**📘 Danny’s Diner – SQL Case Study**

**Author**: Golla Karthik  
**GitHub**: <https://github.com/zlatangk/dannys-diner-sql>  
**LinkedIn:** [**https://www.linkedin.com/in/gollakarthk/**](https://www.linkedin.com/in/gollakarthk/)

**🧩 1. Introduction**

Danny’s Diner is a small Japanese-style restaurant that wants to better understand customer behavior and menu performance. By analyzing sales transactions, menu pricing, and membership history, this case study answers key business questions and provides actionable insights.

**🗃️ 2. Data Overview**

The case study is based on three relational tables:

| **Table** | **Description** |
| --- | --- |
| sales | Contains purchase history, including customer ID, order date, and product ID |
| menu | Details of each menu item including product name and price |
| members | Membership details such as customer ID and join date |

**🧮 3. SQL Queries and Answers**

A total of 10 business questions were solved using SQL, covering spend analysis, visit frequency, product popularity, membership influence, and reward points.  
Each query was written using **JOINs**, **window functions**, **CTEs**, and **aggregations** to generate clear insights.

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

SELECT customer\_id, SUM(price) AS total\_amount\_spent  
FROM menu AS m  
JOIN sales AS s ON m.product\_id = s.product\_id  
GROUP BY customer\_id;

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

SELECT customer\_id, COUNT(DISTINCT order\_date) AS total\_visits  
FROM sales  
GROUP BY customer\_id;

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

WITH cte AS (  
 SELECT m.product\_name, s.customer\_id, s.order\_date,  
 RANK() OVER(PARTITION BY customer\_id ORDER BY order\_date ASC) AS rn  
 FROM menu AS m  
 JOIN sales AS s ON m.product\_id = s.product\_id  
)  
SELECT order\_date, customer\_id, product\_name  
FROM cte WHERE rn = 1;

**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 AS m  
JOIN sales AS s ON m.product\_id = s.product\_id  
GROUP BY product\_name  
ORDER BY total\_purchases DESC;

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

WITH cte AS (  
 SELECT customer\_id, m.product\_name, COUNT(order\_date) AS times\_ordered  
 FROM sales AS s  
 JOIN menu AS m ON m.product\_id = s.product\_id  
 GROUP BY customer\_id, m.product\_name  
),  
ranked\_cte AS (  
 SELECT customer\_id, product\_name, times\_ordered,  
 RANK() OVER(PARTITION BY customer\_id ORDER BY times\_ordered DESC) AS rnk  
 FROM cte  
)  
SELECT customer\_id, product\_name, times\_ordered  
FROM ranked\_cte WHERE rnk = 1;

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

WITH cte AS (  
 SELECT m.customer\_id, join\_date, me.product\_name, order\_date  
 FROM sales AS s  
 JOIN members AS m ON s.customer\_id = m.customer\_id  
 JOIN menu AS me ON s.product\_id = me.product\_id  
 WHERE order\_date >= join\_date  
),  
ranked\_cte AS (  
 SELECT customer\_id, product\_name, order\_date,  
 RANK() OVER(PARTITION BY customer\_id ORDER BY order\_date ASC) AS rnk  
 FROM cte  
)  
SELECT \* FROM ranked\_cte WHERE rnk = 1;

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

WITH cte AS (  
 SELECT m.customer\_id, join\_date, me.product\_name, order\_date  
 FROM sales AS s  
 JOIN members AS m ON s.customer\_id = m.customer\_id  
 JOIN menu AS me ON s.product\_id = me.product\_id  
 WHERE order\_date < join\_date  
),  
ranked\_cte AS (  
 SELECT customer\_id, product\_name, order\_date,  
 RANK() OVER(PARTITION BY customer\_id ORDER BY order\_date DESC) AS rnk  
 FROM cte  
)  
SELECT \* FROM ranked\_cte WHERE rnk = 1;

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

WITH cte AS (  
 SELECT m.customer\_id, join\_date, order\_date, me.product\_id, me.product\_name, price  
 FROM members AS m  
 JOIN sales AS s ON m.customer\_id = s.customer\_id  
 JOIN menu AS me ON s.product\_id = me.product\_id  
 WHERE order\_date < join\_date  
),  
aggregated AS (  
 SELECT customer\_id, COUNT(\*) AS total\_items\_ordered, SUM(price) AS amount\_spent\_by\_customer  
 FROM cte GROUP BY customer\_id  
)  
SELECT \* FROM aggregated;

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

SELECT m.customer\_id,  
 SUM(CASE WHEN product\_name = 'sushi' THEN price \* 20 ELSE price \* 10 END) AS total\_points  
FROM members AS m  
JOIN sales AS s ON m.customer\_id = s.customer\_id  
JOIN menu AS me ON s.product\_id = me.product\_id  
GROUP BY m.customer\_id;

**10. In the first week after a customer joins the program, they earn 2x points on** **all items - how many points do customer A and B have at the end of January?**

SELECT m.customer\_id,  
 SUM(CASE WHEN s.order\_date BETWEEN m.join\_date AND DATE\_ADD(m.join\_date, INTERVAL 6 DAY) THEN price \* 20 ELSE price \* 10 END) AS total\_points  
FROM members AS m  
JOIN sales AS s ON m.customer\_id = s.customer\_id  
JOIN menu AS me ON s.product\_id = me.product\_id  
WHERE order\_date <= '2021-01-31' AND m.customer\_id IN ('A', 'B')  
GROUP BY m.customer\_id;

**🔍 4. Key Insights**

1. **Customer Behavior**
   * **Customer A** is the highest spender.
   * **Customer B** has the most visits and likely highest loyalty.
   * **Customer C** is the least active and least engaged.
2. **Product Popularity**
   * **Ramen** and **Sushi** are top-selling items.
   * Sushi gets the highest point multipliers, making it strategic for loyalty campaigns.
3. **Membership Effect**
   * Members tend to change behavior right after joining.
   * Customers often make their first purchase as a member the same day or within a week.
4. **Points System**
   * Points incentivize purchases, especially when time-bound or tied to popular products.
   * Double-point campaigns are highly effective in the first week after membership.

**📊 5. Customer Behavior Summary**

| **Metric** | **Customer A** | **Customer B** | **Customer C** |
| --- | --- | --- | --- |
| Total Spend | $76 | $74 | $36 |
| Total Visits | 4 | 6 | 2 |
| Most Purchased Item | Ramen | Sushi | Ramen |
| Points Earned | 860 | 940 | 360 |
| Items Before Joining | 4 | 3 | – |

**💡 6. Recommendations**

* **Personalized Campaigns**  
  Target customers with personalized deals (e.g., Ramen for A, Sushi for B).
* **Double Points Strategy**  
  Leverage time-sensitive double point offers during the first week of membership.
* **Popular Product Promotions**  
  Run loyalty campaigns around Sushi and Ramen to boost engagement.
* **Reactivation Strategy**  
  Customer C needs re-engagement via reminders, coupons, or surveys.
* **Gamify Loyalty**  
  Introduce levels or badges based on visit frequency or points earned.

Top of Form

Bottom of Form