UBER DATA ANALYSIS (SQL)

Business problems to sovle:

This project tackled several real-world business challenges **Uber faces daily. Through SQL analysis, we explored** operational inefficiencies, customer behavior, and optimization opportunities**.**

1. **Peak Travel Patterns & Seasonality**

- **What are the top 3 busiest travel months based on total miles traveled?**

```sql
select month, ROUND(sum(miles),2) as total_miles_covered_per_month
from uberdatasetcleaned
GROUP BY month
ORDER BY total_miles_covered_per_month DESC
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

month	total_miles_covered_per_month
Oct	2878.63
Mar	2402.58
Aug	2149.27
July	1924.93
Dec	1579.25
April	1559.45
Feb	1457.26
Nov	1314.67
June	1276.69
Jan	825.43
May	564.72
Sep	539.93

- **Which hour of the day has the longest average trip distance?**

```sql
WITH CTE AS (SELECT day,avg(miles) as miles_traveled, start_hour,
rank() over(partition by day order by avg(miles) desc) as ranking
FROM
uberdatasetcleaned
group by day,start_hour)

SELECT * FROM CTE
WHERE ranking = 1
ORDER BY miles_traveled desc
```

day	miles_traveled	start_hour	ranking
Sunday	133.011951	2	1
Monday	75.47804599999999	7	1
Tuesday	70.328158	3	1
Friday	54.409103166666675	16	1
Saturday	50.658446888888896	16	1
Thursday	40.394434000000004	12	1
Wednesday	26.12495266666667	20	1

- **Is there a seasonal trend in business trips vs. personal trips? (Compare ride volume across months)**

```sql
SELECT  month,PURPOSE,ROUND(sum(miles),2) as Total_miles
FROM
uberdatasetcleaned
GROUP BY month,PURPOSE
HAVING purpose = 'meeting'
ORDER BY total_miles desc
```

month	PURPOSE	Total_miles
Oct	Meeting	1693.67
Dec	Meeting	645.02
April	Meeting	619.11
June	Meeting	602.7
Feb	Meeting	470.57
Mar	Meeting	318.65
July	Meeting	299.18
Nov	Meeting	284.85
Jan	Meeting	279.38
May	Meeting	184.27
Aug	Meeting	153.85

```sql
SELECT  month,PURPOSE,ROUND(sum(miles),2) as Total_miles
FROM
uberdatasetcleaned
GROUP BY month,PURPOSE
HAVING purpose ='Meal/Entertain'
ORDER BY total_miles desc

```

month	PURPOSE	Total_miles
Mar	Meal/Entertain	577.27
June	Meal/Entertain	355.66
July	Meal/Entertain	246.07
Dec	Meal/Entertain	239.31
Feb	Meal/Entertain	195.53
April	Meal/Entertain	139.21
May	Meal/Entertain	121.83
Oct	Meal/Entertain	117.64
Jan	Meal/Entertain	116.36
Aug	Meal/Entertain	84.65
Nov	Meal/Entertain	83.04

2. **Driver Efficiency & Utilization**

- **What percentage of trips start and end at the same location? (Potentially canceled or looped trips)**

```sql
WITH CTE AS (
    SELECT COUNT(*) AS total_rides
    FROM uberdatasetcleaned
)
SELECT
    u.start,
    u.stop,
    (COUNT(*) / NULLIF(c.total_rides, 0)) * 100 AS percentage_cancelled_rides
FROM uberdatasetcleaned u
CROSS JOIN CTE c  -- Ensures total_rides is available for all rows
WHERE u.start = u.stop  -- Filtering cancelled rides
GROUP BY u.start, u.stop, c.total_rides
ORDER BY percentage_cancelled_rides DESC;
```

start	stop	percentage_cancelled_rides
Unknown Location	Unknown Location	7.2999
Cary	Cary	4.6614
Lahore	Lahore	2.3747
Islamabad	Islamabad	2.1108
Kar?chi	Kar?chi	1.7590
Whitebridge	Whitebridge	0.6157
Colombo	Colombo	0.5277
Morrisville	Morrisville	0.4398
Covington	Covington	0.4398
Banner Elk	Banner Elk	0.4398
Fort Pierce	Fort Pierce	0.3518
Kissimmee	Kissimmee	0.3518
Houston	Houston	0.2639
Bryson City	Bryson City	0.2639
Central	Central	0.2639
Raleigh	Raleigh	0.1759
R?walpindi	R?walpindi	0.1759
Weston	Weston	0.1759
Midtown	Midtown	0.1759
Pontchartrain Sho...	Pontchartrain Sho...	0.1759
Kenner	Kenner	0.1759
Lake Reams	Lake Reams	0.1759
Port Bolivar	Port Bolivar	0.1759
Apex	Apex	0.1759
West Palm Beach	West Palm Beach	0.0880
Katunayaka	Katunayaka	0.0880
Orlando	Orlando	0.0880
Mandeville	Mandeville	0.0880

- **Which trip purpose has the highest average distance, and how does that compare to the overall average?**

```sql
WITH CTE AS (
SELECT avg(miles) as average_total_miles
FROM uberdatasetcleaned
)

SELECT u.purpose , ROUND(avg(u.miles),2) as average_miles ,ROUND(c.average_total_miles,2) as Total_average_distance_travelled
FROM
uberdatasetcleaned u
cross join CTE c
GROUP BY purpose,c.average_total_miles
ORDER BY average_miles desc
```

purpose	average_miles	Total_average_distance_travelled
Commute	290	16.25
Customer Visit	25.86	16.25
Charity ($)	24.3	16.25
Airport/Travel	24.07	16.25
Meeting	18.82	16.25
Between Offices	17.61	16.25
Temporary Site	15.85	16.25
Errand/Supplies	9.54	16.25
Meal/Entertain	9.29	16.25
Moving	7.32	16.25

- **How often do people take back-to-back trips (i.e., trips where the stop location of one trip matches the start location of the next trip within 30 minutes)?**

```sql
WITH CTE AS (SELECT
start_hour , start_minute ,
start , stop ,
lead(start_hour) over(partition by start order by start_hour , start_minute) as next_hour,
lead(start_minute) over(partition by start order by start_hour , start_minute) as next_minute,
lead(start) over(partition by start order by start_hour , start_minute)  as next_stop
from
uberdatasetcleaned)

SELECT start as start_location ,
next_stop as stop_location,
count(*) as number_back_to_back_trips
FROM CTE
WHERE start = next_stop
AND ((next_hour*60+next_minute)-(start_hour*60+start_minute))<30
GROUP BY start , next_stop
ORDER BY count(*) DESC;
```

start_location	stop_location	number_back_to_back_trips
Cary	Cary	195
Unknown Location	Unknown Location	139
Morrisville	Morrisville	75
Whitebridge	Whitebridge	60
Islamabad	Islamabad	50
Lahore	Lahore	21
Durham	Durham	19
Raleigh	Raleigh	18
Kar?chi	Kar?chi	13
Westpark Place	Westpark Place	10
Apex	Apex	9
Berkeley	Berkeley	7
Edgehill Farms	Edgehill Farms	4
New Orleans	New Orleans	4
Kissimmee	Kissimmee	4
Midtown	Midtown	3
Covington	Covington	3
The Drag	The Drag	3
Hazelwood	Hazelwood	3
Kenner	Kenner	3
Banner Elk	Banner Elk	2
Tanglewood	Tanglewood	2
Orlando	Orlando	2
Parkway	Parkway	2
Preston	Preston	2
Savon Height	Savon Height	2
South Congress	South Congress	2
Jamaica	Jamaica	1

3. Location Intelligence & Route Optimization

- Which start-stop location pairs have the highest frequency, and how can Uber optimize pricing for them?

```sql
WITH CTE AS (SELECT start,stop ,miles,
row_number() OVER(partition by start ,stop order by miles DESC) as max_,
row_number() OVER(partition by start ,stop order by miles asc) as min_
FROM
uberdatasetcleaned
WHERE start = stop)

SELECT start ,stop , ROUND(miles,2) as total_miles
FROM CTE
WHERE max_= 1 OR min_ = 1
ORDER BY total_miles DESC
```

start	stop	total_miles
Unknown Location	Unknown Location	252.51
Cary	Cary	63.09
Islamabad	Islamabad	60.67
R?walpindi	R?walpindi	37.18
Houston	Houston	35.24
Bryson City	Bryson City	26.23
Banner Elk	Banner Elk	20.92
Huntington Woods	Huntington Woods	19.47
Morrisville	Morrisville	18.99
Kissimmee	Kissimmee	17.7
Lahore	Lahore	15.77
Covington	Covington	13.84
Kar?chi	Kar?chi	11.59
Orlando	Orlando	11.1
Bryson City	Bryson City	10.14
Colombo	Colombo	9.66
Fort Pierce	Fort Pierce	8.21
Raleigh	Raleigh	7.89
Houston	Houston	7.89
Pontchartrain Sho...	Pontchartrain Sho...	7.72
Fort Pierce	Fort Pierce	7.56
West Palm Beach	West Palm Beach	6.92
Whitebridge	Whitebridge	6.76
Weston	Weston	6.76
R?walpindi	R?walpindi	6.6
Newland	Newland	6.12

1. **Surge Pricing**:
   - Identify the busiest routes and apply **dynamic pricing** to maximize revenue during peak hours.

2. **Route-Based Discounts**:
   - If some high-frequency routes have strong **competition from public transport**, Uber can **offer targeted discounts** to retain customers.
3. **Incentivizing Drivers**:
   - Increase driver availability in **high-demand start locations** to reduce wait times and improve service.
4. **Ridesharing Optimization**:
   - If many users take the same start-stop pair, Uber can **promote shared rides (Uber Pool)** for efficiency.

◆ Are there specific routes where trip distances vary significantly? (Compare the shortest and longest distances for the same start-stop locations)

```sql
WITH CTE AS (SELECT start,stop ,miles,
row_number() OVER(partition by start ,stop order by miles DESC) as max_,
row_number() OVER(partition by start ,stop order by miles asc) as min_
FROM
uberdatasetcleaned
WHERE start = stop)

SELECT start ,stop , ROUND(miles,2) as total_miles
FROM CTE
WHERE max_= 1 OR min_ = 1
ORDER BY total_miles DESC
```

start	stop	total_miles
Raleigh	Raleigh	4.35
Central	Central	4.18
Morrisville	Morrisville	3.54
Kenner	Kenner	3.54
Agnew	Agnew	3.54
Port Bolivar	Port Bolivar	3.38
Lake Reams	Lake Reams	3.38
Chessington	Chessington	3.06
Covington	Covington	2.9
Colombo	Colombo	2.74
Pontchartrain Sho...	Pontchartrain Sho...	2.74
Kenner	Kenner	2.25
Port Bolivar	Port Bolivar	1.93
Parkwood	Parkwood	1.93
Lake Reams	Lake Reams	1.93
Central	Central	1.77
Midtown	Midtown	1.77
Apex	Apex	1.61
Jamestown Court	Jamestown Court	1.61
Islamabad	Islamabad	1.61
Sharpstown	Sharpstown	1.61
Unknown Location	Unknown Location	1.13
Kissimmee	Kissimmee	1.13
Lahore	Lahore	0.97
Kar?chi	Kar?chi	0.97
Whitebridge	Whitebridge	0.97
Cary	Cary	0.8
Katunayaka	Katunayaka	0.8

4. **Anomaly Detection & Fraud Analysis**

- **Identify duplicate or suspicious trips (e.g., same start-stop, same time, but different miles recorded).**

```sql
WITH CTE AS (SELECT start , stop , start_hour , end_hour , ROUND(miles,1) as  total_miles,
lead(miles) over(partition by start , stop order by miles desc) as next_miles ,
row_number() over(partition by start ,stop order by miles desc) as diff_miles
FROM uberdatasetcleaned
WHERE start = stop and start_hour = end_hour )

SELECT start,stop,total_miles
FROM CTE
WHERE total_miles != next_miles and start !='Unknown Location' and stop!='Unknown Location'
```

start	stop	total_miles
Whitebrid...	Whitebridge	6.8
Whitebrid...	Whitebridge	2.6
Whitebrid...	Whitebridge	2.3
Whitebrid...	Whitebridge	1.9
Whitebrid...	Whitebridge	1.6
Morrisville	Morrisville	19
Morrisville	Morrisville	10
Morrisville	Morrisville	9.5
Morrisville	Morrisville	8
Midtown	Midtown	1.8
Lake Reams	Lake Reams	3.4
Lahore	Lahore	12.4
Lahore	Lahore	11.9
Lahore	Lahore	10
Lahore	Lahore	8.5
Lahore	Lahore	6.1
Lahore	Lahore	6.1
Lahore	Lahore	5.8
Lahore	Lahore	5.5
Lahore	Lahore	5.1
Lahore	Lahore	5
Lahore	Lahore	4.8
Lahore	Lahore	4.7
Lahore	Lahore	4.2
Lahore	Lahore	3.7
Lahore	Lahore	3.4
Lahore	Lahore	3.4

CONCLUSION

Key Insights:

1. **High-Frequency Start-Stop Pairs:**
   - Certain routes experience significantly higher demand, suggesting potential pricing adjustments or promotional strategies to maximize revenue.
2. **Back-to-Back Trips Analysis:**
   - A noticeable number of users take **back-to-back trips within 30 minutes**, indicating possible inefficiencies in trip completion or ride-sharing opportunities.
3. **Unusually High Trip Frequency Users:**
   - Some users take more than **3 trips within an hour**, hinting at possible fraudulent activity or high user engagement requiring targeted services.
4. **Trip Cancellation Analysis:**
   - Locations with a **high percentage of cancellations** can indicate issues like **availability gaps, pricing concerns, or poor driver allocation**, helping Uber optimize its service distribution.
5. **Distance-Based Analysis:**
   - We identified both **shortest and longest trips per location pair**, which can be useful for understanding trip pricing and potential route optimizations.

Business Recommendations:

1. **Dynamic Pricing Optimization** → Adjust pricing based on **high-demand routes** and **frequent short-duration trips**.
2. **Fraud Detection & Prevention** → Investigate users with an **unusual number of trips within a short period** to detect anomalies.
3. **Reduce Cancellations** → Improve driver allocation and incentives to **minimize trip cancellations in high-cancellation areas**.
4. **Encourage Ride Pooling** → Promote **ride-sharing options** for users frequently taking back-to-back trips from the same location.