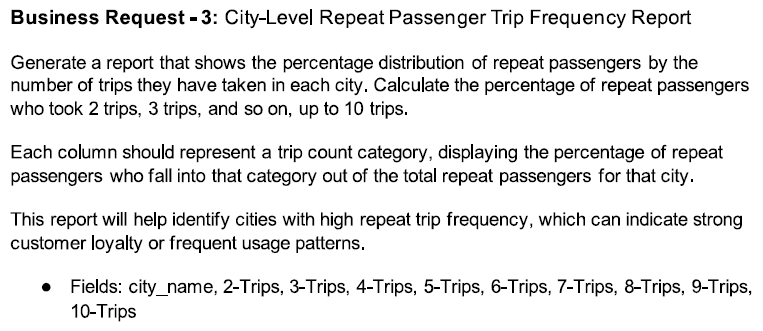
AND target.monthy = actual.monthy;









1. ****

Ans with cte1 as (SELECT

dc.city\_id,

dc.city\_name,

SUM(rt.repeat\_passenger\_count) AS Total\_Trips,

SUM(CASE WHEN rt.trip\_count = '2-Trips' THEN rt.repeat\_passenger\_count ELSE 0 END) AS Trip2\_Count,

ROUND(

100.0 \* SUM(CASE WHEN rt.trip\_count = '2-Trips' THEN rt.repeat\_passenger\_count ELSE 0 END)

/ SUM(rt.repeat\_passenger\_count), 0

) AS Trip2\_Percentage,

SUM(CASE WHEN rt.trip\_count = '3-Trips' THEN rt.repeat\_passenger\_count ELSE 0 END) AS Trip3\_Count,

ROUND(

100.0 \* SUM(CASE WHEN rt.trip\_count = '3-Trips' THEN rt.repeat\_passenger\_count ELSE 0 END)

/ SUM(rt.repeat\_passenger\_count), 0

) AS Trip3\_Percentage,

SUM(CASE WHEN rt.trip\_count = '4-Trips' THEN rt.repeat\_passenger\_count ELSE 0 END) AS Trip4\_Count,

ROUND(

100.0 \* SUM(CASE WHEN rt.trip\_count = '4-Trips' THEN rt.repeat\_passenger\_count ELSE 0 END)

/ SUM(rt.repeat\_passenger\_count), 0

) AS Trip4\_Percentage,

SUM(CASE WHEN rt.trip\_count = '5-Trips' THEN rt.repeat\_passenger\_count ELSE 0 END) AS Trip5\_Count,

ROUND(

100.0 \* SUM(CASE WHEN rt.trip\_count = '5-Trips' THEN rt.repeat\_passenger\_count ELSE 0 END)

/ SUM(rt.repeat\_passenger\_count), 0

) AS Trip5\_Percentage,

SUM(CASE WHEN rt.trip\_count = '6-Trips' THEN rt.repeat\_passenger\_count ELSE 0 END) AS Trip6\_Count,

ROUND(

100.0 \* SUM(CASE WHEN rt.trip\_count = '6-Trips' THEN rt.repeat\_passenger\_count ELSE 0 END)

/ SUM(rt.repeat\_passenger\_count), 0

) AS Trip6\_Percentage,

SUM(CASE WHEN rt.trip\_count = '7-Trips' THEN rt.repeat\_passenger\_count ELSE 0 END) AS Trip7\_Count,

ROUND(

100.0 \* SUM(CASE WHEN rt.trip\_count = '7-Trips' THEN rt.repeat\_passenger\_count ELSE 0 END)

/ SUM(rt.repeat\_passenger\_count), 0

) AS Trip7\_Percentage,

SUM(CASE WHEN rt.trip\_count = '8-Trips' THEN rt.repeat\_passenger\_count ELSE 0 END) AS Trip8\_Count,

ROUND(

100.0 \* SUM(CASE WHEN rt.trip\_count = '8-Trips' THEN rt.repeat\_passenger\_count ELSE 0 END)

/ SUM(rt.repeat\_passenger\_count), 0

) AS Trip8\_Percentage,

SUM(CASE WHEN rt.trip\_count = '9-Trips' THEN rt.repeat\_passenger\_count ELSE 0 END) AS Trip9\_Count,

ROUND(

100.0 \* SUM(CASE WHEN rt.trip\_count = '9-Trips' THEN rt.repeat\_passenger\_count ELSE 0 END)

/ SUM(rt.repeat\_passenger\_count), 0

) AS Trip9\_Percentage,

SUM(CASE WHEN rt.trip\_count = '10-Trips' THEN rt.repeat\_passenger\_count ELSE 0 END) AS Trip10\_Count,

ROUND(

100.0 \* SUM(CASE WHEN rt.trip\_count = '10-Trips' THEN rt.repeat\_passenger\_count ELSE 0 END)

/ SUM(rt.repeat\_passenger\_count), 0

) AS Trip10\_Percentage

FROM

dim\_repeat\_trip\_distribution rt

JOIN

dim\_city dc ON dc.city\_id = rt.city\_id

GROUP BY

dc.city\_id,

dc.city\_name

ORDER BY

Total\_Trips DESC)

select city\_id,city\_name,Total\_Trips,concat(Trip2\_Percentage,'%')as Trips\_2\_pct,

concat(Trip3\_Percentage,'%')as Trips\_3\_pct,concat(Trip4\_Percentage,'%')as Trips\_4\_pct,

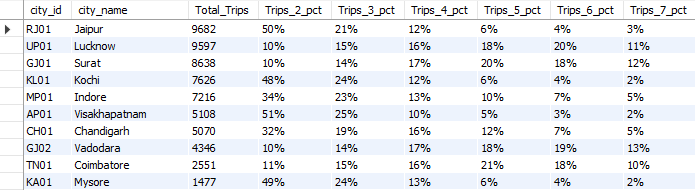
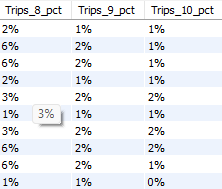
concat(Trip5\_Percentage,'%')as Trips\_5\_pct,concat(Trip6\_Percentage,'%')as Trips\_6\_pct,

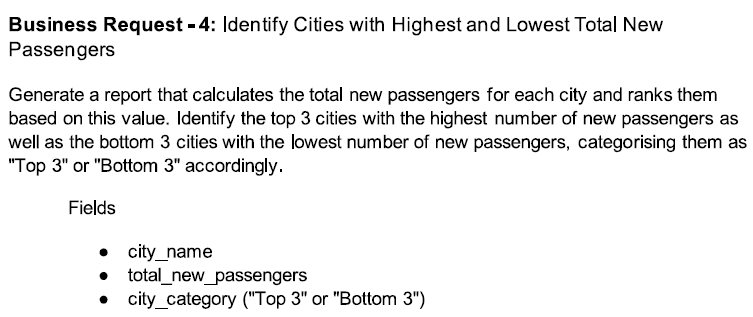
concat(Trip7\_Percentage,'%')as Trips\_7\_pct,concat(Trip8\_Percentage,'%')as Trips\_8\_pct,

concat(Trip9\_Percentage,'%')as Trips\_9\_pct,concat(Trip10\_Percentage,'%')as Trips\_10\_pct

from cte1

order by total\_trips desc

1. ****

Ans WITH ranked\_passengers AS (

SELECT

dc.city\_name,

SUM(ps.new\_passengers) AS total\_new\_passenger,

RANK() OVER (ORDER BY SUM(ps.new\_passengers) DESC) AS rank\_desc,

RANK() OVER (ORDER BY SUM(ps.new\_passengers) ASC) AS rank\_asc

FROM

fact\_passenger\_summary ps

JOIN

dim\_city dc

ON

dc.city\_id = ps.city\_id

GROUP BY

dc.city\_name

)

SELECT

city\_name,

total\_new\_passenger,

CASE

WHEN rank\_desc <= 3 THEN 'Top 3'

WHEN rank\_asc <= 3 THEN 'Bottom 3'

END AS city\_category

FROM

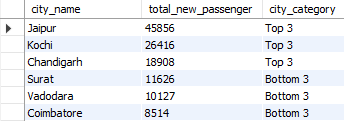
ranked\_passengers

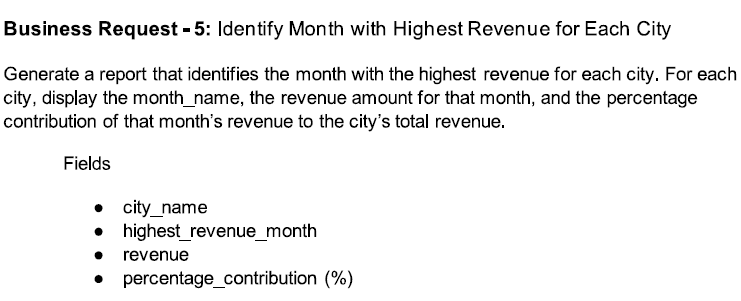
WHERE

rank\_desc <= 3 OR rank\_asc <= 3

ORDER BY

total\_new\_passenger DESC;



1. ****

Ans WITH cte1 AS (

SELECT

dc.city\_name,

MONTHNAME(ft.date) AS Month,

SUM(ft.fare\_amount) AS total\_revenue,

DENSE\_RANK() OVER (PARTITION BY dc.city\_name ORDER BY SUM(ft.fare\_amount) DESC) AS Ranke

FROM

fact\_trips ft

JOIN

dim\_city dc

ON

ft.city\_id = dc.city\_id

GROUP BY

dc.city\_name, Month

),

cte2 AS (

SELECT

cte1.city\_name,

cte1.Month,

cte1.total\_revenue AS Highest\_revenue

FROM

cte1

WHERE

cte1.Ranke = 1

),

cte3 AS (

SELECT

city\_name,

SUM(total\_revenue) AS total\_city\_revenue

FROM

cte1

GROUP BY

city\_name

)

SELECT

cte2.city\_name,

cte2.Month,

cte2.highest\_revenue,

concat (round((cte2.highest\_revenue \* 100.0 / cte3.total\_city\_revenue),2),'%') AS Pct\_Distribution

FROM

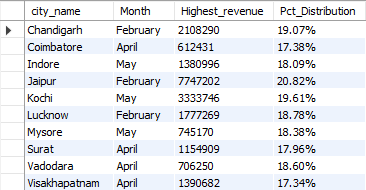
cte2

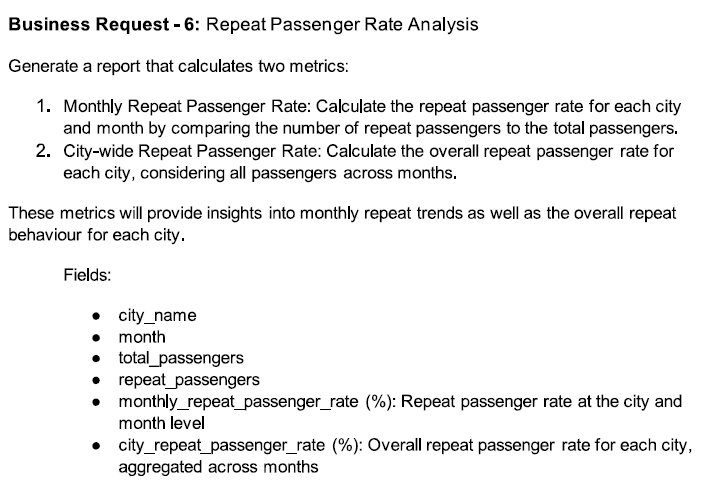
JOIN

cte3

ON

cte2.city\_name = cte3.city\_name;



1. ****

Ans WITH cte1 AS (

SELECT

dc.city\_name,

MONTHNAME(ps.month) AS Month,

SUM(ps.total\_passengers) AS Total\_Passenger,

SUM(ps.repeat\_passengers) AS Total\_Repeat,

CONCAT(ROUND(SUM(ps.repeat\_passengers) \* 100 / SUM(ps.total\_passengers), 2), '%') AS Monthly\_Repeat\_Passenger\_Rate

FROM

fact\_passenger\_summary ps

JOIN

dim\_city dc

ON

ps.city\_id = dc.city\_id

GROUP BY

dc.city\_name, MONTHNAME(ps.month)

),

cte2 AS (

SELECT

dc.city\_name,

SUM(ps.total\_passengers) AS Total\_Passenger,

SUM(ps.repeat\_passengers) AS Total\_Repeat,

CONCAT(ROUND(SUM(ps.repeat\_passengers) \* 100 / SUM(ps.total\_passengers), 2), '%') AS City\_Repeat\_Passenger\_Rate

FROM

fact\_passenger\_summary ps

JOIN

dim\_city dc

ON

ps.city\_id = dc.city\_id

GROUP BY

dc.city\_name

)

SELECT

cte1.city\_name,

cte1.Month,

cte1.Total\_Passenger,

cte1.Total\_Repeat,

cte1.Monthly\_Repeat\_Passenger\_Rate,

cte2.City\_Repeat\_Passenger\_Rate

FROM

cte1

JOIN

cte2

ON

cte1.city\_name = cte2.city\_name

;

