Case Study-1

OYO Business

Name: **E.R Harish**

1. Average Room Rates of Different Cities

```sql
-- 1. Average Room Rates of Different Cities
WITH CityBookings AS (
    SELECT os.hotel_id, oc.city, os.amount - os.discount AS net_amount
    FROM Oyo_Sales os
    JOIN Oyo_City oc ON os.hotel_id = oc.hotel_id
    WHERE os.status != 'Cancelled'
)
SELECT city, AVG(net_amount) AS avg_room_rate FROM CityBookings GROUP BY city ORDER BY avg_room_rate DESC;
```

```
city
----------------------------------------------------------------------------------------------------
Mumbai
Pune
Hyderabad
Kolkata
Delhi
Bangalore
Chennai
Jaipur
Noida
Gurgaon
```

2. Number of Bookings in January, February, and March by City

```sql
-- 2. Number of Bookings in January, February, and March by City
SELECT oc.city, MONTH(os.check_in) AS booking_month, COUNT(*) AS booking_count
FROM Oyo_Sales os
JOIN Oyo_City oc ON os.hotel_id = oc.hotel_id
WHERE MONTH(os.check_in) IN (1, 2, 3)
GROUP BY oc.city, MONTH(os.check_in)
ORDER BY oc.city, booking_month;
```

```
city
--------------------------------------------------------------------------------
Bangalore
Bangalore
Bangalore
Chennai
Chennai
Chennai
Delhi
Delhi
Delhi
Gurgaon
Gurgaon
Gurgaon
Hyderabad
Hyderabad
Hyderabad
Jaipur
Jaipur
Jaipur
Kolkata
Kolkata
Kolkata
Mumbai
Mumbai
Mumbai
Noida
Noida
Noida
Pune
Pune
Pune

(30 rows affected)
```

3. Frequency of Early Bookings Prior to Check-In

```sql
-- 3. Frequency of Early Bookings Prior to Check-In
SELECT DATEDIFF(day, os.date_of_booking, os.check_in) AS days_prior, COUNT(*) AS frequency
FROM Oyo_Sales os WHERE os.status != 'Cancelled'
GROUP BY DATEDIFF(day, os.date_of_booking, os.check_in)
ORDER BY days_prior;
```

days_prior	frequency
0	1088
1	403
2	132
3	72
4	50
5	36
6	17
7	16
8	17
9	12
10	11
11	12
12	9
13	7
14	4
15	2
16	2
17	4
19	3
20	2
21	2
22	6
23	4
24	2
27	3
28	3
30	1
31	4
33	3
34	2
36	1
40	2
43	1
84	1
90	1
93	1

4. Frequency of Bookings by Number of Rooms

```sql
-- 4. Frequency of Bookings by Number of Rooms
SELECT no_of_rooms, COUNT(*) AS booking_count
FROM Oyo_Sales WHERE status != 'Cancelled' GROUP BY no_of_rooms
ORDER BY booking_count DESC;
```

```
no_of_rooms booking_count
----------- -------------
1           1833
2           83
3           14
4           4
6           1
7           1

(6 rows affected)
```

5. New Customers in January

```sql
-- 5. New Customers in January
SELECT customer_id, COUNT(*) AS bookings_in_jan
FROM Oyo_Sales WHERE MONTH(date_of_booking) = 1 GROUP BY customer_id HAVING COUNT(*) = 1;
```

```
customer_id bookings_in_jan
----------- ---------------
765         1
2162        1
2827        1
3145        1
3424        1
3452        1
3580        1
3585        1
3770        1
4043        1
4675        1
5049        1
5977        1
6014        1
6033        1
6284        1
6513        1
6989        1
7121        1
7282        1
7701        1
7731        1
7936        1
8066        1
8240        1
8244        1
8353        1
8380        1
8605        1
8760        1
8861        1
8873        1
8875        1
9133        1
9198        1
9338        1
```

6. Net Revenue to the Company (Excluding Canceled Bookings)

```sql
-- 6. Net Revenue to the Company (Excluding Canceled Bookings)
SELECT SUM(amount - discount) AS net_revenue FROM Oyo_Sales WHERE status != 'Cancelled';
```

```
net_revenue
---------------------------------------
5780940.0000000000

(1 row affected)
```

7. Gross Revenue to the Company

```sql
-- 7. Gross Revenue to the Company
SELECT SUM(amount) AS gross_revenue FROM Oyo_Sales;
```

```
gross_revenue
---------------------------------------
11917462.0000000000

(1 row affected)
```

8. Average Room Rates of Different Cities

```sql
-- 8. Average Room Rates of Different Cities
WITH CityBookings AS (
    SELECT os.hotel_id, oc.city, os.amount - os.discount AS net_amount
    FROM Oyo_Sales os
    JOIN Oyo_City oc ON os.hotel_id = oc.hotel_id
    WHERE os.status != 'Cancelled'
)
SELECT city, AVG(net_amount) AS avg_room_rate
FROM CityBookings GROUP BY city ORDER BY avg_room_rate DESC;
```

```
city
-----------------------------------------------------------------
Mumbai
Pune
Hyderabad
Kolkata
Delhi
Bangalore
Chennai
Jaipur
Noida
Gurgaon

(10 rows affected)
```

9. Monthly Cancellations and Booking Trends by City with Percentage of Cancellations

```sql
-- 9. Monthly Cancellations and Booking Trends by City with Percentage of Cancellations
WITH MonthlyBookings AS (
    SELECT
        oc.city,
        MONTH(os.check_in) AS booking_month,
        COUNT(CASE WHEN os.status = 'Cancelled' THEN 1 END) AS cancellations,
        COUNT(*) AS total_bookings
    FROM Oyo_Sales os
    JOIN Oyo_City oc ON os.hotel_id = oc.hotel_id
    GROUP BY oc.city, MONTH(os.check_in)
)
SELECT city,booking_month,cancellations,total_bookings,
ROUND((CAST(cancellations AS FLOAT) / total_bookings) * 100, 2) AS cancellation_percentage
FROM MonthlyBookings ORDER BY cancellation_percentage DESC, city, booking_month;
```

```
city
----------
Chennai
Delhi
Gurgaon
Hyderabad
Noida
Noida
Pune
Gurgaon
Hyderabad
Bangalore
Kolkata
Delhi
Delhi
Noida
Hyderabad
Chennai
Jaipur
Mumbai
Delhi
Hyderabad
Gurgaon
Noida
Delhi
Jaipur
Bangalore
Mumbai
Noida
Bangalore
Gurgaon
Kolkata
Hyderabad
Gurgaon
Mumbai
Bangalore
Mumbai
```

10. Top 3 Cities by Revenue with Average Room Rate and Cancellation Impact

```sql
-- 10. Top 3 Cities by Revenue with Average Room Rate and Cancellation Impact
WITH CityRevenue AS (
    SELECT
        oc.city,
        SUM(CASE WHEN os.status != 'Cancelled' THEN os.amount - os.discount ELSE 0 END) AS net_revenue,
        AVG(CASE WHEN os.status != 'Cancelled' THEN os.amount - os.discount END) AS avg_room_rate,
        SUM(CASE WHEN os.status = 'Cancelled' THEN os.amount - os.discount ELSE 0 END) AS canceled_revenue
    FROM Oyo_Sales os
    JOIN Oyo_City oc ON os.hotel_id = oc.hotel_id
    GROUP BY oc.city
)
SELECT
    city,
    net_revenue,
    avg_room_rate,
    canceled_revenue,
    (net_revenue + canceled_revenue) AS potential_gross_revenue
FROM CityRevenue
ORDER BY net_revenue DESC
OFFSET 0 ROWS FETCH NEXT 3 ROWS ONLY;
```

```
city
--------------------------------------------------------------------------------
Gurgaon
Delhi
Bangalore
Warning: Null value is eliminated by an aggregate or other SET operation.

(3 rows affected)
```

11. Customers with the Most Last-Minute Bookings and Their Total Spend

```sql
-- 11. Customers with the Most Last-Minute Bookings and Their Total Spend
WITH LastMinuteBookings AS (
    SELECT
        customer_id,
        COUNT(*) AS last_minute_count,
        SUM(amount - discount) AS total_spend
    FROM Oyo_Sales
    WHERE DATEDIFF(day, date_of_booking, check_in) = 0
    AND status != 'Cancelled'
    GROUP BY customer_id
)
SELECT
    customer_id,
    last_minute_count,
    total_spend,
    RANK() OVER (ORDER BY last_minute_count DESC, total_spend DESC) AS rank
FROM LastMinuteBookings
WHERE last_minute_count >= 5
ORDER BY rank;
```

customer_id	last_minute_count	total_spend	rank
140493	17	64302.0000000000	1
137492	16	26360.0000000000	2
78365	11	20439.0000000000	3
137032	9	11570.0000000000	4
146622	8	13773.0000000000	5
143338	7	11550.0000000000	6
57801	7	10618.0000000000	7
126588	6	17574.0000000000	8
193175	6	16211.0000000000	9
168301	6	10605.0000000000	10
10362	6	9819.0000000000	11
3207	6	9662.0000000000	12
77179	5	27762.0000000000	13
8704	5	23762.0000000000	14
45644	5	21102.0000000000	15
9858	5	11756.0000000000	16
45412	5	9939.0000000000	17
26389	5	9374.0000000000	18
113889	5	7923.0000000000	19
172114	5	6741.0000000000	20

(20 rows affected)

12. Booking Patterns for Single vs. Multiple Room Stays by City

```sql
-- 12. Booking Patterns for Single vs. Multiple Room Stays by City
WITH RoomBookingPatterns AS (
    SELECT
        oc.city,
        CASE WHEN os.no_of_rooms = 1 THEN 'Single Room' ELSE 'Multiple Rooms' END AS room_type,
        COUNT(*) AS booking_count,
        SUM(os.amount - os.discount) AS total_revenue,
        AVG(DATEDIFF(day, os.check_in, os.check_out)) AS avg_stay_length
    FROM Oyo_Sales os
    JOIN Oyo_City oc ON os.hotel_id = oc.hotel_id
    WHERE os.status != 'Cancelled'
    GROUP BY oc.city, CASE WHEN os.no_of_rooms = 1 THEN 'Single Room' ELSE 'Multiple Rooms' END
)
SELECT
    city,
    room_type,
    booking_count,
    total_revenue,
    avg_stay_length
FROM RoomBookingPatterns
ORDER BY city, room_type;
```

```
city
----------------------------------------------------
Bangalore
Bangalore
Chennai
Delhi
Delhi
Gurgaon
Gurgaon
Hyderabad
Hyderabad
Jaipur
Jaipur
Kolkata
Mumbai
Mumbai
Noida
Noida
Pune
Pune

(18 rows affected)
```

13. Top Returning Customers with Average Booking Interval and Total Spend

```sql
-- 13. Top Returning Customers with Average Booking Interval and Total Spend
WITH CustomerBookings AS (
    SELECT
        customer_id,
        check_in,
        amount - discount AS net_amount,
        LAG(check_in) OVER (PARTITION BY customer_id ORDER BY check_in) AS previous_check_in
    FROM Oyo_Sales
    WHERE status != 'Cancelled'
),
CustomerIntervals AS (
    SELECT
        customer_id,
        COUNT(*) AS total_bookings,
        SUM(net_amount) AS total_spend,
        AVG(DATEDIFF(day, previous_check_in, check_in)) AS avg_booking_interval
    FROM CustomerBookings
    WHERE previous_check_in IS NOT NULL
    GROUP BY customer_id
)
SELECT
    customer_id,
    total_bookings,
    total_spend,
    avg_booking_interval,
    RANK() OVER (ORDER BY total_spend DESC) AS spend_rank
FROM CustomerIntervals
WHERE total_bookings > 1
ORDER BY spend_rank;
```

customer_id	total_bookings	total_spend	avg_booking_interval	spend_rank
140493	21	77304.0000000000	4	1
77179	9	73891.0000000000	5	2
158427	5	45624.0000000000	0	3
78365	17	31503.0000000000	3	4
122584	5	30223.0000000000	12	5
66504	3	29407.0000000000	2	6
57801	18	26937.0000000000	1	7
193036	4	25988.0000000000	14	8
137492	15	25128.0000000000	4	9
8704	5	24842.0000000000	7	10
170174	5	18594.0000000000	2	11
45644	4	17313.0000000000	17	12
126588	6	17266.0000000000	5	13
133399	3	17247.0000000000	-32	14
74860	4	16902.0000000000	15	15
58198	2	16821.0000000000	3	16
71382	5	16204.0000000000	7	17
193175	5	14612.0000000000	7	18
55143	3	14292.0000000000	5	19
196829	2	13931.0000000000	10	20
37423	3	12864.0000000000	11	21
106210	5	12536.0000000000	10	22
73006	3	12345.0000000000	2	23
146622	7	12271.0000000000	12	24
154793	2	12220.0000000000	0	25
66762	5	12193.0000000000	14	26
143338	7	11643.0000000000	2	27
26389	6	11371.0000000000	9	28
130908	2	11217.0000000000	3	29
8380	2	10954.0000000000	23	30
188854	4	10660.0000000000	20	31
168301	6	10605.0000000000	-2	32
9858	4	10420.0000000000	12	33
137032	8	10276.0000000000	3	34
45412	5	9939.0000000000	3	35
149015	2	9856.0000000000	1	36