8WEEKSQLCHALLENGE.COM

CASE STUDY #1

THE TASTE OF SUCCESS

DATAWITHDANNY.COM

Case Study Questions

Each of the following case study questions can be answered using a single SQL statement:

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

Bonus Questions

Join All the Things

11. The following questions are related creating basic data tables that Danny and his team can use to quickly derive insights without needing to join the underlying tables using SQL.

Rank All the Things

12. Danny also requires further information about the ranking of customer products, but he purposely does not need the ranking for non-member purchases so he expects null ranking values for the records when customers are not yet part of the loyalty program.

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

customer_id	total_amount_spent
A	76
B	74
C	36

```sql
-- 2. How many days has each customer visited the restaurant?
SELECT
    customer_id,
    COUNT(DISTINCT order_date) AS customer_visited_days
FROM
    sales
GROUP BY customer_id
```

customer_id	customer_visited_days
A	4
B	6
C	2

```sql
-- 3. What was the first item from the menu purchased by each customer?
WITH cte
AS (SELECT customer_id,
           product_name,
           order_date,
           RANK() OVER (PARTITION BY customer_id ORDER BY order_date ASC) AS rnk
    FROM sales s
        JOIN menu m
            ON s.product_id = m.product_id
   )
SELECT customer_id,
       product_name,
       order_date
FROM cte
WHERE rnk = 1
```

customer_id	product_name	order_date
A	sushi	2021-01-01
A	curry	2021-01-01
B	curry	2021-01-01
C	ramen	2021-01-01
C	ramen	2021-01-01

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

SELECT
    product_name AS most_purchased_item,
    COUNT(*) AS no_of_orders
FROM
    sales s
        JOIN
    menu m ON s.product_id = m.product_id
GROUP BY product_name
ORDER BY no_of_orders DESC
LIMIT 1
```

most_purchased_item	no_of_orders
ramen	8

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

WITH cte
AS (SELECT customer_id,
           product_name,
           count(*) AS orders,
           RANK() OVER (PARTITION BY customer_id ORDER BY count(*) DESC) AS rnk
    FROM sales s
        JOIN menu m
            ON s.product_id = m.product_id
    GROUP BY customer_id,
             product_name
   )
SELECT customer_id,
       product_name AS most_popular_product_for_customer
FROM cte
WHERE rnk = 1
```

customer_id	most_popular_product_for_customer
A	ramen
B	curry
B	sushi
B	ramen
C	ramen

```sql
-- 6. Which item was purchased first by the customer after they became a member?
WITH cte
AS (SELECT s.customer_id AS customer_id,
           s.product_id AS product_id,
           product_name,
           order_date,
           RANK() OVER (PARTITION BY s.customer_id ORDER BY order_date) as rnk
    FROM sales s
        JOIN members me
            ON s.customer_id = me.customer_id
        JOIN menu m
            ON s.product_id = m.product_id
    where order_date >= join_date
   )
SELECT customer_id,
       product_id,
       product_name
from cte
WHERE rnk = 1
```

customer_id	product_id	product_name
A	2	curry
B	1	sushi

```sql
-- 7. Which item was purchased just before the customer became a member?
WITH cte
AS (SELECT s.customer_id AS customer_id,
           s.product_id AS product_id,
           product_name,
           order_date,
           RANK() OVER (PARTITION BY s.customer_id ORDER BY order_date DESC) AS rnk
    FROM sales s
        JOIN members me
            ON s.customer_id = me.customer_id
        JOIN menu m
            ON s.product_id = m.product_id
    where order_date < join_date
   )
SELECT customer_id,
       product_id,
       product_name,
       order_date
FROM cte
WHERE rnk = 1
```

customer_id	product_id	product_name	order_date
A	1	sushi	2021-01-01
A	2	curry	2021-01-01
B	1	sushi	2021-01-04

```sql
-- 8. What is the total items and amount spent for each member
-- before they became a member?
SELECT s.customer_id AS customer_id,
       COUNT(order_date) AS total_items,
       sum(price) as amount_spent
FROM sales s
    JOIN members me
        ON s.customer_id = me.customer_id
    JOIN menu m
        ON s.product_id = m.product_id
WHERE order_date < join_date
GROUP BY s.customer_id
```

customer_id	total_items	amount_spent
B	3	40
A	2	25

```sql
-- 9.  If each $1 spent equates to 10 points and sushi has a 2x points multiplier - how
    many points would each customer have?
SELECT customer_id,
       sum(price) AS amount_spent,
       sum(   CASE
                  WHEN product_name = 'sushi' THEN
                      price * 2 * 10
                  ELSE
                      price * 10
              END
          ) AS points
FROM sales s
    JOIN menu m
        ON s.product_id = m.product_id
GROUP BY s.customer_id
```

	customer_id	amount_spent	points
▶	A	76	860
	B	74	940
	C	36	360

```sql
-- 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(price) AS amount_spent,
  SUM(
    CASE
      WHEN order_date BETWEEN join_date AND ADDDATE(join_date,INTERVAL 6 DAY) THEN price * 10 * 2
      WHEN product_name = 'sushi' THEN price * 10 * 2
      ELSE price * 10
    END
  ) as points
FROM sales s
JOIN menu m
ON s.product_id = m.product_id
JOIN members me
ON me.customer_id = s.customer_id
where MONTH(order_date)=1
GROUP BY s.customer_id
ORDER BY points DESC
```

	customer_id	amount_spent	points
▶	A	76	1370
	B	62	820

```sql
-- 11. Joining all the things
SELECT s.customer_id AS customer_id,
       order_date,
       product_name,
       price,
       CASE
           WHEN join_date is null
                OR order_date < join_date THEN
               'N'
           ELSE
               'Y'
       END AS member
FROM sales s
    LEFT JOIN members me
        ON s.customer_id = me.customer_id
    LEFT JOIN menu m
        ON s.product_id = m.product_id
```

customer_id	order_date	product_name	price	member
A	2021-01-01	sushi	10	N
A	2021-01-01	curry	15	N
A	2021-01-07	curry	15	Y
A	2021-01-10	ramen	12	Y
A	2021-01-11	ramen	12	Y
A	2021-01-11	ramen	12	Y
B	2021-01-01	curry	15	N
B	2021-01-02	curry	15	N
B	2021-01-04	sushi	10	N
B	2021-01-11	sushi	10	Y
B	2021-01-16	ramen	12	Y
B	2021-02-01	ramen	12	Y
C	2021-01-01	ramen	12	N
C	2021-01-01	ramen	12	N
C	2021-01-07	ramen	12	N

```sql
-- 12. Ranking

WITH cte
AS (SELECT s.customer_id AS customer_id,
           order_date,
           product_name,
           price,
           CASE
               WHEN join_date is null
                    OR order_date < join_date THEN
                   'N'
               ELSE
                   'Y'
           END AS member
    FROM sales s
        LEFT JOIN members me
            ON s.customer_id = me.customer_id
        LEFT JOIN menu m
            ON s.product_id = m.product_id
   )
SELECT *,
       CASE
           WHEN member = 'N' THEN
               null
           ELSE
               RANK() OVER (PARTITION BY customer_id, member ORDER BY order_date)
       END AS ranking
FROM cte
```

customer_id	order_date	product_name	price	member	ranking
A	2021-01-01	sushi	10	N	NULL
A	2021-01-01	curry	15	N	NULL
A	2021-01-07	curry	15	Y	1
A	2021-01-10	ramen	12	Y	2
A	2021-01-11	ramen	12	Y	3
A	2021-01-11	ramen	12	Y	3
B	2021-01-01	curry	15	N	NULL
B	2021-01-02	curry	15	N	NULL
B	2021-01-04	sushi	10	N	NULL
B	2021-01-11	sushi	10	Y	1
B	2021-01-16	ramen	12	Y	2
B	2021-02-01	ramen	12	Y	3
C	2021-01-01	ramen	12	N	NULL
C	2021-01-01	ramen	12	N	NULL
C	2021-01-07	ramen	12	N	NULL