Booking data analysis SQl

1. Retrieve all successful bookings, sorted by booking value in descending order.

```
Select booking_status, booking_value from bookings

where booking_status = 'Success'

order by booking_value Desc;
```

2. Find the average ride distance and total ride distance for each vehicle type.

```
Select Vehicle_type, ROUND(CAST(AVG(Ride_Distance) AS NUMERIC), 2) AS AVG_Distance,
Sum(ride_distance) as Total_distance

from bookings

group by vehicle_type

order by vehicle_type;
```

3. Calculate the cancellation rate for customers and drivers.

```
SELECT

 Round(SUM(CASE WHEN Booking_Status = 'Canceled by Customer' THEN 1 ELSE 0 END) * 100.0 /
COUNT(*) , 2) AS Customer_Cancellation_Rate,

 Round(SUM(CASE WHEN Booking_Status = 'Canceled by Driver' THEN 1 ELSE 0 END) * 100.0 /
COUNT(*) , 2) AS Driver_Cancellation_Rate

FROM

 bookings;

---Count of each

        SELECT

 SUM(CASE WHEN Booking_Status = 'Canceled by Customer' THEN 1 ELSE 0 END) AS
Customer_Cancellation_Rate,

 SUM(CASE WHEN Booking_Status = 'Canceled by Driver' THEN 1 ELSE 0 END) AS
Driver_Cancellation_Rate

FROM

 bookings;
```

4. Identify the most frequently used pickup locations.(same for drop location)

```
Select pickup_location, count(*) as pick from bookings
group by pickup_location
order by pick desc
limit 10;
```

5. Find the top 3 vehicle types with the highest average booking value.

```
Select vehicle_type, Round(Avg(booking_value), 0) as Average_value from bookings
group by vehicle_type
order by Average_value desc
limit 3;
```

6. Calculate the percentage of rides completed successfully for each payment method.

```
SELECT payment_method,
ROUND(COUNT(*) * 100.0 / (SELECT COUNT(*) FROM bookings WHERE booking_status = 'Success'), 2)
AS success_percentage
FROM bookings
WHERE booking_status = 'Success'
GROUP BY payment_method;
```

7. Retrieve all rides canceled by drivers, grouped by reason.

```
Select canceled_rides_by_driver, count(booking_id) as cancel_count from bookings
where booking_status = 'Canceled by Driver'
group by canceled_rides_by_driver
order by cancel_count desc
```

8. Identify the vehicle type with the highest cancellation rate.

```
Select vehicle_type, count(booking_status) as cancel_count from bookings
where booking_status = 'Canceled by Customer' or booking_status = 'Canceled by Driver'
group by Vehicle_type
order by cancel_count DESC;
```

```
WITH cancellation_data AS (
  SELECT
    vehicle_type,
    COUNT(*) AS total_bookings,
    SUM(CASE WHEN booking_status IN ('Canceled by Customer', 'Canceled by Driver') THEN 1 ELSE 0 END) AS canceled_bookings
  FROM bookings
  GROUP BY vehicle_type
)
SELECT
  vehicle_type,
  canceled_bookings AS cancel_count,
  ROUND((canceled_bookings * 100.0 / total_bookings), 2) AS cancel_rate
FROM cancellation_data
ORDER BY cancel_rate DESC;
```

```
SELECT
  vehicle_type,
  COUNT(CASE WHEN booking_status IN ('Canceled by Customer', 'Canceled by Driver') THEN 1 END) AS cancel_count,
  COUNT(*) AS total_count,
  ROUND(COUNT(CASE WHEN booking_status IN ('Canceled by Customer', 'Canceled by Driver') THEN 1 END) * 100.0 / COUNT(*), 2) AS cancel_rate
FROM bookings
```

```
GROUP BY vehicle_type
ORDER BY cancel_rate DESC
LIMIT 1;
```

----------(this one for cancelation rate for vehicle with total)

```
WITH total_canceled_rides AS (
    SELECT COUNT(*) AS total_canceled
    FROM bookings
    WHERE booking_status IN ('Canceled by Customer', 'Canceled by Driver')
        )
SELECT vehicle_type,
    COUNT(*) AS canceled_rides,
    ROUND(COUNT(*) * 100.0 / (SELECT total_canceled FROM total_canceled_rides), 2) AS
cancellation_rate
FROM bookings
WHERE booking_status IN ('Canceled by Customer', 'Canceled by Driver')
GROUP BY vehicle_type
ORDER BY cancellation_rate DESC;
```

9. List all bookings where the booking value exceeds the average value for its vehicle type.

```
WITH avg_booking_value AS (
    SELECT vehicle_type, AVG(booking_value) AS avg_value
    FROM bookings
    GROUP BY vehicle_type
)
SELECT b.*
FROM bookings b
JOIN avg_booking_value abv
ON b.vehicle_type = abv.vehicle_type
WHERE b.booking_value > abv.avg_value;
```

10. Calculate the total booking value and number of rides for each payment method.

```
Select payment_method, count(*) , Sum(booking_value) as Total_booking from bookings
group by payment_method
order by total_booking;
```

11. Identify the busiest days of the week for successful bookings.

```
        WITH date_cte AS (
   SELECT
      date::DATE AS booking_date,
      TO_CHAR(date, 'Day') AS day_of_week
   FROM generate_series('2024-06-01'::DATE, '2024-07-31'::DATE, '1 day'::INTERVAL) AS s(date)
)
SELECT
   d.day_of_week,
   COUNT(b.booking_id) AS total_successful_bookings
FROM date_cte d
LEFT JOIN bookings b ON d.booking_date = b.date
WHERE b.booking_status = 'Success'
GROUP BY d.day_of_week
ORDER BY total_successful_bookings DESC;
```

12. Compare the average booking value for UPI payments versus cash payments.

```
select payment_method, round(avg(booking_value),2) from bookings
where payment_method in ('UPI', 'Cash')
group by payment_method
```

----------if want different column----------

```
SELECT
  Round(AVG(CASE WHEN payment_method = 'UPI' THEN booking_value END), 2) AS upi_average,
  Round(AVG(CASE WHEN payment_method = 'Cash' THEN booking_value END), 2) AS cash_average
FROM bookings
WHERE payment_method IN ('UPI', 'Cash');
```

13. Identify trends in booking status over time (e.g., monthly or weekly trends).

```
SELECT
  DATE_TRUNC('week', date) AS weeks,
  booking_status,
  COUNT(*) AS total_bookings
FROM bookings
GROUP BY DATE_TRUNC('week', date), booking_status
ORDER BY weeks, booking_status;
```

14. Calculate the average ride distance for rides canceled due to driver-related issues.(it will show 0 as per our data since for canceled rodes we donot have ride distance value)

```
Select booking_status, Avg(ride_distance) from bookings
where booking_status = 'Canceled by Driver'
group by booking_status
```