Introduction

The restaurant has captured some very basic data from their few months of operation but have no idea how to use their data to help them run the business.

Problem Statement

We want to use the data to answer a few simple questions about his customers, especially about their

- visiting patterns,
- how much money they've spent, and
- which menu items are their favourite. Having this deeper connection with his customers will help him deliver a better and more personalised experience for his loyal customers.

The data set contains the following 3 tables:

Cust_id	Cust_Name	Join_date
1	Yash	2023-01-12
2	Sanskar	2023-02-18
3	Kunal	2023-03-14
4	Anuj	2023-04-20
5	Vivek	2023-05-16

Menu_id	Menu_name	Price
101	Jumbo Burger	150
102	Pizza	200
103	Cold coffee	100

Cust_id	Menu_id	Order_date
1	102	2023-04-12
1	101	2023-04-12
1	103	2023-02-08
1	102	2023-06-06
2	101	2023-03-15
2	101	2023-03-15
2	103	2023-02-16
3	101	2023-09-17
3	103	2023-09-13
4	103	2023-07-19
4	102	2023-09-19
4	102	2023-06-16
5	101	2023-04-19
5	101	2023-04-22

Case Study Questions

1. What is the total amount each customer spent at the restaurant?

2. How many days has each customer visited the restaurant?

3. What was the first item from the menu purchased by each customer?

4. What is the most purchased item on the menu and how many times was it purchased by all customers?

5. Which item was the most popular for each customer?

6. Which item was purchased first by the customer after they became a member?

7. Which item was purchased just before the customer became a member?

8. What is the total items and amount spent for each member before they became a member?

9. If each Ruppe 1 spent equates to 10 points and Jumbo Burger has a 2x points multiplier - how many points would each customer have?

Solution: Click here to open (.sql) file

```
create database Evershine_FastFood;
drop database Evershine_FastFood;
use Evershine_FastFood;

create table Customer(
Cust_id int primary key ,
Cust_Name varchar(20) not null,
Join_date date);
drop table Customer;

create table Menu(
Menu_id int primary key,
Menu_name varchar(20),
Price float);
drop table Menu;
```

```
create table Sales(
Cust_id int ,
Menu_id int ,
Order_date date,
foreign key (Cust_id) references customer(Cust_id),
foreign key (Menu_id) references menu(Menu_id)
on update cascade
on delete cascade);
drop table Sales;

insert into Customer (Cust_id, Cust_Name, Join_date) values
(1, "Yash" , "2023-01-12"),
(2, "Sanskar" , "2023-02-18"),
(3, "Kunal" , "2023-03-14"),
(4, "Anuj" , "2023-04-20"),
(5, "Vivek" , "2023-05-16");
truncate table Customer;
select * from Customer;

insert into Menu (Menu_id, Menu_name, Price) values
(101, "Jumbo Burger", 150),
(102, "Pizza", 200 ),
(103, "Cold coffee", 100);
truncate table Menu;
select * from Menu;

insert into Sales (Cust_id, Menu_id, Order_date) values
(1, 102, "2023-04-12"),
(1, 101, "2023-04-12"),
(1, 103, "2023-02-08"),
(1, 102, "2023-06-06"),
(2, 101, "2023-03-15"),
(2, 101, "2023-03-15"),
(2, 103, "2023-02-16"),
(3, 101, "2023-09-17"),
(3, 103, "2023-09-13"),
(4, 103, "2023-07-19"),
(4, 102, "2023-09-19"),
(4, 102, "2023-06-16"),
(5, 101, "2023-04-19"),
(5, 101, "2023-04-22");
truncate table Sales;
select * from Sales;
```

-- 1. What is the total amount each customer spent at the restaurant?

```
SELECT s.Cust_id, c.Cust_Name, SUM(Price) AS Total_Spent
FROM Sales AS s
JOIN Customer AS c
ON s.Cust_id = c.Cust_id
JOIN Menu AS m
ON s.Menu_id = m.Menu_id
GROUP BY Cust_id;
```

Cust_id	Cust_Name	Total_Spent
1	Yash	650
2	Sanskar	400
3	Kunal	250
5	Vivek	300
4	Anuj	500

-- 2. How many days has each customer visited the restaurant?

```
SELECT s.Cust_id, c.Cust_Name, COUNT(DISTINCT(s.Order_date)) AS Number_of_Visits
FROM Sales s
JOIN Customer c
ON s.Cust_id = c.Cust_id
GROUP BY Cust_id;
```

Cust_id	Cust_Name	Number_of_Visits
1	Yash	3
2	Sanskar	2
3	Kunal	2
4	Anuj	3
5	Vivek	2

-- 3. What was the first item from the menu purchased by each customer?

```
SELECT s.Cust_id, c.Cust_Name, s.Order_date, m.Menu_name,
DENSE_RANK() OVER (PARTITION BY Cust_id ORDER BY Order_date) as item_rank
FROM Sales s
JOIN Customer c
ON s.Cust_id = c.Cust_id
JOIN Menu m
ON s.Menu_id = m.Menu_id; -- Rank table (table with item_rank by order_date) - subquery
```

Cust_id	Cust_Name	Order_date	Menu_name	item_rank
1	Yash	2023-02-08	Cold coffee	1
1	Yash	2023-04-12	Pizza	2
1	Yash	2023-04-12	Jumbo Burger	2
1	Yash	2023-06-06	Pizza	3
2	Sanskar	2023-02-16	Cold coffee	1
2	Sanskar	2023-03-15	Jumbo Burger	2
2	Sanskar	2023-03-15	Jumbo Burger	2
3	Kunal	2023-09-13	Cold coffee	1
3	Kunal	2023-09-17	Jumbo Burger	2
4	Anuj	2023-06-16	Pizza	1
4	Anuj	2023-07-19	Cold coffee	2
4	Anuj	2023-09-19	Pizza	3
5	Vivek	2023-04-19	Jumbo Burger	1
5	Vivek	2023-04-22	Jumbo Burger	2

```
SELECT Cust_id, Cust_Name, Menu_name as First_order
FROM
(
	SELECT s.Cust_id, c.Cust_Name, s.Order_date, m.Menu_name,
	DENSE_RANK() OVER (PARTITION BY Cust_id ORDER BY Order_date) as item_rank
	FROM Sales s
	JOIN Customer c
	ON s.Cust_id = c.Cust_id
	JOIN Menu m
	ON s.Menu_id = m.Menu_id
) as rank_table
WHERE rank_table.item_rank=1
GROUP BY Cust_id, Menu_name;
```

Cust_id	Cust_Name	First_order
1	Yash	Cold coffee
2	Sanskar	Cold coffee
3	Kunal	Cold coffee
4	Anuj	Pizza
5	Vivek	Jumbo Burger

-- 4. What is the most purchased item on the menu and how many times was it purchased by all customers?

```
SELECT  s.Menu_id, Menu_name as Most_purchased_Item, (COUNT(s.Menu_id)) AS
Number_of_times_purchased
FROM Sales AS s
JOIN Menu AS m
ON s.Menu_id = m.Menu_id
GROUP BY s.Menu_id, Menu_name
ORDER BY Number_of_times_purchased DESC
LIMIT 1;
```

	Menu_id	Most_purchased_Item	Number_of_times_purchased
▶	101	Jumbo Burger	6

-- 5. Which item was the most popular for each customer?

```
SELECT s.Cust_id, c.Cust_Name, m.Menu_name, COUNT(m.Menu_id) AS order_count,
DENSE_RANK() OVER(PARTITION BY s.Cust_id ORDER BY COUNT(s.Menu_id)DESC) as rnk
FROM Sales as s
JOIN Menu as m ON s.Menu_id = m.Menu_id
JOIN Customer c
ON s.Cust_id = c.Cust_id
GROUP BY Cust_id, Menu_name; -- order rank (table with order_count and their
ranks) - subquery
```

Cust_id	Cust_Name	Menu_name	order_count	rnk
1	Yash	Pizza	2	1
1	Yash	Jumbo Burger	1	2
1	Yash	Cold coffee	1	2
2	Sanskar	Jumbo Burger	2	1
2	Sanskar	Cold coffee	1	2
3	Kunal	Jumbo Burger	1	1
3	Kunal	Cold coffee	1	1
4	Anuj	Pizza	2	1
4	Anuj	Cold coffee	1	2
5	Vivek	Jumbo Burger	2	1

```
SELECT Cust_id, Cust_Name, Menu_name,order_count
FROM
(
	SELECT s.Cust_id, c.Cust_Name, m.Menu_name, COUNT(m.Menu_id) AS
order_count,
	DENSE_RANK() OVER(PARTITION BY s.Cust_id ORDER BY COUNT(s.Menu_id)DESC) as
rnk
	FROM Sales as s
	JOIN Menu as m ON s.Menu_id = m.Menu_id
	JOIN Customer c
	ON s.Cust_id = c.Cust_id
	GROUP BY Cust_id, Menu_name
) AS order_rnk
WHERE RNK=1
GROUP BY Cust_id, Menu_name;
```

Cust_id	Cust_Name	Menu_name	order_count
1	Yash	Pizza	2
2	Sanskar	Jumbo Burger	2
3	Kunal	Jumbo Burger	1
3	Kunal	Cold coffee	1
4	Anuj	Pizza	2
5	Vivek	Jumbo Burger	2

-- 6. Which item was purchased first by the customer after they became a member?

```
SELECT s.Cust_id, c.Cust_Name, c.Join_date, s.Order_date, s.Menu_id,
RANK() OVER(PARTITION BY s.Cust_id ORDER BY s.Order_date) as rnk
FROM Sales as s
JOIN Customer as c
ON s.Cust_id=c.Cust_id
WHERE s.Order_date > c.Join_date; -- subquery
```

Cust_id	Cust_Name	Join_date	Order_date	Menu_id	rnk
1	Yash	2023-01-12	2023-02-08	103	1
1	Yash	2023-01-12	2023-04-12	102	2
1	Yash	2023-01-12	2023-04-12	101	2
1	Yash	2023-01-12	2023-06-06	102	4
2	Sanskar	2023-02-18	2023-03-15	101	1
2	Sanskar	2023-02-18	2023-03-15	101	1
3	Kunal	2023-03-14	2023-09-13	103	1
3	Kunal	2023-03-14	2023-09-17	101	2
4	Anuj	2023-04-20	2023-06-16	102	1
4	Anuj	2023-04-20	2023-07-19	103	2
4	Anuj	2023-04-20	2023-09-19	102	3

```
WITH after_mem_orders AS
(
	SELECT s.Cust_id, c.Cust_Name, s.Menu_id, s.Order_date,
	RANK() OVER(PARTITION BY s.Cust_id ORDER BY s.Order_date) as rnk
	FROM Sales as s
	JOIN Customer as c
	ON s.Cust_id=c.Cust_id
	WHERE s.Order_date > c.Join_date
)
SELECT amo.Cust_id, amo.Cust_Name, Menu.Menu_name, amo.Order_date
FROM after_mem_orders as amo
JOIN Menu
ON Menu.Menu_id=amo.Menu_id
WHERE rnk=1;
```

Cust_id	Cust_Name	Menu_name	Order_date
2	Sanskar	Jumbo Burger	2023-03-15
2	Sanskar	Jumbo Burger	2023-03-15
4	Anuj	Pizza	2023-06-16
1	Yash	Cold coffee	2023-02-08
3	Kunal	Cold coffee	2023-09-13

-- 7. Which item was purchased just before the customer became a member?

```
SELECT s.Cust_id, c.Cust_Name, c.Join_date, s.Order_date, s.Menu_id,
RANK() OVER(PARTITION BY s.Cust_id ORDER BY s.Order_date DESC) as rnk
FROM Sales as s
JOIN Customer as c
ON s.Cust_id=c.Cust_id
WHERE s.Order_date < c.Join_date; -- subquery
```

Cust_id	Cust_Name	Join_date	Order_date	Menu_id	rnk
2	Sanskar	2023-02-18	2023-02-16	103	1
5	Vivek	2023-05-16	2023-04-22	101	1
5	Vivek	2023-05-16	2023-04-19	101	2

```
WITH prior_mem_orders AS
(
	SELECT s.Cust_id, c.Cust_Name, c.Join_date, s.Order_date, s.Menu_id,
	RANK() OVER(PARTITION BY s.Cust_id ORDER BY s.Order_date DESC) as rnk
	FROM Sales as s
	JOIN Customer as c
	ON s.Cust_id=c.Cust_id
	WHERE s.Order_date < c.Join_date
)
SELECT pmo.Cust_id, pmo.Cust_Name, Menu.Menu_name, pmo.Order_date
FROM prior_mem_orders as pmo
JOIN Menu
ON Menu.Menu_id = pmo.Menu_id
WHERE rnk = 1;
```

Cust_id	Cust_Name	Menu_name	Order_date
5	Vivek	Jumbo Burger	2023-04-22
2	Sanskar	Cold coffee	2023-02-16

-- 8. What is the total items and amount spent for each member before they became a member?

```
SELECT s.Cust_id, Cust_Name, COUNT(s.Menu_id) as Total_orders_before_join_date,
SUM(m.Price) as Amount_Spent
FROM Customer as c
JOIN Sales as s
ON s.Cust_id = c.Cust_id
JOIN Menu as m
ON m.Menu_id = s.Menu_id
WHERE s.Order_date < c.Join_date
GROUP BY s.Cust_id;
```

Cust_id	Cust_Name	Total_orders_before_join_date	Amount_Spent
5	Vivek	2	300
2	Sanskar	1	100

-- 9. If each Ruppe 1 spent equates to 10 points and Jumbo Burger has a 2x points multiplier - how many points would each customer have?

```
SELECT s.Cust_id, Cust_Name,
SUM(CASE WHEN m.Menu_name = 'Jumbo Burger' THEN Price*20 ELSE Price*10 END) as
Total_Points
FROM Sales as s
JOIN Menu as m
ON m.Menu_id = s.Menu_id
JOIN Customer as c
ON s.Cust_id = c.Cust_id
GROUP BY s.Cust_id
ORDER BY s.Cust_id;
```

Cust_id	Cust_Name	Total_Points
1	Yash	8000
2	Sanskar	7000
3	Kunal	4000
4	Anuj	5000
5	Vivek	6000