SQL-Case Study

Submitted By-

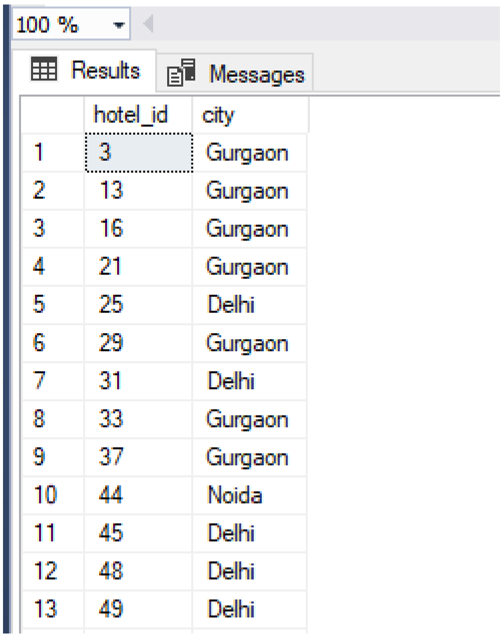
Subrat Shukla, DE Batch1

Creating an 'OYO_Business' database-

```
create database OYO_Business;
use OYO_Business;
```

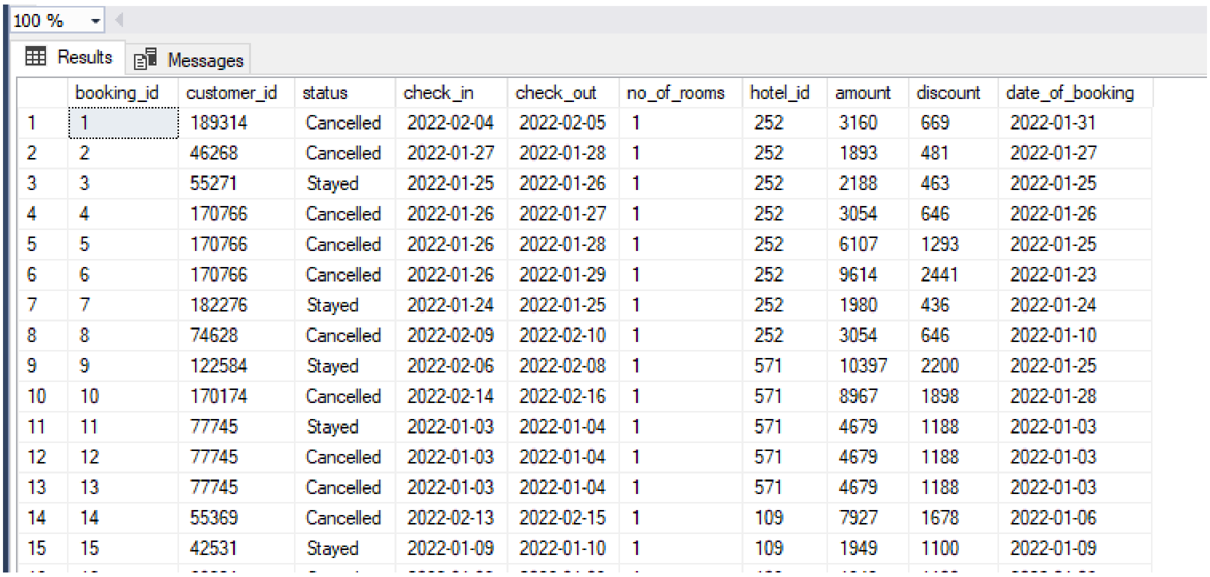
Overview of database tables-

```
select *from [dbo].[Oyo_City_CSV]
```



	hotel_id	city
1	3	Gurgaon
2	13	Gurgaon
3	16	Gurgaon
4	21	Gurgaon
5	25	Delhi
6	29	Gurgaon
7	31	Delhi
8	33	Gurgaon
9	37	Gurgaon
10	44	Noida
11	45	Delhi
12	48	Delhi
13	49	Delhi

```
select *from [dbo].[Oyo_Sales_CSV]
```



	booking_id	customer_id	status	check_in	check_out	no_of_rooms	hotel_id	amount	discount	date_of_booking
1	1	189314	Cancelled	2022-02-04	2022-02-05	1	252	3160	669	2022-01-31
2	2	46268	Cancelled	2022-01-27	2022-01-28	1	252	1893	481	2022-01-27
3	3	55271	Stayed	2022-01-25	2022-01-26	1	252	2188	463	2022-01-25
4	4	170766	Cancelled	2022-01-26	2022-01-27	1	252	3054	646	2022-01-26
5	5	170766	Cancelled	2022-01-26	2022-01-28	1	252	6107	1293	2022-01-25
6	6	170766	Cancelled	2022-01-26	2022-01-29	1	252	9614	2441	2022-01-23
7	7	182276	Stayed	2022-01-24	2022-01-25	1	252	1980	436	2022-01-24
8	8	74628	Cancelled	2022-02-09	2022-02-10	1	252	3054	646	2022-01-10
9	9	122584	Stayed	2022-02-06	2022-02-08	1	571	10397	2200	2022-01-25
10	10	170174	Cancelled	2022-02-14	2022-02-16	1	571	8967	1898	2022-01-28
11	11	77745	Stayed	2022-01-03	2022-01-04	1	571	4679	1188	2022-01-03
12	12	77745	Cancelled	2022-01-03	2022-01-04	1	571	4679	1188	2022-01-03
13	13	77745	Cancelled	2022-01-03	2022-01-04	1	571	4679	1188	2022-01-03
14	14	55369	Cancelled	2022-02-13	2022-02-15	1	109	7927	1678	2022-01-06
15	15	42531	Stayed	2022-01-09	2022-01-10	1	109	1949	1100	2022-01-09

Given Insights-

1. Banglore , gurgaon & delhi were popular in the bookings, whereas Kolkata is less popular in bookings
2. Nature of Bookings:

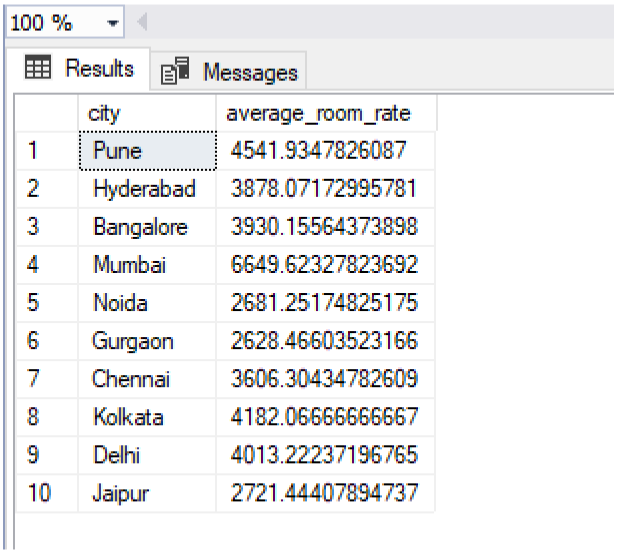
   • Nearly 50 % of the bookings were made on the day of check in only.
   • Nearly 85 % of the bookings were made with less than 4 days prior to the date of check in.
   • Very few no.of bookings were made in advance(i.e over a 1 month or 2 months).
   • Most of the bookings involved only a single room.
   • Nearly 80% of the bookings involved a stay of 1 night only.

3. Oyo should acquire more hotels in the cities of Pune, Kolkata & Mumbai. Because their average room rates are comparatively higher so more revenue will come.
4. The % cancellation Rate is high on all 9 cities except pune , so Oyo should focus on finding reasons about cancellation.

SQL Queries to find-

1. Average Room Rates of Different Cities

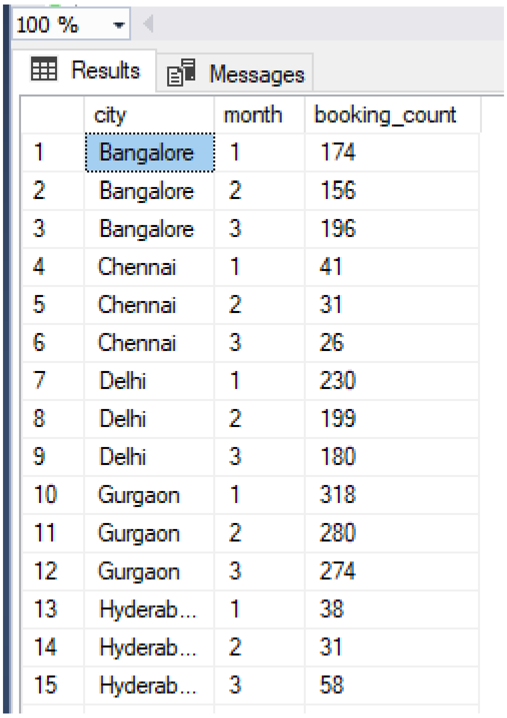
```sql
WITH CityAverage AS (
    SELECT
        oc.city,
        AVG(os.amount / NULLIF(os.no_of_rooms, 0)) AS average_room_rate
    FROM Oyo_Sales_CSV os
    JOIN Oyo_City_CSV oc ON os.hotel_id = oc.hotel_id
    WHERE os.status != 'Cancelled'
    GROUP BY oc.city
)
SELECT * FROM CityAverage;
```

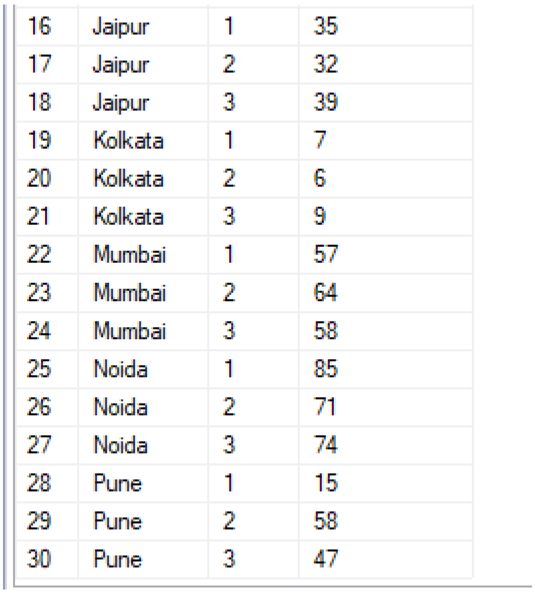


	city	average_room_rate
1	Pune	4541.9347826087
2	Hyderabad	3878.07172995781
3	Bangalore	3930.15564373898
4	Mumbai	6649.62327823692
5	Noida	2681.25174825175
6	Gurgaon	2628.46603523166
7	Chennai	3606.30434782609
8	Kolkata	4182.06666666667
9	Delhi	4013.22237196765
10	Jaipur	2721.44407894737

2. Number of Bookings in Different Cities for January, February, and March

```
SELECT
    oc.city,
    MONTH(os.date_of_booking) AS month,
    COUNT(os.booking_id) AS booking_count
FROM Oyo_Sales_CSV os
JOIN Oyo_City_CSV oc ON os.hotel_id = oc.hotel_id
WHERE MONTH(os.date_of_booking) IN (1, 2, 3)
GROUP BY oc.city, MONTH(os.date_of_booking)
ORDER BY oc.city, month;
```

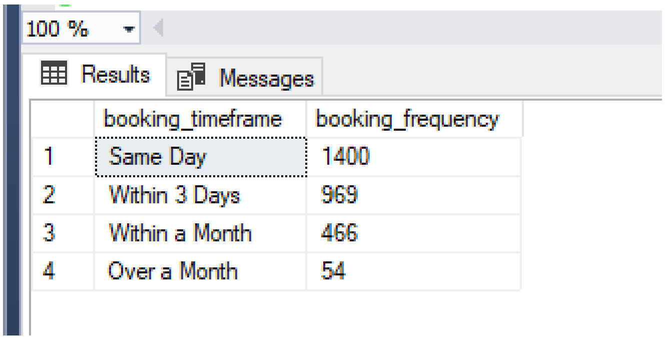




	city	month	booking_count
1	Bangalore	1	174
2	Bangalore	2	156
3	Bangalore	3	196
4	Chennai	1	41
5	Chennai	2	31
6	Chennai	3	26
7	Delhi	1	230
8	Delhi	2	199
9	Delhi	3	180
10	Gurgaon	1	318
11	Gurgaon	2	280
12	Gurgaon	3	274
13	Hyderab...	1	38
14	Hyderab...	2	31
15	Hyderab...	3	58
16	Jaipur	1	35
17	Jaipur	2	32
18	Jaipur	3	39
19	Kolkata	1	7
20	Kolkata	2	6
21	Kolkata	3	9
22	Mumbai	1	57
23	Mumbai	2	64
24	Mumbai	3	58
25	Noida	1	85
26	Noida	2	71
27	Noida	3	74
28	Pune	1	15
29	Pune	2	58
30	Pune	3	47

3. Frequency of Early Bookings Prior to Check-in

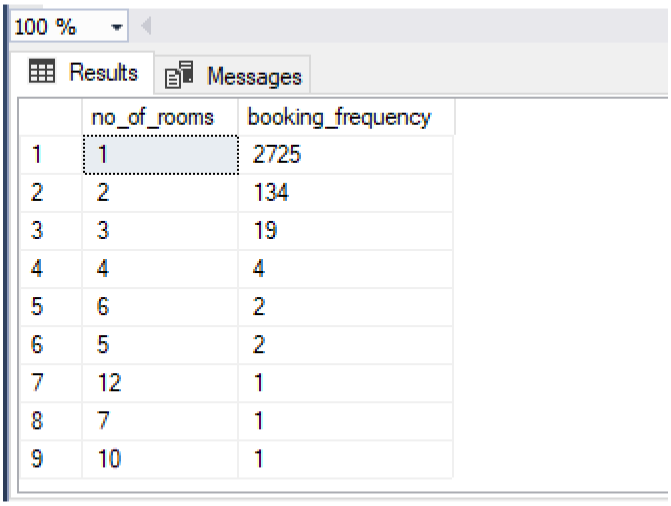
```
SELECT
    CASE
        WHEN DATEDIFF(DAY, os.date_of_booking, os.check_in) = 0 THEN 'Same Day'
        WHEN DATEDIFF(DAY, os.date_of_booking, os.check_in) <= 3 THEN 'Within 3 Days'
        WHEN DATEDIFF(DAY, os.date_of_booking, os.check_in) BETWEEN 4 AND 30 THEN
'Within a Month'
        WHEN DATEDIFF(DAY, os.date_of_booking, os.check_in) > 30 THEN 'Over a Month'
    END AS booking_timeframe,
    COUNT(os.booking_id) AS booking_frequency
FROM Oyo_Sales_CSV os
GROUP BY
    CASE
        WHEN DATEDIFF(DAY, os.date_of_booking, os.check_in) = 0 THEN 'Same Day'
        WHEN DATEDIFF(DAY, os.date_of_booking, os.check_in) <= 3 THEN 'Within 3 Days'
        WHEN DATEDIFF(DAY, os.date_of_booking, os.check_in) BETWEEN 4 AND 30 THEN
'Within a Month'
        WHEN DATEDIFF(DAY, os.date_of_booking, os.check_in) > 30 THEN 'Over a Month'
    END
ORDER BY booking_frequency DESC;
```



	booking_timeframe	booking_frequency
1	Same Day	1400
2	Within 3 Days	969
3	Within a Month	466
4	Over a Month	54

4. Frequency of Bookings Based on Number of Rooms

```
SELECT
    os.no_of_rooms,
    COUNT(os.booking_id) AS booking_frequency
FROM Oyo_Sales_CSV os
GROUP BY os.no_of_rooms
ORDER BY booking_frequency DESC;
```

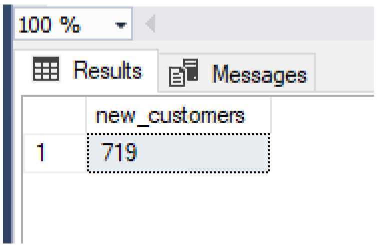


	no_of_rooms	booking_frequency
1	1	2725
2	2	134
3	3	19
4	4	4
5	6	2
6	5	2
7	12	1
8	7	1
9	10	1

5. New Customers in January

Assuming that a "new customer" is identified by a customer ID with no previous bookings before January.

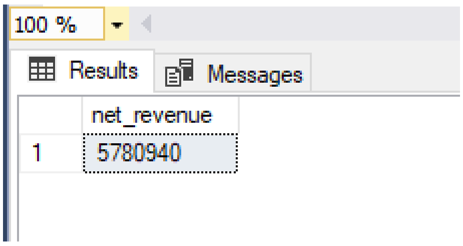
```
WITH JanuaryBookings AS (
    SELECT customer_id, MIN(date_of_booking) AS first_booking_date
    FROM Oyo_Sales_CSV
    GROUP BY customer_id
)
SELECT COUNT(customer_id) AS new_customers
FROM JanuaryBookings
WHERE MONTH(first_booking_date) = 1;
```



	new_customers
1	719

6. Net Revenue to Company (Excluding Cancelled Bookings)

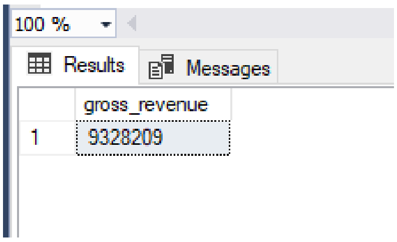
```
SELECT
    SUM(os.amount - os.discount) AS net_revenue
FROM Oyo_Sales_CSV os
WHERE os.status != 'Cancelled';
```



	net_revenue
1	5780940

7. Gross Revenue to Company (Including All Bookings)

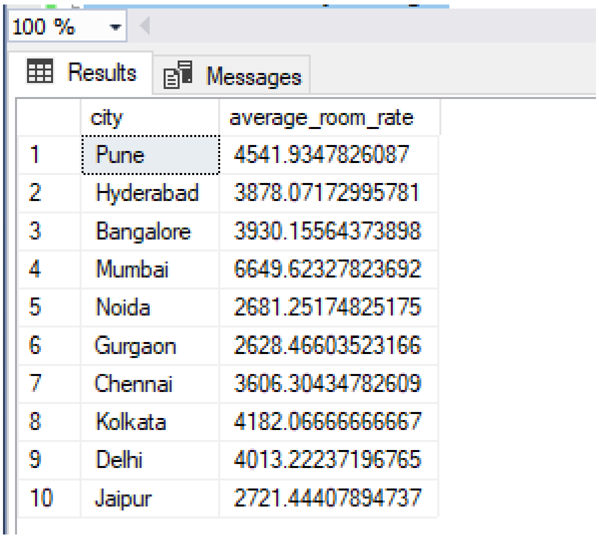
```
SELECT
    SUM(os.amount - os.discount) AS gross_revenue
FROM Oyo_Sales_CSV os;
```



	gross_revenue
1	9328209

8. Average Room Rates of Different Cities (Repeated for Clarity)

```
WITH CityAverage AS (
    SELECT
        oc.city,
        AVG(os.amount / NULLIF(os.no_of_rooms, 0)) AS average_room_rate
    FROM Oyo_Sales_CSV os
    JOIN Oyo_City_CSV oc ON os.hotel_id = oc.hotel_id
    WHERE os.status != 'Cancelled'
    GROUP BY oc.city
)
SELECT * FROM CityAverage;
```



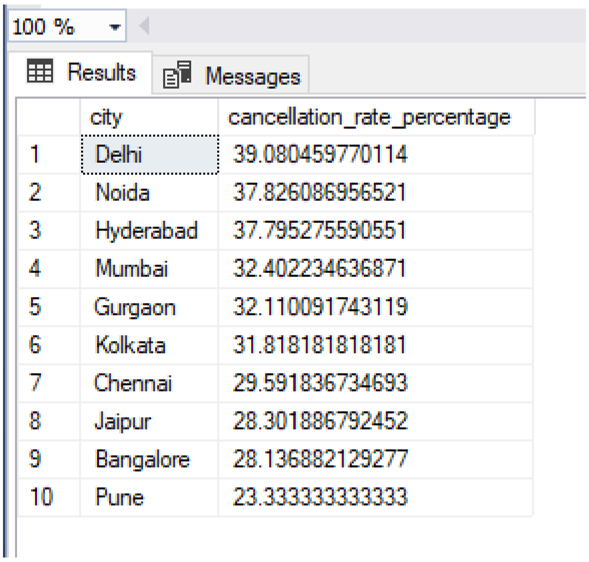
	city	average_room_rate
1	Pune	4541.9347826087
2	Hyderabad	3878.07172995781
3	Bangalore	3930.15564373898
4	Mumbai	6649.62327823692
5	Noida	2681.25174825175
6	Gurgaon	2628.46603523166
7	Chennai	3606.30434782609
8	Kolkata	4182.06666666667
9	Delhi	4013.22237196765
10	Jaipur	2721.44407894737

**Additional 5 self-written queries **

1. Find the Percentage of Cancellations for Each City

This query calculates the cancellation rate for each city by dividing the number of cancelled bookings by the total bookings in that city.

```
SELECT
    oc.city,
    COUNT(CASE WHEN os.status = 'Cancelled' THEN 1 END) * 100.0 / COUNT(*)
       AS cancellation_rate_percentage
FROM Oyo_Sales_CSV os
JOIN Oyo_City_CSV oc ON os.hotel_id = oc.hotel_id
GROUP BY oc.city
ORDER BY cancellation_rate_percentage DESC;
```

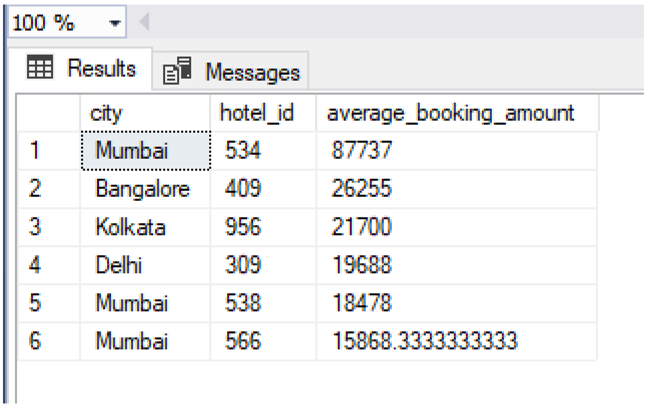


	city	cancellation_rate_percentage
1	Delhi	39.080459770114
2	Noida	37.826086956521
3	Hyderabad	37.795275590551
4	Mumbai	32.402234636871
5	Gurgaon	32.110091743119
6	Kolkata	31.818181818181
7	Chennai	29.591836734693
8	Jaipur	28.301886792452
9	Bangalore	28.136882129277
10	Pune	23.333333333333

2. List All Hotels in Cities with High Average Booking Amount (Above a Certain Threshold)

This query lists hotel IDs in cities where the average booking amount exceeds a specified threshold (e.g., 15000).

```
SELECT
    oc.city,
    os.hotel_id,
    AVG(os.amount) AS average_booking_amount
FROM Oyo_Sales_CSV os
JOIN Oyo_City_CSV oc ON os.hotel_id = oc.hotel_id
GROUP BY oc.city, os.hotel_id
HAVING AVG(os.amount) > 15000
ORDER BY average_booking_amount DESC;
```

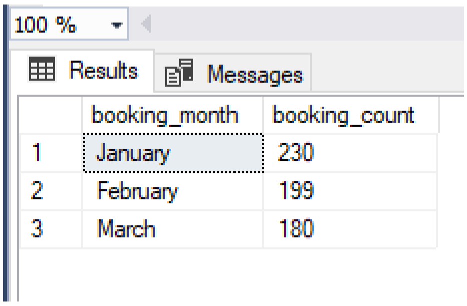


	city	hotel_id	average_booking_amount
1	Mumbai	534	87737
2	Bangalore	409	26255
3	Kolkata	956	21700
4	Delhi	309	19688
5	Mumbai	538	18478
6	Mumbai	566	15868.3333333333

3. Count of Bookings by Month for a Specific City (e.g., 'Delhi')

```
SELECT

    DATENAME(MONTH, os.date_of_booking) AS booking_month,
    COUNT(os.booking_id) AS booking_count
FROM Oyo_Sales_CSV os
JOIN Oyo_City_CSV oc ON os.hotel_id = oc.hotel_id
WHERE oc.city = 'Delhi'
GROUP BY DATENAME(MONTH, os.date_of_booking), MONTH(os.date_of_booking)
ORDER BY MONTH(os.date_of_booking);
```

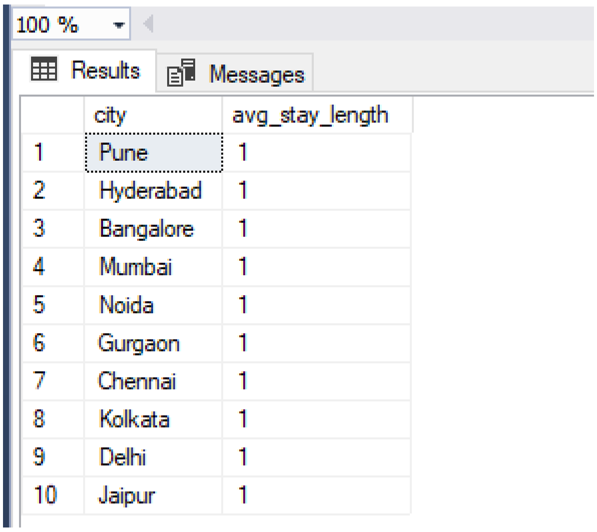


	booking_month	booking_count
1	January	230
2	February	199
3	March	180

4. Average Length of Stay for Each City

This query calculates the average stay duration (in days) for each city by using the DATEDIFF function on the check_in and check_out dates.

```
SELECT
    oc.city,
    AVG(DATEDIFF(DAY, os.check_in, os.check_out)) AS avg_stay_length
FROM Oyo_Sales_CSV os
JOIN Oyo_City_CSV oc ON os.hotel_id = oc.hotel_id
WHERE os.status != 'Cancelled'
GROUP BY oc.city
ORDER BY avg_stay_length DESC;
```

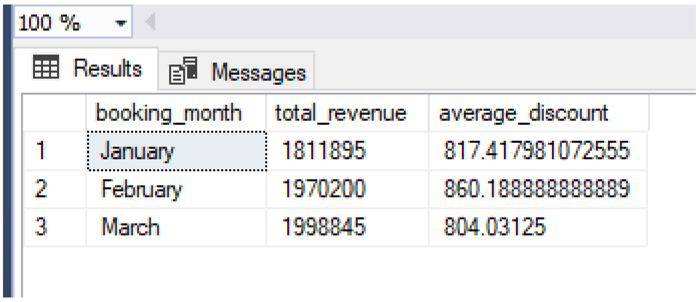


	city	avg_stay_length
1	Pune	1
2	Hyderabad	1
3	Bangalore	1
4	Mumbai	1
5	Noida	1
6	Gurgaon	1
7	Chennai	1
8	Kolkata	1
9	Delhi	1
10	Jaipur	1

5. Total Revenue and Average Discount Given for Each Month

This query calculates the total revenue and average discount given per month across all bookings (excluding cancelled bookings).

```
SELECT
    DATENAME(MONTH, os.date_of_booking) AS booking_month,
    SUM(os.amount - os.discount) AS total_revenue,
    AVG(os.discount) AS average_discount
FROM Oyo_Sales_CSV os
WHERE os.status != 'Cancelled'
GROUP BY DATENAME(MONTH, os.date_of_booking), MONTH(os.date_of_booking)
ORDER BY MONTH(os.date_of_booking);
```



	booking_month	total_revenue	average_discount
1	January	1811895	817.417981072555
2	February	1970200	860.188888888889
3	March	1998845	804.03125

-Thank You!