





















Ad Hoc SQL Analysis Report – Goodcabs

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

```
use gc_trips;
select c.city_name, count(t.trip_id) as total_trips,
round(sum(t.fare_amount) / sum(t.distance_travelled_km), 2) as avg_fare_per_km,
round(avg(t.fare_amount), 2) as avg_fare_per_trip,
round((count(t.trip_id) / (select count(*) from fact_trips)) * 100, 2) as pct_contribution_to_total_trips
from fact_trips t
join dim_city c
on t.city_id = c.city_id
group by c.city_name;
```

City Name	Total Trips	Avg fare per km	Avg fare per trip	% Contribution
Chandigarh	 38981	₹ 12.06	₹ 283.69	 9.15%
Coimbatore	 21104	₹ 11.15	₹ 166.98	 4.96%
Indore	 42456	₹ 10.90	₹ 179.84	 9.97%
Jaipur	 76888	₹ 16.12	₹ 483.92	 18.05%
Kochi	 50702	₹ 13.93	₹ 335.25	 11.90%
Lucknow	 64299	₹ 11.76	₹ 147.18	 15.10%
Mysore	 16238	₹ 15.14	₹ 249.71	 3.81%
Surat	 54843	₹ 10.66	₹ 117.27	 12.88%
Vadodara	 32026	₹ 10.29	₹ 118.57	 7.52%
Visakhapatnam	 28366	₹ 12.53	₹ 282.67	 6.66%

(SQL Query & Power BI Visualization)

Insight - Overall, Jaipur and Lucknow have the highest trip volumes, while cities like Mysore and Coimbatore have fewer trips but higher fare per km, showing that demand and pricing vary from city to city.

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.

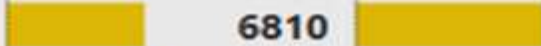

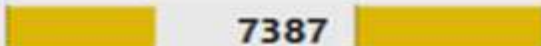



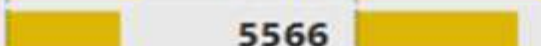



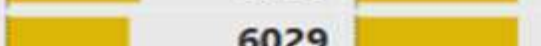

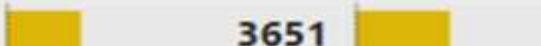

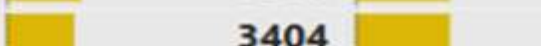

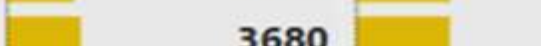



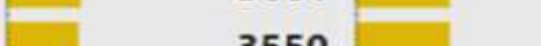


Fields: City_name, month name, actual_trips, target_trips, performance_status, % difference

```
use gc_trips;
select dc.city_name, dd.month_name,
count(ft.trip_id) as actual_trips,
mt.total_target_trips as target_trips,

    case
    when count(ft.trip_id) > mt.total_target_trips then "Above Target"
    else "Below Target"
    end as performance_status,

    round(((count(ft.trip_id) - mt.total_target_trips) / mt.total_target_trips) * 100 , 2) as pct_difference

from gc_trips.fact_trips ft
join gc_trips.dim_city dc on ft.city_id = dc.city_id
join gc_trips.dim_date dd on ft.date = dd.date
join gc_targets.monthly_target_trips mt on ft.city_id = mt.city_id AND dd.start_of_month = mt.month
group by dc.city_id, dd.start_of_month, dd.month_name;
```


City Name	Actual Trips	Target Trips	Performance Status	% Difference
<input type="checkbox"/> Chandigarh	38981	39000	Below Target	-0.05
Jan 2024	 6810	 7000	Below Target	-2.71
Feb 2024	 7387	 7000	Above Target	5.53
Mar 2024	 6569	 7000	Below Target	-6.16
Apr 2024	 5566	 6000	Below Target	-7.23
May 2024	 6620	 6000	Above Target	10.33
Jun 2024	 6029	 6000	Above Target	0.48
<input type="checkbox"/> Coimbatore	21104	21000	Above Target	0.50
Jan 2024	 3651	 3500	Above Target	4.31
Feb 2024	 3404	 3500	Below Target	-2.74
Mar 2024	 3680	 3500	Above Target	5.14
Apr 2024	 3661	 3500	Above Target	4.60
May 2024	 3550	 3500	Above Target	1.43
Jun 2024	 3158	 3500	Below Target	-9.77
<input type="checkbox"/> Indore	42456	43500	Below Target	-2.40
<input type="checkbox"/> Jaipur	76888	67500	Above Target	13.91
<input type="checkbox"/> Kochi	50702	49500	Above Target	2.43
<input type="checkbox"/> Lucknow	64299	72000	Below Target	-10.70
<input type="checkbox"/> Mysore	16238	13500	Above Target	20.28
<input type="checkbox"/> Surat	54843	57000	Below Target	-3.78
<input type="checkbox"/> Vadodara	32026	37500	Below Target	-14.60
<input type="checkbox"/> Visakhapatnam	28366	28500	Below Target	-0.47

(SQL Query & Power BI Visualization)

Insight - City-wise trip performance reveals a mix of trends across months. While Jaipur and Mysore consistently outperformed their targets, cities like Lucknow and Vadodara stayed below target throughout. Cities such as Chandigarh, Coimbatore, and Kochi displayed fluctuating performance, with some months exceeding targets and others falling short—highlighting varying demand or operational shifts over time.

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

```
use gc_trips;
with rp_by_city_and_tripcount as
  (select dc.city_name, drt.trip_count,
    sum(drt.repeat_passenger_count) as repeat_count
  from dim_repeat_trip_distribution drt
  join dim_city dc
  on drt.city_id = dc.city_id
  where drt.trip_count in('2-Trips', '3-Trips', '4-Trips', '5-Trips', '6-Trips',
    '7-Trips', '8-Trips', '9-Trips', '10-Trips')
  group by dc.city_name, drt.trip_count),

total_rp_by_city as
  (select city_name,
    sum(repeat_count) as total_repeat_passengers
  from rp_by_city_and_tripcount
  group by city_name)
```

```
select rpct.city_name,
  round(sum(case when rpct.trip_count = '2-Trips' then rpct.repeat_count else 0 end) * 100 / trp.total_repeat_passengers, 2) as '2-Trips',
  round(sum(case when rpct.trip_count = '3-Trips' then rpct.repeat_count else 0 end) * 100 / trp.total_repeat_passengers, 2) as '3-Trips',
  round(sum(case when rpct.trip_count = '4-Trips' then rpct.repeat_count else 0 end) * 100 / trp.total_repeat_passengers, 2) as '4-Trips',
  round(sum(case when rpct.trip_count = '5-Trips' then rpct.repeat_count else 0 end) * 100 / trp.total_repeat_passengers, 2) as '5-Trips',
  round(sum(case when rpct.trip_count = '6-Trips' then rpct.repeat_count else 0 end) * 100 / trp.total_repeat_passengers, 2) as '6-Trips',
  round(sum(case when rpct.trip_count = '7-Trips' then rpct.repeat_count else 0 end) * 100 / trp.total_repeat_passengers, 2) as '7-Trips',
  round(sum(case when rpct.trip_count = '8-Trips' then rpct.repeat_count else 0 end) * 100 / trp.total_repeat_passengers, 2) as '8-Trips',
  round(sum(case when rpct.trip_count = '9-Trips' then rpct.repeat_count else 0 end) * 100 / trp.total_repeat_passengers, 2) as '9-Trips',
  round(sum(case when rpct.trip_count = '10-Trips' then rpct.repeat_count else 0 end) * 100 / trp.total_repeat_passengers, 2) as '10-Trips'
from rp_by_city_and_tripcount rpct
join total_rp_by_city trp
on rpct.city_name = trp.city_name
group by rpct.city_name
order by rpct.city_name;
```

City Name	2-Trips	3-Trips	4-Trips	5-Trips	6-Trips	7-Trips	8-Trips	9-Trips	10-Trips
Chandigarh	32.31%	19.25%	15.74%	12.21%	7.42%	5.48%	3.47%	2.33%	1.79%
Coimbatore	11.21%	14.82%	15.56%	20.62%	17.64%	10.47%	6.15%	2.31%	1.22%
Indore	34.34%	22.69%	13.40%	10.34%	6.85%	5.24%	3.26%	2.38%	1.51%
Jaipur	50.14%	20.73%	12.12%	6.29%	4.13%	2.52%	1.90%	1.20%	0.97%
Kochi	47.67%	24.35%	11.81%	6.48%	3.91%	2.11%	1.65%	1.21%	0.81%
Lucknow	9.66%	14.77%	16.20%	18.42%	20.18%	11.33%	6.43%	1.91%	1.10%
Mysore	48.75%	24.44%	12.73%	5.82%	4.06%	1.76%	1.42%	0.54%	0.47%
Surat	9.76%	14.26%	16.55%	19.75%	18.45%	11.89%	6.24%	1.74%	1.35%
Vadodara	9.87%	14.17%	16.52%	18.06%	19.08%	12.86%	5.78%	2.05%	1.61%
Visakhapatnam	51.25%	24.96%	9.98%	5.44%	3.19%	1.98%	1.39%	0.88%	0.92%

(SQL Query & Power BI Visualization)

Insight - Across most cities, 2-trip passengers make up the largest share of repeat users, especially in Visakhapatnam (51.25%) and Jaipur (50.14%). In contrast, cities like Lucknow, Surat, and Vadodara have a more even distribution across higher trip counts, indicating a relatively stronger base of frequent repeat users.

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")

```
use gc_trips;

> with new_passengers as(
    select ps.city_id, dc.city_name,
    sum(ps.new_passengers) as total_new_passengers
    from fact_passenger_summary ps
    join dim_city dc
    on ps.city_id = dc.city_id
    group by ps.city_id, dc.city_name),

> city_ranks as(
    select city_name, total_new_passengers,
    dense_rank() over (order by total_new_passengers desc) as top_rank,
    dense_rank() over (order by total_new_passengers asc) as bottom_rank
    from new_passengers),

> top_bottom_cities as(
    select city_name, total_new_passengers,
    'Top 3' as city_category
    from city_ranks
    where top_rank <= 3
    union all
    select city_name, total_new_passengers,
    'Bottom 3' as city_category
    from city_ranks
    where bottom_rank <= 3)

select * from top_bottom_cities
order by city_category, total_new_passengers desc;
```

City Name	Total New Passengers	City Category
Jaipur	45856	Top 3
Kochi	26416	Top 3
Chandigarh	18908	Top 3
Lucknow	16260	
Indore	14863	
Visakhapatnam	12747	
Mysore	11681	
Surat	11626	Bottom 3
Vadodara	10127	Bottom 3
Coimbatore	8514	Bottom 3

(SQL Query & Power BI Visualization)

Insight - Jaipur brought in the highest number of new passengers, followed by Kochi and Chandigarh. On the flip side, Surat, Vadodara, and Coimbatore show the lowest new passenger counts, showing a clear gap in new user growth across cities.

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 (%)

```
use gc_trips;
> with monthly_revenue as(
    select dc.city_name, dd.month_name, dd.start_of_month,
    sum(ft.fare_amount) as revenue
    from fact_trips ft
    join dim_city dc on ft.city_id = dc.city_id
    join dim_date dd on ft.date = dd.date
    group by dc.city_name, dd.month_name, dd.start_of_month),

> total_revenue_by_city as(
    select city_name,
    sum(revenue) as total_revenue
    from monthly_revenue
    group by city_name),

> ranked_revenue as(
    select mr.city_name, mr.month_name, mr.revenue, trc.total_revenue,
    rank() over(partition by mr.city_name order by mr.revenue desc) as revenue_rank
    from monthly_revenue mr
    join total_revenue_by_city trc
    on mr.city_name = trc.city_name)

select city_name, month_name as highest_revenue_month, revenue,
round((revenue / total_revenue) * 100, 2) as pct_contribution
from ranked_revenue
where revenue_rank = 1
order by city_name;
```

City Name	Max Revenue Month	Revenue	% Contribution
Chandigarh	February	₹ 21,08,290	19.07%
Coimbatore	April	₹ 6,12,431	17.38%
Indore	May	₹ 13,80,996	18.09%
Jaipur	February	₹ 77,47,202	20.82%
Kochi	May	₹ 33,33,746	19.61%
Lucknow	February	₹ 17,77,269	18.78%
Mysore	May	₹ 7,45,170	18.38%
Surat	April	₹ 11,54,909	17.96%
Vadodara	April	₹ 7,06,250	18.60%
Visakhapatnam	April	₹ 13,90,682	17.34%

(SQL Query & Power BI Visualization)

Insight - The highest monthly revenue came from Jaipur in February with ₹77.47 lakhs, contributing 20.82% to its total revenue. Kochi followed with ₹33.34 lakhs in May (19.61%), and Chandigarh with ₹21.08 lakhs in February (19.07%). These cities had the strongest monthly revenue spikes across all.

Business Request - 6: Repeat Passenger Rate Analysis

Generate a report that calculates the metric:

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.

This metric will provide insights into monthly repeat trends.

Fields : city_name month total_passengers repeat_passengers

monthly_repeat_passenger_rate (%): Repeat passenger rate at the city and month level

```
use gc_trips;
select dc.city_name, fps.month, fps.total_passengers, fps.repeat_passengers,
round(fps.repeat_passengers / fps.total_passengers * 100, 2) as monthly_repeat_passenger_rate
from fact_passenger_summary fps
join dim_city dc
on fps.city_id = dc.city_id
order by dc.city_name, fps.month;
```


City Name	Total Passengers	Repeat Passengers	Monthly Repeat Passenger Rate
<input checked="" type="checkbox"/> Chandigarh	23978	5070	21.14%
<input checked="" type="checkbox"/> Coimbatore	11065	2551	23.05%
<input checked="" type="checkbox"/> Indore	22079	7216	32.68%
<input checked="" type="checkbox"/> Jaipur	55538	9682	17.43%
<input checked="" type="checkbox"/> Kochi	34042	7626	22.40%
<input type="checkbox"/> Lucknow	25857	9597	37.12%
Jan 2024	4896	1431	29.23%
Feb 2024	5188	1659	31.98%
Mar 2024	4781	1622	33.93%
Apr 2024	3807	1496	39.30%
May 2024	3487	1662	47.66%
Jun 2024	3698	1727	46.70%
<input checked="" type="checkbox"/> Mysore	13158	1477	11.23%
<input type="checkbox"/> Surat	20264	8638	42.63%
Jan 2024	3616	1184	32.74%
Feb 2024	3567	1313	36.81%
Mar 2024	3440	1494	43.43%
Apr 2024	3394	1551	45.70%
May 2024	3217	1606	49.92%
Jun 2024	3030	1490	49.17%
<input checked="" type="checkbox"/> Vadodara	14473	4346	30.03%
<input checked="" type="checkbox"/> Visakhapatnam	17855	5108	28.61%

(SQL Query & Power BI Visualization)

Insight - Across most cities, April, May, and June stood out with the highest monthly repeat passenger rates. These months saw a noticeable rise in returning users, pointing to stronger engagement and retention during this period compared to the rest of the year.

Business Request - 7: Repeat Passenger Rate Analysis

Generate a report that calculates the metric:

City-wise Repeat Passenger Rate: Calculate the overall repeat passenger rate for each city, considering all passengers across months.

This metric will provide insights into overall repeat behaviour for each city.

Fields : city_name total_passengers repeat_passengers

city_repeat_passenger_rate (%): Overall repeat passenger rate for each city, aggregated across months

```
use gc_trips;

select dc.city_name, sum(fps.total_passengers) as total_passengers ,
sum(fps.repeat_passengers) as repeat_passengers,
round(sum(fps.repeat_passengers) / sum(fps.total_passengers) * 100, 2) as city_repeat_passenger_rate

from fact_passenger_summary fps
join dim_city dc
on fps.city_id = dc.city_id
group by dc.city_name
order by city_repeat_passenger_rate desc;
```

City Name	Total Passengers	Repeat Passengers	Repeat Passenger Rate
Chandigarh	<div><div></div></div> 23978	<div><div></div></div> 5070	21.14%
Coimbatore	<div><div></div></div> 11065	<div><div></div></div> 2551	23.05%
Indore	<div><div></div></div> 22079	<div><div></div></div> 7216	32.68%
Jaipur	<div><div></div></div> 55538	<div><div></div></div> 9682	17.43%
Kochi	<div><div></div></div> 34042	<div><div></div></div> 7626	22.40%
Lucknow	<div><div></div></div> 25857	<div><div></div></div> 9597	37.12%
Mysore	<div><div></div></div> 13158	<div><div></div></div> 1477	11.23%
Surat	<div><div></div></div> 20264	<div><div></div></div> 8638	42.63%
Vadodara	<div><div></div></div> 14473	<div><div></div></div> 4346	30.03%
Visakhapatnam	<div><div></div></div> 17855	<div><div></div></div> 5108	28.61%

(SQL Query & Power BI Visualization)

Insight - Surat, Lucknow, and Indore recorded the highest repeat passenger rates at 42.63%, 37.12%, and 32.68% respectively, showing strong retention in these cities. In contrast, Mysore, Jaipur, and Chandigarh had the lowest rates, with less than 22% of their total passengers being repeat users.