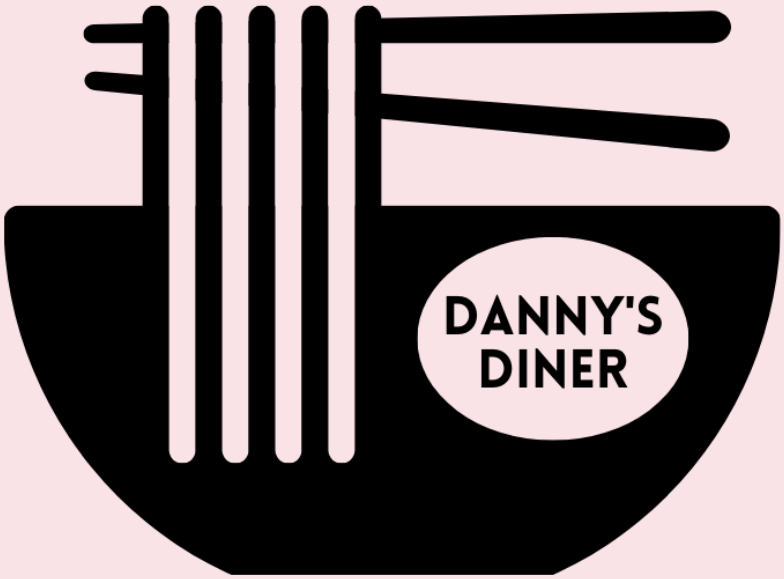
Case Study #1 - Danny's Diner

8WEEKSQLCHALLENGE.COM

CASE STUDY #1



THE TASTE OF SUCCESS

DATAWITHDANNY.COM

Problem Statement

Danny wants 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 also 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.

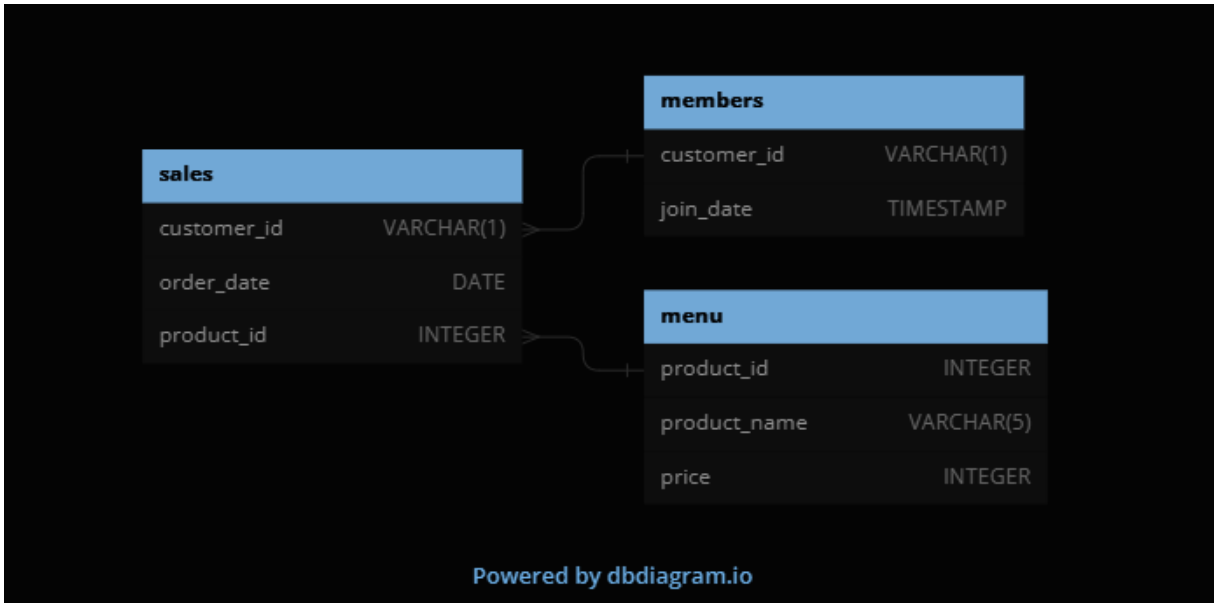
He plans on using these insights to help him decide whether he should expand the existing customer loyalty program - additionally he needs help to generate some basic datasets so his team can easily inspect the data without needing to use SQL.

Danny has provided you with a sample of his overall customer data due to privacy issues - but he hopes that these examples are enough for you to write fully functioning SQL queries to help him answer his questions!

Danny has shared with you 3 key datasets for this case study:

- Sales
- Menu
- Members

You can inspect the entity relationship diagram and example data below.



Case Study Questions

Q1. What is the total amount each customer spent at the restaurant?

```
--Q1.What is the total amount each customer spent at the restaurant?
SELECT s.customer_id, SUM(m.price) AS total_amount
FROM sales AS s
INNER JOIN menu AS m ON s.product_id = m.product_id
GROUP BY customer_id
ORDER BY customer_id;
```

	customer_id character varying (1)	total_amount bigint
1	A	76
2	B	74
3	C	36

Q2. How many days has each customer visited the restaurant?

```
--Q2.How many days has each customer visited the restaurant?
SELECT customer_id, COUNT(DISTINCT(order_date)) AS total_days
FROM sales
GROUP BY customer_id
ORDER BY customer_id;
```

	customer_id character varying (1)	total_days bigint
1	A	4
2	B	6
3	C	2

Q3. What was the first item from the menu purchased by each customer?

```
--Q3.What was the first item from the menu purchased by each customer?
SELECT DISTINCT(s.customer_id), m.product_name, s.order_date
FROM sales AS s
INNER JOIN menu AS m ON s.product_id = m.product_id
WHERE s.order_date = (Select MIN(order_date) FROM sales)
ORDER BY s.customer_id;
```

	customer_id character varying (1)	product_name character varying (5)	order_date date
1	A	curry	2021-01-01
2	A	sushi	2021-01-01
3	B	curry	2021-01-01
4	C	ramen	2021-01-01

Q4. What is the most purchased item on the menu and how many times was it purchased by all customers?

```
--Q4.What is the most purchased item on the menu and how many times was it
--purchased by all customers?
SELECT m.product_name, COUNT(*) AS purchase_count
FROM sales s
JOIN menu m ON s.product_id = m.product_id
GROUP BY m.product_name
ORDER BY purchase_count DESC
LIMIT 1;
```

	product_name character varying (5)	purchase_count bigint
1	ramen	8

Q5. Which item was the most popular for each customer?

```sql
--Q5.Which item was the most popular for each customer?
WITH customer_most_popular_item AS (
  SELECT
    s.customer_id,
    m.product_name,
    COUNT(*) AS purchase_count,
    RANK() OVER (PARTITION BY s.customer_id ORDER BY COUNT(*) DESC) AS item_rank
  FROM
    sales s
    JOIN menu m ON s.product_id = m.product_id
  GROUP BY
    s.customer_id, m.product_name
)

SELECT
  customer_id,
  product_name AS most_popular_item,
  purchase_count
FROM
  customer_most_popular_item
WHERE
  item_rank = 1;
```

	customer_id character varying (1)	most_popular_item character varying (5)	purchase_count bigint
1	A	ramen	3
2	B	sushi	2
3	B	curry	2
4	B	ramen	2
5	C	ramen	3

Q6. Which item was purchased first by the customer after they became a member?

```sql
--Q6.Which item was purchased first by the customer after they became a member?
WITH customer_first_purchase AS (
  SELECT
    s.customer_id,
    m.product_name,
    s.order_date,
    mem.join_date,
    ROW_NUMBER() OVER (PARTITION BY s.customer_id ORDER BY s.order_date) AS purchase_rank
  FROM
    sales s
    JOIN menu m ON s.product_id = m.product_id
    JOIN members mem ON s.customer_id = mem.customer_id
  WHERE
    s.order_date >= mem.join_date
)

SELECT
  customer_id,
  product_name AS first_purchase_item,
  order_date AS purchase_date
FROM
  customer_first_purchase
WHERE
  purchase_rank = 1;
```

	customer_id character varying (1)	first_purchase_item character varying (5)	purchase_date date
1	A	curry	2021-01-07
2	B	sushi	2021-01-11

Q7.Which item was purchased just before the customer became a member?

```sql
--Q7.Which item was purchased just before the customer became a member?
WITH customer_last_purchase_before_membership AS (
  SELECT
    s.customer_id,
    m.product_name,
    s.order_date,
    LAG(mem.join_date) OVER (PARTITION BY s.customer_id ORDER BY s.order_date) AS previous_membership_date
  FROM
    sales s
    JOIN menu m ON s.product_id = m.product_id
    LEFT JOIN members mem ON s.customer_id = mem.customer_id
)

SELECT
  customer_id,
  product_name AS last_purchase_before_membership_item,
  order_date AS last_purchase_date
FROM
  customer_last_purchase_before_membership
WHERE
  order_date = previous_membership_date;
```

	customer_id character varying (1)	last_purchase_before_membership_item character varying (5)	last_purchase_date date
1	A	curry	2021-01-07

Q8. What is the total items and amount spent for each member before they became a member?

```sql
--Q8.What is the total items and amount spent for each member before they became a member?
WITH member_purchase_summary AS (
  SELECT
    s.customer_id,
    COUNT(*) AS total_items,
    SUM(m.price) AS total_amount_spent
  FROM
    sales s
    JOIN menu m ON s.product_id = m.product_id
    LEFT JOIN members mem ON s.customer_id = mem.customer_id
  WHERE
    s.order_date < mem.join_date OR mem.join_date IS NULL
  GROUP BY
    s.customer_id
)

SELECT
  customer_id,
  total_items,
  total_amount_spent
FROM
  member_purchase_summary;
```

	customer_id character varying (1)	total_items bigint	total_amount_spent bigint
1	B	3	40
2	C	3	36
3	A	2	25

Q9.If each $1 spent equates to 10 points and sushi has a 2x points multiplier - how many points would each customer have?

```sql
--Q9.If each $1 spent equates to 10 points and sushi has a 2x points multiplier - how many points would each customer have?
WITH customer_points AS (
  SELECT
    s.customer_id,
    SUM(CASE WHEN m.product_name = 'sushi' THEN 2 * m.price ELSE m.price END) AS total_amount_spent
  FROM
    sales s
    JOIN menu m ON s.product_id = m.product_id
  GROUP BY
    s.customer_id
)

SELECT
  customer_id,
  total_amount_spent * 10 AS total_points
FROM
  customer_points;
```

	customer_id character varying (1)	total_points bigint
1	B	940
2	C	360
3	A	860

Q10.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?

```sql
WITH customer_points AS (
  SELECT
    s.customer_id,
    SUM(
      CASE
        WHEN s.order_date < mem.join_date + INTERVAL '7 days' THEN 2 * m.price
        ELSE m.price
      END
    ) AS total_amount_spent
  FROM
    sales s
    JOIN menu m ON s.product_id = m.product_id
    LEFT JOIN members mem ON s.customer_id = mem.customer_id
  WHERE
    s.order_date <= '2021-01-31' AND s.order_date >= mem.join_date
  GROUP BY
    s.customer_id
)
SELECT
  customer_id,
  total_amount_spent * 10 AS total_points
FROM
  customer_points
WHERE
  customer_id IN ('A', 'B');
```

	customer_id character varying (1)	total_points bigint
1	A	1020
2	B	320