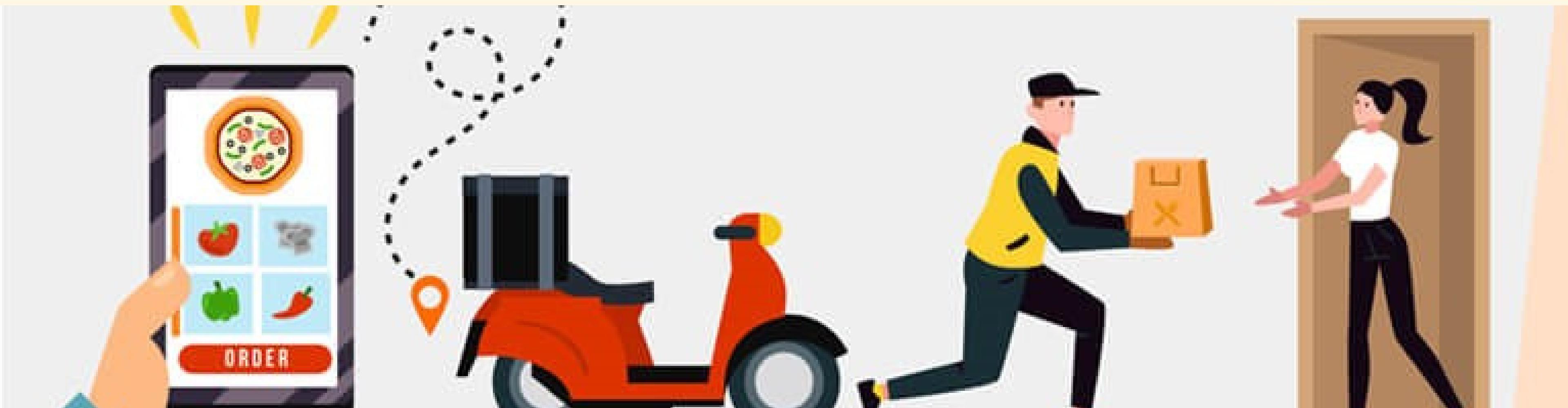


BUSINESS INSIGHTS

Yumm Food Delivery Service

From Restaurant to Your Residence





Introduction

In a small town, Priya and her friends have launched a startup named "**Yum Food Delivery Service**," and they've tapped into a compelling issue: the lack of food delivery services. The town's residents favor convenient food delivery options. The lack of existing delivery services only highlighted the demand for a solution.

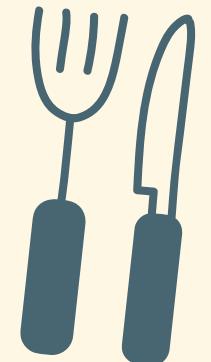
Focused efforts led Priya to secure partnerships with five local hotels an achievement that lays the groundwork for future growth. In two days, the startup delivered seven orders.

To continue driving growth and make informed decisions, Priya recognizes the importance of data. She plans to analyze survey results from her customers, seeking valuable insights that can fuel the Marketing team's efforts. Priya aims to extract meaningful trends and patterns from the data, allowing her to optimize her startup's operations. This case study underscores the effective utilization of Priya's business to address crucial inquiries within the realm of food delivery services.



Datasets

The user IDs correspond to customers who have placed orders with a hotel. Each user is associated with a unique ID, along with their respective names and email addresses



Customers have placed food orders with a specific hotel, and each order includes the hotel's ID, name, and the type of cuisine the customers have ordered

Users

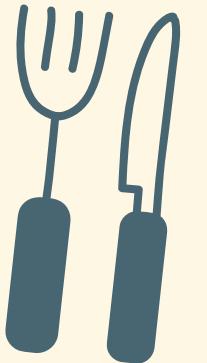
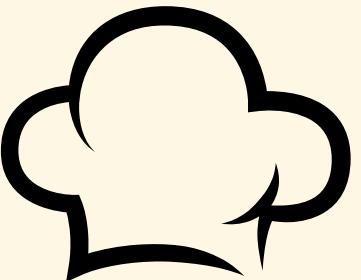
User_Id	Name	Email
1	Siya	siya@gmail.com
2	Sneha	Sneha@gmail.com
3	Aryan	Aryan@gmail.com
4	Naveen	Naveen@gmail.com
5	Shresha	Shresha@gmail.com
6	Neil	Neil@gmail.com
7	Aarav	Aarav@gmail.com

Hotel

Hotel_Id	Hotel_Name	Cuisine
1	Juice Junction	Eastern Indian Cuisine
2	Swagat	Western Indian Cuisine
3	Rasoi	North Indian
4	Dosa Plaza	South Indian
5	China Town	Chinese

Datasets

Customers have placed food orders, and each order includes the respective food's name, ID, and type



The partner ID and partner name refer to the individuals responsible for delivering the food orders. Each delivery is associated with a specific partner's ID and name

Food

Food_Id	Food_Name	Food_Type
1	Non-veg Pizza	Non-veg
2	Veg Pizza	Veg
3	Choco Lava cake	Veg
4	Chicken Wings	Non-veg
5	Chicken Popcorn	Non-veg
6	Rice Meal	Veg
7	Roti meal	Veg
8	Masala Dosa	Veg
9	Rava Idli	Veg
10	Schezwan Noodles	Veg
11	Veg Manchurian	Veg

Delivery Partner

Partner_Id	Partner_Name
1	Anish
2	Kavya
3	Ajay
4	Ariana
5	Gagan

Datasets

Menu

The menu ID corresponds to a specific hotel, and for each menu ID, there are associated food items with their respective food IDs and prices



Menu_Id	Hotel_Id	Food_Id	Price
1	1	1	450
2	1	2	400
3	1	3	100
4	2	3	115
5	2	4	230
6	2	5	300
7	3	3	80
8	3	6	160
9	3	7	140
10	4	6	230
11	4	8	180
12	4	9	120
13	5	6	250
14	5	10	220
15	5	11	180

Datasets

"The 'hotel rating' represents the rating of the hotel from which the customer has placed an order. 'Delivery time' signifies the estimated time it takes for the ordered food to arrive. 'Amount' refers to the total cost of the food items ordered by customers. The 'User ID' serves as the unique identifier for the customer placing the order, while the 'Order ID' is a distinct identifier for each order placed at a hotel. 'Hotel ID' corresponds to the unique identifier of the hotel where the customer has placed the order, and 'date' signifies the specific period during which the customer initiated the order.

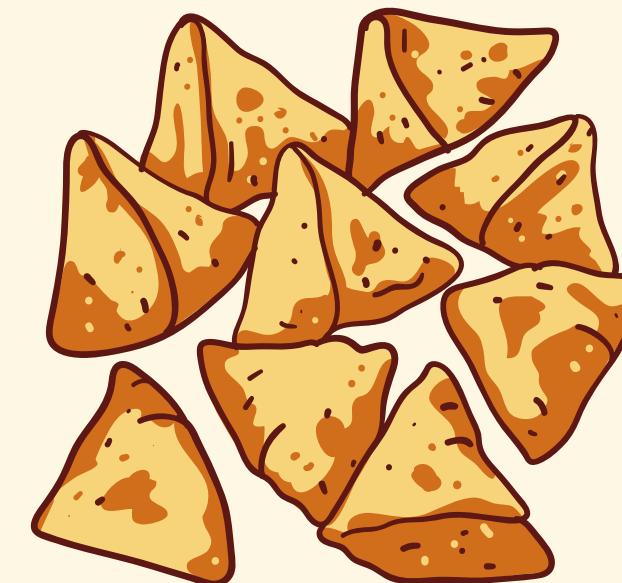


Orders

Order_Id	User_Id	Hotel_Id	Amount	Date	Partner_Id	Delivery_Time	Delivery_Rating	Hotel_Rating
1001	1	1	550	2022-05-10	1	25	5	3
1002	1	2	415	2022-05-26	1	19	5	2
1003	1	3	240	2022-06-15	5	29	4	
1004	1	3	240	2022-06-29	4	42	3	5
1005	1	3	220	2022-07-10	1	58	1	4
1006	2	1	950	2022-06-10	2	16	5	
1007	2	2	530	2022-06-23	3	60	1	5
1008	2	3	240	2022-07-07	5	33	4	5
1009	2	4	300	2022-07-17	4	41	1	
1010	2	5	650	2022-07-31	1	67	1	4
1011	3	1	450	2022-05-10	2	25	3	1
1012	3	4	180	2022-05-20	5	33	4	1
1013	3	2	230	2022-05-30	4	45	3	
1014	3	2	230	2022-06-11	2	55	1	2
1015	3	2	230	2022-06-22	3	21	5	
1016	4	4	300	2022-05-15	3	31	5	5
1017	4	4	300	2022-05-30	1	50	1	
1018	4	4	400	2022-06-15	2	40	3	5
1019	4	5	400	2022-06-30	1	70	2	4
1020	4	5	400	2022-07-15	3	26	5	3
1021	5	1	550	2022-07-01	5	22	2	
1022	5	1	550	2022-07-08	1	34	5	1
1023	5	2	645	2022-07-15	4	38	5	
1024	5	2	645	2022-07-21	2	58	2	1
1025	5	2	645	2022-07-28	2	44	4	

Datasets

The 'Order ID' is associated with a 'Food ID,' indicating the specific food item included in each order.



Orders

Id	Order_Id	Food_Id
1	1001	1
2	1001	3
3	1002	4
4	1002	3
5	1003	6
6	1003	3
7	1004	6
8	1004	3
9	1005	7
10	1005	3
11	1006	1
12	1006	2
13	1006	3
14	1007	4
15	1007	3
16	1008	6
17	1008	3
18	1009	8
19	1009	9
20	1010	10
21	1010	11
22	1010	6
23	1011	1
24	1012	8
25	1013	4
26	1014	4
27	1015	4
28	1016	8
29	1016	9
30	1017	8
31	1017	9
32	1018	10
33	1018	11
34	1019	10
35	1019	11
36	1020	10
37	1020	11
38	1021	1
39	1021	3
40	1022	1
41	1022	3
42	1023	3
43	1023	4
44	1023	5
45	1024	3
46	1024	4
47	1024	5
48	1025	3
49	1025	4
50	1025	5

Case Study Questions



IDENTIFY CUSTOMERS WHO HAVEN'T PLACED ANY ORDERS.

INPUT:

• `SELECT Name as "Customers Names" FROM users WHERE user_id NOT IN (SELECT User_Id FROM orders) ;`

OUTPUT:

Customers Names
Neil
Aarav



Case Study Questions

CALCULATE THE AVERAGE PRICE OF A DISH.

INPUT:

```
SELECT  
    f.Food_Id, f.Food_Name, AVG(Price) AS 'Average price'  
FROM  
    menu m  
        JOIN  
    food f ON m.Food_Id = f.Food_Id  
GROUP BY f.Food_Id , f.Food_Name;
```

OUTPUT:

Food_Id	Food_Name	Average price
1	Non-veg Pizza	450.0000
2	Veg Pizza	400.0000
3	Choco Lava cake	98.3333
4	Chicken Wings	230.0000
5	Chicken Popcorn	300.0000
6	Rice Meal	213.3333
7	Roti meal	140.0000
8	Masala Dosa	180.0000
9	Rava Idli	120.0000
10	Schezwan Noodles	220.0000
11	Veg Manchurian	180.0000



YOUR PARAGRAPH TEXT

Case Study Questions

DETERMINE THE RESTAURANT WITH THE HIGHEST ORDER COUNT FOR A SPECIFIC MONTH NUMBER.

INPUT:

```
SELECT Hotel_Id, count(*) AS "Month Number" FROM orders WHERE monthname(date) LIKE "June" GROUP BY Hotel_Id;
```

INPUT:

Hotel_Id	Month
3	2
1	1
2	3
4	1
5	1



Case Study Questions

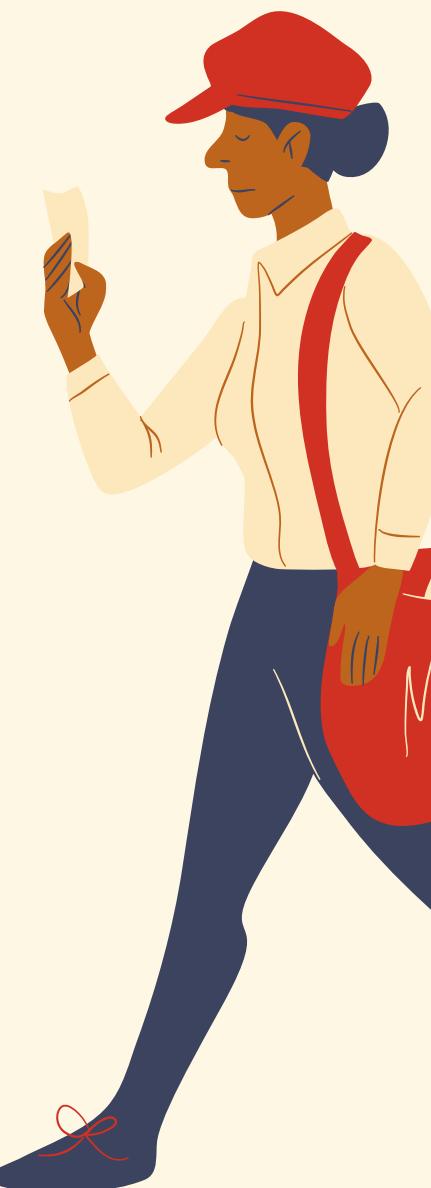
RETRIEVE RESTAURANTS WITH MONTHLY SALES EXCEEDING A SET THRESHOLD.

INPUT:

```
SELECT h.Hotel_Name,sum(o.amount) AS "Revenue" FROM orders o JOIN hotel h ON o.Hotel_Id=h.Hotel_Id  
WHERE monthname(date)="June" GROUP BY h.Hotel_Id HAVING revenue >500 ;
```

OUTPUT:

Hotel_Name	Revenue
Juice Junction	950
Swagat	990



Case Study Questions



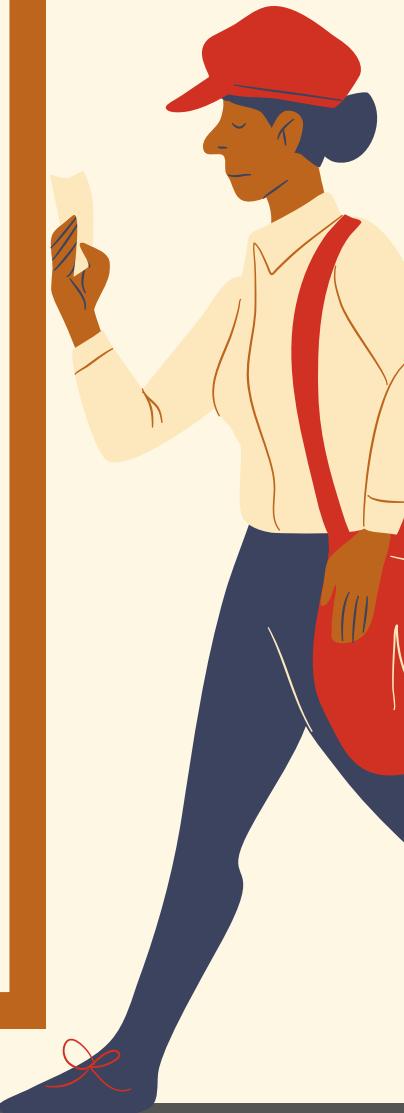
DISPLAY ALL ORDER INFORMATION, INCLUDING ORDER DETAILS, FOR A SPECIFIC CUSTOMER WITHIN A SPECIFIED DATE RANGE.

INPUT:

```
SELECT
    o.Order_Id, h.Hotel_Name, f.Food_Name
FROM
    orders o
        JOIN
    hotel h ON h.Hotel_Id = o.Hotel_Id
        JOIN
    order_details od ON o.Order_Id = od.Order_Id
        JOIN
    food f ON f.Food_Id = od.Food_Id
WHERE
    User_Id = (SELECT
        User_Id
    FROM
        users
    WHERE
        Name = 'Siya')
    AND (Date > '2022-06-10'
    AND Date < '2022-07-10');
```

OUTPUT

Order_Id	Hotel_Name	Food_Name
1004	Rasoi	Choco Lava cake
1003	Rasoi	Choco Lava cake
1004	Rasoi	Rice Meal
1003	Rasoi	Rice Meal



Case Study Questions

LOCATE RESTAURANTS THAT HAVE THE HIGHEST NUMBER OF RECURRING CUSTOMERS.

INPUT:

```
SELECT h.Hotel_Name, count(*) AS Loyal_Customers FROM (SELECT Hotel_Id, User_Id, count(*) AS "Visits" FROM orders  
GROUP BY Hotel_Id, User_Id HAVING Visits > 1) T  
JOIN hotel h ON h.Hotel_Id = T.Hotel_Id GROUP BY T.Hotel_Id ORDER BY Loyal_Customers DESC LIMIT 1;
```

OUTPUT:

Hotel_Name	Loyal_Customers
Swagat	2



Case Study Questions



CALCULATE THE MONTH-TO-MONTH REVENUE GROWTH FOR SWIGGY.

INPUT:

```
WITH Temp AS(
SELECT o.User_Id,od.Food_Id,count(*) AS "Frequency"  FROM orders o JOIN order_details od ON o.Order_Id=od.Order_Id GROUP BY  o.User_Id,od.Food_Id)
SELECT u.Name,f.Food_Name,T1.Frequency  FROM Temp T1 JOIN users u  ON u.User_Id=T1.User_Id JOIN food f ON f.Food_Id=T1.Food_Id
WHERE T1.Frequency = (SELECT max(Frequency) FROM Temp T2 WHERE T2.User_Id=T1.User_Id );
```

OUTPUT:

Month	Month Over Month Revenue
May	NULL
June	32.7835
July	50.4658



Case Study Questions



IDENTIFY THE PREFERRED FOOD CHOICES OF CUSTOMERS.

INPUT:

```
WITH Temp AS(
SELECT o.User_Id,od.Food_Id,count(*) AS "Frequency"  FROM orders o JOIN order_details od ON o.Order_Id=od.Order_Id GROUP BY  o.User_Id,od.Food_Id)
SELECT u.Name,f.Food_Name,T1.Frequency  FROM Temp T1 JOIN users u  ON u.User_Id=T1.User_Id JOIN food f ON f.Food_Id=T1.Food_Id
WHERE T1.Frequency = (SELECT max(Frequency) FROM Temp T2 WHERE T2.User_Id=T1.User_Id );
```

OUTPUT:

Name	Food_Name	Frequency
Shresha	Choco Lava cake	5
Sneha	Choco Lava cake	3
Siya	Choco Lava cake	5
Aryan	Chicken Wings	3
Naveen	Schezwan Noodles	3
Naveen	Veg Manchurian	3



Why Cook When You Can Order Online?

ORDER FOOD AT YUMM FOOD DELIVERY SERVICE

Thank you !

