**SQL-Case Study**

**Submitted By-**

**Subrat Shukla, DE Batch1**

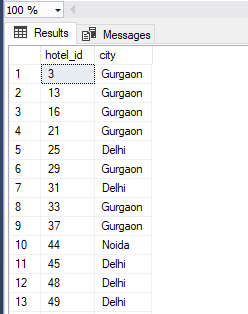
**Creating an ‘OYO\_Business’ database-**

create database OYO\_Business;

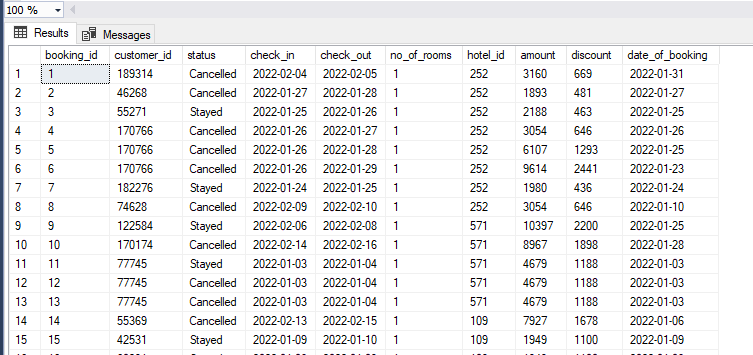
use OYO\_Business;

**Overview of database tables-**

select \*from [dbo].[Oyo\_City\_CSV]



select \*from [dbo].[Oyo\_Sales\_CSV]



**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.

1. 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.
2. 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**

WITH CityAverage AS (

SELECT

oc.city,

AVG(os.amount / NULLIF(os.no\_of\_rooms, 0)) AS average\_room\_rate

FROM Oyo\_Sales\_CSV os

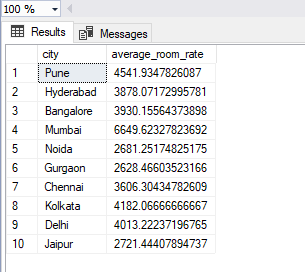
JOIN Oyo\_City\_CSV oc ON os.hotel\_id = oc.hotel\_id

WHERE os.status != 'Cancelled'

GROUP BY oc.city

)

SELECT \* FROM CityAverage;



**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

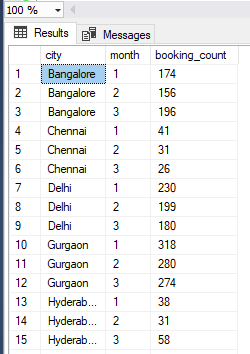
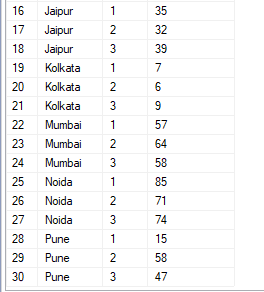
FROM Oyo\_Sales\_CSV os

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;

**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'

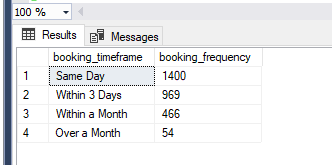
WHEN DATEDIFF(DAY, os.date\_of\_booking, os.check\_in) <= 3 THEN 'Within 3 Days'

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;



**4. Frequency of Bookings Based on Number of Rooms**

SELECT

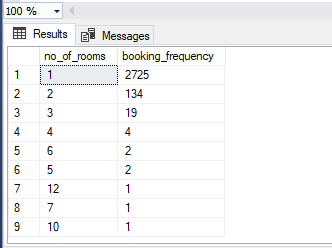
os.no\_of\_rooms,

COUNT(os.booking\_id) AS booking\_frequency

FROM Oyo\_Sales\_CSV os

GROUP BY os.no\_of\_rooms

ORDER BY booking\_frequency DESC;



**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

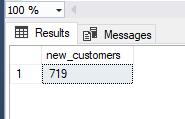
GROUP BY customer\_id

)

SELECT COUNT(customer\_id) AS new\_customers

FROM JanuaryBookings

WHERE MONTH(first\_booking\_date) = 1;



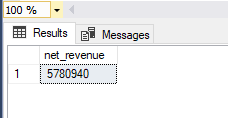
**6. Net Revenue to Company (Excluding Cancelled Bookings)**

SELECT

SUM(os.amount - os.discount) AS net\_revenue

FROM Oyo\_Sales\_CSV os

WHERE os.status != 'Cancelled';

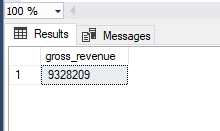


**7. Gross Revenue to Company (Including All Bookings)**

SELECT

SUM(os.amount - os.discount) AS gross\_revenue

FROM Oyo\_Sales\_CSV os;



**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

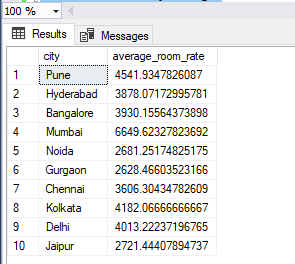
JOIN Oyo\_City\_CSV oc ON os.hotel\_id = oc.hotel\_id

WHERE os.status != 'Cancelled'

GROUP BY oc.city

)

SELECT \* FROM CityAverage;



**\*\*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(\*)

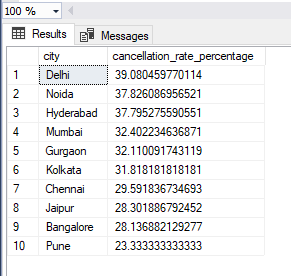
AS cancellation\_rate\_percentage

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;



**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

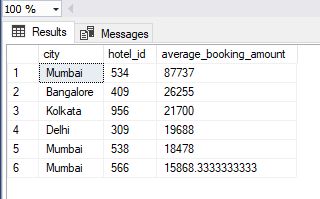
FROM Oyo\_Sales\_CSV os

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;



**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

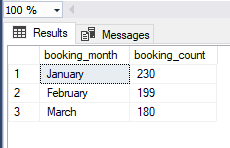
FROM Oyo\_Sales\_CSV os

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);



**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

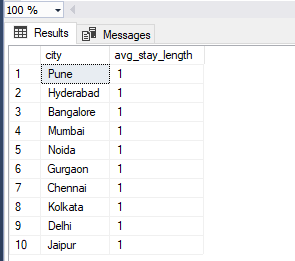
FROM Oyo\_Sales\_CSV os

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;



**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,

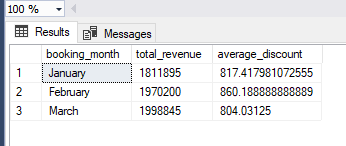
AVG(os.discount) AS average\_discount

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);



**-Thank You!**