

OLA SQL Project Report

Project Overview

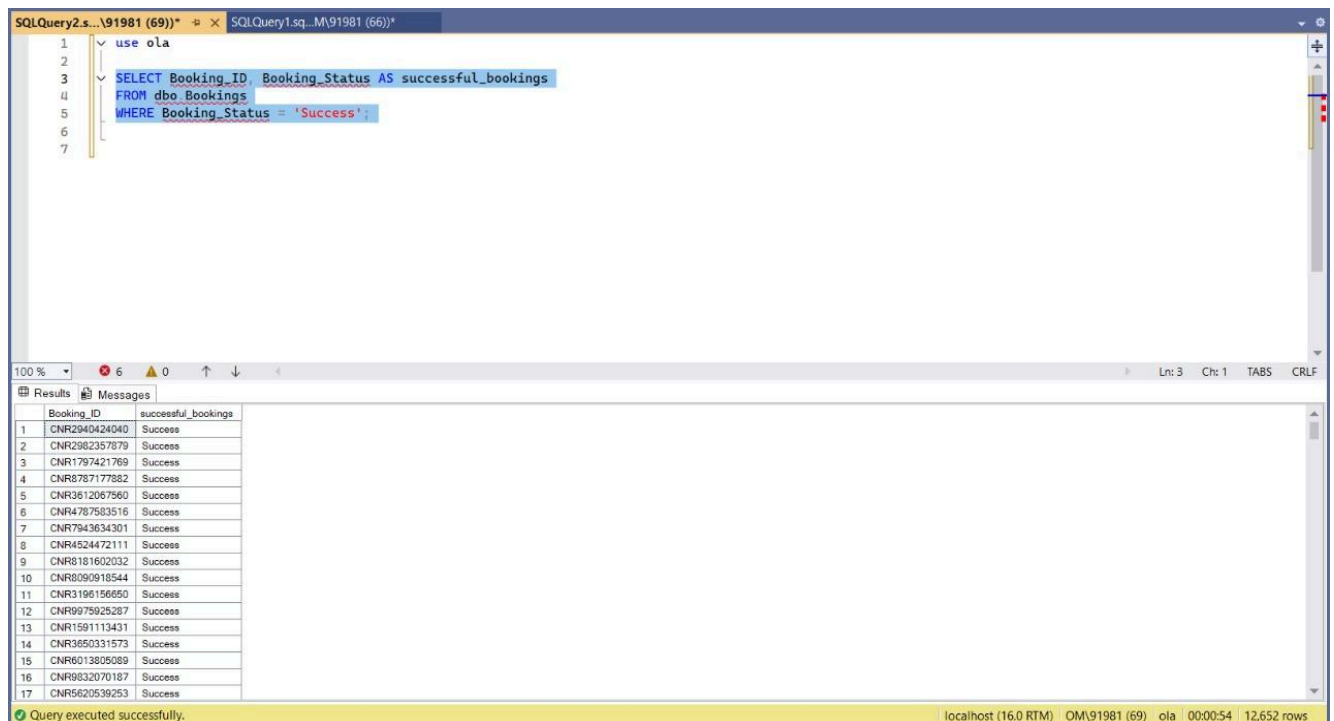
This project presents an analysis of the Ola ride booking system using SQL. Various queries were written to retrieve, filter, and aggregate data to gain insights into ride performance, customer behavior, and operational statistics. The data is assumed to come from a table named 'Bookings' in the 'ola' database.

1. Successful Bookings

This query retrieves all bookings that were marked as successfully completed.

SQL Query:

```
SELECT Booking_ID, Booking_Status AS successful_bookings
FROM dbo.Bookings
WHERE Booking_Status = 'Success';
```



The screenshot shows a SQL Server Enterprise Manager window with a query editor and a results pane. The query editor contains the following SQL code:

```
1 use ola
2
3 SELECT Booking_ID, Booking_Status AS successful_bookings
4 FROM dbo.Bookings
5 WHERE Booking_Status = 'Success';
6
7
```

The results pane displays the output of the query, showing a table with two columns: Booking_ID and successful_bookings. The table contains 17 rows of data, all with a status of 'Success'.

Booking_ID	successful_bookings
CNR2940424040	Success
CNR2982357879	Success
CNR1797421769	Success
CNR8787177882	Success
CNR3612067960	Success
CNR4787583516	Success
CNR7943634301	Success
CNR4524472111	Success
CNR8181602032	Success
CNR8090918544	Success
CNR3196156650	Success
CNR9975925287	Success
CNR1591113431	Success
CNR3650331573	Success
CNR6013805089	Success
CNR9832070187	Success
CNR5620539253	Success

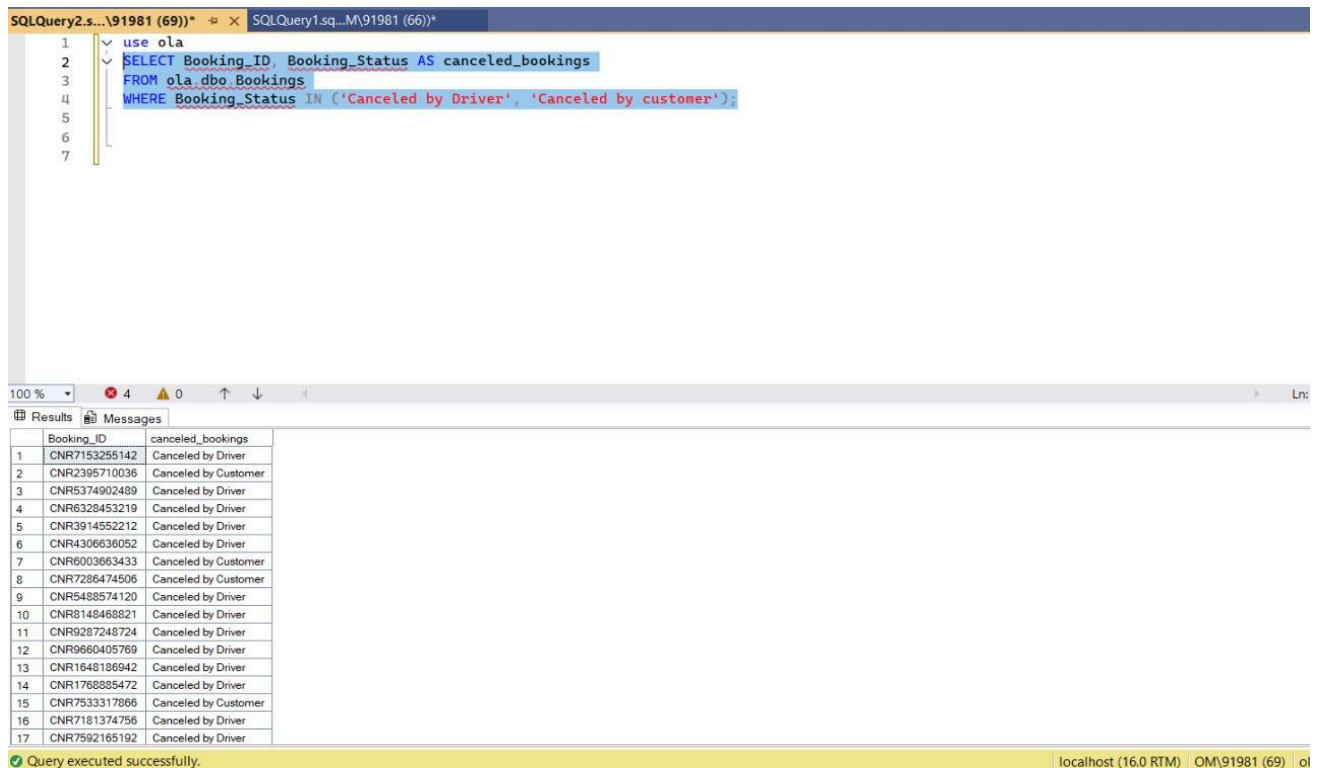
The status bar at the bottom indicates: Query executed successfully. | localhost (16.0 RTM) | OM\91981 (69) | ola | 00:00:54 | 12,652 rows

2. Canceled Bookings

This query filters out all canceled bookings and categorizes them based on who canceled the ride.

SQL Query:

```
SELECT Booking_ID, Booking_Status AS canceled_bookings
FROM dbo.Bookings
WHERE Booking_Status IN ('Canceled by Driver', 'Canceled by customer');
```



The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a query window with the following SQL code:

```
1 use ola
2 SELECT Booking_ID, Booking_Status AS canceled_bookings
3 FROM ola.dbo.Bookings
4 WHERE Booking_Status IN ('Canceled by Driver', 'Canceled by customer');
5
6
7
```

The bottom pane shows the results of the query, which is a table with two columns: Booking_ID and canceled_bookings. The results are as follows:

	Booking_ID	canceled_bookings
1	CNR7153255142	Canceled by Driver
2	CNR2395710036	Canceled by Customer
3	CNR5374902489	Canceled by Driver
4	CNR6328453219	Canceled by Driver
5	CNR3914552212	Canceled by Driver
6	CNR4306636052	Canceled by Driver
7	CNR6003663433	Canceled by Customer
8	CNR7286474506	Canceled by Customer
9	CNR5488574120	Canceled by Driver
10	CNR8148468821	Canceled by Driver
11	CNR9287248724	Canceled by Driver
12	CNR9660405769	Canceled by Driver
13	CNR1648186942	Canceled by Driver
14	CNR1768885472	Canceled by Driver
15	CNR7533317866	Canceled by Customer
16	CNR7181374756	Canceled by Driver
17	CNR7592165192	Canceled by Driver

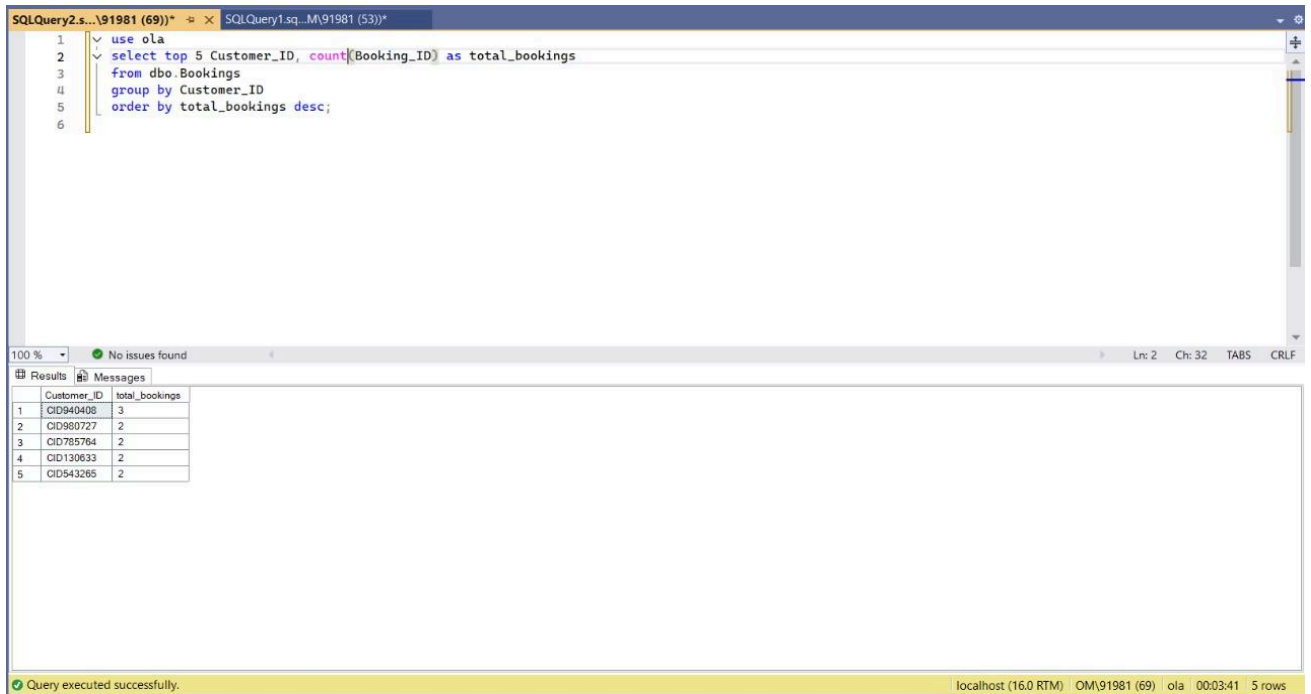
At the bottom of the interface, a status bar indicates: "Query executed successfully. localhost (16.0 RTM) OM\91981 (69) ol".

3. Top 5 Customers by Bookings

Displays the top 5 customers who made the highest number of bookings.

SQL Query:

```
SELECT TOP 5 Customer_ID, COUNT(Booking_ID) AS total_bookings
FROM dbo.Bookings
GROUP BY Customer_ID
ORDER BY total_bookings DESC;
```



The screenshot shows a SQL Server Enterprise Manager window with a query editor and a results pane. The query editor contains the following SQL code:

```
1 use ola
2 select top 5 Customer_ID, count(Booking_ID) as total_bookings
3 from dbo.Bookings
4 group by Customer_ID
5 order by total_bookings desc;
6
```

The results pane displays the following data:

	Customer_ID	total_bookings
1	OD940408	3
2	OD980727	2
3	OD785764	2
4	OD130633	2
5	OD543265	2

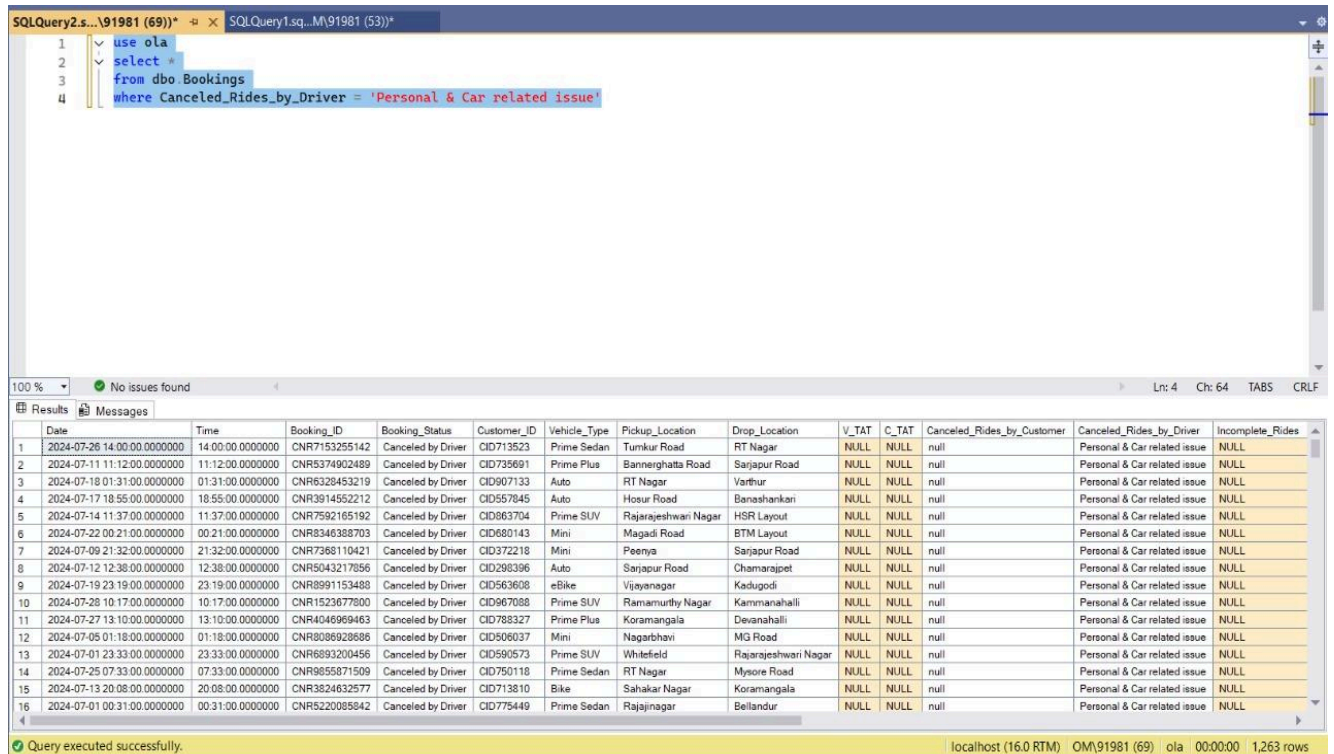
The status bar at the bottom indicates: Query executed successfully. localhost (16.0 RTM) OM(91981 (69)) ola 00:03:41 5 rows

4. Bookings Canceled Due to Car Issues

Retrieves bookings that were canceled by the driver due to personal or vehicle issues.

SQL Query:

```
SELECT * FROM dbo.Bookings
WHERE Canceled_Rides_by_Driver = 'Personal & Car related issue';
```



The screenshot shows a SQL Server Enterprise Manager window with a query executed successfully. The query is: `use ola; select * from dbo.Bookings where Canceled_Rides_by_Driver = 'Personal & Car related issue';` The results pane displays 16 rows of data. The status bar at the bottom indicates 'Query executed successfully.' and 'localhost (16.0 RTM) OM\91981 (69) ola 00:00:00 1,263 rows'.

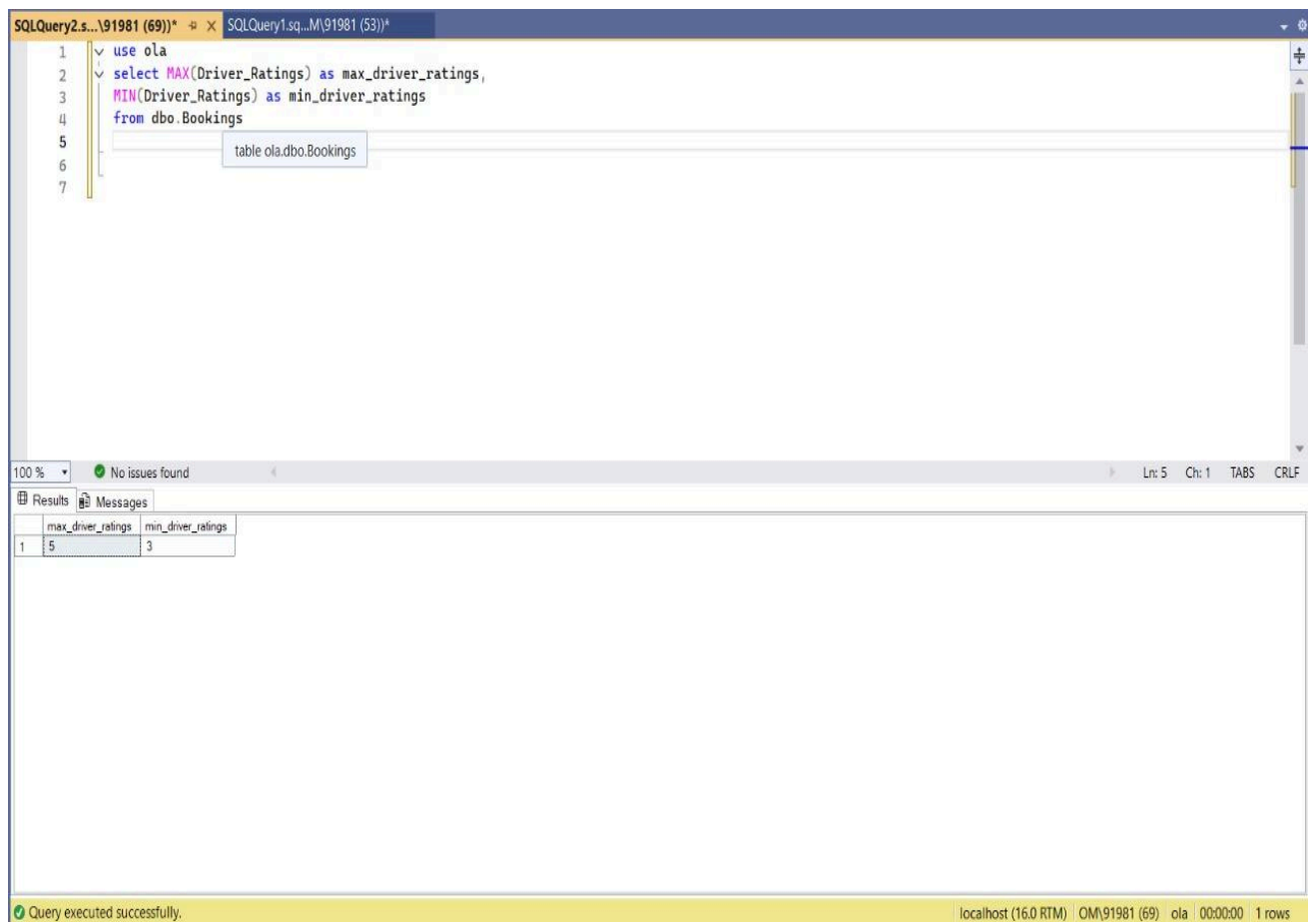
	Date	Time	Booking_ID	Booking_Status	Customer_ID	Vehicle_Type	Pickup_Location	Drop_Location	V_TAT	C_TAT	Canceled_Rides_by_Customer	Canceled_Rides_by_Driver	Incomplete_Rides
1	2024-07-26	14:00:00.0000000	CNR7153255142	Canceled by Driver	CID713523	Prime Sedan	Tumkur Road	RT Nagar	NULL	NULL	null	Personal & Car related issue	NULL
2	2024-07-11	11:12:00.0000000	CNR5374902489	Canceled by Driver	CID735691	Prime Plus	Bannerghatta Road	Sarjapur Road	NULL	NULL	null	Personal & Car related issue	NULL
3	2024-07-18	01:31:00.0000000	CNR6328453219	Canceled by Driver	CID907133	Auto	RT Nagar	Varthur	NULL	NULL	null	Personal & Car related issue	NULL
4	2024-07-17	18:55:00.0000000	CNR3914552212	Canceled by Driver	CID557845	Auto	Hosur Road	Banashankari	NULL	NULL	null	Personal & Car related issue	NULL
5	2024-07-14	11:37:00.0000000	CNR7592165192	Canceled by Driver	CID863704	Prime SUV	Rajarajeshwari Nagar	HSR Layout	NULL	NULL	null	Personal & Car related issue	NULL
6	2024-07-22	00:21:00.0000000	CNR8346388703	Canceled by Driver	CID680143	Mini	Magadi Road	BTM Layout	NULL	NULL	null	Personal & Car related issue	NULL
7	2024-07-09	21:32:00.0000000	CNR7368110421	Canceled by Driver	CID372218	Mini	Peenya	Sarjapur Road	NULL	NULL	null	Personal & Car related issue	NULL
8	2024-07-12	12:38:00.0000000	CNR5043217856	Canceled by Driver	CID298396	Auto	Sarjapur Road	Chamarajpet	NULL	NULL	null	Personal & Car related issue	NULL
9	2024-07-19	23:19:00.0000000	CNR8991153488	Canceled by Driver	CID563608	eBike	Vijayanagar	Kadugodi	NULL	NULL	null	Personal & Car related issue	NULL
10	2024-07-28	10:17:00.0000000	CNR1523677800	Canceled by Driver	CID967088	Prime SUV	Ramamurthy Nagar	Kammanahalli	NULL	NULL	null	Personal & Car related issue	NULL
11	2024-07-27	13:10:00.0000000	CNR4046969463	Canceled by Driver	CID788327	Prime Plus	Koramangala	Devanahalli	NULL	NULL	null	Personal & Car related issue	NULL
12	2024-07-05	01:18:00.0000000	CNR8086928886	Canceled by Driver	CID506037	Mini	Nagarbhavi	MG Road	NULL	NULL	null	Personal & Car related issue	NULL
13	2024-07-01	23:33:00.0000000	CNR6893200456	Canceled by Driver	CID590573	Prime SUV	Whitefield	Rajarajeshwari Nagar	NULL	NULL	null	Personal & Car related issue	NULL
14	2024-07-25	07:33:00.0000000	CNR6855871509	Canceled by Driver	CID750118	Prime Sedan	RT Nagar	Mysore Road	NULL	NULL	null	Personal & Car related issue	NULL
15	2024-07-13	20:08:00.0000000	CNR3824632577	Canceled by Driver	CID713810	Bike	Sahakar Nagar	Koramangala	NULL	NULL	null	Personal & Car related issue	NULL
16	2024-07-01	00:31:00.0000000	CNR5220085842	Canceled by Driver	CID775449	Prime Sedan	Rajajinagar	Bellandur	NULL	NULL	null	Personal & Car related issue	NULL

5. Maximum and Minimum Driver Ratings

Displays the highest and lowest driver ratings from all completed rides.

SQL Query:

```
SELECT MAX(Driver_Ratings) AS max_driver_ratings, MIN(Driver_Ratings) AS  
min_driver_ratings  
FROM dbo.Bookings;
```



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays a query in the 'SQLQuery2.s...' window. The query is as follows:

```
1 use ola
2 select MAX(Driver_Ratings) as max_driver_ratings,
3 MIN(Driver_Ratings) as min_driver_ratings
4 from dbo.Bookings
5
6
7
```

The bottom pane shows the 'Results' tab with a single row of data:

	max_driver_ratings	min_driver_ratings
1	5	3

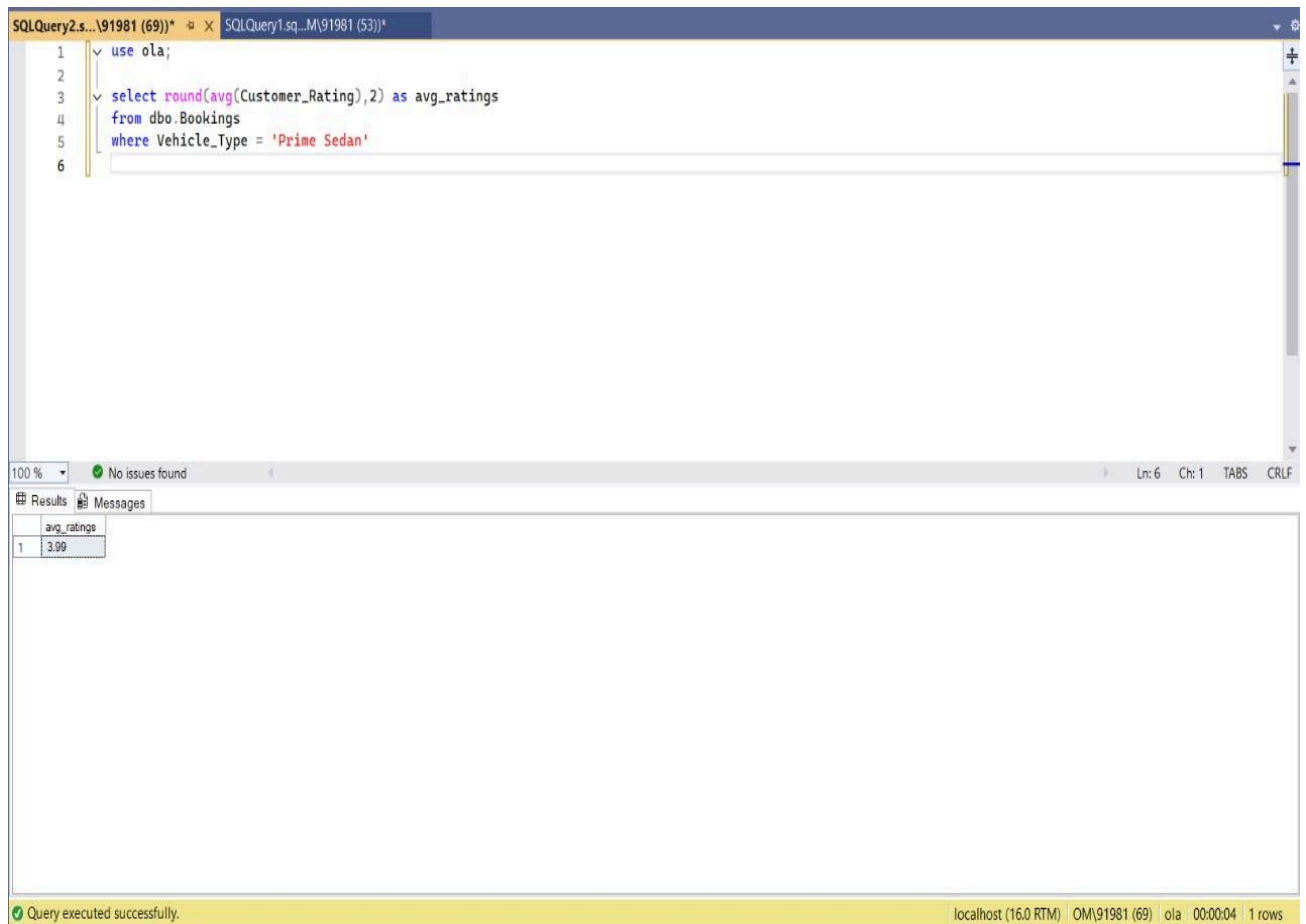
The status bar at the bottom indicates 'Query executed successfully.' and 'localhost (16.0 RTM) | OM\91981 (69) | ola | 00:00:00 | 1 rows'.

6. Average Customer Rating for Prime Sedan

Calculates the average rating given by customers specifically for the 'Prime Sedan' vehicle type.

SQL Query:

```
SELECT ROUND(AVG(Customer_Rating), 2) AS avg_ratings
FROM dbo.Bookings
WHERE Vehicle_Type = 'Prime Sedan';
```



The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a query window with the following SQL code:

```
1 use ola;
2
3 select round(avg(Customer_Rating),2) as avg_ratings
4 from dbo.Bookings
5 where Vehicle_Type = 'Prime Sedan'
6
```

The bottom pane shows the results of the query. The status bar indicates "No issues found". The results are displayed in a table with one row and one column:

avg_ratings
3.99

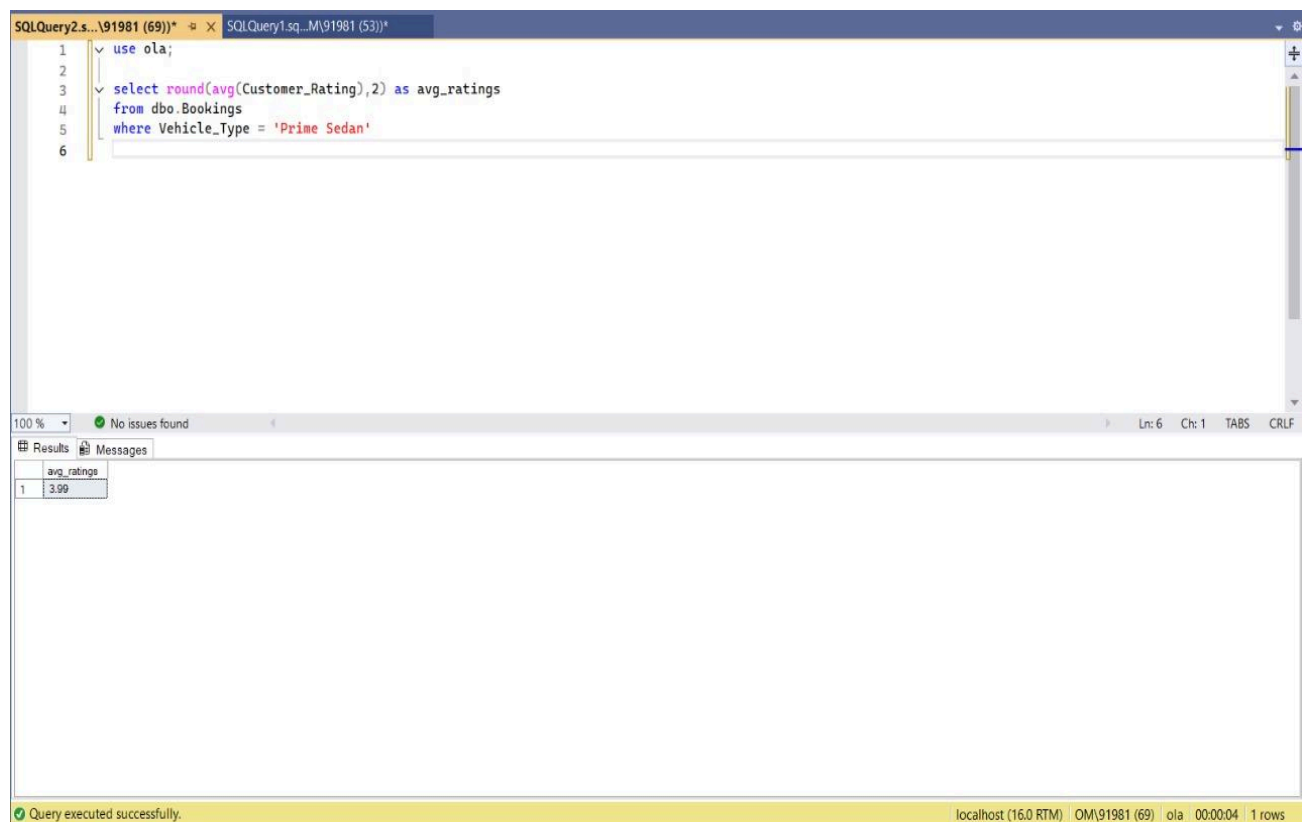
The status bar at the bottom indicates "Query executed successfully." and provides additional details: "localhost (16.0 RTM) OM\91981 (69) ola 00:00:04 1 rows".

7. Bike Rides to Shivajinagar

Lists customers who used a bike and dropped off at Shivajinagar.

SQL Query:

```
SELECT Customer_ID, Vehicle_Type, Drop_Location AS place
FROM dbo.Bookings
WHERE Drop_Location = 'Shivajinagar' AND Vehicle_Type = 'Bike';
```



The screenshot shows a SQL Server Enterprise Manager window with two tabs: 'SQLQuery2.s... \91981 (69))' and 'SQLQuery1.sq...M(91981 (53))'. The active tab displays the following SQL query:

```
1 use ola;
2
3 select round(avg(Customer_Rating),2) as avg_ratings
4 from dbo.Bookings
5 where Vehicle_Type = 'Prime Sedan'
6
```

Below the query editor, the 'Results' tab is selected, showing a single row of data:

avg_ratings
3.99

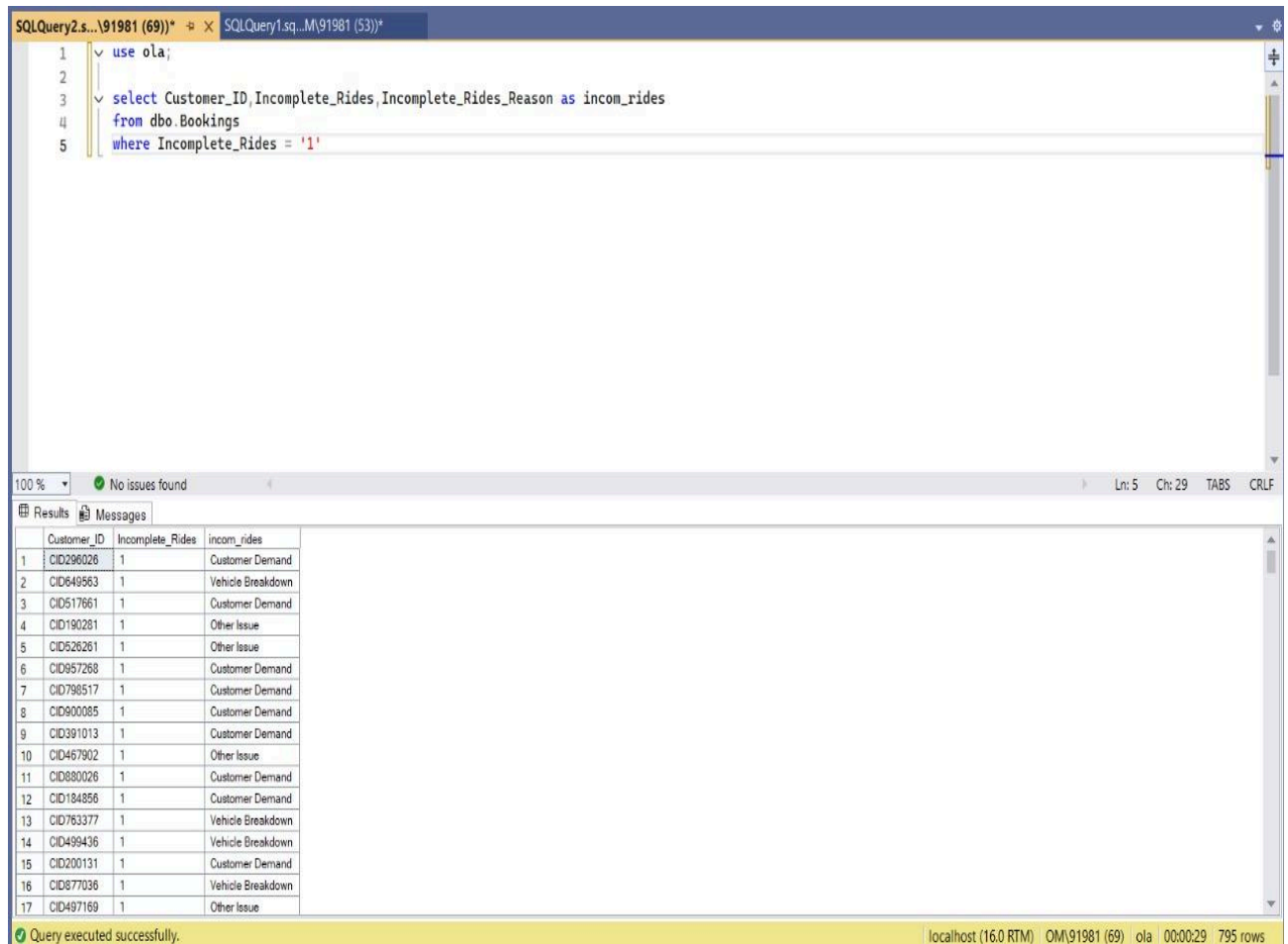
The status bar at the bottom indicates 'Query executed successfully.' and 'localhost (16.0 RTM) OM\91981 (69) ola 00:00:04 1 rows'.

8. Incomplete Rides and Reasons

Shows all incomplete rides along with the reason provided.

SQL Query:

```
SELECT Customer_ID, Incomplete_Rides, Incomplete_Rides_Reason AS incom_rides
FROM dbo.Bookings
WHERE Incomplete_Rides = '1';
```



The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a query window with the following SQL code:

```
1 use ola;
2
3 select Customer_ID, Incomplete_Rides, Incomplete_Rides_Reason as incom_rides
4 from dbo.Bookings
5 where Incomplete_Rides = '1';
```

The bottom pane shows the results of the query, which is a table with three columns: Customer_ID, Incomplete_Rides, and incom_rides. The table contains 17 rows of data.

	Customer_ID	Incomplete_Rides	incom_rides
1	CID296026	1	Customer Demand
2	CID649563	1	Vehicle Breakdown
3	CID517661	1	Customer Demand
4	CID190281	1	Other Issue
5	CID526261	1	Other Issue
6	CID957268	1	Customer Demand
7	CID798517	1	Customer Demand
8	CID900085	1	Customer Demand
9	CID391013	1	Customer Demand
10	CID467902	1	Other Issue
11	CID680026	1	Customer Demand
12	CID184856	1	Customer Demand
13	CID763377	1	Vehicle Breakdown
14	CID499436	1	Vehicle Breakdown
15	CID200131	1	Customer Demand
16	CID877036	1	Vehicle Breakdown
17	CID497169	1	Other Issue

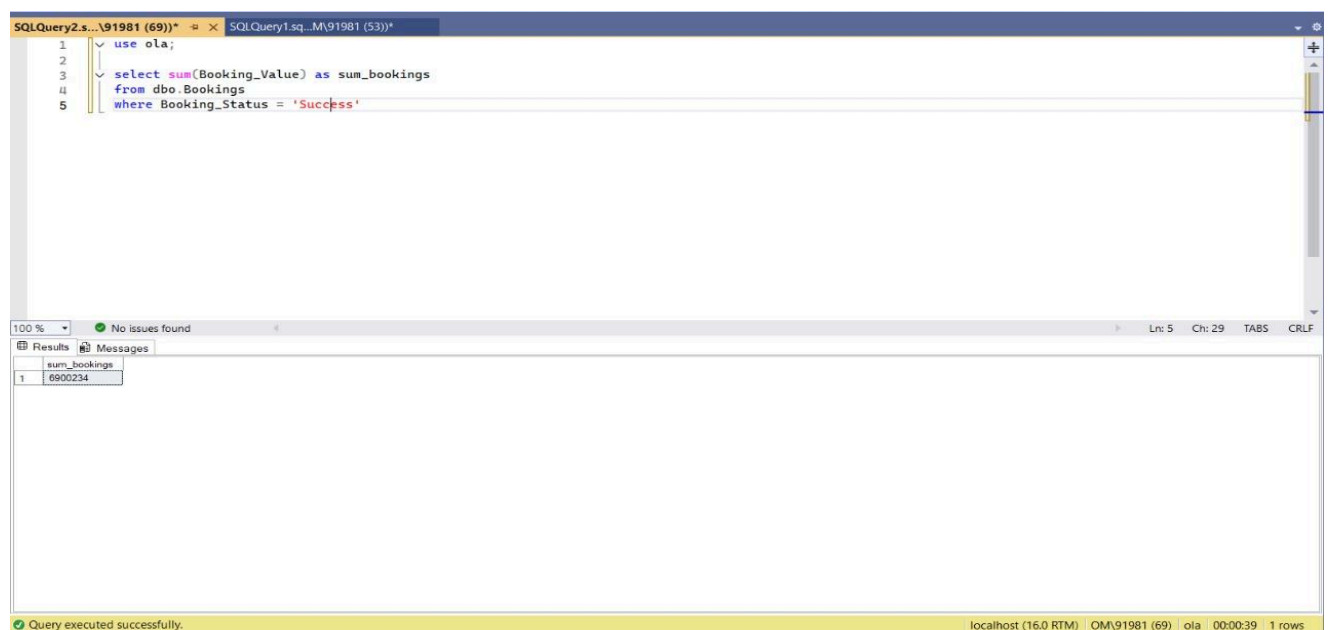
The status bar at the bottom indicates that the query was executed successfully on the 'ola' database, returning 795 rows.

9. Total Revenue from Successful Bookings

Calculates the total revenue generated from all successfully completed bookings.

SQL Query:

```
SELECT SUM(Booking_Value) AS sum_bookings
FROM dbo.Bookings
WHERE Booking_Status = 'Success';
```



Conclusion

The SQL queries executed in this project helped derive meaningful insights about Ola's booking data. They included statistics related to successful and canceled rides, customer behavior, and overall ride ratings.

Report generated by: OM WADIKAR