



SQL Project

Provide Insights to Chief of Operations in Transportation Domain

Domain: Transportation & Mobility

Function: Operations

Tools: SQL (Primary), Excel (Visualization)

Project Overview



Goodcabs, a cab service company operating in ten tier-2 cities in India, aims to improve passenger satisfaction and business growth by analyzing key performance metrics such as:

- **Trip Volume**
- **Passenger Satisfaction**
- **Repeat Passenger Rate**
- **Trip Distribution**
- **New vs. Repeat Passengers**

The Chief of Operations, **Bruce Haryali**, needs these insights **urgently**. However, since the analytics manager Tony is occupied, **I (Data Analyst)** am assigned to generate these insights using **SQL for data extraction & analysis** and **Excel for visualization**.

This project provides **actionable insights** to help Goodcabs make **data-driven decisions** for business improvement in 2024.

Data Understanding



- **Datasets Overview**

- trips_db → Contains trip details, passenger data, and repeat trip patterns. (Operational Data)
 - ~ dim_city → City details for location-based insights.
 - ~ dim_date → Time-based trip grouping (months, weekdays vs. weekends).
 - ~ fact_trips → Trip-level data (distance, fare, ratings).
 - ~ fact_passenger_summary → New vs. repeat passenger trends.
 - ~ dim_repeat_trip_distribution → Repeat trip frequency analysis.
- targets_db → Stores monthly performance targets for Goodcabs. (Performance Targets)
 - ~ city_target_passenger_rating → Target passenger ratings per city.
 - ~ monthly_target_new_passengers → Target new passenger growth per city.
 - ~ monthly_target_trips → Target trip volume per city and month.

- **Data Relationships**

- Trips and passengers are linked to cities and dates.
- Targets provide benchmarks for evaluating performance.

- **Considerations**

- Ensure accurate date formats.
- Validate passenger and trip aggregations.
- Compare actual performance with targets.

Key Metrics & Performance Indicators

426K

Total Trips

108M

Total Revenue

8.1M km

Total
Dist. Travelled

7.66

Avg. Passenger
Rating

7.83

Avg. Driver Rating

₹ 254.02

Avg. Trip Cost

₹ 13.28

Avg. km Cost

19.13 km

Avg. Trip Distance

Key Metrics & Performance Indicators

45 km

Max Trip Distance

5 km

Min Trip Distance

177K

New Trips

249K

Repeated Trips

238K

Total Passengers

177K

New Passengers

61K

Repeated Passengers

25.7%

Repeat Passenger
Rate

Driving Insights: Business Problem & SQL Analysis



Business Problem: Goodcabs aims to optimize its operations by understanding passenger trends, repeat usage, and revenue growth across tier-2 cities. The challenge is to analyze data effectively to make data-driven decisions on customer retention, revenue strategies, and operational efficiency.

SQL Analysis Highlights:

- **New vs. Repeat Passenger Trends:** Ratio of first-time riders to returning customers.
- **Repeat Passenger Rate (%):** Identifying customer loyalty trends.
- **Revenue Growth Analysis:** Monthly revenue trends to assess business expansion.
- **Target Achievement Metrics:** Performance tracking for trips, new passengers, and customer satisfaction.
- **City-Wise Insights:** Evaluating regional variations in trip patterns and customer behavior.

Visual Elements:

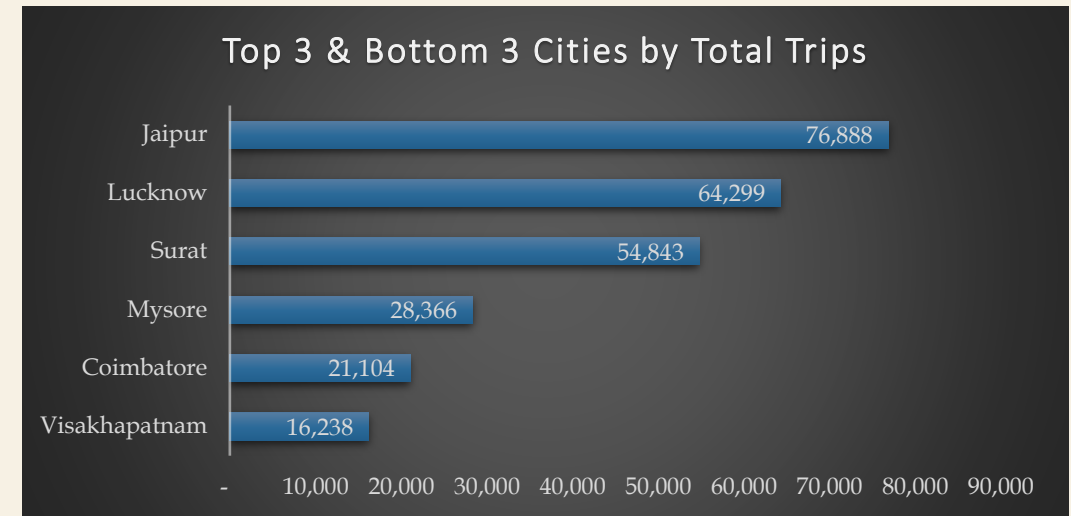
- **Bar Chart** – Revenue growth trends.
- **Line Graph** – Monthly repeat passenger rate.
- **Pie Chart** – Average passenger type.

Top and Bottom Performing Cities

- Identify the top 3 and bottom 3 cities by total trips over the entire analysis period.

Query:

```
(
    SELECT c.city_name, COUNT(f.trip_id) AS total_trips
    FROM fact_trips f
    JOIN dim_city c ON f.city_id = c.city_id
    GROUP BY c.city_name
    ORDER BY total_trips DESC
    LIMIT 3 -- Top 3 Cities
)
UNION ALL
(
    SELECT c.city_name, COUNT(f.trip_id) AS total_trips
    FROM fact_trips f
    JOIN dim_city c ON f.city_id = c.city_id
    GROUP BY c.city_name
    ORDER BY total_trips ASC
    LIMIT 3 -- Bottom 3 Cities
);
```



	city_name	total_trips
▸	Jaipur	76888
	Lucknow	64299
	Surat	54843
	Mysore	16238
	Coimbatore	21104
	Visakhapatnam	28366

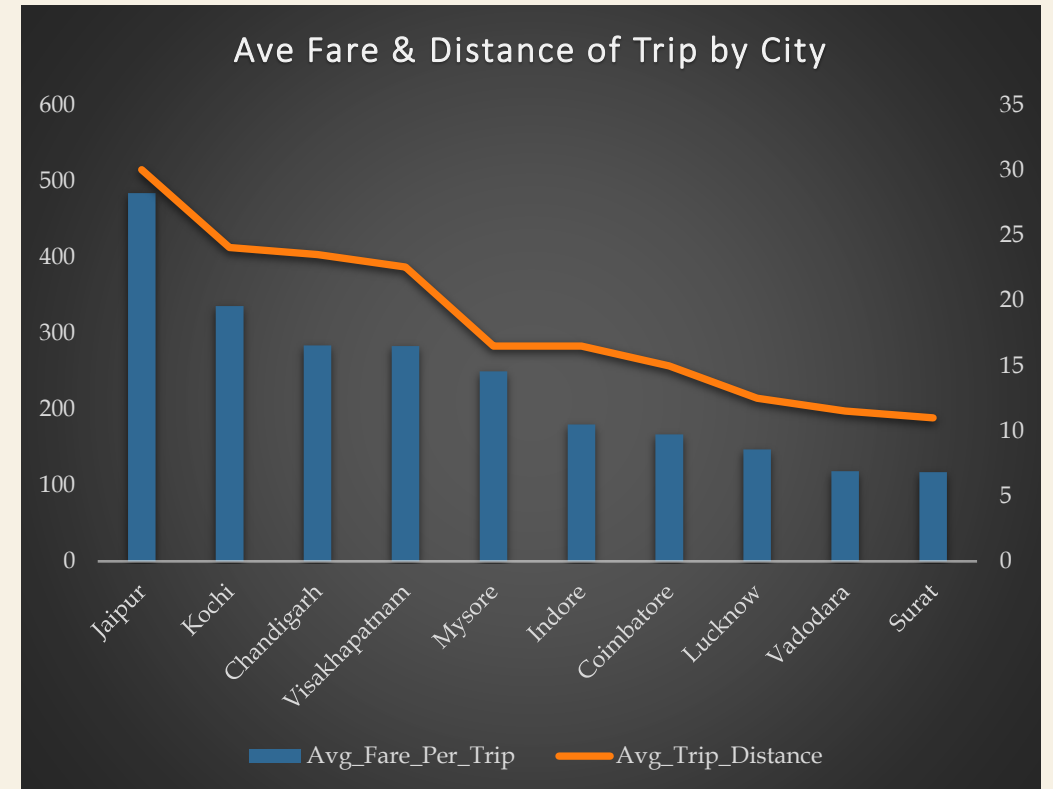
Average Fare Per Trip by City

- Calculate the Average fare per trip by each city and compare it with the city's average trip distance. Identify the cities with the highest and lowest average fare per trip to assess pricing efficiency across locations.

Query:

```
SELECT
    dc.city_name,
    AVG(ft.fare_amount) AS avg_fare_per_trip,
    AVG(ft.distance_travelled_km) AS avg_trip_distance
FROM fact_trips ft
JOIN dim_city dc ON ft.city_id = dc.city_id
GROUP BY dc.city_name
ORDER BY avg_fare_per_trip DESC;
```

city_name	avg_fare_per_trip	avg_trip_distance
Jaipur	483.9181	30.0231
Kochi	335.2451	24.0655
Chandigarh	283.6870	23.5187
Visakhapatnam	282.6723	22.5539
Mysore	249.7072	16.4969
Indore	179.8386	16.5025
Coimbatore	166.9822	14.9792
Lucknow	147.1804	12.5130
Vadodara	118.5662	11.5177
Surat	117.2729	10.9972



Average Ratings by City & Passenger Type

- (i) Calculate the Average passenger and driver ratings for each city, segmented by passenger type.

Query:

```
SELECT
    ft.city_id,
    dc.city_name,
    ft.passenger_type,
    ROUND(AVG(ft.passenger_rating), 2) AS avg_passenger_rating,
    ROUND(AVG(ft.driver_rating), 2) AS avg_driver_rating
FROM fact_trips ft
JOIN dim_city dc ON ft.city_id = dc.city_id
GROUP BY ft.city_id, dc.city_name, ft.passenger_type
ORDER BY avg_passenger_rating DESC;
```

city_id	city_name	passenger_type	avg_passenger_rating	avg_driver_rating
KL01	Kochi	repeated	8.00	8.99
AP01	Visakhap...	repeated	7.99	8.99
RJ01	Jaipur	repeated	7.99	8.98
KA01	Mysore	repeated	7.98	8.97
CH01	Chandig...	repeated	7.49	7.47
TN01	Coimbat...	repeated	7.48	7.48
MP01	Indore	repeated	7.47	7.48
GJ01	Surat	repeated	6.00	6.48
UP01	Lucknow	repeated	5.99	6.49
GJ02	Vadodara	repeated	5.98	6.48

city_id	city_name	passenger_type	avg_passenger_rating	avg_driver_rating
KL01	Kochi	new	8.99	8.99
RJ01	Jaipur	new	8.99	8.99
AP01	Visakhap...	new	8.98	8.98
KA01	Mysore	new	8.98	8.98
CH01	Chandig...	new	8.49	7.99
MP01	Indore	new	8.49	7.97
TN01	Coimbat...	new	8.49	7.99
GJ01	Surat	new	7.98	6.99
GJ02	Vadodara	new	7.98	7.00
UP01	Lucknow	new	7.98	6.99

Average Ratings by City & Passenger Type

- (ii) Identify cities with the highest and lowest average ratings.

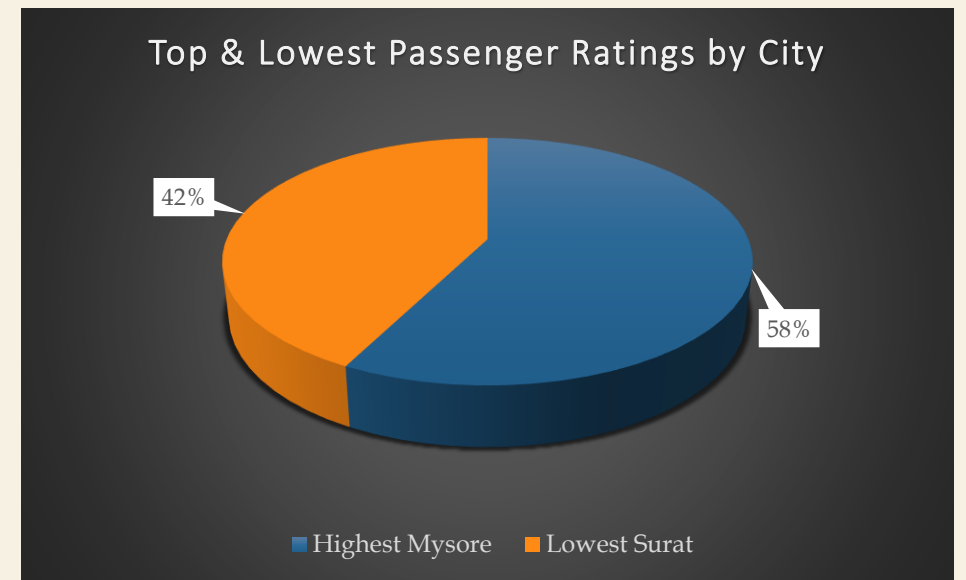
Query:

```
SELECT 'Highest' AS category, city_name, avg_passenger_rating
FROM (
  SELECT
    dc.city_name,
    ROUND(AVG(ft.passenger_rating), 2) AS avg_passenger_rating
  FROM fact_trips ft
  JOIN dim_city dc ON ft.city_id = dc.city_id
  GROUP BY dc.city_name
  ORDER BY avg_passenger_rating DESC
  LIMIT 1
) highest

UNION ALL

SELECT 'Lowest' AS category, city_name, avg_passenger_rating
FROM (
  SELECT
    dc.city_name,
    ROUND(AVG(ft.passenger_rating), 2) AS avg_passenger_rating
  FROM fact_trips ft
  JOIN dim_city dc ON ft.city_id = dc.city_id
  GROUP BY dc.city_name
  ORDER BY avg_passenger_rating ASC
  LIMIT 1
) lowest;
```

category	city_name	avg_passenger_rating
Highest	Mysore	8.70
Lowest	Surat	6.42



Peak & Low Demand Months by City

- For each city, Identify the month with the highest total trips (peak demand) and the month with the lowest total trips (low demand). This analysis will help Goodcabs understand seasonal patterns and adjust resources accordingly.

Query:

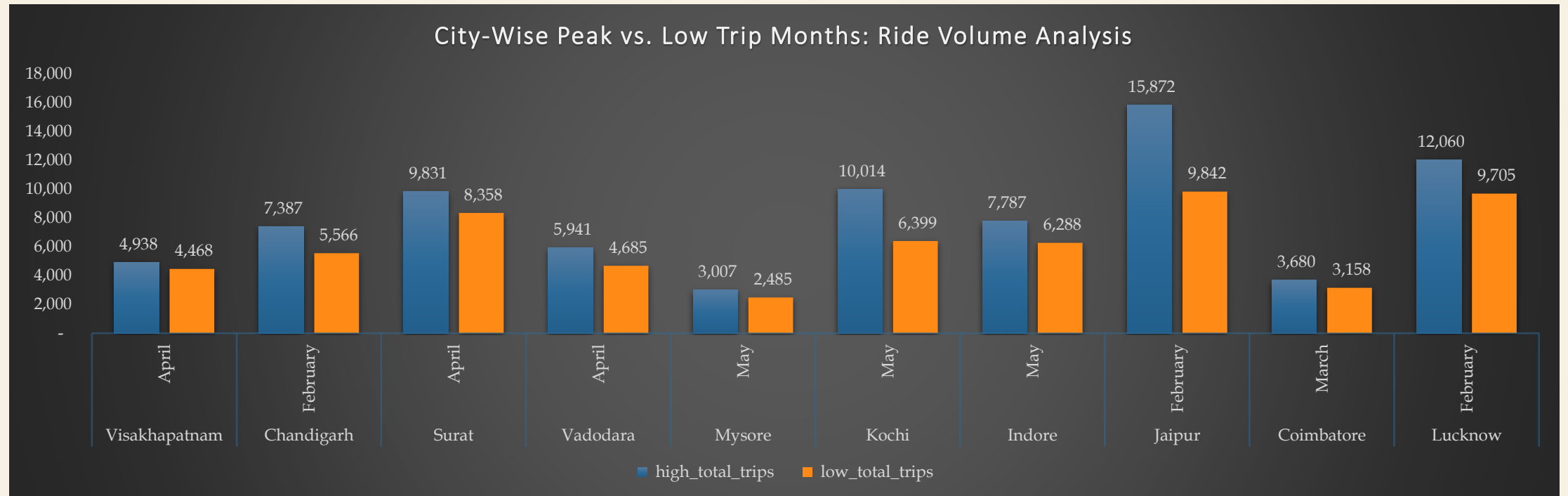
```
SELECT
    city_id,
    city_name,
    MAX(CASE WHEN peak_rank = 1 THEN month END) AS peak_month,
    MAX(CASE WHEN peak_rank = 1 THEN total_trips END) AS high_total_trips,
    MAX(CASE WHEN low_rank = 1 THEN total_trips END) AS low_total_trips
FROM (
    SELECT
        ft.city_id,
        dc.city_name,
        DATE_FORMAT(ft.date, '%M') AS month, -- Extracts only the month name
        COUNT(ft.trip_id) AS total_trips,
        RANK() OVER (PARTITION BY ft.city_id ORDER BY COUNT(ft.trip_id) DESC) AS peak_rank,
        RANK() OVER (PARTITION BY ft.city_id ORDER BY COUNT(ft.trip_id) ASC) AS low_rank
    FROM fact_trips ft
    JOIN dim_city dc ON ft.city_id = dc.city_id
    GROUP BY ft.city_id, dc.city_name, month
) ranked_data
WHERE peak_rank = 1 OR low_rank = 1
GROUP BY city_id, city_name
ORDER BY city_id;
```

city_id	city_name	peak_month	high_total_trips	low_total_trips
AP01	Visakhapatnam	April	4938	4468
CH01	Chandigarh	February	7387	5566
GJ01	Surat	April	9831	8358
GJ02	Vadodara	April	5941	4685
KA01	Mysore	May	3007	2485
KL01	Kochi	May	10014	6399
MP01	Indore	May	7787	6288
RJ01	Jaipur	February	15872	9842
TN01	Coimbatore	March	3680	3158
UP01	Lucknow	February	12060	9705

Peak & Low Demand Months by City

- For each city, Identify the month with the highest total trips (peak demand) and the month with the lowest total trips (low demand). This analysis will help Goodcabs understand seasonal patterns and adjust resources accordingly.

[Visualization Part]

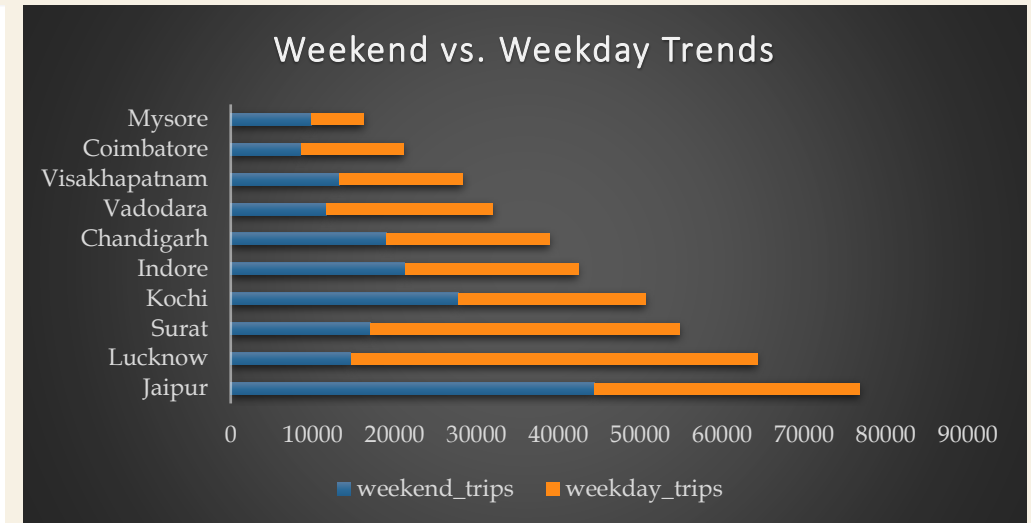


Weekend vs. Weekday Trip Demand by City

- Compare the total trips taken on weekdays versus weekends for each city over the six-month period. Identify cities with a strong preference for either weekend or weekday trips to understand demand variations.

Query:

```
SELECT
    dc.city_id,
    dc.city_name,
    SUM(CASE WHEN dd.day_type = 'Weekend' THEN 1 ELSE 0 END) AS weekend_trips,
    SUM(CASE WHEN dd.day_type = 'Weekday' THEN 1 ELSE 0 END) AS weekday_trips,
    CASE
        WHEN SUM(CASE WHEN dd.day_type = 'Weekend' THEN 1 ELSE 0 END)
            > SUM(CASE WHEN dd.day_type = 'Weekday' THEN 1 ELSE 0 END)
        THEN 'Weekend Preference'
        ELSE 'Weekday Preference'
    END AS preference
FROM fact_trips ft
JOIN dim_date dd ON ft.date = dd.date
JOIN dim_city dc ON ft.city_id = dc.city_id
GROUP BY dc.city_id, dc.city_name
ORDER BY (weekend_trips + weekday_trips) DESC;
```



city_id	city_name	weekend_trips	weekday_trips	preference
RJ01	Jaipur	44397	32491	Weekend Preference
UP01	Lucknow	14682	49617	Weekday Preference
GJ01	Surat	17050	37793	Weekday Preference
KL01	Kochi	27787	22915	Weekend Preference
MP01	Indore	21258	21198	Weekend Preference
CH01	Chandigarh	19067	19914	Weekday Preference
GJ02	Vadodara	11716	20310	Weekday Preference
AP01	Visakhapatnam	13266	15100	Weekday Preference
TN01	Coimbatore	8528	12576	Weekday Preference
KA01	Mysore	9814	6424	Weekend Preference

Repeat Passenger Frequency & City Contribution Analysis

- Analyse the frequency of trips taken by repeat passengers in each city. Identify which cities contribute most to higher trip frequencies among repeat passengers, and examine if there are distinguishable patterns between tourism-focused and business-focused cities.

Query:

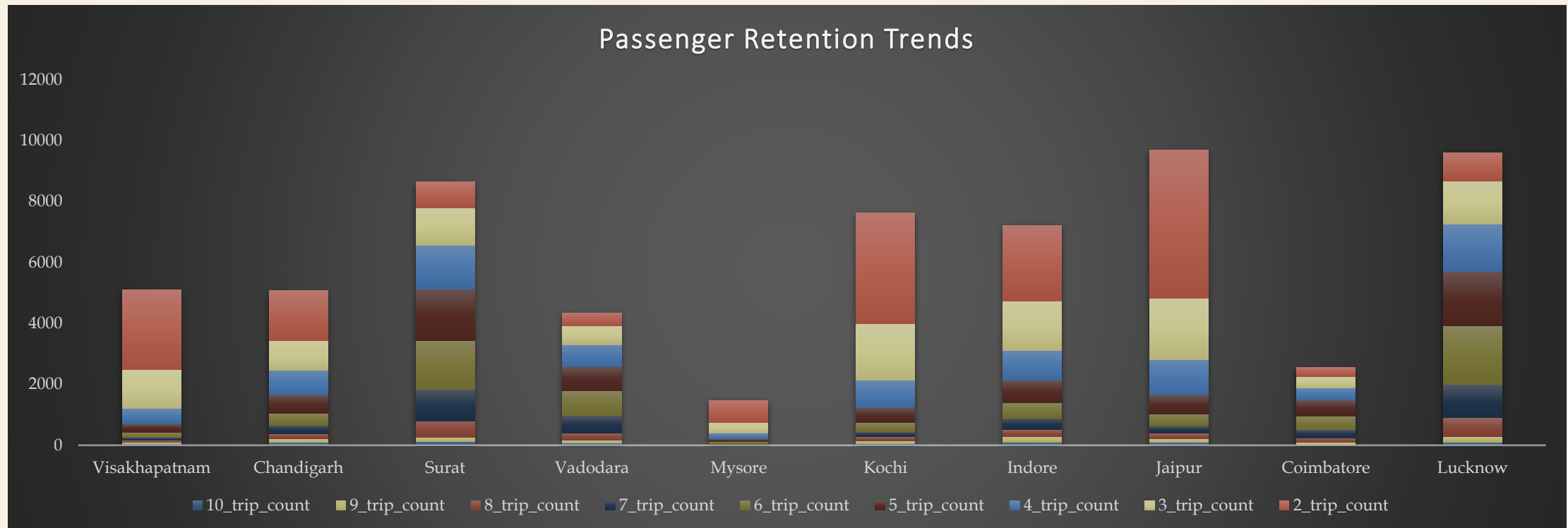
```
SELECT
    d.city_id,
    c.city_name,
    SUM(CASE WHEN d.trip_count = 10 THEN d.repeat_passenger_count ELSE 0 END) AS "10_trip_count",
    SUM(CASE WHEN d.trip_count = 9 THEN d.repeat_passenger_count ELSE 0 END) AS "9_trip_count",
    SUM(CASE WHEN d.trip_count = 8 THEN d.repeat_passenger_count ELSE 0 END) AS "8_trip_count",
    SUM(CASE WHEN d.trip_count = 7 THEN d.repeat_passenger_count ELSE 0 END) AS "7_trip_count",
    SUM(CASE WHEN d.trip_count = 6 THEN d.repeat_passenger_count ELSE 0 END) AS "6_trip_count",
    SUM(CASE WHEN d.trip_count = 5 THEN d.repeat_passenger_count ELSE 0 END) AS "5_trip_count",
    SUM(CASE WHEN d.trip_count = 4 THEN d.repeat_passenger_count ELSE 0 END) AS "4_trip_count",
    SUM(CASE WHEN d.trip_count = 3 THEN d.repeat_passenger_count ELSE 0 END) AS "3_trip_count",
    SUM(CASE WHEN d.trip_count = 2 THEN d.repeat_passenger_count ELSE 0 END) AS "2_trip_count",
    SUM(d.repeat_passenger_count) AS repeat_passenger_count,
    SUM(d.repeat_passenger_count) AS total_repeat_passengers,
    ROUND((SUM(d.repeat_passenger_count) * 100.0) / NULLIF(SUM(d.repeat_passenger_count), 0), 2) AS repeat_percentage
FROM
    dim_repeat_trip_distribution d
JOIN
    dim_city c ON d.city_id = c.city_id
GROUP BY
    d.city_id, c.city_name
ORDER BY
    d.city_id;
```

city_id	city_name	10_trip	9_trip	8_trip	7_trip	6_trip	5_trip	4_trip	3_trip	2_trip	repeat_p	total_rep	repeat_p
AP01	Visakhap...	47	45	71	101	163	278	510	1275	2618	5108	5108	100.00
CH01	Chandig...	91	118	176	278	376	619	798	976	1638	5070	5070	100.00
GJ01	Surat	117	150	539	1027	1594	1706	1430	1232	843	8638	8638	100.00
GJ02	Vadodara	70	89	251	559	829	785	718	616	429	4346	4346	100.00
KA01	Mysore	7	8	21	26	60	86	188	361	720	1477	1477	100.00
KL01	Kochi	62	92	126	161	298	494	901	1857	3635	7626	7626	100.00
MP01	Indore	109	172	235	378	494	746	967	1637	2478	7216	7216	100.00
RJ01	Jaipur	94	116	184	244	400	609	1173	2007	4855	9682	9682	100.00
TN01	Coimbat...	31	59	157	267	450	526	397	378	286	2551	2551	100.00
UP01	Lucknow	106	183	617	1087	1937	1768	1555	1417	927	9597	9597	100.00

Repeat Passenger Frequency & City Contribution Analysis

- Analyse the frequency of trips taken by repeat passengers in each city. Identify which cities contribute most to higher trip frequencies among repeat passengers, and examine if there are distinguishable patterns between tourism-focused and business-focused cities.

[Visualization Part]



Key Findings

(Insights from SQL Analysis & Visualizations)

1. Passenger Ratings Across Cities:

- **Mysore** has the highest average passenger rating (8.70), indicating a superior customer experience.
- **Surat** has the lowest rating (6.42), suggesting potential service or infrastructure issues.

2. Peak Travel Trends Across Cities:

- **Jaipur** (15,872), **Lucknow** (12,060), and **Kochi** (10,014) have the highest trips in their peak months.
- **May** is a peak travel month for most cities, indicating seasonal trends, possibly due to summer vacations.
- **Chandigarh** and **Jaipur** peak in **February**, suggesting tourism or local events driving demand.

3. Weekend vs. Weekday Travel Patterns:

- Tourist-friendly cities like **Jaipur**, **Kochi**, and **Mysore** see higher weekend trips, suggesting leisure travel dominance.
- Business hubs like **Surat**, **Lucknow**, and **Vadodara** show higher weekday trips, indicating strong work-related commutes.

4. Trip Frequency Analysis:

- **Jaipur** and **Lucknow** have the highest repeat travelers, showing strong customer retention.
- Higher trip frequency (8+ trips) is more common in business hubs like **Surat** and **Lucknow**.
- **Mysore** and **Coimbatore** have fewer repeat travelers, indicating potential service issues or alternative travel options.

Business Recommendations

(Based on Insights)

1. Improve Customer Experience in Low-Rating Cities (Surat & Others):

- Conduct customer feedback surveys and address key concerns.
- Improve driver training and vehicle quality in Surat.
- Offer promotional discounts to boost positive reviews.

2. Leverage Seasonality Trends for Targeted Marketing:

- Increase marketing efforts in May (peak season for most cities) to maximize bookings.
- Run special discounts in February for Jaipur and Chandigarh due to increased demand.

3. Enhance Weekend Travel Experience in Tourist Cities:

- Partner with hotels and travel agencies in Jaipur, Kochi, and Mysore for bundled offers.
- Increase vehicle availability on weekends in these cities to meet demand.

4. Optimize Fleet Management for Business Cities:

- Deploy more vehicles in Surat, Lucknow, and Vadodara on weekdays to accommodate work commuters.
- Introduce loyalty programs for frequent travelers in these cities.

5. Expand Repeat Traveler Engagement Programs:

- Jaipur and Lucknow have strong repeat travelers – introduce loyalty rewards and referral programs.
- For Mysore and Coimbatore, analyze reasons for lower repeat trips and improve service offerings.



Thank You!
