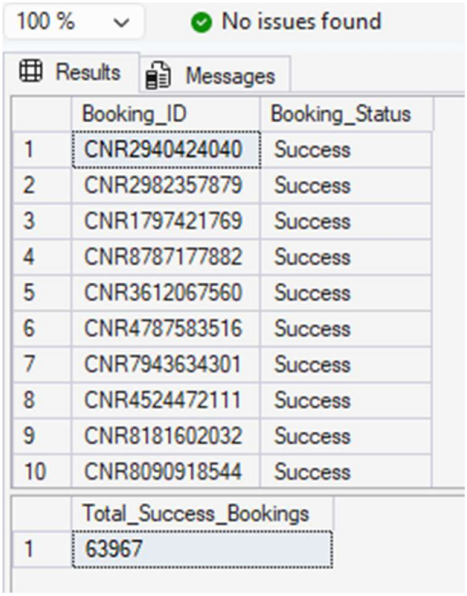
1. Retrieve all successful bookings

```
SELECT Booking_ID, Booking_Status FROM [Uber Bookings]
WHERE Booking_Status = 'Success'
SELECT COUNT(Booking_ID) AS Total_Success_Bookings FROM
[Uber Bookings]
WHERE Booking_Status = 'Success'
```



100 % No issues found

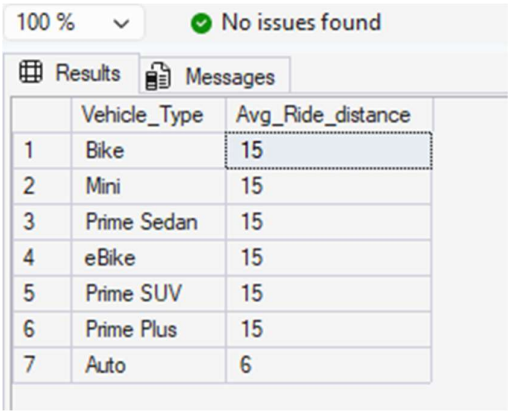
Results Messages

	Booking_ID	Booking_Status
1	CNR2940424040	Success
2	CNR2982357879	Success
3	CNR1797421769	Success
4	CNR8787177882	Success
5	CNR3612067560	Success
6	CNR4787583516	Success
7	CNR7943634301	Success
8	CNR4524472111	Success
9	CNR8181602032	Success
10	CNR8090918544	Success

	Total_Success_Bookings
1	63967

2. Find the average ride distance for each vehicle type

```
SELECT Vehicle_Type, AVG(Ride_Distance) AS
Avg_Ride_distance FROM [Uber Bookings]
GROUP BY Vehicle_Type
```



100 % No issues found

Results Messages

	Vehicle_Type	Avg_Ride_distance
1	Bike	15
2	Mini	15
3	Prime Sedan	15
4	eBike	15
5	Prime SUV	15
6	Prime Plus	15
7	Auto	6

3. Get the total number of cancelled rides by customers

```
SELECT Booking_ID, Booking_Status FROM [Uber Bookings]
WHERE Booking_Status = 'Canceled by Customer'
SELECT COUNT(Booking_Status) FROM [Uber Bookings]
WHERE Booking_Status = 'Canceled by Customer'
```

100 % No issues found

Results Messages

	Booking_ID	Booking_Status
1	CNR2395710036	Canceled by Customer
2	CNR6003663433	Canceled by Customer
3	CNR7286474506	Canceled by Customer
4	CNR7533317866	Canceled by Customer
5	CNR3780555542	Canceled by Customer
6	CNR8014852782	Canceled by Customer
7	CNR5508853708	Canceled by Customer
8	CNR1903850514	Canceled by Customer
9	CNR1281921513	Canceled by Customer
10	CNR8072285783	Canceled by Customer

	(No column name)
1	10499

4. List the top 5 customers who booked the highest number of rides

```
SELECT TOP 5 Customer_ID, COUNT(Booking_ID) AS
Total_Bookings, AVG(Booking_Value) AS Paid_Price FROM
[Uber Bookings]
GROUP BY Customer_ID
ORDER BY COUNT(Booking_ID) DESC,AVG(Booking_Value) DESC
```

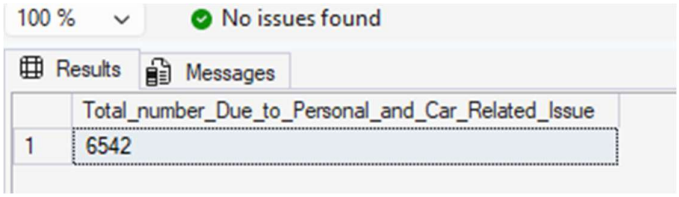
100 % 2 0

Results Messages

	Customer_ID	Total_Bookings	Paid_Price
1	CID954071	5	404
2	CID836942	4	1504
3	CID387617	4	1003
4	CID309168	4	958
5	CID635963	4	710

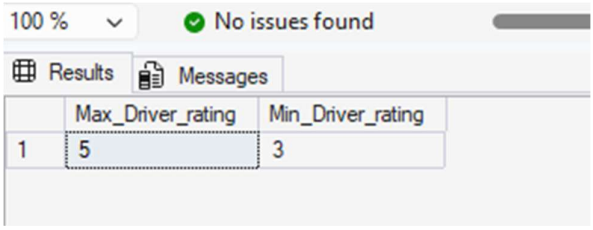
5. Get the number of rides cancelled by drivers due to personal and car-related issues

```
SELECT COUNT(Canceled_Rides_by_Driver) AS
Total_number_Due_to_Personal_and_Car_Related_Issue FROM
[Uber Bookings]
WHERE Canceled_Rides_by_Driver = 'Personal & Car
related    issue'
```



6. Find the maximum and minimum driver ratings for Prime Sedan bookings

```
SELECT MAX(CAST(Driver_Ratings AS FLOAT)) AS
Max_Driver_rating, MIN(CAST(Driver_Ratings AS FLOAT))
AS Min_Driver_rating FROM [Uber Bookings]
WHERE Driver_Ratings <> 'null' AND Vehicle_Type =
'Prime   Sedan'
```



7. Retrieve all rides where payment was made using UPI

```
SELECT * FROM [Uber Bookings]
SELECT Booking_ID,Payment_Method FROM [Uber Bookings]
WHERE Payment_Method = 'UPI'
```

```sql
SELECT COUNT(Payment_Method) AS
Total_no_of_Payment_by_UPI FROM [Uber Bookings]
WHERE Payment_Method = 'UPI'
```

100 % No issues found

Results Messages

	Booking_ID	Payment_Method
1	CNR2982357879	UPI
2	CNR8787177882	UPI
3	CNR4524472111	UPI
4	CNR8181602032	UPI
5	CNR8090918544	UPI
6	CNR9975925287	UPI
7	CNR4443921904	UPI
8	CNR7194303296	UPI
9	CNR6494005067	UPI
10	CNR7142279862	UPI

	Total_no_of_Payment_by_UPI
1	25881

8. Find the average customer rating per vehicle type

```sql
SELECT Vehicle_Type, CAST(SUM(CAST(Customer_Rating AS
FLOAT))/(COUNT(Customer_Rating)) AS DECIMAL(10,2)) AS
Avg_Rating FROM [Uber Bookings]
WHERE Customer_Rating <> 'null'
GROUP BY Vehicle_Type
```

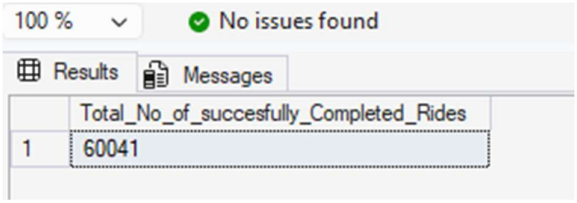
100 % No issues found

Results Messages

	Vehicle_Type	Avg_Rating
1	Bike	3.99
2	Mini	4.00
3	Prime Sedan	4.00
4	eBike	3.99
5	Prime SUV	4.00
6	Prime Plus	4.01
7	Auto	4.00

9. Calculate the total booking value of rides completed successfully

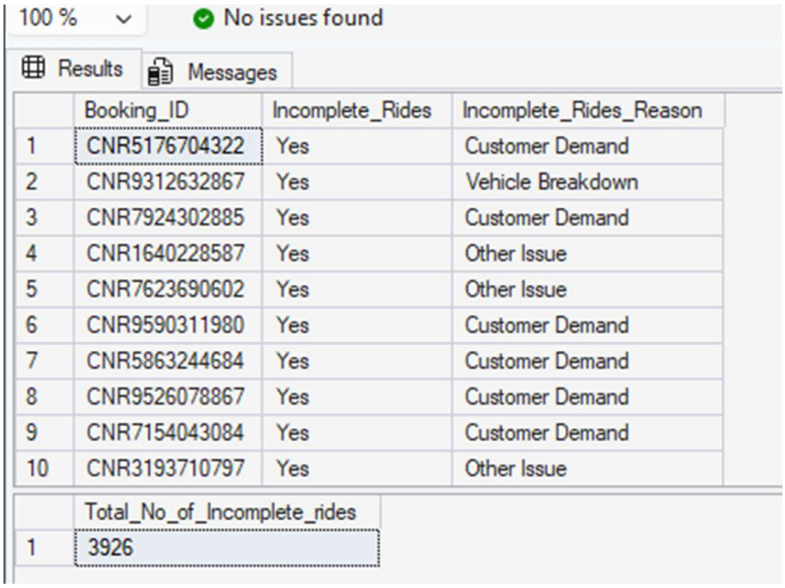
```
SELECT COUNT(Incomplete_Rides) AS
Total_No_of_succesfully_Completed_Rides FROM [Uber
Bookings]
WHERE Incomplete_Rides = 'No'
```



	Total_No_of_succesfully_Completed_Rides
1	60041

10. List all incomplete rides along with the reason

```
SELECT Booking_ID, Incomplete_Rides,
Incomplete_Rides_Reason FROM [Uber Bookings]
WHERE Incomplete_Rides = 'Yes'
SELECT COUNT(Incomplete_Rides) AS
Total_No_of_Incomplete_rides FROM [Uber Bookings]
 WHERE Incomplete_Rides = 'Yes'
```



	Booking_ID	Incomplete_Rides	Incomplete_Rides_Reason
1	CNR5176704322	Yes	Customer Demand
2	CNR9312632867	Yes	Vehicle Breakdown
3	CNR7924302885	Yes	Customer Demand
4	CNR1640228587	Yes	Other Issue
5	CNR7623690602	Yes	Other Issue
6	CNR9590311980	Yes	Customer Demand
7	CNR5863244684	Yes	Customer Demand
8	CNR9526078867	Yes	Customer Demand
9	CNR7154043084	Yes	Customer Demand
10	CNR3193710797	Yes	Other Issue

	Total_No_of_Incomplete_rides
1	3926