CASE STUDY – Customer Behaviour Analysis on Dummy Data

It focuses on examining patterns, trends, and factors influencing customer spending in order to gain insights into their preferences, purchasing habits, and potential areas for improvement in menu offerings or marketing strategies in a dining establishment.

Background:

Fazrina seriously loves Japanese food so in the beginning of 2021, he decides to embark upon a risky venture and opens up a cute little restaurant that sells his 3 favourite foods: sushi, curry and ramen. Danny's Diner is in need of your assistance to help the restaurant stay afloat - 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.

```
SELECT *
FROM members
SELECT*
FROM sales
SELECT *
FROM menu
```

```
--1. What is the total amount each customer spent at the restaurant?
SELECT customer_id,sum(price) AS total_spent
FROM sales s
INNER JOIN menu m
ON s.product_id=m.product_id
GROUP BY customer_id
```

Results Messages

	customer_id	total_spent
1	A	76
2	B	74
3	C	36

```
--2. How many days has each customer visited the restaurant?
SELECT customer_id,COUNT(DISTINCT order_date) AS VISIT
FROM sales s
INNER JOIN menu m
ON s.product_id=m.product_id
GROUP BY customer_id
```

Results Messages

	customer_id	VISIT
1	A	4
2	B	6
3	C	2

```
--3. What was the first item from the menu purchased by each customer?
WITH cutomer_first AS
(SELECT customer_id,MIN(order_date) AS First_date
FROM sales
GROUP BY customer_id)
SELECT cf.customer_id,cf.First_date ,m.product_name
FROM cutomer_first cf
INNER JOIN sales s
ON s.customer_id=cf.customer_id
AND cf.First_date=s.order_date
INNER JOIN menu m ON m.product_id=s.product_id
```

Results | Messages

	customer_id	First_date	product_name
1	A	2021-01-01	sushi
2	A	2021-01-01	curry
3	B	2021-01-01	curry
4	C	2021-01-01	ramen
5	C	2021-01-01	ramen

```sql
-- 4. What is the most purchased item on the menu and how many times was it purchased
by all customers?
SELECT m.product_name,COUNT(product_name) Total_pucrchased
FROM sales s
INNER JOIN menu m
ON s.product_id=m.product_id
GROUP by product_name
ORDER by Total_pucrchased DESC
```

Results | Messages

	product_name	Total_pucrchased
1	ramen	8
2	curry	4
3	sushi	3

```sql
-- 5. Which item was the most popular for each customer?
WITH customer_popularity AS (
 SELECT s.customer_id, m.product_name, COUNT(*) AS purchase_count,
 DENSE_RANK() OVER (PARTITION BY s.customer_id ORDER BY COUNT(*) DESC) AS rank
 FROM sales s
 INNER JOIN menu m ON s.product_id = m.product_id
 GROUP BY s.customer_id, m.product_name
)
SELECT customer_id, product_name, purchase_count
FROM customer_popularity
WHERE rank = 1;
```

Results | Messages

	customer_id	product_name	purchase_count
1	A	ramen	3
2	B	sushi	2
3	B	curry	2
4	B	ramen	2
5	C	ramen	3

```sql
-- 6. Which item was purchased first by the customer after they became a member?
WITH first_purchase_after_membership AS (
 SELECT s.customer_id, MIN(s.order_date) AS first_purchase_date
 FROM dbo.sales s
 INNER JOIN dbo.members mb ON s.customer_id = mb.customer_id
 WHERE s.order_date >= mb.join_date
 GROUP BY s.customer_id
```

```
)
SELECT fpam.customer_id, m.product_name
FROM first_purchase_after_membership fpam
INNER JOIN dbo.sales s ON fpam.customer_id = s.customer_id
AND fpam.first_purchase_date = s.order_date
INNER JOIN dbo.menu m ON s.product_id = m.product_id;
```

	customer_id	product_name
1	A	curry
2	B	sushi

```
-- 7. Which item was purchased just before the customer became a member?
WITH last_purchase_after_membership AS (
 SELECT s.customer_id, Max(s.order_date) AS last_purchase_date
 FROM dbo.sales s
 INNER JOIN dbo.members mb ON s.customer_id = mb.customer_id
 WHERE s.order_date < mb.join_date
 GROUP BY s.customer_id
)
SELECT lpam.customer_id, m.product_name
FROM last_purchase_after_membership lpam
INNER JOIN dbo.sales s ON lpam.customer_id = s.customer_id
AND lpam.last_purchase_date = s.order_date
INNER JOIN dbo.menu m ON s.product_id = m.product_id;
```

Results Messages

	customer_id	product_name
1	A	sushi
2	A	curry
3	B	sushi

```
-- 8. What is the total items and amount spent by each member before they became a
member?
SELECT s.customer_id,COUNT(*) ABC,SUM(m.price) AS spent
FROM sales s
INNER JOIN menu m
ON s.product_id=m.product_id
INNER JOIN  members mb on s.customer_id=mb.customer_id
WHERE s.order_date < mb.join_date
GROUP BY s.customer_id ;
```

	customer_id	ABC	spent
1	A	2	25
2	B	3	40

```
-- 9. If each $1 spent equates to 10 points and sushi has a 2x points multiplier
-- how many points would each customer have?
SELECT s.customer_id, SUM(
CASE
```

```
WHEN m.product_name = 'sushi' THEN m.price*20
ELSE m.price*10 END) AS total_points
FROM dbo.sales s
INNER JOIN dbo.menu m ON s.product_id = m.product_id
GROUP BY s.customer_id;
```

Results Messages

	customer_id	total_points
1	A	860
2	B	940
3	C	360

```
/* 10. In the first week after a customer joins the program (including their join
date) they earn 2x points on all items, not just sushi
--how many points do customer A and B have at the end of January?*/
SELECT s.customer_id, SUM(
 CASE
 WHEN s.order_date BETWEEN mb.join_date AND DATEADD(day, 7, mb.join_date) THEN
m.price*20
 WHEN m.product_name = 'sushi' THEN m.price*20
 ELSE m.price*10
 END) AS total_points
FROM dbo.sales s
INNER JOIN dbo.menu m ON s.product_id = m.product_id
LEFT JOIN dbo.members mb ON s.customer_id = mb.customer_id
WHERE s.customer_id IN ('A', 'B') AND s.order_date <= '2021-01-31'
--WHERE s.customer_id = mb.customer_id AND s.order_date <= '2021-01-31'
GROUP BY s.customer_id;
```

Results Messages

	customer_id	total_points
1	A	1370
2	B	940

```
--Q11 show the customer A and B order the same product and on same date?
SELECT s.order_date, s.product_id, s.customer_id, m.product_name, m.price
FROM sales s
JOIN menu m ON s.product_id = m.product_id
WHERE s.order_date IN (
 SELECT order_date
 FROM sales
 WHERE customer_id = 'A'
)
AND s.order_date IN (
 SELECT order_date
 FROM sales
 WHERE customer_id = 'B'
)
AND s.product_id IN (
 SELECT product_id
 FROM sales
 WHERE customer_id = 'A'
 GROUP BY product_id
 HAVING COUNT(DISTINCT order_date) > 1
)
AND s.customer_id IN ('A', 'B');
```

Results Messages

	order_date	product_id	customer_id	product_name	price
1	2021-01-01	2	A	curry	15
2	2021-01-01	2	B	curry	15
3	2021-01-11	3	A	ramen	12
4	2021-01-11	3	A	ramen	12