[1]:

```python
import pandas as pd
import mysql.connector

# Database connection
db = mysql.connector.connect(
    host="localhost",
    user="root",
    password="AA7077@nak",
    database="rapido"
)
# Function to execute a query and return the result as a DataFrame
def fetch_data(query):
    cur = db.cursor(dictionary=True)
    cur.execute(query)
    result = cur.fetchall()
    return pd.DataFrame(result)
```

[2]:

```python
query1 = "SELECT * FROM mumbai_booking_data WHERE Booking_Status = 'Success';"
successful_bookings = fetch_data(query1)
print("Successful Bookings:")
successful_bookings
```

```
Successful Bookings:
```

[2]:

	Date	Time	Booking_ID	Booking_Status	Customer_ID	Vehicle_Type	Pickup_Location	Drop_Location	Avg_VTAT	Avg_CTAT	Cancelled_Rides_by_Customer
0	2025-01-21	14:55	CNR2373430247	Success	CUST907817	eBike	Area 3	Area 26	12.76	17.36	0
1	2025-01-13	14:46	CNR8825424090	Success	CUST359317	Bike	Area 3	Area 35	4.95	8.60	0
2	2025-01-17	12:50	CNR2899177378	Success	CUST338656	Prime Sedan	Area 21	Area 5	12.24	12.38	0
3	2025-01-08	02:13	CNR4467207253	Success	CUST495153	eBike	Area 21	Area 18	9.69	6.06	0
4	2025-01-26	07:00	CNR9464572995	Success	CUST492121	Bike	Area 32	Area 31	10.90	17.09	0
...	...	...	...	...	...	...	...	...	...	...	...
66828	2025-01-06	16:42	CNR6199834671	Success	CUST322703	Auto	Area 31	Area 34	12.02	5.11	0
66829	2025-01-18	04:01	CNR9270824454	Success	CUST819149	eBike	Area 42	Area 25	3.58	21.88	0
66830	2025-01-28	12:42	CNR8878295435	Success	CUST828854	Mini	Area 26	Area 21	9.64	16.66	0
66831	2025-01-08	09:40	CNR2447490578	Success	CUST796716	Auto	Area 18	Area 49	1.76	14.50	0
66832	2025-01-27	22:57	CNR6112472209	Success	CUST519046	Bike	Area 22	Area 14	14.54	18.01	0

01-08

66833 rows × 20 columns

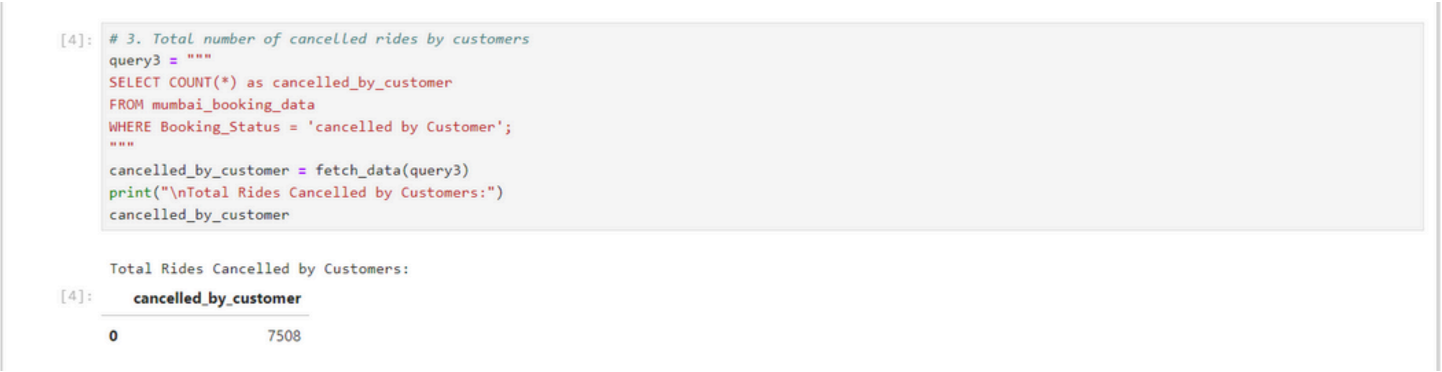
[3]:

```python
# 2. Average ride distance for each vehicle type
query2 = """
SELECT Vehicle_Type, AVG(Ride_Distance) as avg_distance
FROM mumbai_booking_data
GROUP BY Vehicle_Type;
"""
avg_ride_distance = fetch_data(query2)
print("\nAverage Ride Distance by Vehicle Type:")
avg_ride_distance
```

```
Average Ride Distance by Vehicle Type:
```

[3]:

	Vehicle_Type	avg_distance
0	eBike	45.785770
1	Bike	45.583207
2	Prime Sedan	45.265887
3	Auto	45.742474
4	Prime Plus	45.618667
5	Prime SUV	45.816331
6	Mini	46.093986

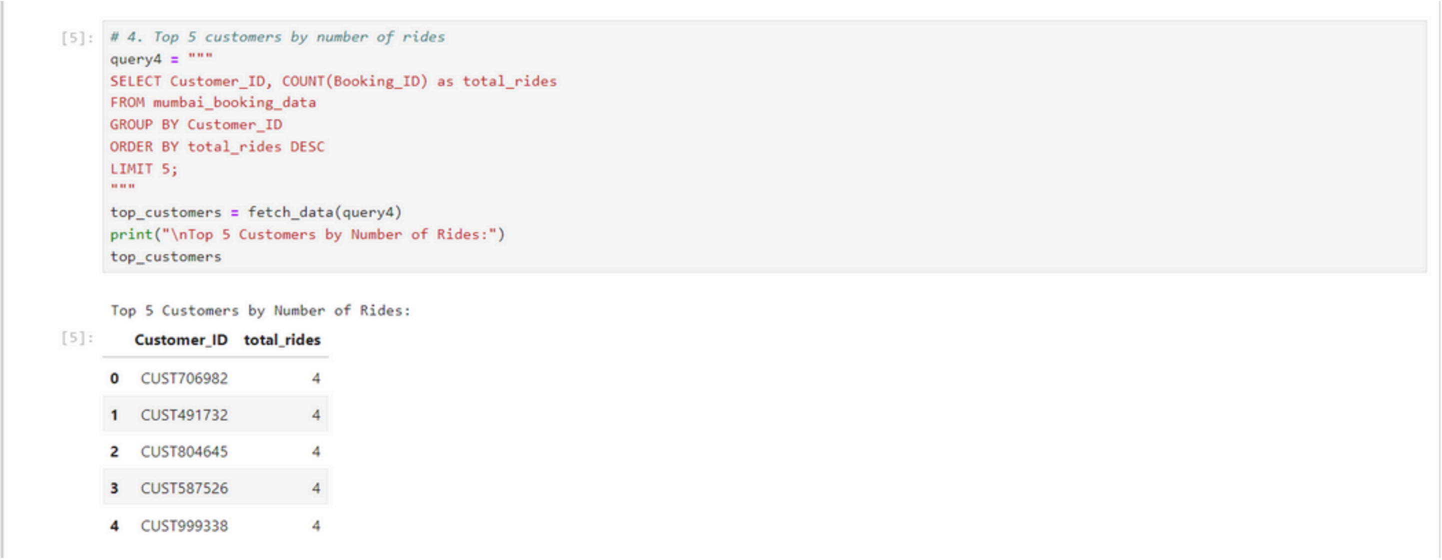


```
[4]: # 3. Total number of cancelled rides by customers
     query3 = """
     SELECT COUNT(*) as cancelled_by_customer
     FROM mumbai_booking_data
     WHERE Booking_Status = 'cancelled by Customer';
     """
     cancelled_by_customer = fetch_data(query3)
     print("\nTotal Rides Cancelled by Customers:")
     cancelled_by_customer
```

```
Total Rides Cancelled by Customers:
```

[4]:

	cancelled_by_customer
0	7508

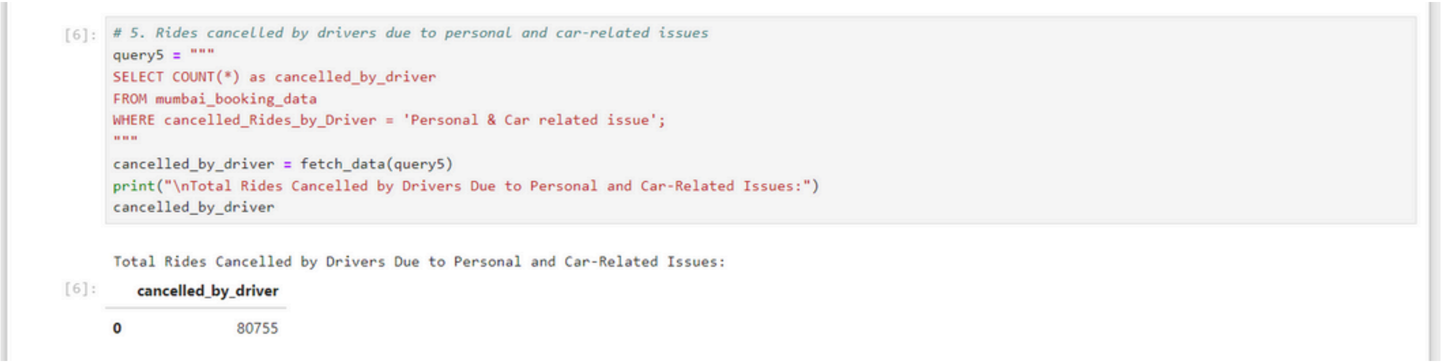


```
[5]: # 4. Top 5 customers by number of rides
     query4 = """
     SELECT Customer_ID, COUNT(Booking_ID) as total_rides
     FROM mumbai_booking_data
     GROUP BY Customer_ID
     ORDER BY total_rides DESC
     LIMIT 5;
     """
     top_customers = fetch_data(query4)
     print("\nTop 5 Customers by Number of Rides:")
     top_customers
```

```
Top 5 Customers by Number of Rides:
```

[5]:

	Customer_ID	total_rides
0	CUST706982	4
1	CUST491732	4
2	CUST804645	4
3	CUST587526	4
4	CUST999338	4

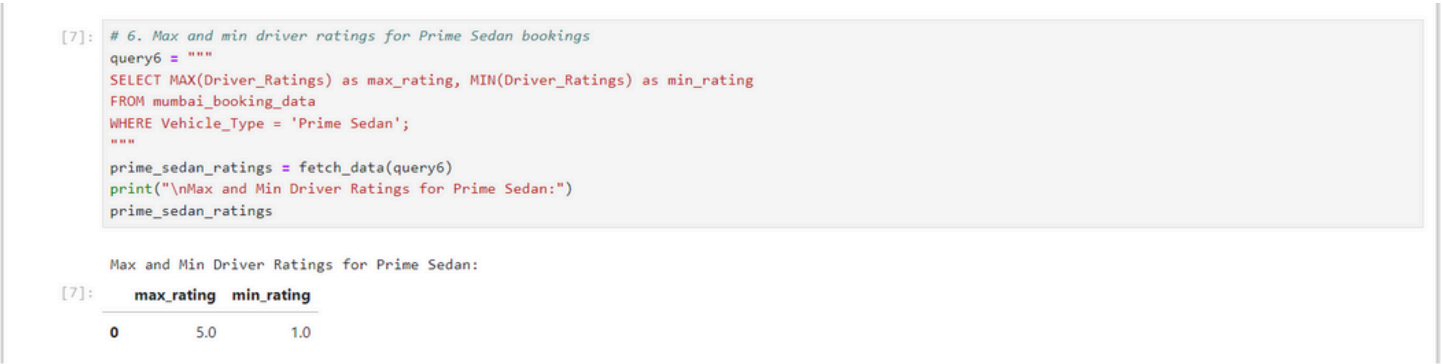


```
[6]: # 5. Rides cancelled by drivers due to personal and car-related issues
     query5 = """
     SELECT COUNT(*) as cancelled_by_driver
     FROM mumbai_booking_data
     WHERE cancelled_Rides_by_Driver = 'Personal & Car related issue';
     """
     cancelled_by_driver = fetch_data(query5)
     print("\nTotal Rides Cancelled by Drivers Due to Personal and Car-Related Issues:")
     cancelled_by_driver
```

```
Total Rides Cancelled by Drivers Due to Personal and Car-Related Issues:
```

[6]:

	cancelled_by_driver
0	80755

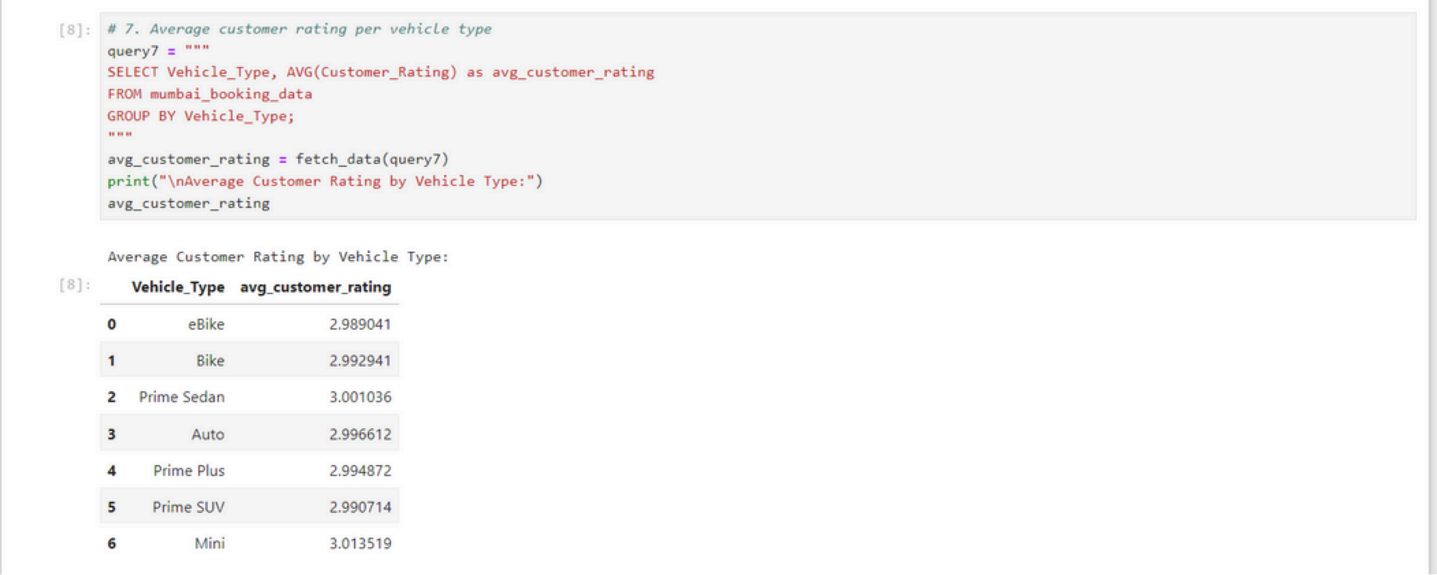


```
[7]: # 6. Max and min driver ratings for Prime Sedan bookings
     query6 = """
     SELECT MAX(Driver_Ratings) as max_rating, MIN(Driver_Ratings) as min_rating
     FROM mumbai_booking_data
     WHERE Vehicle_Type = 'Prime Sedan';
     """
     prime_sedan_ratings = fetch_data(query6)
     print("\nMax and Min Driver Ratings for Prime Sedan:")
     prime_sedan_ratings
```

```
Max and Min Driver Ratings for Prime Sedan:
```

[7]:

	max_rating	min_rating
0	5.0	1.0

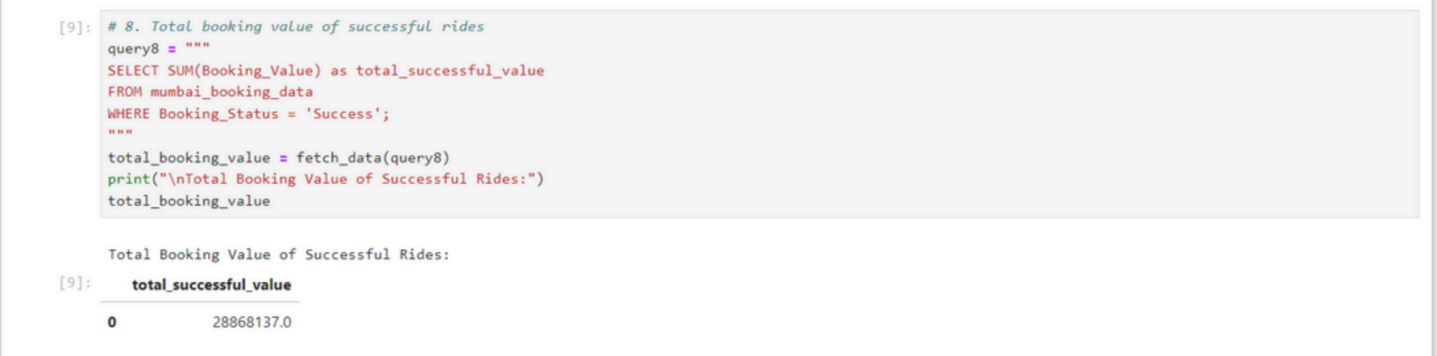


```
[8]: # 7. Average customer rating per vehicle type
query7 = """
SELECT Vehicle_Type, AVG(Customer_Rating) as avg_customer_rating
FROM mumbai_booking_data
GROUP BY Vehicle_Type;
"""
avg_customer_rating = fetch_data(query7)
print("\nAverage Customer Rating by Vehicle Type:")
avg_customer_rating
```

```
Average Customer Rating by Vehicle Type:
```

[8]:

	Vehicle_Type	avg_customer_rating
0	eBike	2.989041
1	Bike	2.992941
2	Prime Sedan	3.001036
3	Auto	2.996612
4	Prime Plus	2.994872
5	Prime SUV	2.990714
6	Mini	3.013519

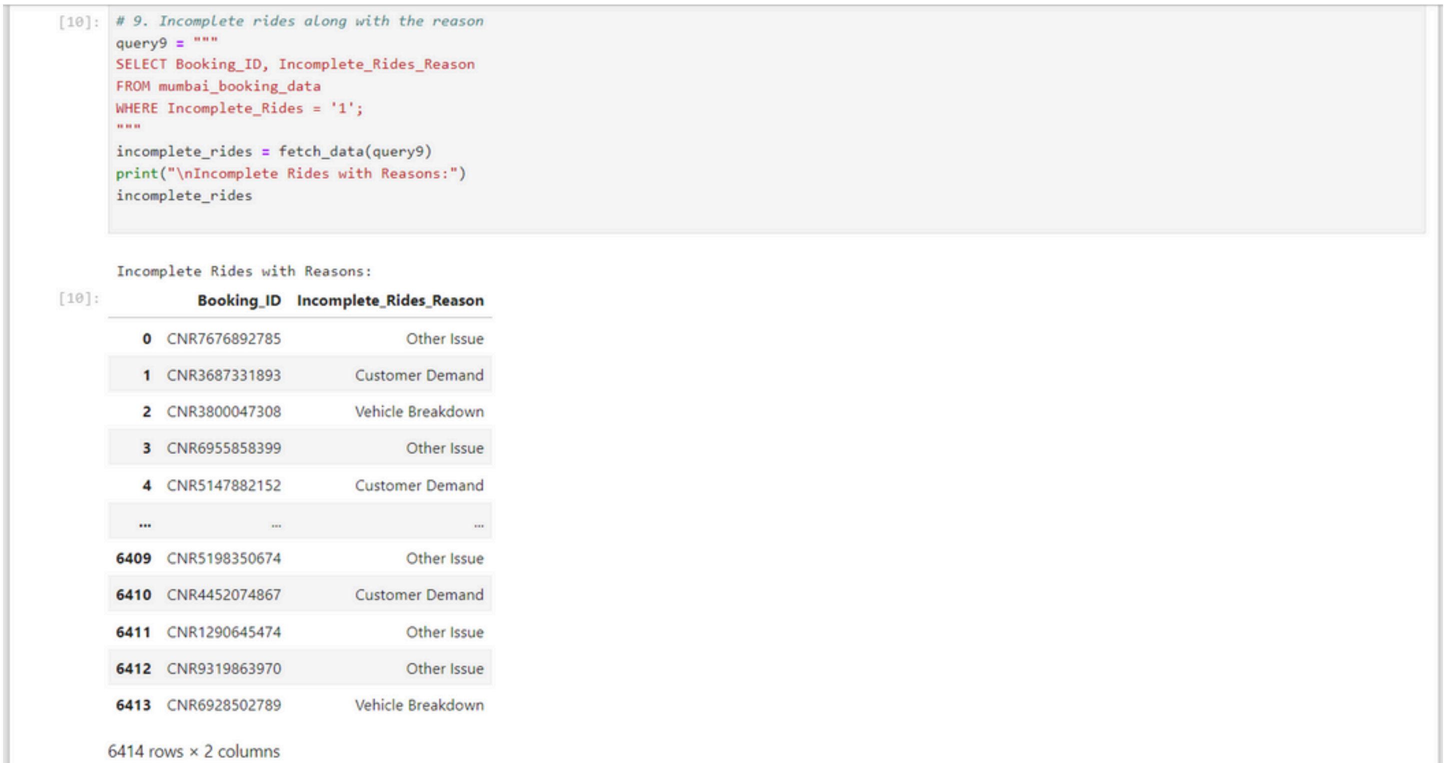


```
[9]: # 8. Total booking value of successful rides
query8 = """
SELECT SUM(Booking_Value) as total_successful_value
FROM mumbai_booking_data
WHERE Booking_Status = 'Success';
"""
total_booking_value = fetch_data(query8)
print("\nTotal Booking Value of Successful Rides:")
total_booking_value
```

```
Total Booking Value of Successful Rides:
```

[9]:

	total_successful_value
0	28868137.0



```
[10]: # 9. Incomplete rides along with the reason
query9 = """
SELECT Booking_ID, Incomplete_Rides_Reason
FROM mumbai_booking_data
WHERE Incomplete_Rides = '1';
"""
incomplete_rides = fetch_data(query9)
print("\nIncomplete Rides with Reasons:")
incomplete_rides
```

```
Incomplete Rides with Reasons:
```

[10]:

	Booking_ID	Incomplete_Rides_Reason
0	CNR7676892785	Other Issue
1	CNR3687331893	Customer Demand
2	CNR3800047308	Vehicle Breakdown
3	CNR6955858399	Other Issue
4	CNR5147882152	Customer Demand
...	...	...
6409	CNR5198350674	Other Issue
6410	CNR4452074867	Customer Demand
6411	CNR1290645474	Other Issue
6412	CNR9319863970	Other Issue
6413	CNR6928502789	Vehicle Breakdown

6414 rows × 2 columns