8WEEKSQLCHALLENGE.COM

CASE STUDY #1

THE TASTE OF SUCCESS

DATAWITHDANNY.COM

Introduction

- Danny 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.

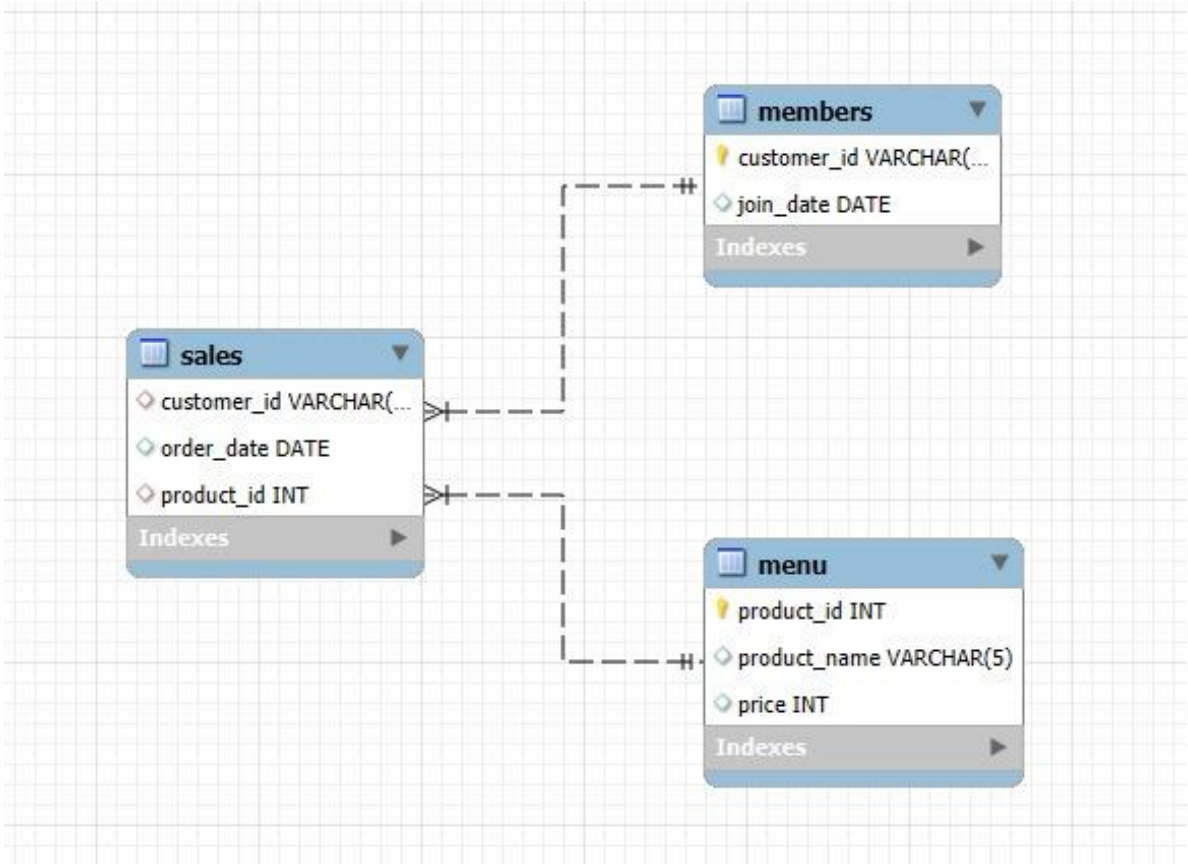
Problem Statement

Danny wants to analyze customer behavior using data from sales, menu, and members tables. He aims to understand visit patterns, spending habits, and popular menu items to personalize customer experiences. These insights will guide his decision on expanding the loyalty program. He needs SQL queries and basic datasets for easy team access.

Case Study Questions

- What is the total amount each customer spent at the restaurant?
- How many days has each customer visited the restaurant?
- What was the first item from the menu purchased by each customer?
- What is the most purchased item on the menu and how many times was it purchased by all customers?
- Which item was the most popular for each customer?
- Which item was purchased first by the customer after they became a member?
- Which item was purchased just before the customer became a member?
- What is the total items and amount spent for each member before they became a member?
- If each $1 spent equates to 10 points and sushi has a 2x points multiplier - how many points would each customer have?
- 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?

Entity Relationship Diagram



Creating database

```sql
-- 1. Create the database
CREATE DATABASE danny_diner;
USE danny_diner;
```

```sql
-- 2. Create the menu table
CREATE TABLE menu (
    product_id INTEGER PRIMARY KEY,
    product_name VARCHAR(5),
    price INTEGER
);
```

Creating Menu Table

```sql
-- 3. Insert data into menu
INSERT INTO menu (product_id, product_name, price)
VALUES
  (1, 'sushi', 10),
  (2, 'curry', 15),
  (3, 'ramen', 12);
```

Insert Values in Menu Table

```sql
-- 4. Create the members table
CREATE TABLE members (
    customer_id VARCHAR(1) PRIMARY KEY,
    join_date DATE
);
```

Creating Members Table

```sql
-- 5. Insert data into members
INSERT INTO members (customer_id, join_date)
VALUES
  ('A', '2021-01-07'),
  ('B', '2021-01-09');
```

Insert Values in members Table

```sql
-- 6. Create the sales table
CREATE TABLE sales (
  customer_id  VARCHAR(1),
  order_date DATE,
  product_id INTEGER,
  FOREIGN KEY (customer_id) REFERENCES members(customer_id),
  FOREIGN KEY (product_id) REFERENCES menu(product_id)
);
```

Creating Sales Table

```sql
-- 7. Insert data into sales
INSERT INTO sales (customer_id, order_date, product_id)
VALUES
  ('A', '2021-01-01', 1),
  ('A', '2021-01-01', 2),
  ('A', '2021-01-07', 2),
  ('A', '2021-01-10', 3),
  ('A', '2021-01-11', 3),
  ('A', '2021-01-11', 3),
  ('B', '2021-01-01', 2),
  ('B', '2021-01-02', 2),
  ('B', '2021-01-04', 1),
  ('B', '2021-01-11', 1),
  ('B', '2021-01-16', 3),
  ('B', '2021-02-01', 3);
```

Insert Values in sales Table

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

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

Result Grid | Filter Rows:

customer_id	Total_Amount
A	76
B	74

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

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

Result Grid | Filter Rows:

customer_id	visit_days
A	4
B	6

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

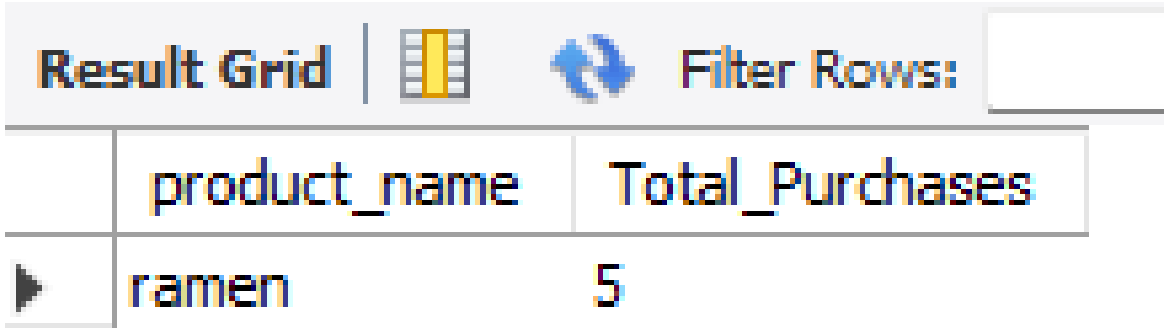
```sql
-- 3. What was the first item from the menu purchased by each customer?
SELECT
    product_name, customer_id, order_date, ROW_NUMBER() OVER
    (PARTITION BY customer_id ORDER BY order_date) as rank_

FROM
    menu
        JOIN
    sales ON menu.product_id = sales.product_id
```

Result Grid | Filter Rows: | Exp

product_name	customer_id	order_date	rank_
sushi	A	2021-01-01	1
curry	A	2021-01-01	2
curry	A	2021-01-07	3
ramen	A	2021-01-10	4
ramen	A	2021-01-11	5
ramen	A	2021-01-11	6
curry	B	2021-01-01	1
curry	B	2021-01-02	2
sushi	B	2021-01-04	3
sushi	B	2021-01-11	4
ramen	B	2021-01-16	5
ramen	B	2021-02-01	6

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

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



product_name	Total_Purchases
ramen	5

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

```sql
-- 5. Which item was the most popular for each customer?
select customer_id, product_name, order_count
from
(
SELECT
    customer_id, product_name, count(*) as order_count, row_number()
    over (partition by customer_id order by count(*)  desc) as rank_
FROM
    menu
        JOIN
    sales ON menu.product_id = sales.product_id
group by customer_id, product_name ) as t1
where rank_ = 1
```

Result Grid | Filter Rows:

customer_id	product_name	order_count
A	ramen	3
B	sushi	2

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

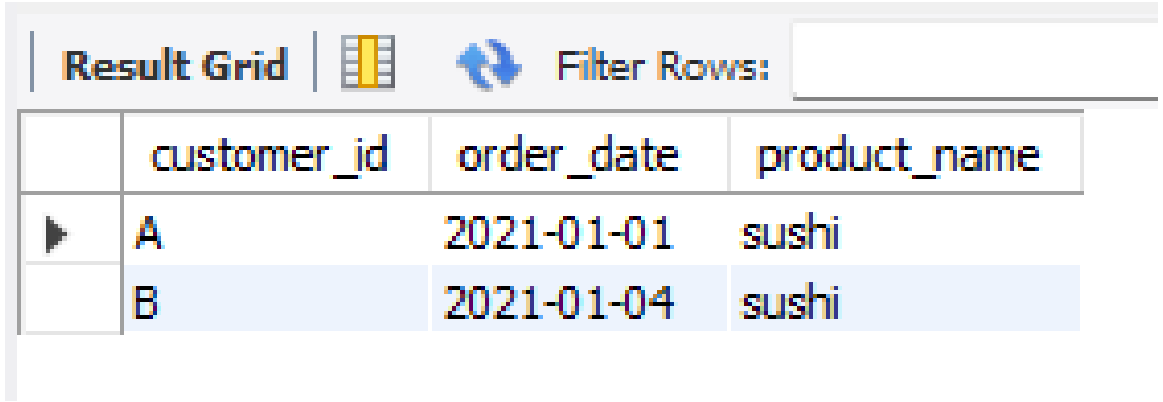
```sql
-- 6. Which item was purchased first by the customer after they became a member?
SELECT customer_id, order_date, product_name
FROM
(
SELECT
    s.customer_id, mem.join_date, m.product_name, s.order_date, row_number()
    over(partition by s.customer_id order by s.order_date) as rank_
FROM
    sales s
        JOIN
    menu m ON m.product_id = s.product_id
    join
    members mem on s.customer_id = mem.customer_id
WHERE s.order_date >= mem.join_date ) as t1
where rank_ = 1
```

Result Grid | Filter Rows:

customer_id	order_date	product_name
A	2021-01-07	curry
B	2021-01-11	sushi

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

```sql
-- 7. Which item was purchased just before the customer became a member?
SELECT customer_id, order_date, product_name
FROM
(
SELECT
    s.customer_id, mem.join_date, m.product_name, s.order_date, row_number()
    over(partition by s.customer_id order by s.order_date desc) as rank_
FROM
    sales s
        JOIN
    menu m ON m.product_id = s.product_id
    join
    members mem on s.customer_id = mem.customer_id
WHERE s.order_date < mem.join_date ) as t1
where rank_ = 1
```



customer_id	order_date	product_name
A	2021-01-01	sushi
B	2021-01-04	sushi

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

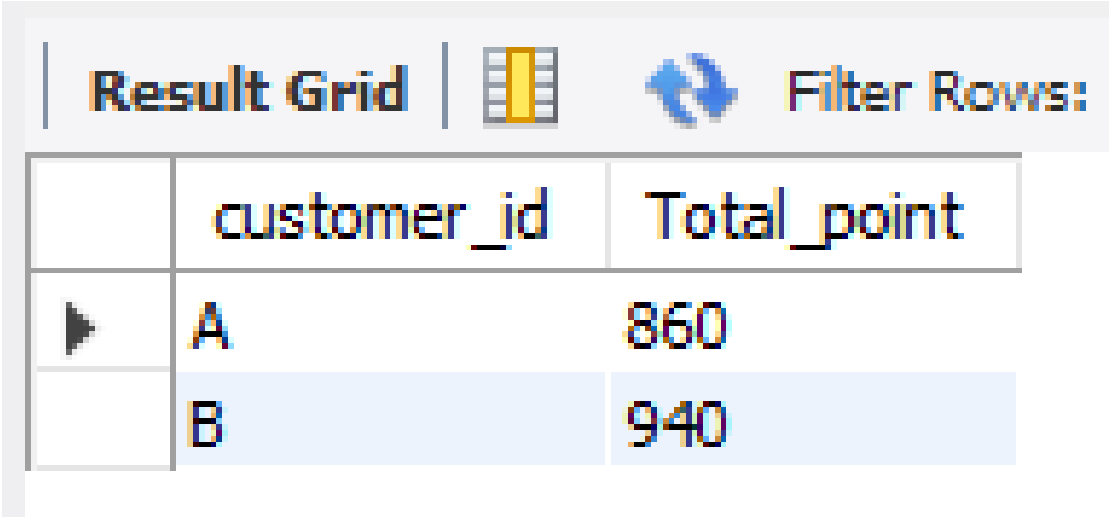
```sql
-- 8. What is the total items and amount spent for each member before they became a member?
SELECT
    s.customer_id, Count(*) as Total_Item, SUM(m.price) as Total_Amount
FROM
    sales s
        JOIN
    menu m ON m.product_id = s.product_id
    join
    members mem on s.customer_id = mem.customer_id
WHERE s.order_date < mem.join_date
group by s.customer_id
```

Result Grid | Filter Rows:

customer_id	Total_Item	Total_Amount
A	2	25
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?

```sql
SELECT
    s.customer_id,
    SUM(
    CASE
    WHEN m.product_name = 'sushi' THEN m.price * 20
    ELSE m.price * 10
    END
    ) AS Total_point
FROM
    menu m
        JOIN
    sales s ON m.product_id = s.product_id
GROUP BY s.customer_id
```



customer_id	Total_point
A	860
B	940

-- 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?

```sql
SELECT
    s.customer_id,
    SUM(
        CASE
            -- Double points for all items in the first 7 days after joining
            WHEN s.order_date BETWEEN mem.join_date AND DATE_ADD(mem.join_date, INTERVAL 6 DAY)
            THEN m.price * 20

            -- Double points only for sushi outside of first 7 days
            WHEN m.product_name = 'sushi' THEN m.price * 20

            -- Regular points otherwise
            ELSE m.price * 10
        END
    ) AS total_points
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 BETWEEN '2021-01-01' AND '2021-01-31'
GROUP BY s.customer_id;
```

Result Grid | Filter Rows:

customer_id	total_points
A	1370
B	820