OLA Data Analyst Project

SQL Questions: OLA Data Analyst Project

1. Retrieve all successful bookings:

2. Find the average ride distance for each vehicle type:

3. Get the total number of cancelled rides by customers:

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

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

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

7. Retrieve all rides where payment was made using UPI:

8. Find the average customer rating per vehicle type:

9. Calculate the total booking value of rides completed successfully:

10. List all incomplete rides along with the reason

OLA Data Analyst Project

SQL Answers:

1. **# 1. Retrieve all successful bookings:**

```
SELECT*FROM Successful_Bookings;
```

2. **# 2. Find the average ride distance for each vehicle type:**

```
SELECT*FROM ride_distance_for_each_vehicle;
```

3. **# 3. Get the total number of cancelled rides by customers**

```
SELECT*FROM Cancelled_rides_by_customers;
```

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

```
SELECT*FROM top_5_customers;
```

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

```
SELECT*FROM rides_cancelled_by_drivers_P_C_ISSUES;
```

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

```
Select*from Max_Min_Ratings;
```

7. **#7. Retrieve all rides where payment was made using UPI:**

```
SELECT*FROM UPI_Payment;
```

8. **# 8. Find the average customer rating per vehicle type:**

```
SELECT*FROM Avg_customer_rating;
```

9. **# 9. Calculate the total booking value of rides completed successfully:**

```
SELECT*FROM Total_successfully_rides_value;
```

10. **#10. List all incomplete rides along with the reason:**

```
SELECT*FROM Incomplete_Rides_Reason;
```

OLA Data Analyst Project

SQL Questions & Answers

```
CREATE DATABASE ola;

USE OLA;

SELECT * FROM ola.`bookings-100000-rows`;
```

# 1. Retrieve all successful bookings:

```
CREATE VIEW Successful_Bookings AS

SELECT * FROM ola.`bookings-100000-rows`

WHERE Booking_Status="success";
```

# 2. Find the average ride distance for each vehicle type:

```
 CREATE VIEW ride_distance_for_each_vehicle as

 SELECT vehicle_type,avg(Ride_Distance)

 as avg_distance FROM `bookings-100000-rows`

 group by vehicle_type;
```

# 3. Get the total number of cancelled rides by customers:

```
  CREATE VIEW Cancelled_rides_by_customers as

 SELECT COUNT(*) FROM `bookings-100000-rows`

 WHERE Booking_Status="Canceled by customers";
```

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

```
 CREATE VIEW top_5_customers as

 SELECT CUSTOMER_ID, COUNT(BOOKING_ID) AS TOTAL_RIDES
```

```
      FROM `bookings-100000-rows`
      GROUP BY CUSTOMER_ID
      ORDER BY  TOTAL_RIDES DESC LIMIT 5;
```

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

```
 CREATE VIEW rides_cancelled_by_drivers_P_C_ISSUES AS
 SELECT COUNT(*)FROM `bookings-100000-rows`
 WHERE canceled_rides_by_driver='personal and car-related issue';
```

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

```
 CREATE view Max_Min_Ratings as
  SELECT MAX(Driver_Ratings) as Max_rating,
  Min(Driver_Ratings) as min_rating
  FROM `bookings-100000-rows` WHERE Vehicle_type='prime sedan';
```

#7. Retrieve all rides where payment was made using UPI:

```
  CREATE VIEW UPI_Payment as
  SELECT*FROM `bookings-100000-rows`
  WHERE Payment_Method='UPI';
```

# 8. Find the average customer rating per vehicle type:

```
   CREATE view Avg_customer_rating as
   SELECT vehicle_Type,avg(customer_Rating ) AS avg_customer_rating
   FROM `bookings-100000-rows`
   group by vehicle_Type;
```

# 9. Calculate the total booking value of rides completed successfully:

```
  CREATE view Total_successfully_rides_value AS
```

```
SELECT SUM(Booking_value) as Total_successfully_rides_value

FROM `bookings-100000-rows`

WHERE Booking_status='Success';
```

#10. List all incomplete rides along with the reason:

```
CREATE VIEW Incomplete_Rides_Reason as

SELECT Booking_id, Incomplete_Rides_Reason

FROM `bookings-100000-rows`

WHERE  Incomplete_Rides='YES';
```

Data Columns

1. Date

2. Time

3. Booking_ID

4. Booking_Status

5. Customer_ID

6. Vehicle_Type

7. Pickup_Location

8. Drop_Location

9. V_TAT

10. C_TAT

11. cancelled_Rides_by_Customer

12. cancelled_Rides_by_Driver

13. Incomplete_Rides

14. Incomplete_Rides_Reason

15. Booking_Value

16. Payment_Method

17. Ride_Distance

18. Driver_Ratings

19. Customer_Rating