

# Danny's Diner Project – MYSQL

## 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 3 key datasets for this case study:

- sales
- menu
- members

Let us start by creating the database.

Step 1: Create Database danny's\_diner;

Step 2: Now let us create our tables.

1. Sales table: Contains fields like customer\_id, order\_date and product\_id.

```
CREATE TABLE sales (  
  customer_id VARCHAR(1),  
  order_date date,  
  product_id int  
);
```

2. Menu table: Contains fields such as product\_id, product\_name and price.

```
CREATE TABLE menu (  
  product_id INT,  
  product_name VARCHAR(10),  
  price INT  
);
```

3. Members table: Contains fields such as customer\_id and join\_date.

```
CREATE TABLE members (  
  customer_id VARCHAR(1),  
  join_date DATE  
);
```

Step 3: Inserting values into the tables.

1. Sales table:

```
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'),  
  ('C', '2021-01-01', '3'),  
  ('C', '2021-01-01', '3'),  
  ('C', '2021-01-07', '3');
```

2. Menu table:

```
INSERT INTO menu  
  (product_id, product_name, price)  
VALUES  
  ('1', 'sushi', '10'),  
  ('2', 'curry', '15'),  
  ('3', 'ramen', '12');
```

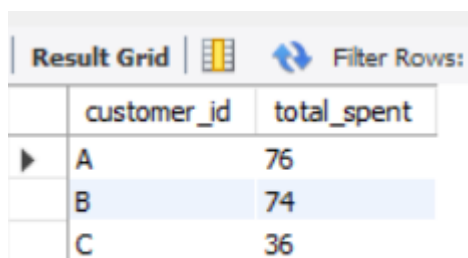
3. Members table:

```
INSERT INTO members
(customer_id, join_date)
VALUES
('A', '2021-01-07'),
('B', '2021-01-09');
```

Questions answered:

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

```
SELECT
  s.customer_id, SUM(m.price) AS total_spent
FROM
  sales s
  INNER JOIN
  menu m ON s.product_id = m.product_id
GROUP BY s.customer_id;
```





The screenshot shows a database interface with a 'Result Grid' tab. It contains a table with two columns: 'customer\_id' and 'total\_spent'. There are three rows of data: Customer A with a total spent of 76, Customer B with 74, and Customer C with 36. The row for Customer B is highlighted in blue.

	customer_id	total_spent
▶	A	76
	B	74
	C	36

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
	C	2

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

```

WITH CTE AS (
  SELECT
    s.customer_id,
    s.order_date,
    m.product_name,
    DENSE_RANK() OVER (
      PARTITION BY s.customer_id
      ORDER BY s.order_date) AS rnk
  FROM sales s
  INNER JOIN menu m
    ON s.product_id = m.product_id
)

SELECT
  customer_id,
  product_name
FROM CTE
WHERE rnk = 1
GROUP BY customer_id, product_name;

```

Result Grid		Filter Rows:
	customer_id	product_name
▶	A	sushi
	A	curry
	B	curry
	C	ramen

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(s.product\_id) AS most\_purchased\_item

FROM

sales s

INNER JOIN

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

GROUP BY m.product\_name

ORDER BY most\_purchased\_item DESC

LIMIT 1;

Result Grid		Filter Rows:
	product_name	most_purchased_item
▶	ramen	8

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

WITH most\_popular 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 rnk

FROM menu m


INNER JOIN sales s

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

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

)

```
SELECT
  customer_id,
  product_name,
  order_count
FROM most_popular
WHERE rnk = 1;
```

Result Grid  Filter Rows: <input data-bbox="683 544 933 600" type="text"/>			
	customer_id	product_name	order_count
▶	A	ramen	3
	B	curry	2
	B	sushi	2
	B	ramen	2
	C	ramen	3