Uber Supply-Demand SQL Insights Report

Tools Used: Excel, SQL, Python (Pandas)

This report contains key insights from the Uber request data using SQL-style logic translated to Python visualizations. These findings are part of the final submission for the Uber Supply-Demand Gap project.

Business Objective

To analyze Uber's ride request data and identify patterns in cancellations and ride unavailability, with the goal of optimizing supply and improving ride fulfillment — especially in problematic time slots and pickup zones.

SQL-Based Insights

• Total Trips by Status: Most ride requests are either cancelled or marked "No Cars Available", showing operational inefficiencies.

• Peak Hours by Pickup Point: Airport sees more activity during Late Night and Early Morning; City peaks during rush hours.

• Cancellation Rate by Time Slot: Early Morning has the highest cancellation rate due to driver refusals.

• Airport Failures: "No Cars Available" spikes at the Airport between 12 AM and 5 AM.

• Trip Completions: Afternoon and Evening time slots show the best service rates.

Import and Load Data

In [15]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('Uber_Request_Data_MySQL.csv')
df['Request_date'] = pd.to_datetime(df['Request_date'], dayfirst=True, errors='coerce').dt.date
df['Drop_date'] = pd.to_datetime(df['Drop_date'], dayfirst=True, errors='coerce').dt.date
df['Trip_completed'] = df['Trip_completed'].apply(lambda x: 1 if x == True or x == 1 else 0)
```

In [16]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Set styles
sns.set(style="whitegrid")
plt.rcParams["figure.figsize"] = (10, 6)

# Load your dataset (adjust path if needed)
df = pd.read_csv("Uber_Request_Data_Cleaned_SQL.csv")

# Generate insights data
status_counts = df["Status"].value_counts()
pickup_status = df.groupby(["Pickup_point", "Status"]).size().unstack()

hourly_demand = df.groupby("Request_hour").size()
completed_per_hour = df[df["Status"] == "Trip Completed"].groupby("Request_hour").size()

time_slot_demand = df["Time_slot"].value_counts()
time_slot_status = df.groupby(["Time_slot", "Status"]).size().unstack().fillna(0)

pickup_demand = df["Pickup_point"].value_counts()
day_status = df.groupby(["Request_day", "Status"]).size().unstack().fillna(0)
```

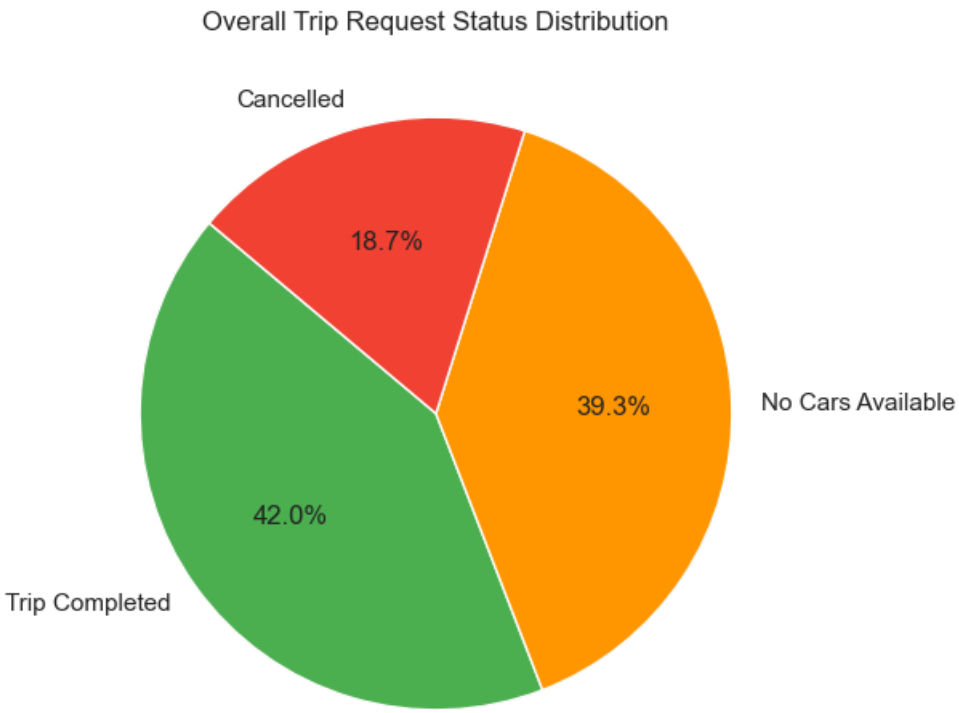
Key Visual Insights

Overall Request Status Distribution

1. ~42% trips were completed.
2. ~39% had no cars available.
3. ~19% were cancelled.

This indicates a significant demand-supply issue, especially due to "No Cars Available".

In [17]:
```
# Plot request status distribution
status_counts.plot(kind="pie", autopct="%1.1f%%", startangle=140, colors=["#4CAF50", "#FF9800", "#F44336"])
plt.title("Overall Trip Request Status Distribution")
plt.ylabel("")
plt.show()
```



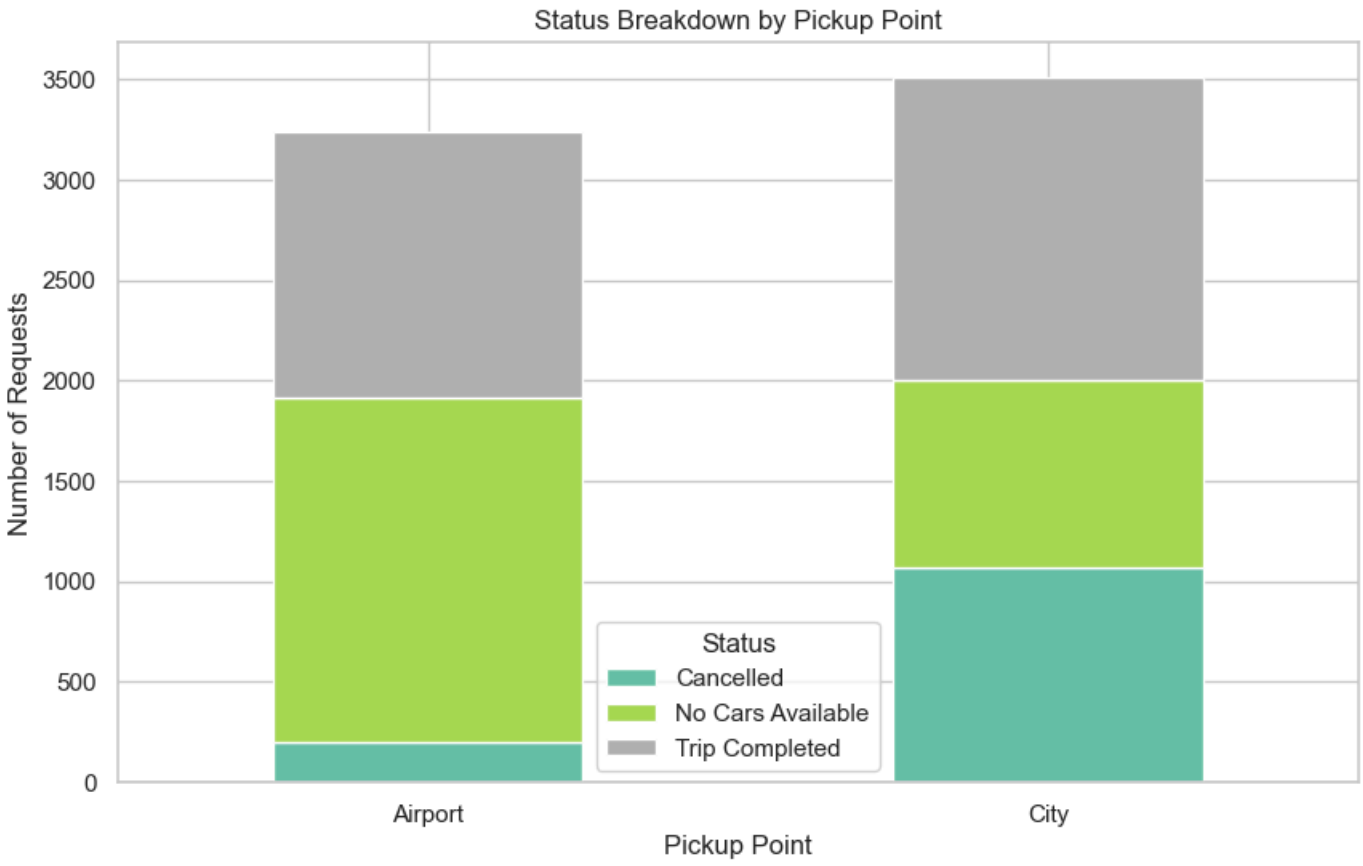
2. Pickup Point vs Status

1.City has more cancellations.

2.Airport has higher "No Cars Available".

Suggests possible driver reluctance or unavailability at the Airport.

In [18]:
```
pickup_status.plot(kind="bar", stacked=True, colormap="Set2")
plt.title("Status Breakdown by Pickup Point")
plt.xlabel("Pickup Point")
plt.ylabel("Number of Requests")
plt.xticks(rotation=0)
plt.show()
```

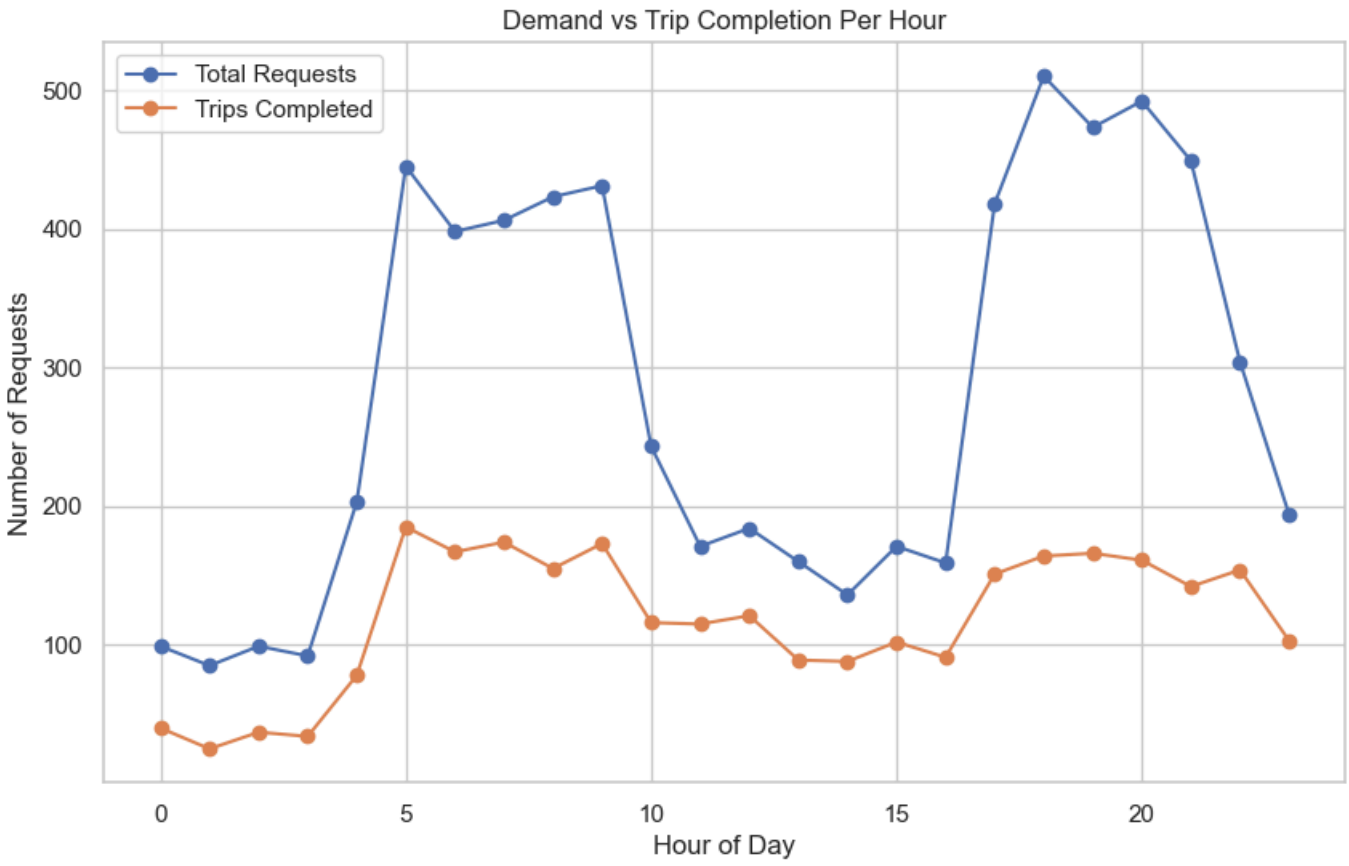


Requests Per Hour (Demand vs Completed)

1.High demand during 8 AM–10 AM and 5 PM–9 PM.

2.Completion rate drops significantly during peak hours.

In [19]:
```
plt.plot(hourly_demand.index, hourly_demand.values, label="Total Requests", marker='o')
plt.plot(completed_per_hour.index, completed_per_hour.values, label="Trips Completed", marker='o')
plt.title("Demand vs Trip Completion Per Hour")
plt.xlabel("Hour of Day")
plt.ylabel("Number of Requests")
plt.legend()
plt.grid(True)
plt.show()
```

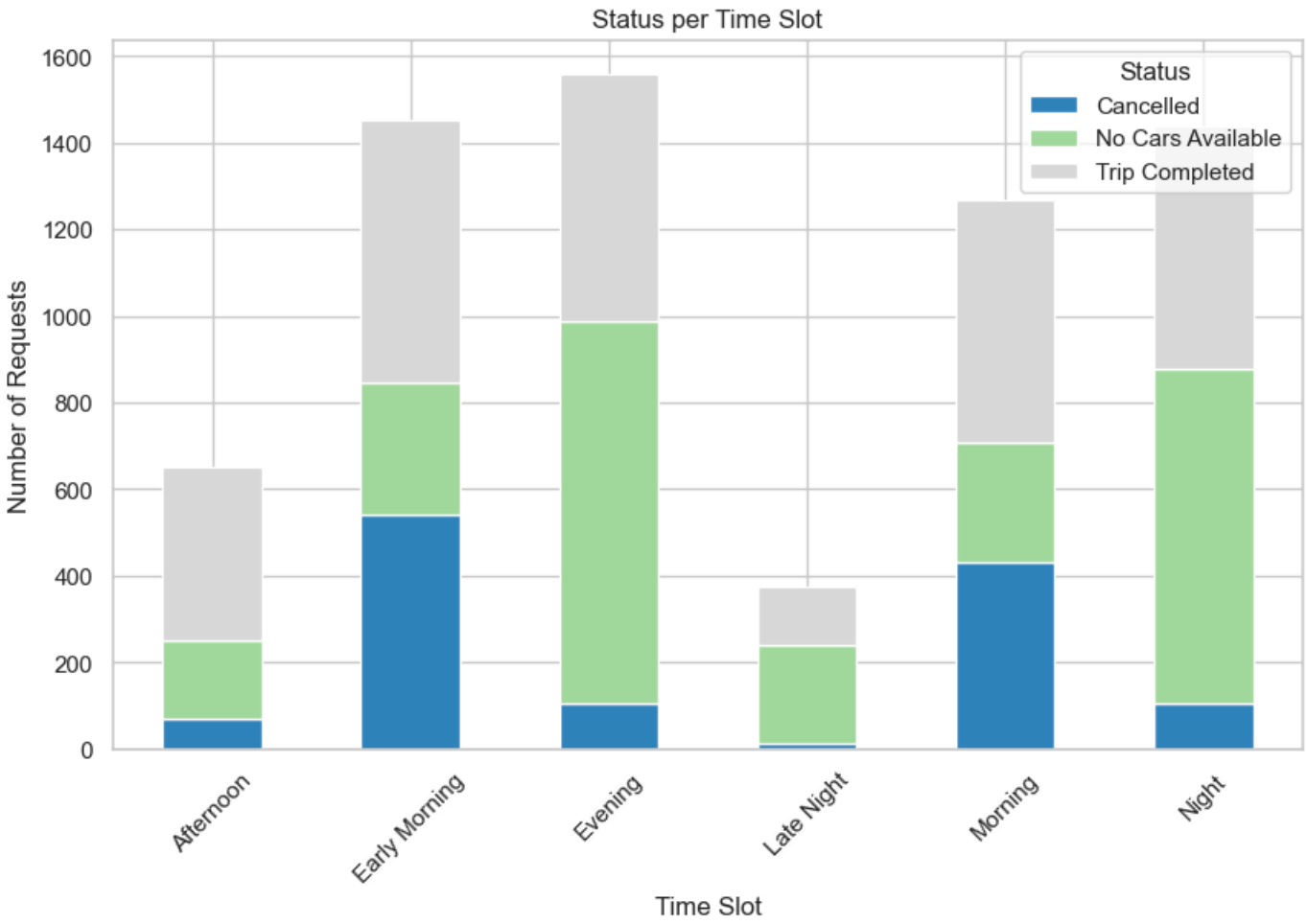


Time Slot vs Status

1.Evening & Night slots show highest “No Cars Available”.

2.Morning has more cancellations.

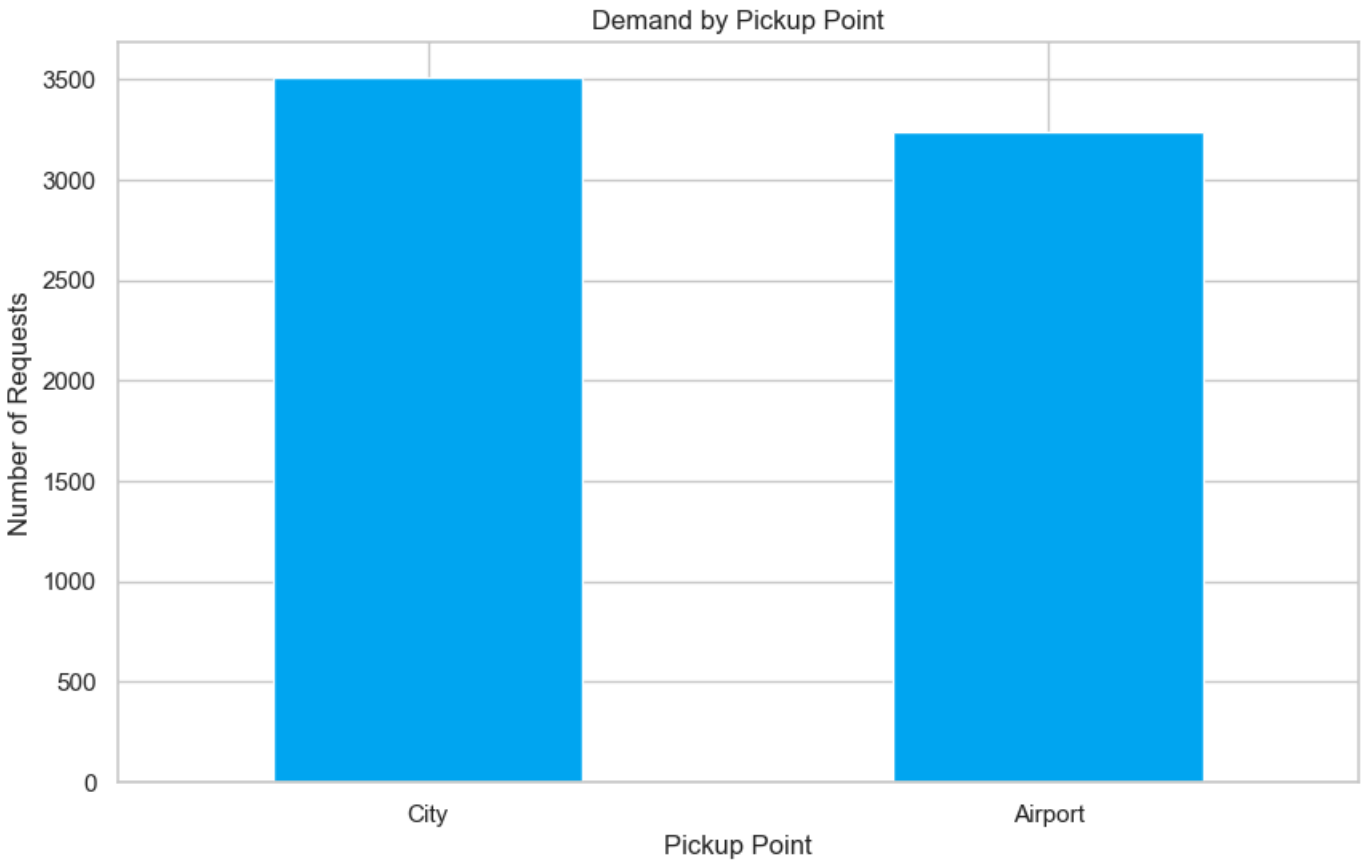
In [20]:
```
time_slot_status.plot(kind="bar", stacked=True, colormap="tab20c")
plt.title("Status per Time Slot")
plt.xlabel("Time Slot")
plt.ylabel("Number of Requests")
plt.xticks(rotation=45)
plt.show()
```



Pickup Point Demand

1.City has slightly more requests than Airport.

In [21]:
```
pickup_demand.plot(kind="bar", color="#03A9F4")
plt.title("Demand by Pickup Point")
plt.xlabel("Pickup Point")
plt.ylabel("Number of Requests")
plt.xticks(rotation=0)
plt.show()
```



Request Status by Day

1.Cancellations are highest on Wednesdays.

2.“No Cars Available” peaks on Thursdays and Fridays.

In [22]:
```
day_status.plot(kind="bar", stacked=True, colormap="Pastel1")
plt.title("Status Breakdown by Day of Week")
plt.xlabel("Day")
plt.ylabel("Number of Requests")
plt.xticks(rotation=45)
plt.show()
```

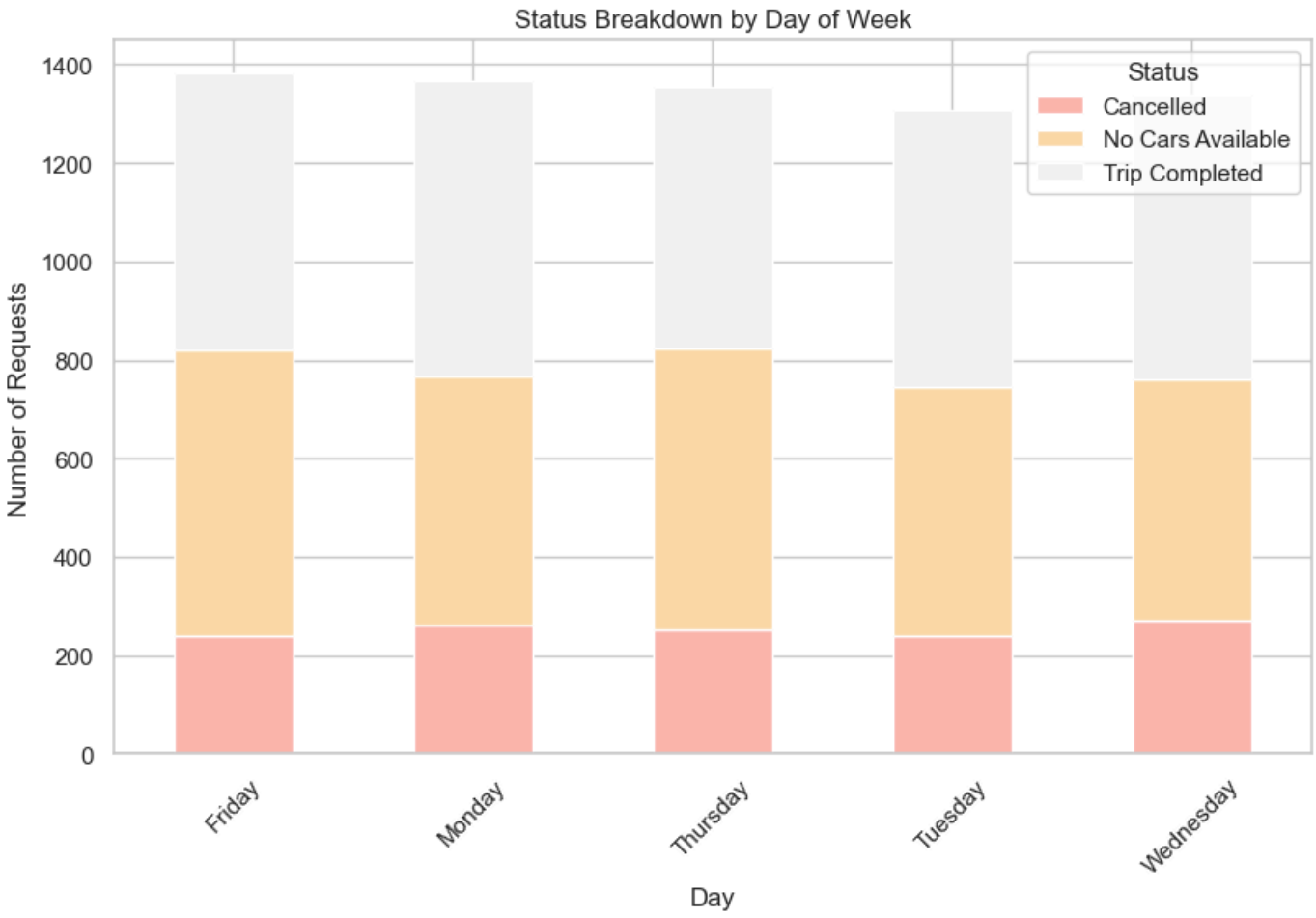


Chart 7. Top 10 Drivers

1.A small number of drivers complete significantly more trips.

2.Indicates driver commitment or optimized positioning.

🔍 Recommendation: Study their patterns and replicate for training or routing.

In [23]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Load the cleaned data
df = pd.read_csv("Uber_Request_Data_Cleaned_SQL.csv")

# Ensure 'Trip_completed' is boolean
df['Trip_completed'] = df['Trip_completed'].astype(bool)

# Filter only completed trips
completed_trips = df[df['Trip_completed'] == True]

# Top 10 drivers by trip count
top_10_drivers = (
    completed_trips.groupby('Driver id')
    .size()
    .reset_index(name='Trips_Completed')
    .sort_values(by='Trips_Completed', ascending=False)
    .head(10)
)

# Plot
plt.figure(figsize=(10, 6))
plt.bar(top_10_drivers['Driver id'].astype(str), top_10_drivers['Trips_Completed'], color='skyblue')
plt.xlabel('Driver ID')
plt.ylabel('Trips Completed')
plt.title('Top 10 Drivers by Number of Completed Trips')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

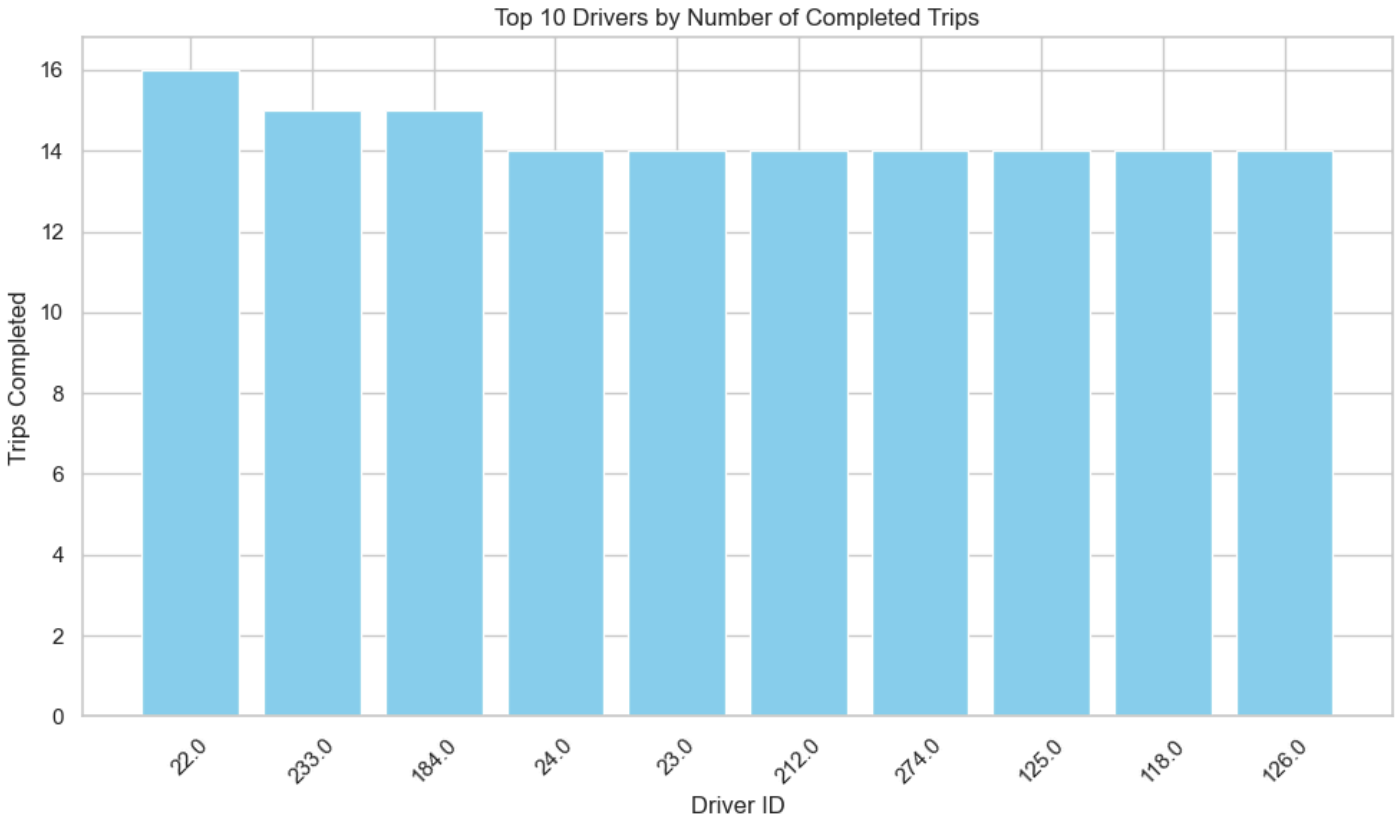


Chart 9. Missing Drop-offs by Hour

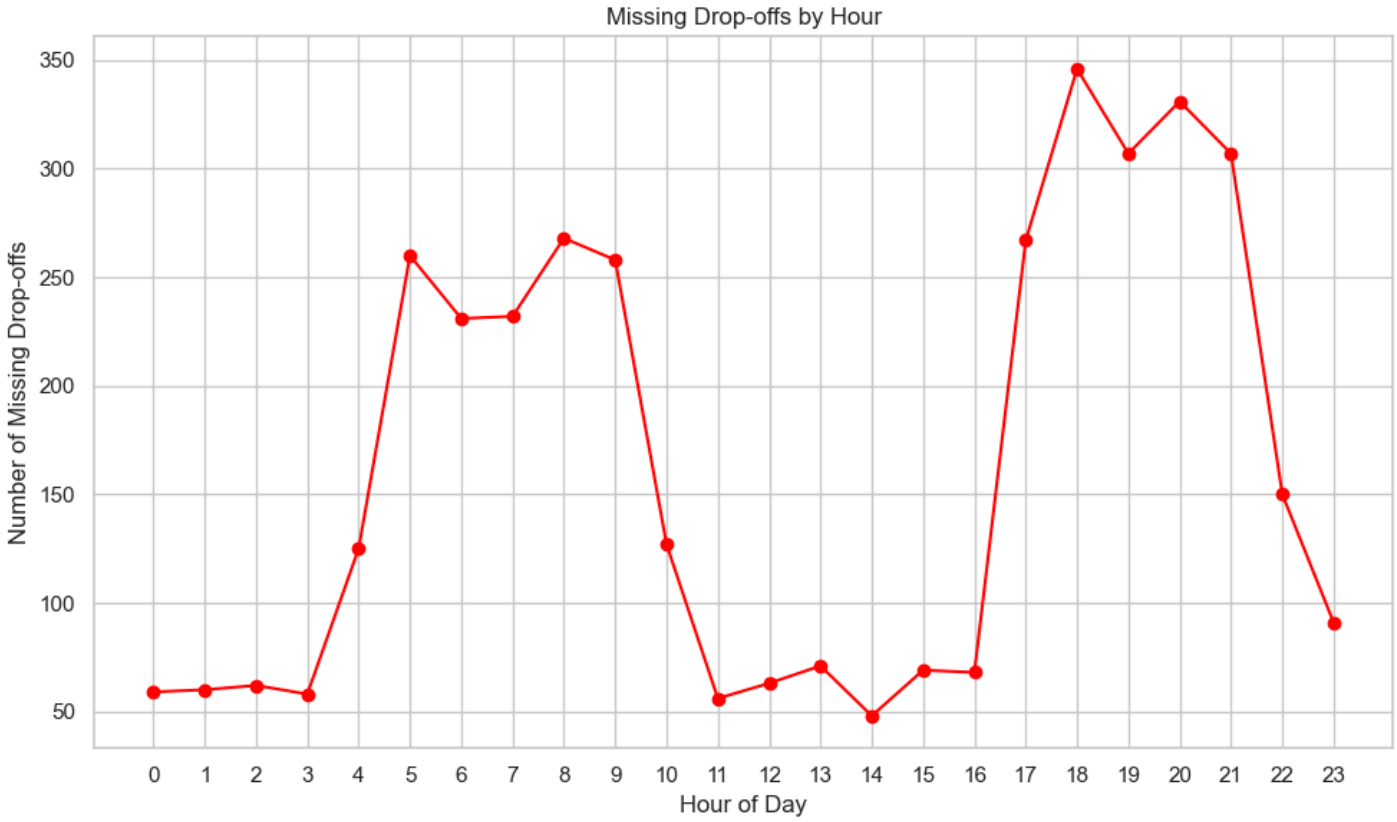
1.More missing drop-off data in late night/early morning.

2.May indicate unfulfilled trips or missing system logs.

🔍 Recommendation: Review system tracking or incident logs for these hours.

In [24]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_csv("Uber_Request_Data_Cleaned_SQL.csv")

# Ensure 'Trip_completed' is boolean
df['Trip_completed'] = df['Trip_completed'].astype(bool)

# Filter for trips that were NOT completed (missing drop-offs)
missing_dropoffs = df[df['Trip_completed'] == False]

# Group by hour of request and count
missing_by_hour = (
    missing_dropoffs.groupby('Request_hour')
    .size()
    .reset_index(name='Missing_Trips')
    .sort_values(by='Request_hour')
)

# Print table (optional)
print(missing_by_hour)

# Plotting
plt.figure(figsize=(10, 6))
plt.plot(missing_by_hour['Request_hour'], missing_by_hour['Missing_Trips'], marker='o', color='red')
plt.xlabel('Hour of Day')
plt.ylabel('Number of Missing Drop-offs')
plt.title('Missing Drop-offs by Hour')
plt.grid(True)
plt.xticks(range(0, 24))
plt.tight_layout()
plt.show()
```

	Request_hour	Missing_Trips
0	0	59
1	1	60
2	2	62
3	3	58
4	4	125
5	5	260
6	6	231
7	7	232
8	8	268
9	9	258
10	10	127
11	11	56
12	12	63
13	13	71
14	14	48
15	15	69
16	16	68
17	17	267
18	18	346
19	19	307
20	20	331
21	21	307
22	22	150
23	23	91



Summary

The Uber platform experiences peak demand in the evenings, faces most challenges with cancellations and driver unavailability, and has untapped improvement potential in night hours and driver distribution.

Recommendations

Time-Slot Based Driver Scheduling: - Increase driver availability during Late Night (12–5 AM) and Early Morning (4–8 AM). Pickup-Specific Solutions: - Airport: Night-shift coverage + bonuses. - City: Address driver cancellation behavior, especially early hours. Dynamic Pricing + Incentives: - Use surge pricing and guaranteed fares for unserved periods. Predictive Modeling: - Use ML to forecast demand by hour and pickup point.

Conclusion

Uber is facing clear supply-demand mismatches — most notably: - High “No Cars Available” during Late Night at the Airport - Frequent cancellations in Early Morning in the City By implementing time-specific driver incentives, better forecasting, and optimized shift planning, Uber can significantly improve ride completion rates, reduce customer churn, and unlock lost revenue opportunities.