**K GAUTHAM**

**SQL CASE STUDY**

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

SELECT

s.customer\_id,

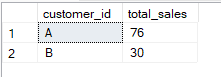
SUM(price) AS total\_sales

FROM dbo.sales AS s

JOIN dbo.menu AS m

ON s.product\_id = m.product\_id

GROUP BY customer\_id;



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

SELECT

customer\_id,

COUNT(DISTINCT(order\_date)) AS visit\_count

FROM dbo.sales

GROUP BY customer\_id;

Table

Description automatically generated

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

WITH ordered\_sales\_cte AS

(

SELECT

customer\_id,

order\_date,

product\_name,

DENSE\_RANK() OVER(PARTITION BY s.customer\_id ORDER BY s.order\_date) AS rank

FROM dbo.sales AS s

JOIN dbo.menu AS m

ON s.product\_id = m.product\_id

)

SELECT

customer\_id,

product\_name

FROM ordered\_sales\_cte

WHERE rank = 1

GROUP BY customer\_id, product\_name;

Table

Description automatically generated

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

SELECT

TOP 1 (COUNT(s.product\_id)) AS most\_purchased,

product\_name

FROM dbo.sales AS s

JOIN dbo.menu AS m

ON s.product\_id = m.product\_id

GROUP BY s.product\_id, product\_name

ORDER BY most\_purchased DESC;

Table

Description automatically generated

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

WITH fav\_item\_cte AS

(

SELECT

s.customer\_id,

m.product\_name,

COUNT(m.product\_id) AS order\_count,

DENSE\_RANK() OVER(PARTITION BY s.customer\_id ORDER BY COUNT(s.customer\_id) DESC) AS rank

FROM dbo.menu AS m

JOIN dbo.sales AS s

ON m.product\_id = s.product\_id

GROUP BY s.customer\_id, m.product\_name

)

SELECT

customer\_id,

product\_name,

order\_count

FROM fav\_item\_cte

WHERE rank = 1;

Graphical user interface, text, application

Description automatically generated

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

WITH member\_sales\_cte AS

(

SELECT

s.customer\_id,

m.join\_date,

s.order\_date,

s.product\_id,

DENSE\_RANK() OVER(PARTITION BY s.customer\_id ORDER BY s.order\_date) AS rank

FROM sales AS s

JOIN members AS m

ON s.customer\_id = m.customer\_id

WHERE s.order\_date >= m.join\_date)

SELECT

s.customer\_id,

s.order\_date,

m2.product\_name

FROM member\_sales\_cte AS s

JOIN menu AS m2

ON s.product\_id = m2.product\_id

WHERE rank = 1;

Table

Description automatically generated

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

WITH prior\_member\_purchased\_cte AS

(

SELECT

s.customer\_id,

m.join\_date,

s.order\_date,

s.product\_id,

DENSE\_RANK() OVER(PARTITION BY s.customer\_id ORDER BY s.order\_date DESC) AS rank

FROM sales AS s

JOIN members AS m

ON s.customer\_id = m.customer\_id

WHERE s.order\_date < m.join\_date

)

SELECT

s.customer\_id,

s.order\_date,

m2.product\_name

FROM prior\_member\_purchased\_cte AS s

JOIN menu AS m2

ON s.product\_id = m2.product\_id

WHERE rank = 1;

Table

Description automatically generated

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

SELECT

s.customer\_id,

COUNT(DISTINCT s.product\_id) AS unique\_menu\_item,

SUM(mm.price) AS total\_sales

FROM sales AS s

JOIN members AS m

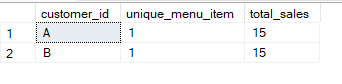
ON s.customer\_id = m.customer\_id

JOIN menu AS mm

ON s.product\_id = mm.product\_id

WHERE s.order\_date < m.join\_date

GROUP BY s.customer\_id



9)If each $1 spent equates to 10 points and sushi has a 2x points multiplier - how many points would each customer have?

WITH price\_points\_cte AS

(

SELECT \*,

CASE WHEN product\_name = 'sushi' THEN price \* 20

ELSE price \* 10 END AS points

FROM menu

)

SELECT

s.customer\_id,

SUM(p.points) AS total\_points

FROM price\_points\_cte AS p

JOIN sales AS s

ON p.product\_id = s.product\_id

GROUP BY s.customer\_id

Table

Description automatically generated

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?

-- 1. Find member validity date of each customer and get last date of January

-- 2. Use CASE WHEN to allocate points by date and product id

-- 3. SUM price and points

WITH dates\_cte AS

(

SELECT

\*,

DATEADD(DAY, 6, join\_date) AS valid\_date,

EOMONTH('2023/04/05') AS last\_date

FROM members AS m

)

SELECT

d.customer\_id, s.order\_date, d.join\_date, d.valid\_date, d.last\_date, m.product\_name,

m.price,

SUM(

CASE WHEN m.product\_name = 'sushi' THEN 2 \* 10 \* m.price

WHEN s.order\_date BETWEEN d.join\_date AND d.valid\_date THEN 2 \* 10 \* m.price

ELSE 10 \* m.price END) AS points

FROM dates\_cte AS d

JOIN sales AS s

ON d.customer\_id = s.customer\_id

JOIN menu AS m

ON s.product\_id = m.product\_id

WHERE s.order\_date < d.last\_date

GROUP BY d.customer\_id, s.order\_date, d.join\_date, d.valid\_date, d.last\_date, m.product\_name, m.price

