Ola Project from Top Versity

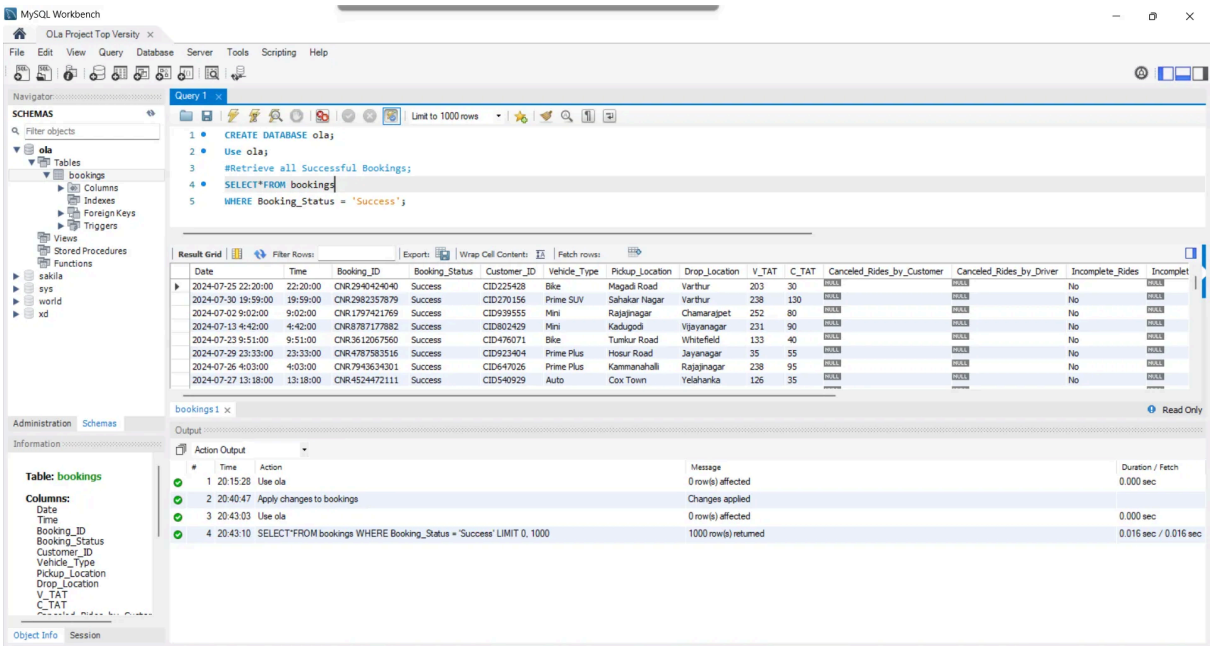
⇒ Cleaning and Filtering (Better on Google SpreadSheet)

⇒ Exporting data as CSV for Analysing on SQL

⇒ Adding data on SQL (Create DB (Ola) ⇒ Table (Right Click) ⇒ Table data import Wizard)

Qns(1) ⇒ Retrieve all Successful Bookings

```
Create View Successful_Bookings As
SELECT*FROM bookings
WHERE Booking_Status = 'Success';
Select *From Successful_Bookings;
```



What is a View in SQL?

A **view** in SQL is a virtual table that is based on the result of a SQL query. Unlike a regular table, a view does not store data itself; instead, it displays the data stored in other tables.

Views are created using the `CREATE VIEW` statement and can be queried just like a regular table.

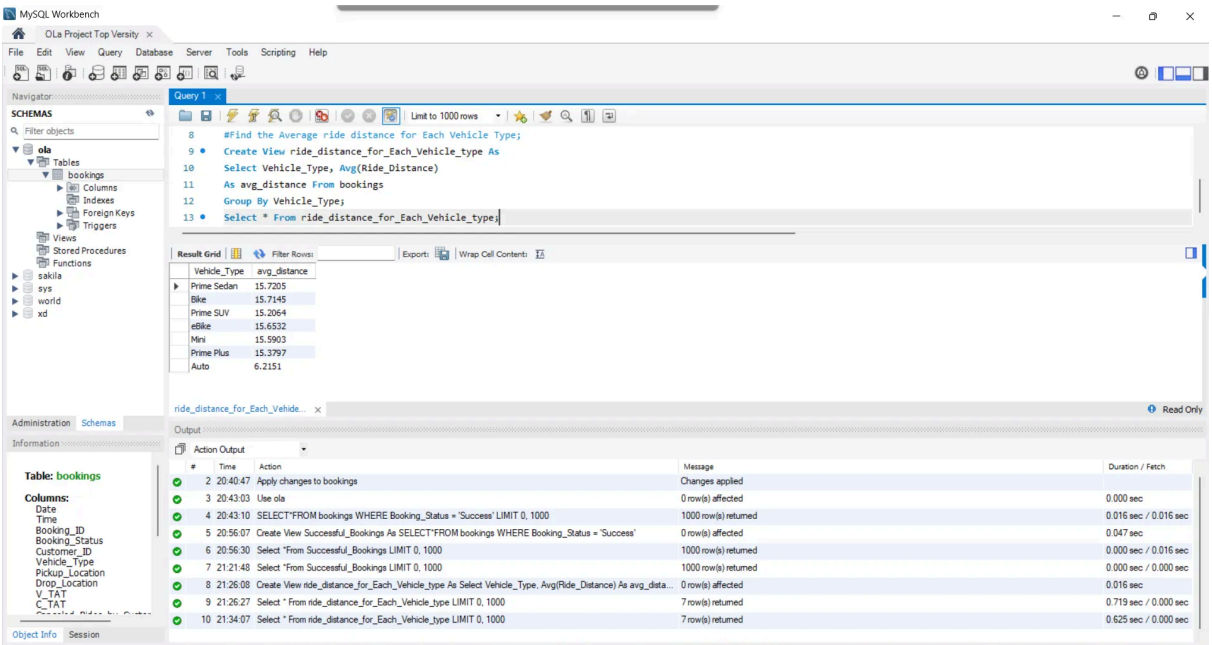
Role of Views in Data Analysis

Views play a significant role in data analysis by:

1. **Simplifying Complex Queries**: They can encapsulate complex SQL queries, making them reusable and easier to understand.
2. **Data Security**: Views can restrict access to specific columns or rows, exposing only the required data to users.
3. **Data Abstraction**: Views provide a layer of abstraction, so users don't need to know the underlying table structures.
4. **Aggregation and Summarization**: Views can summarize data (e.g., using `GROUP BY`) for analytical purposes.
5. **Enhancing Performance**: Although views themselves don't enhance performance, materialized views (a special type) can store query results for faster access.

#Qns(2) ⇒ Find the Average ride distance for Each Vehicle Type:

```
Create View ride_distance_for_Each_Vehicle_type As
Select Vehicle_Type, Avg(Ride_Distance)
As avg_distance From bookings
Group By Vehicle_Type;
Select * From ride_distance_for_Each_Vehicle_type;
```



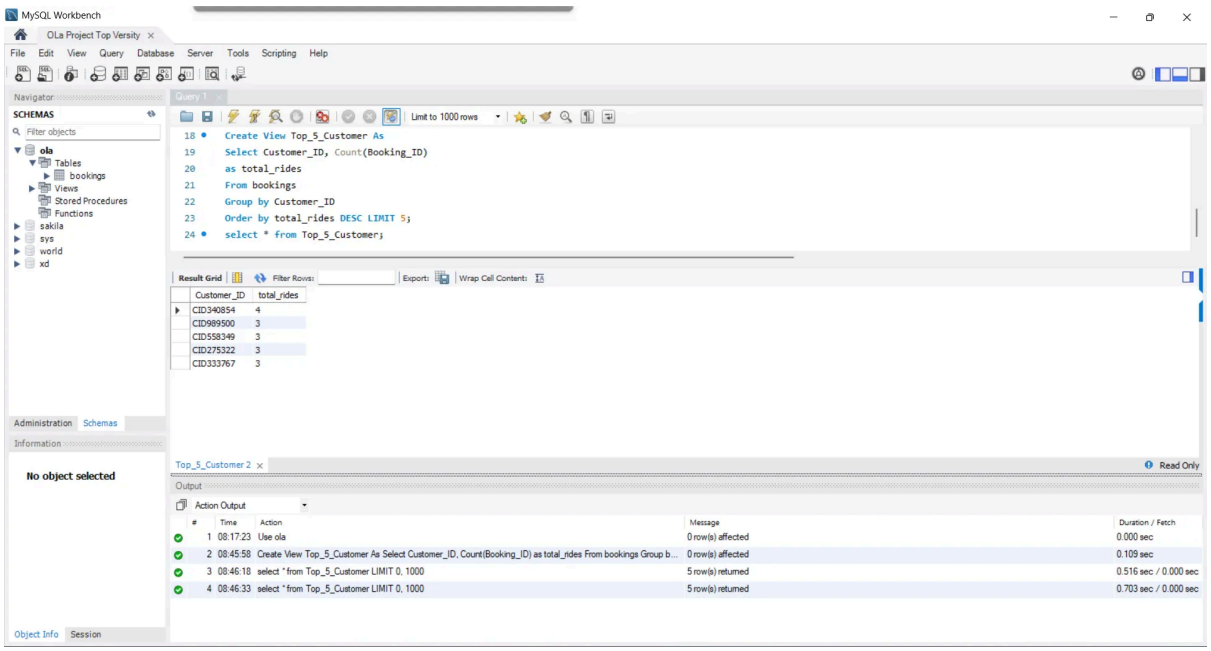
Qns(3) ⇒ Get the total number of Canceled Rides by Customers:

```
Select Count(*) From bookings
Where Booking_Status = 'Canceled by Customer';
```

Qns(4) ⇒ List the top 5 customers who booked the highest number of rides

#top 5 customers who booked the highest number of rides

```
Create View Top_5_Customer As
Select Customer_ID, Count(Booking_ID)
as total_rides
From bookings
Group by Customer_ID
Order by total_rides DESC LIMIT 5;
select * from Top_5_Customer;
```



- **`Create View Top_5_Customer As`** :
  - This creates a **view** named `Top_5_Customer`. A view is a virtual table based on the result set of a SQL query. It does not store data physically but provides a way to save and reuse queries.
- **`Select Customer_ID, Count(Booking_ID) as total_rides`** :
  - **`Customer_ID`** : Selects the customer identifier from the `bookings` table.
  - **`Count(Booking_ID) as total_rides`** : Counts the number of bookings (`Booking_ID`) for each customer and labels this count as `total_rides` .
- **`From bookings`** :

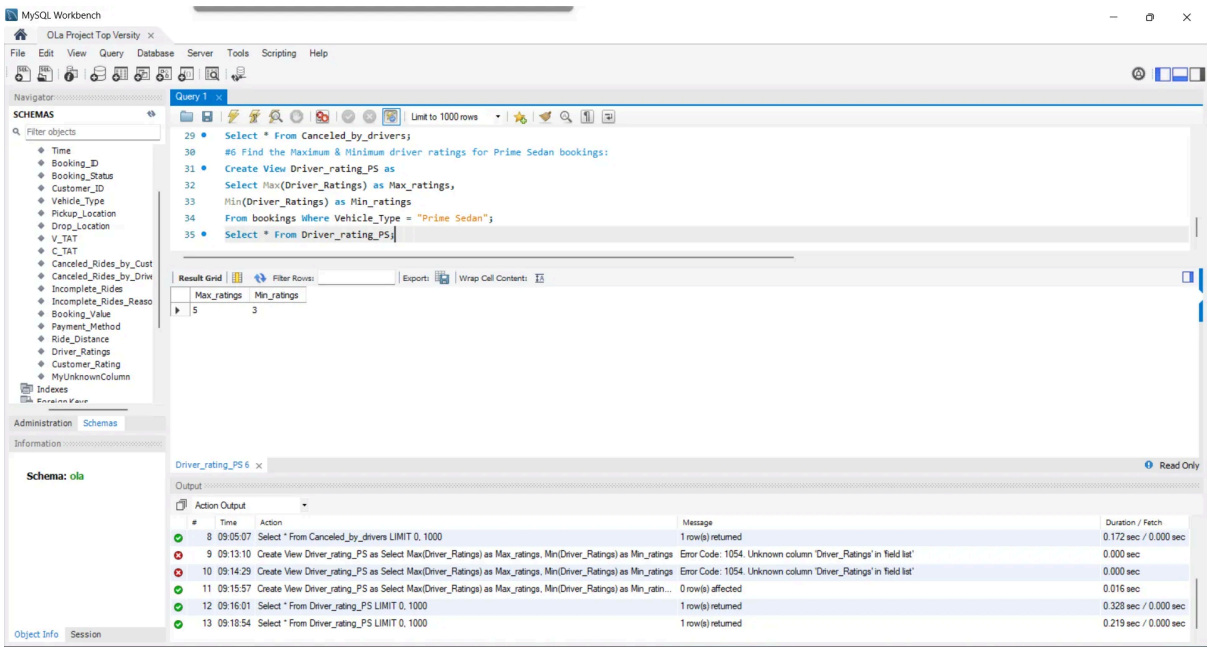
    - Specifies the source table, which is `bookings`.
- **`Group by Customer_ID`**:
    - Groups the data by `Customer_ID`, ensuring that the count of bookings (`Booking_ID`) is calculated per customer.
- **`Order by total_rides DESC`**:
    - Orders the grouped results in descending order based on the total number of rides (`total_rides`), so customers with the highest number of rides appear first.
- **`LIMIT 5`**:
    - Restricts the output to only the top 5 rows, effectively returning the top 5 customers with the most rides.

Qns(5)⇒ Get the number of rides canceled by drivers due to personal and car-related issues:

```
Create View Canceled_by_drivers As
SELECT Count(*) From bookings
Where Canceled_Rides_by_driver = 'Personal & Car related issue';
Select * From Canceled_by_drivers;
```

Qns(6)⇒ Find the Maximum & Minimum driver ratings for Prime Sedan bookings:

```
Create View Driver_rating_PS as
Select Max(Driver_Ratings) as Max_ratings,
Min(Driver_Ratings) as Min_ratings
From bookings Where Vehicle_Type = "Prime Sedan";
Select * From Driver_rating_PS;
```

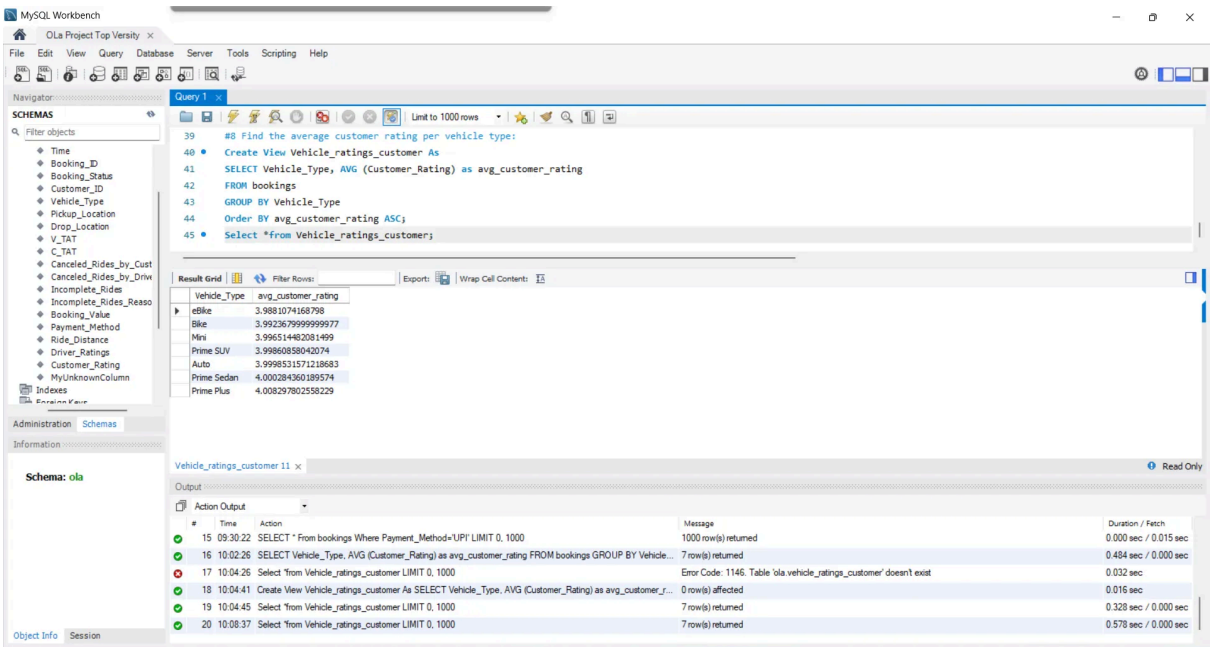


Qns(7)⇒ Retrieve all rides where payment was made using UPI:

```
SELECT * FROM bookings
WHERE Payment Method = 'UPI';
```

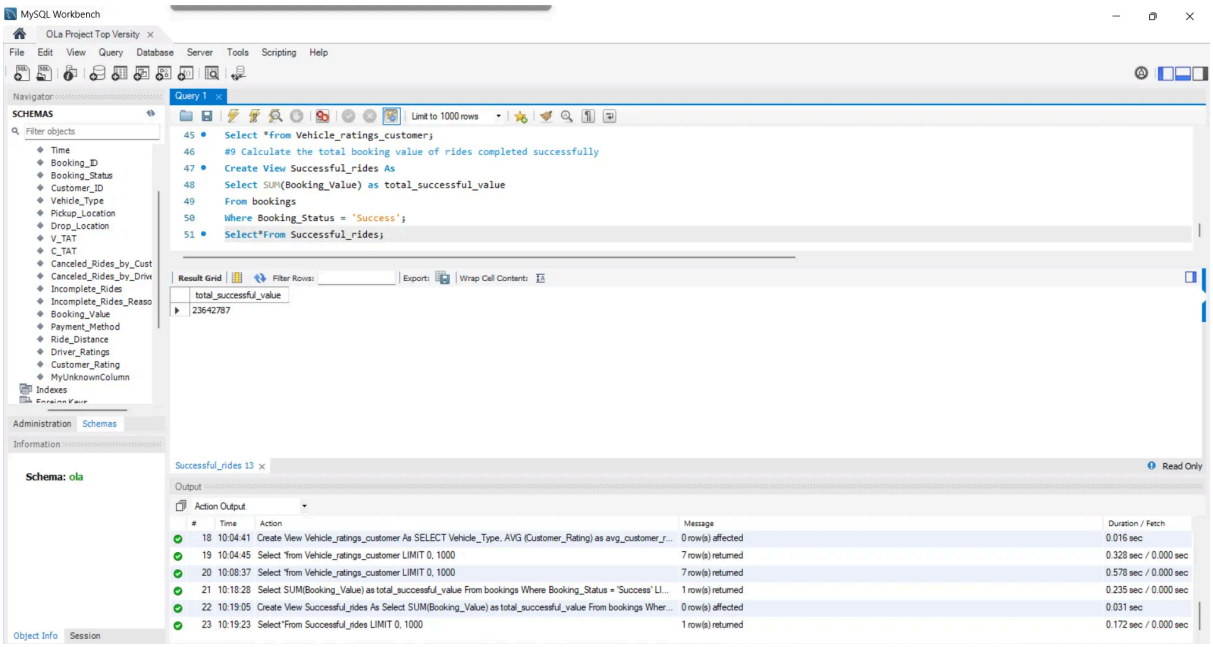
#Qns(8)⇒ Find the average customer rating per vehicle type:

```
Create View Vehicle_ratings_customer As
SELECT Vehicle_Type, AVG (Customer_Rating) as avg_customer_rating
FROM bookings
GROUP BY Vehicle_Type
Order BY avg_customer_rating ASC;
Select *from Vehicle_ratings_customer;
```



#Qns(9)⇒ Calculate the total booking value of rides completed successfully

```
Create View Successful_rides As
Select SUM(Booking_Value) as total_successful_value
From bookings
Where Booking_Status = 'Success';
Select*From Successful_rides;
```



#Qns(10)⇒ List all incomplete rides along with the reason:

```
CREATE VIEW incomplete_rides_list As
SELECT Booking_ID, Incomplete_Rides_Reason
FROM bookings
Where Incomplete_Rides = 'Yes';
SELECT*FROM incomplete_rides_list;
```