Uber Supply Demand Analysis using SQLite

- More people ask for Ubers in the Evening and Morning than at other times.
- These are the busiest times for Uber, like rush hour for rides.
- In the Morning, lots of people can't get a car ("No Cars Available"). In the Evening, many people cancel their ride after asking for one.
- The problem in the Morning is not enough drivers. The problem in the Evening might be long waits or drivers canceling.
- Over half the time, people at the Airport can't get an Uber. In the City, it's less than a third of the time.
- There's a big shortage of Uber cars specifically at the Airport.
- The most cancellations happen in the Morning (40%).
- A lot of people booking Ubers in the morning end up canceling, probably because they're waiting too long or find another ride.
- Only about a quarter of requested trips from the Airport in the Evening actually get completed. This is the lowest rate.
- The Airport during the Evening is the worst time and place for Uber to successfully complete a ride request.
- The hours around 2 AM and then several hours in the late Evening (7 PM - 10 PM) have the fewest successful trips.
- These specific hours are when Uber struggles the most to get people rides, showing a clear mismatch between when people want rides and when drivers are available.
- The number of people asking for rides jumps significantly during the usual morning and evening commute hours.
- This shows the predictable rush hour pattern for Uber demand throughout the day.
- Even though lots of people want rides in the Morning and Evening, fewer rides are completed than requested, especially in the Morning.
- Uber is missing opportunities to complete rides during the busiest times because there aren't enough drivers to handle all the requests.

Upload and Load Data

```
# Step 1: Upload cleaned dataset
from google.colab import files
uploaded = files.upload()
```

Choose Files uber_data_sql.csv

- **uber_data_sql.csv**(text/csv) - 725617 bytes, last modified: 6/22/2025 - 100% done

```
Saving uber_data_sql.csv to uber_data_sql.csv
```

```
# Step 2: Load CSV into pandas
import pandas as pd

df = pd.read_csv('uber_data_sql.csv')
df.head()
```

	Request_id	Pickup_point	Driver_id	Status	Request_DateTime	Drop_DateTime	Request_Date	Request_Time	Drop_Date	Drop_Time	Request_Hour
0	1	Airport	285.0	Trip Completed	11/7/2016 0:20	11/7/2016 0:51	11/7/2016	12:20:00 AM	11/7/2016	12:51:00 AM	0
1	2	Airport	NaN	No Cars Available	11/7/2016 0:23	NaN	11/7/2016	12:23:00 AM	NaN	NaN	0
2	3	Airport	80.0	Trip Completed	11/7/2016 0:24	11/7/2016 1:31	11/7/2016	12:24:00 AM	11/7/2016	1:31:00 AM	0
3	4	City	NaN	No Cars Available	11/7/2016 0:37	NaN	11/7/2016	12:37:00 AM	NaN	NaN	0
4	5	Airport	264.0	Trip Completed	11/7/2016 0:36	11/7/2016 1:35	11/7/2016	12:36:00 AM	11/7/2016	1:35:00 AM	0

Next steps: Generate code with df View recommended plots New interactive sheet

Set Up SQLite

```
# Step 3: Set up in-memory SQLite database
import sqlite3

conn = sqlite3.connect(':memory:')  # Creates a temporary DB in RAM
df.to_sql('uber_requests', conn, index=False, if_exists='replace')
```

```
6745
```

SQL Queries with Insights

1. Total Requests by Time Slot

```
query = '''
        SELECT Time_Slot, COUNT(*) AS total_requests
        FROM uber_requests
        GROUP BY Time_Slot
        ORDER BY total_requests DESC;
        '''
pd.read_sql(query, conn)
```

	Time_Slot	total_requests
0	Evening	2840
1	Morning	2103
2	Day	1224
3	Late Night	578

It shows that demand for Uber services is not uniform throughout the day. The "Evening" time slot (likely covering the evening commute and post-work hours) and the "Morning" time slot (representing the morning commute) are the periods with the highest volume of ride requests. This highlights the critical importance of ensuring adequate driver supply during these peak hours to meet customer needs.

2. Status Breakdown per Time Slot

```
query = '''
        SELECT Time_Slot, Status, COUNT(*) AS count
        FROM uber_requests
        GROUP BY Time_Slot, Status
        ORDER BY Time_Slot;
        '''
pd.read_sql(query, conn)
```

	Time_Slot	Status	count
0	Day	Cancelled	168
1	Day	No Cars Available	334
2	Day	Trip Completed	722
3	Evening	Cancelled	188
4	Evening	No Cars Available	1611
5	Evening	Trip Completed	1041
6	Late Night	Cancelled	65
7	Late Night	No Cars Available	299
8	Late Night	Trip Completed	214
9	Morning	Cancelled	843
10	Morning	No Cars Available	406
11	Morning	Trip Completed	854

3. % of “No Cars Available” by Pickup Point

```
query = '''
        SELECT Pickup_point,
        ROUND(SUM(CASE WHEN status = 'No Cars Available' THEN 1 ELSE 0 END)*100.0/COUNT(*), 2) AS no_car_percent
        FROM uber_requests
        GROUP BY Pickup_point;
        '''
pd.read_sql(query, conn)
```

	Pickup_point	no_car_percent
0	Airport	52.90
1	City	26.72

Airport has a significantly higher no-car percentage than the city. There is a clear supply shortage issue at the Airport.

4. % of Cancellations by Time Slot

```
query = '''
        SELECT Time_Slot,
        ROUND(SUM(CASE WHEN status = 'Cancelled' THEN 1 ELSE 0 END)*100.0/COUNT(*), 2) AS cancel_percent
        FROM uber_requests
        GROUP BY Time_Slot;
        '''
pd.read_sql(query, conn)
```

	Time_Slot	cancel_percent
0	Day	13.73
1	Evening	6.62
2	Late Night	11.25
3	Morning	40.09

The Morning time slot exhibits the highest cancellation percentage (40.09%). This could indicate issues with driver availability during peak morning hours or passengers canceling due to long wait times. The Day slot also shows a notable cancellation rate (13.73%), which could be influenced by factors like increased traffic or driver fatigue.

5. Trip Completion Rate by Pickup Point and Time Slot

```
query = '''
        SELECT Pickup_point, Time_Slot,
        ROUND(SUM(CASE WHEN status = 'Trip Completed' THEN 1 ELSE 0 END)*100.0/COUNT(*), 2) AS trip_completion_percent
        FROM uber_requests
        GROUP BY Pickup_point, Time_Slot
        ORDER BY trip_completion_percent ASC;
        '''
pd.read_sql(query, conn)
```

	Pickup_point	Time_Slot	trip_completion_percent
0	Airport	Evening	24.75
1	City	Morning	28.15
2	City	Late Night	34.15
3	Airport	Late Night	40.71
4	City	Day	52.95
5	Airport	Day	68.41
6	City	Evening	69.30
7	Airport	Morning	89.67

Airport - Evening combination has lowest trip completion. This should be focus on to put efforts here to reduce customer churn.

6. Top 5 Worst Performing Hours (Lowest Completion Rates)

```
query = '''
        SELECT Request_Hour+1 AS hour_of_day,
        COUNT(*) AS total_requests,
        SUM(CASE WHEN status = 'Trip Completed' THEN 1 ELSE 0 END) AS completed_trips,
        ROUND(SUM(CASE WHEN status = 'Trip Completed' THEN 1 ELSE 0 END)*100.0/COUNT(*), 2) AS completion_rate
        FROM uber_requests
        GROUP BY Request_Hour
        ORDER BY completion_rate ASC
        LIMIT 5;
        '''
pd.read_sql(query, conn)
```

	hour_of_day	total_requests	completed_trips	completion_rate
0	2	85	25	29.41
1	22	449	142	31.63
2	19	510	164	32.16
3	21	492	161	32.72
4	20	473	166	35.10

The hours with the lowest trip completion rates are 2, 22, 19, 21, and 20. These hours, particularly those in the evening (19, 20, 21, 22), indicate significant challenges in matching riders with available drivers, leading to a high number of unfulfilled requests. This suggests a supply-demand mismatch during these specific times.

In simple terms, Uber is having the hardest time completing rides during specific hours: 2 AM and then a block of hours in the evening (7 PM, 8 PM, 9 PM, and 10 PM). This means during these times, many people are asking for rides, but not enough drivers are available, leading to a lot of unfulfilled requests. Focusing on getting more drivers on the road during these hours, especially in the evening, could significantly improve service.

7. Request Distribution by Hour (Demand Curve)

```
query = '''
        SELECT request_hour, COUNT(*) AS total_requests
        FROM uber_requests
        GROUP BY request_hour
        ORDER BY total_requests DESC
        LIMIT 10;
        '''
pd.read_sql(query, conn)
```

	Request_Hour	total_requests
0	18	510
1	20	492
2	19	473
3	21	449
4	5	445
5	9	431
6	8	423
7	17	418
8	7	406
9	6	398

Requests are heavily clustered in Morning and Evening hours — classic rush hour trend.

8. Trip Completion Count per Time Slot (Actual Fulfillment)

```
query = '''
        SELECT time_slot, COUNT(*) AS total_requests,
        SUM(CASE WHEN status = 'Trip Completed' THEN 1 ELSE 0 END) AS trips_completed
        FROM uber_requests
        GROUP BY time_slot
        ORDER BY trips_completed ASC;
        '''
pd.read_sql(query, conn)
```

	Time_Slot	total_requests	trips_completed
0	Late Night	578	214
1	Day	1224	722
2	Morning	2103	854
3	Evening	2840	1041

While earlier queries showed high demand in Morning and Evening, this query highlights the gap between demand and fulfilled trips. The number of completed trips in the Morning slot is relatively low compared to the total requests, reinforcing the idea of a significant supply shortage during this peak time. Even in the Evening, while more trips are completed than in the morning, the number of uncompleted requests (total minus completed) is still substantial. This metric shows the actual success rate of converting demand into completed rides.

```
_
```