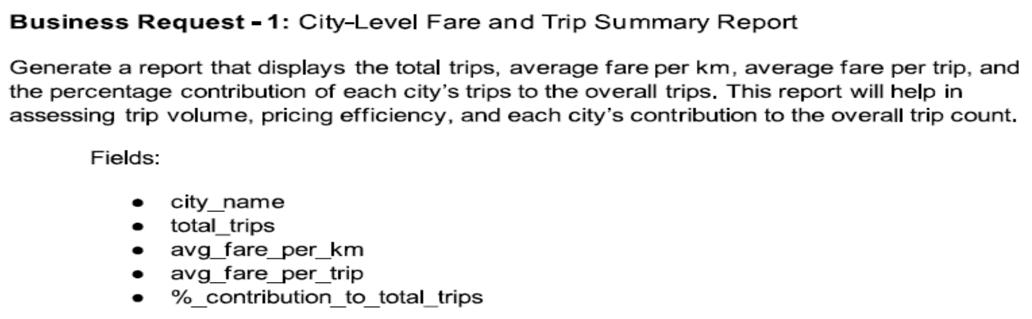
SQL QUERY FOR GOODCABS PROJECT



Business Request - 1: City-Level Fare and Trip Summary Report

Generate a report that displays the total trips, average fare per km, average fare per trip, and the percentage contribution of each city's trips to the overall trips. This report will help in assessing trip volume, pricing efficiency, and each city's contribution to the overall trip count.

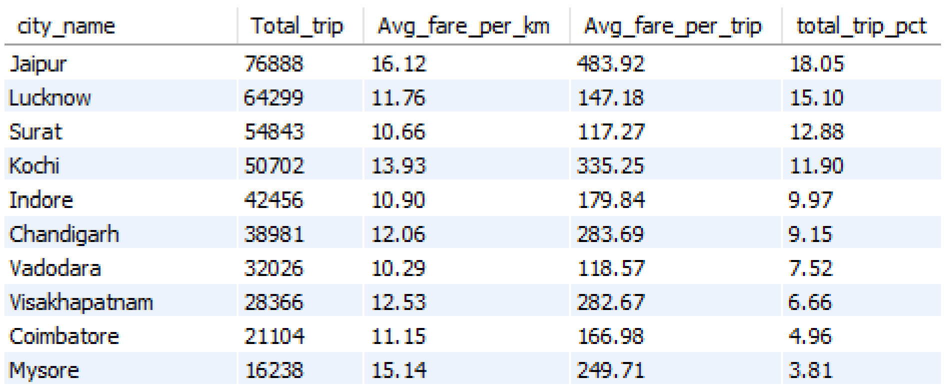
Fields:

- city_name
- total_trips
- avg_fare_per_km
- avg_fare_per_trip
- %_contribution_to_total_trips

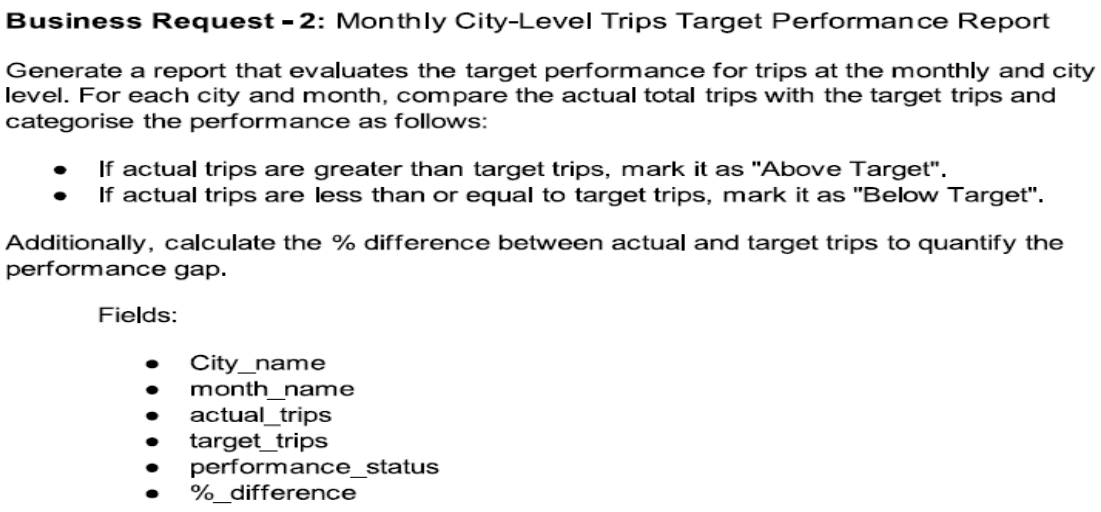
1.

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;
```



city_name	Total_trip	Avg_fare_per_km	Avg_fare_per_trip	total_trip_pct
Jaipur	76888	16.12	483.92	18.05
Lucknow	64299	11.76	147.18	15.10
Surat	54843	10.66	117.27	12.88
Kochi	50702	13.93	335.25	11.90
Indore	42456	10.90	179.84	9.97
Chandigarh	38981	12.06	283.69	9.15
Vadodara	32026	10.29	118.57	7.52
Visakhapatnam	28366	12.53	282.67	6.66
Coimbatore	21104	11.15	166.98	4.96
Mysore	16238	15.14	249.71	3.81



Business Request - 2: Monthly City-Level Trips Target Performance Report

Generate a report that evaluates the target performance for trips at the monthly and city level. For each city and month, compare the actual total trips with the target trips and categorise the performance as follows:

- If actual trips are greater than target trips, mark it as "Above Target".
- If actual trips are less than or equal to target trips, mark it as "Below Target".

Additionally, calculate the % difference between actual and target trips to quantify the performance gap.

Fields:

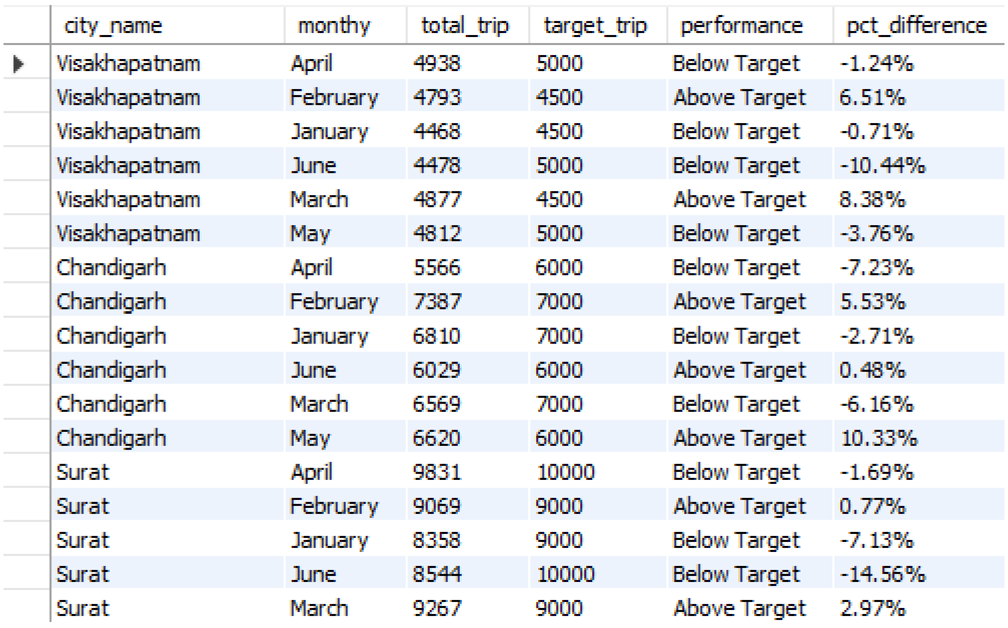
- City_name
- month_name
- actual_trips
- target_trips
- performance_status
- %_difference

2.

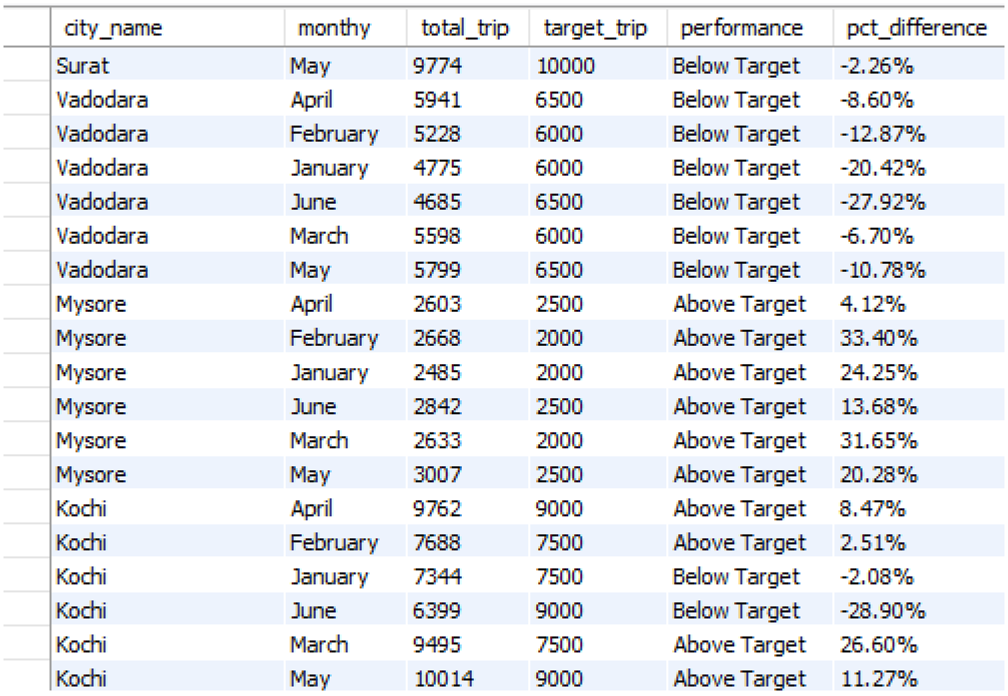
Ans -- Step 1: Actual trip counts per city and month

```
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)
),
```

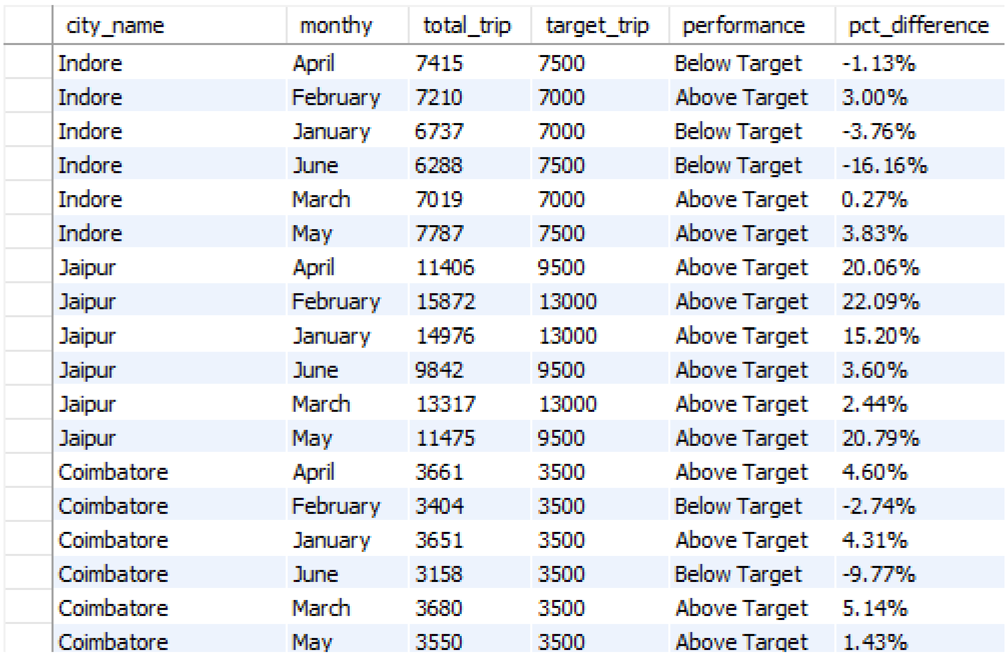
```sql
-- Step 2: Target trip counts per city and month

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;
```



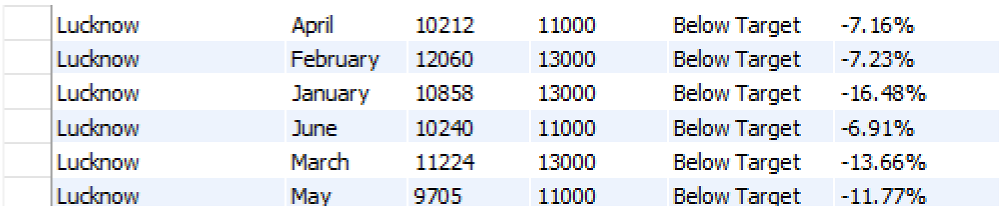
city_name	monthy	total_trip	target_trip	performance	pct_difference
Visakhapatnam	April	4938	5000	Below Target	-1.24%
Visakhapatnam	February	4793	4500	Above Target	6.51%
Visakhapatnam	January	4468	4500	Below Target	-0.71%
Visakhapatnam	June	4478	5000	Below Target	-10.44%
Visakhapatnam	March	4877	4500	Above Target	8.38%
Visakhapatnam	May	4812	5000	Below Target	-3.76%
Chandigarh	April	5566	6000	Below Target	-7.23%
Chandigarh	February	7387	7000	Above Target	5.53%
Chandigarh	January	6810	7000	Below Target	-2.71%
Chandigarh	June	6029	6000	Above Target	0.48%
Chandigarh	March	6569	7000	Below Target	-6.16%
Chandigarh	May	6620	6000	Above Target	10.33%
Surat	April	9831	10000	Below Target	-1.69%
Surat	February	9069	9000	Above Target	0.77%
Surat	January	8358	9000	Below Target	-7.13%
Surat	June	8544	10000	Below Target	-14.56%
Surat	March	9267	9000	Above Target	2.97%



city_name	monthy	total_trip	target_trip	performance	pct_difference
Surat	May	9774	10000	Below Target	-2.26%
Vadodara	April	5941	6500	Below Target	-8.60%
Vadodara	February	5228	6000	Below Target	-12.87%
Vadodara	January	4775	6000	Below Target	-20.42%
Vadodara	June	4685	6500	Below Target	-27.92%
Vadodara	March	5598	6000	Below Target	-6.70%
Vadodara	May	5799	6500	Below Target	-10.78%
Mysore	April	2603	2500	Above Target	4.12%
Mysore	February	2668	2000	Above Target	33.40%
Mysore	January	2485	2000	Above Target	24.25%
Mysore	June	2842	2500	Above Target	13.68%
Mysore	March	2633	2000	Above Target	31.65%
Mysore	May	3007	2500	Above Target	20.28%
Kochi	April	9762	9000	Above Target	8.47%
Kochi	February	7688	7500	Above Target	2.51%
Kochi	January	7344	7500	Below Target	-2.08%
Kochi	June	6399	9000	Below Target	-28.90%
Kochi	March	9495	7500	Above Target	26.60%
Kochi	May	10014	9000	Above Target	11.27%



city_name	monthy	total_trip	target_trip	performance	pct_difference
Indore	April	7415	7500	Below Target	-1.13%
Indore	February	7210	7000	Above Target	3.00%
Indore	January	6737	7000	Below Target	-3.76%
Indore	June	6288	7500	Below Target	-16.16%
Indore	March	7019	7000	Above Target	0.27%
Indore	May	7787	7500	Above Target	3.83%
Jaipur	April	11406	9500	Above Target	20.06%
Jaipur	February	15872	13000	Above Target	22.09%
Jaipur	January	14976	13000	Above Target	15.20%
Jaipur	June	9842	9500	Above Target	3.60%
Jaipur	March	13317	13000	Above Target	2.44%
Jaipur	May	11475	9500	Above Target	20.79%
Coimbatore	April	3661	3500	Above Target	4.60%
Coimbatore	February	3404	3500	Below Target	-2.74%
Coimbatore	January	3651	3500	Above Target	4.31%
Coimbatore	June	3158	3500	Below Target	-9.77%
Coimbatore	March	3680	3500	Above Target	5.14%
Coimbatore	May	3550	3500	Above Target	1.43%



Lucknow	April	10212	11000	Below Target	-7.16%
Lucknow	February	12060	13000	Below Target	-7.23%
Lucknow	January	10858	13000	Below Target	-16.48%
Lucknow	June	10240	11000	Below Target	-6.91%
Lucknow	March	11224	13000	Below Target	-13.66%
Lucknow	May	9705	11000	Below Target	-11.77%

Business Request - 3: City-Level Repeat Passenger Trip Frequency Report

Generate a report that shows the percentage distribution of repeat passengers by the number of trips they have taken in each city. Calculate the percentage of repeat passengers who took 2 trips, 3 trips, and so on, up to 10 trips.

Each column should represent a trip count category, displaying the percentage of repeat passengers who fall into that category out of the total repeat passengers for that city.

This report will help identify cities with high repeat trip frequency, which can indicate strong customer loyalty or frequent usage patterns.

- Fields: city_name, 2-Trips, 3-Trips, 4-Trips, 5-Trips, 6-Trips, 7-Trips, 8-Trips, 9-Trips, 10-Trips

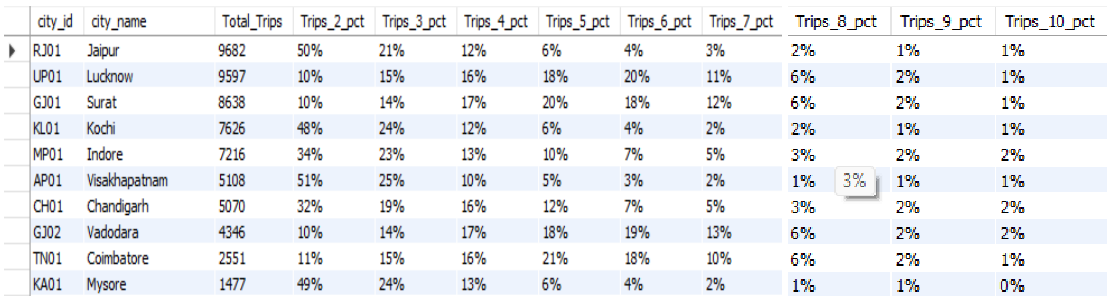
3.

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
```



city_id	city_name	Total_Trips	Trips_2_pct	Trips_3_pct	Trips_4_pct	Trips_5_pct	Trips_6_pct	Trips_7_pct	Trips_8_pct	Trips_9_pct	Trips_10_pct
RJ01	Jaipur	9682	50%	21%	12%	6%	4%	3%	2%	1%	1%
UP01	Lucknow	9597	10%	15%	16%	18%	20%	11%	6%	2%	1%
GJ01	Surat	8638	10%	14%	17%	20%	18%	12%	6%	2%	1%
KL01	Kochi	7626	48%	24%	12%	6%	4%	2%	2%	1%	1%
MP01	Indore	7216	34%	23%	13%	10%	7%	5%	3%	2%	2%
AP01	Visakhapatnam	5108	51%	25%	10%	5%	3%	2%	1%	1%	1%
CH01	Chandigarh	5070	32%	19%	16%	12%	7%	5%	3%	2%	2%
GJ02	Vadodara	4346	10%	14%	17%	18%	19%	13%	6%	2%	2%
TN01	Coimbatore	2551	11%	15%	16%	21%	18%	10%	6%	2%	1%
KA01	Mysore	1477	49%	24%	13%	6%	4%	2%	1%	1%	0%

Business Request - 4: Identify Cities with Highest and Lowest Total New Passengers

Generate a report that calculates the total new passengers for each city and ranks them based on this value. Identify the top 3 cities with the highest number of new passengers as well as the bottom 3 cities with the lowest number of new passengers, categorising them as "Top 3" or "Bottom 3" accordingly.

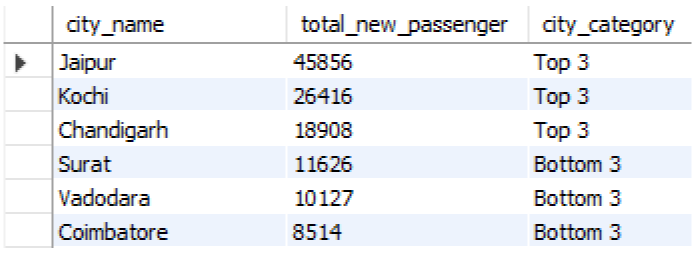
Fields

- city_name
- total_new_passengers
- city_category ("Top 3" or "Bottom 3")

4.

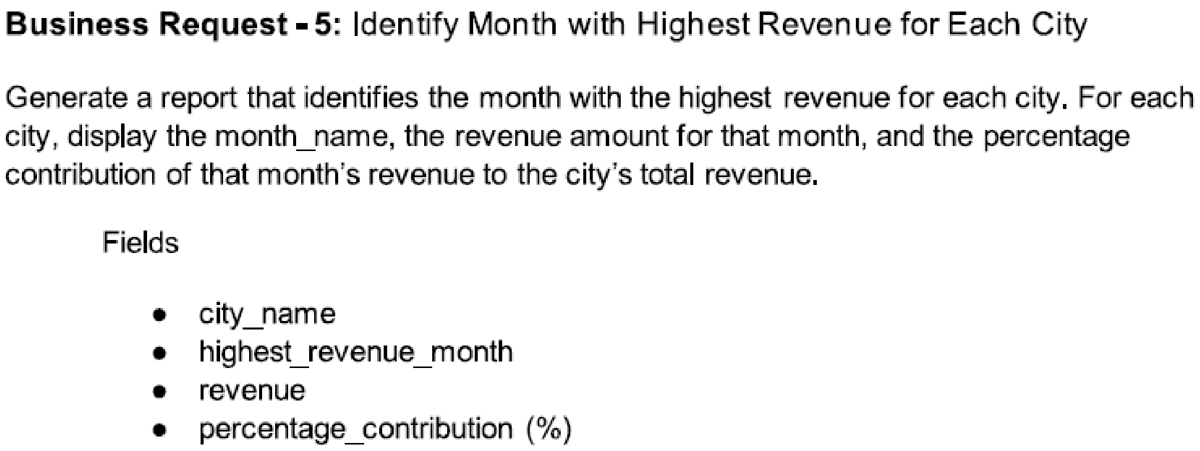
Ans

```
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;
```



	city_name	total_new_passenger	city_category
▶	Jaipur	45856	Top 3
	Kochi	26416	Top 3
	Chandigarh	18908	Top 3
	Surat	11626	Bottom 3
	Vadodara	10127	Bottom 3
	Coimbatore	8514	Bottom 3

5.



Business Request - 5: Identify Month with Highest Revenue for Each City

Generate a report that identifies the month with the highest revenue for each city. For each city, display the month_name, the revenue amount for that month, and the percentage contribution of that month's revenue to the city's total revenue.

Fields

- city_name
- highest_revenue_month
- revenue
- percentage_contribution (%)

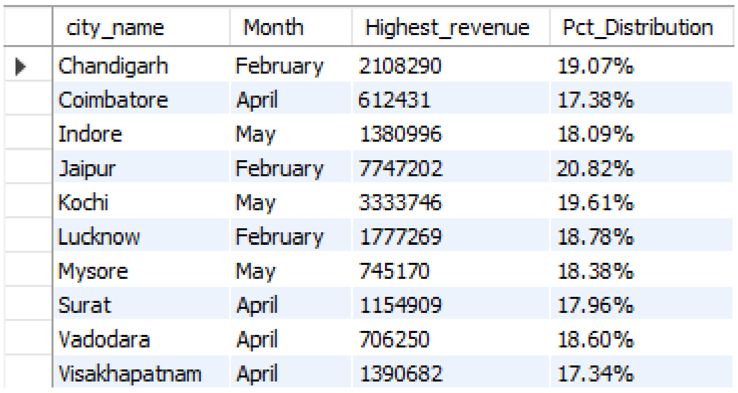
Ans

```
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
```

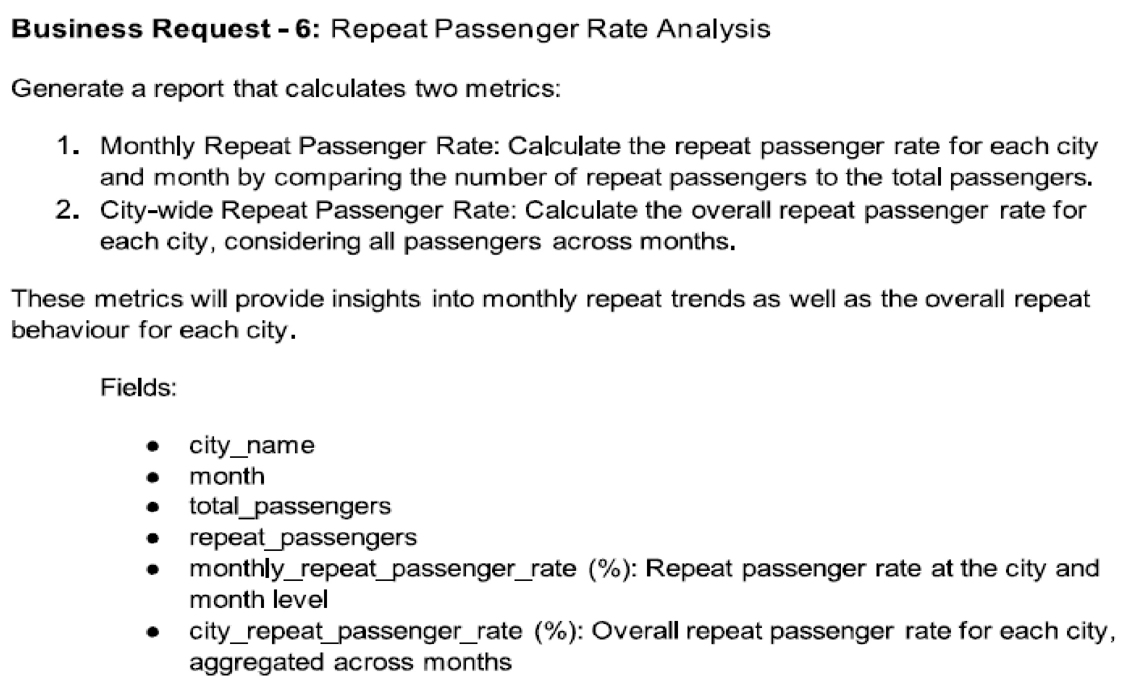
```
  GROUP BY
    dc.city_name, Month
),
cte2 AS (
  SELECT
    cte1.city_name,
    cte1.Month,
    cte1.total_revenue AS Highest_revenue
  FROM
    cte1
  WHERE
    cte1.Ranke = 1
),
cte3 AS (
  SELECT
    city_name,
    SUM(total_revenue) AS total_city_revenue
  FROM
    cte1
  GROUP BY
    city_name
)
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;



city_name	Month	Highest_revenue	Pct_Distribution
Chandigarh	February	2108290	19.07%
Coimbatore	April	612431	17.38%
Indore	May	1380996	18.09%
Jaipur	February	7747202	20.82%
Kochi	May	3333746	19.61%
Lucknow	February	1777269	18.78%
Mysore	May	745170	18.38%
Surat	April	1154909	17.96%
Vadodara	April	706250	18.60%
Visakhapatnam	April	1390682	17.34%

6. 

Business Request - 6: Repeat Passenger Rate Analysis

Generate a report that calculates two metrics:

1. Monthly Repeat Passenger Rate: Calculate the repeat passenger rate for each city and month by comparing the number of repeat passengers to the total passengers.
2. City-wide Repeat Passenger Rate: Calculate the overall repeat passenger rate for each city, considering all passengers across months.

These metrics will provide insights into monthly repeat trends as well as the overall repeat behaviour for each city.

Fields:

- city_name
- month
- total_passengers
- repeat_passengers
- monthly_repeat_passenger_rate (%): Repeat passenger rate at the city and month level
- city_repeat_passenger_rate (%): Overall repeat passenger rate for each city, aggregated across months

Ans WITH cte1 AS (

SELECT

dc.city_name,

MONTHNAME(ps.month) AS Month,

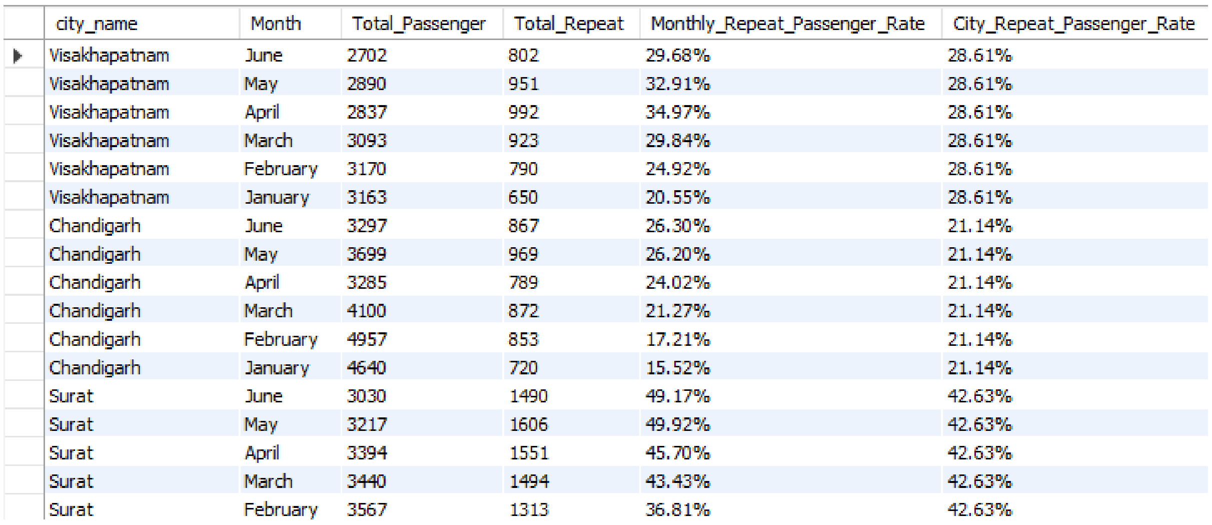
SUM(ps.total_passengers) AS Total_Passenger,

SUM(ps.repeat_passengers) AS Total_Repeat,

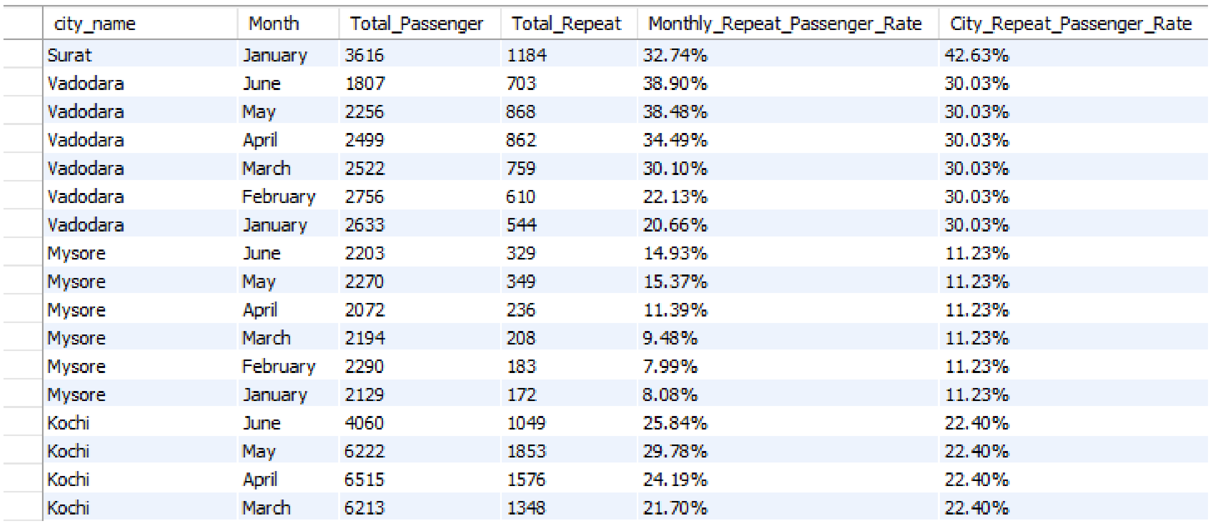
CONCAT(ROUND(SUM(ps.repeat_passengers) * 100 / SUM(ps.total_passengers), 2), '%') AS Monthly_Repeat_Passenger_Rate

FROM

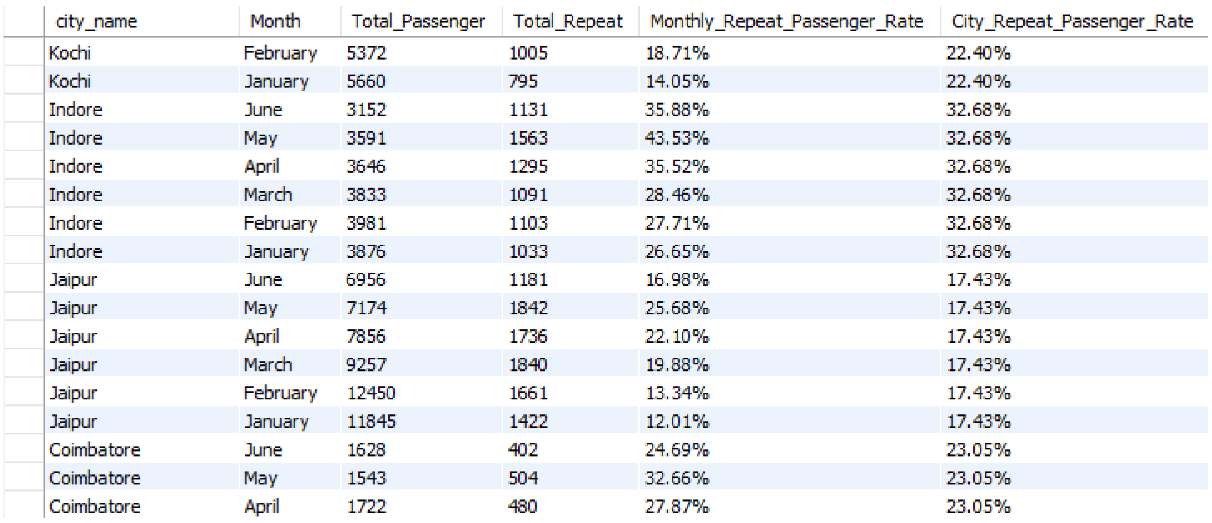
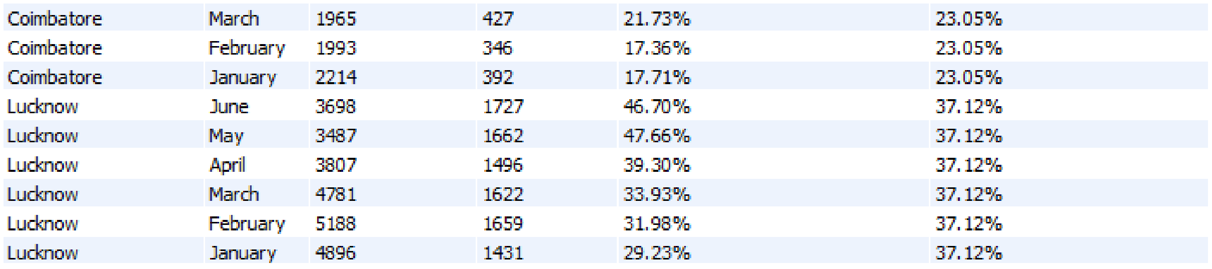
```
        fact_passenger_summary ps
    JOIN
        dim_city dc
    ON
        ps.city_id = dc.city_id
    GROUP BY
        dc.city_name, MONTHNAME(ps.month)
),
cte2 AS (
    SELECT
        dc.city_name,
        SUM(ps.total_passengers) AS Total_Passenger,
        SUM(ps.repeat_passengers) AS Total_Repeat,
        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
;
```



city_name	Month	Total_Passenger	Total_Repeat	Monthly_Repeat_Passenger_Rate	City_Repeat_Passenger_Rate
Visakhapatnam	June	2702	802	29.68%	28.61%
Visakhapatnam	May	2890	951	32.91%	28.61%
Visakhapatnam	April	2837	992	34.97%	28.61%
Visakhapatnam	March	3093	923	29.84%	28.61%
Visakhapatnam	February	3170	790	24.92%	28.61%
Visakhapatnam	January	3163	650	20.55%	28.61%
Chandigarh	June	3297	867	26.30%	21.14%
Chandigarh	May	3699	969	26.20%	21.14%
Chandigarh	April	3285	789	24.02%	21.14%
Chandigarh	March	4100	872	21.27%	21.14%
Chandigarh	February	4957	853	17.21%	21.14%
Chandigarh	January	4640	720	15.52%	21.14%
Surat	June	3030	1490	49.17%	42.63%
Surat	May	3217	1606	49.92%	42.63%
Surat	April	3394	1551	45.70%	42.63%
Surat	March	3440	1494	43.43%	42.63%
Surat	February	3567	1313	36.81%	42.63%



city_name	Month	Total_Passenger	Total_Repeat	Monthly_Repeat_Passenger_Rate	City_Repeat_Passenger_Rate
Surat	January	3616	1184	32.74%	42.63%
Vadodara	June	1807	703	38.90%	30.03%
Vadodara	May	2256	868	38.48%	30.03%
Vadodara	April	2499	862	34.49%	30.03%
Vadodara	March	2522	759	30.10%	30.03%
Vadodara	February	2756	610	22.13%	30.03%
Vadodara	January	2633	544	20.66%	30.03%
Mysore	June	2203	329	14.93%	11.23%
Mysore	May	2270	349	15.37%	11.23%
Mysore	April	2072	236	11.39%	11.23%
Mysore	March	2194	208	9.48%	11.23%
Mysore	February	2290	183	7.99%	11.23%
Mysore	January	2129	172	8.08%	11.23%
Kochi	June	4060	1049	25.84%	22.40%
Kochi	May	6222	1853	29.78%	22.40%
Kochi	April	6515	1576	24.19%	22.40%
Kochi	March	6213	1348	21.70%	22.40%

city_name	Month	Total_Passenger	Total_Repeat	Monthly_Repeat_Passenger_Rate	City_Repeat_Passenger_Rate
Kochi	February	5372	1005	18.71%	22.40%
Kochi	January	5660	795	14.05%	22.40%
Indore	June	3152	1131	35.88%	32.68%
Indore	May	3591	1563	43.53%	32.68%
Indore	April	3646	1295	35.52%	32.68%
Indore	March	3833	1091	28.46%	32.68%
Indore	February	3981	1103	27.71%	32.68%
Indore	January	3876	1033	26.65%	32.68%
Jaipur	June	6956	1181	16.98%	17.43%
Jaipur	May	7174	1842	25.68%	17.43%
Jaipur	April	7856	1736	22.10%	17.43%
Jaipur	March	9257	1840	19.88%	17.43%
Jaipur	February	12450	1661	13.34%	17.43%
Jaipur	January	11845	1422	12.01%	17.43%
Coimbatore	June	1628	402	24.69%	23.05%
Coimbatore	May	1543	504	32.66%	23.05%
Coimbatore	April	1722	480	27.87%	23.05%
Coimbatore	March	1965	427	21.73%	23.05%
Coimbatore	February	1993	346	17.36%	23.05%
Coimbatore	January	2214	392	17.71%	23.05%
Lucknow	June	3698	1727	46.70%	37.12%
Lucknow	May	3487	1662	47.66%	37.12%
Lucknow	April	3807	1496	39.30%	37.12%
Lucknow	March	4781	1622	33.93%	37.12%
Lucknow	February	5188	1659	31.98%	37.12%
Lucknow	January	4896	1431	29.23%	37.12%