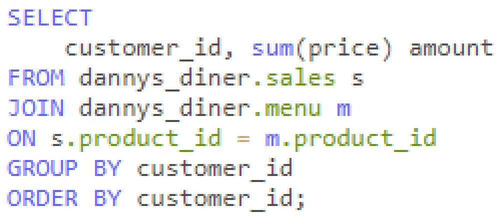
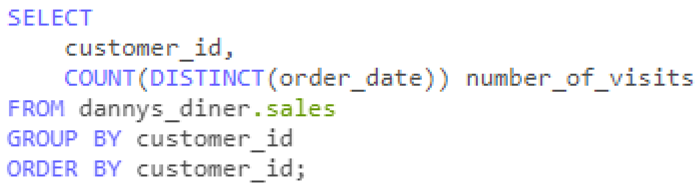
Q1 - What is the total amount each customer spent at the restaurant?



```
SELECT
    customer_id, sum(price) amount
FROM dannys_diner.sales s
JOIN dannys_diner.menu m
ON s.product_id = m.product_id
GROUP BY customer_id
ORDER BY customer_id;
```

customer_id	amount
A	76
B	74
C	36

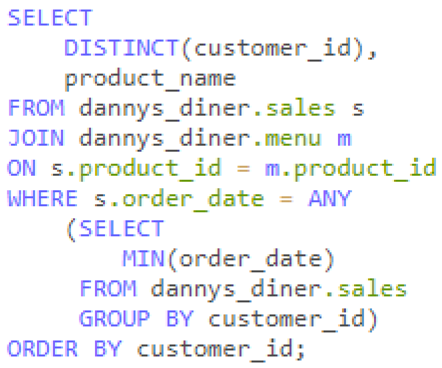
Q2 - How many days has each customer visited the restaurant?



```
SELECT
    customer_id,
    COUNT(DISTINCT(order_date)) number_of_visits
FROM dannys_diner.sales
GROUP BY customer_id
ORDER BY customer_id;
```

customer_id	number_of_visits
A	4
B	6
C	2

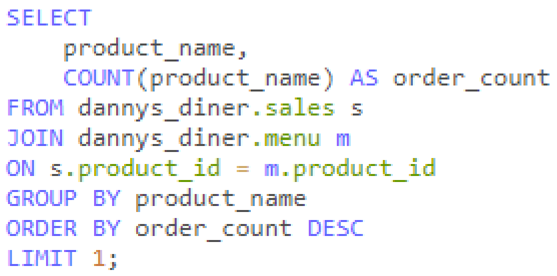
Q3 - What was the first item from the menu purchased by each customer?



```
SELECT
    DISTINCT(customer_id),
    product_name
FROM dannys_diner.sales s
JOIN dannys_diner.menu m
ON s.product_id = m.product_id
WHERE s.order_date = ANY
    (SELECT
        MIN(order_date)
     FROM dannys_diner.sales
     GROUP BY customer_id)
ORDER BY customer_id;
```

customer_id	product_name
A	curry
A	sushi
B	curry
C	ramen

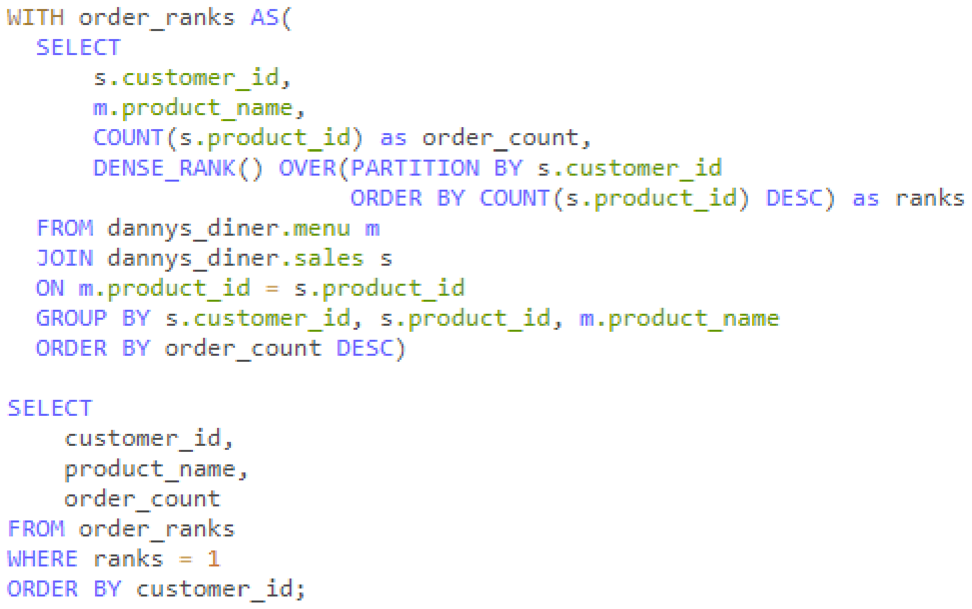
Q4 - What is the most purchased item on the menu and how many times was it purchased by all customers?



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

product_name	order_count
ramen	8

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

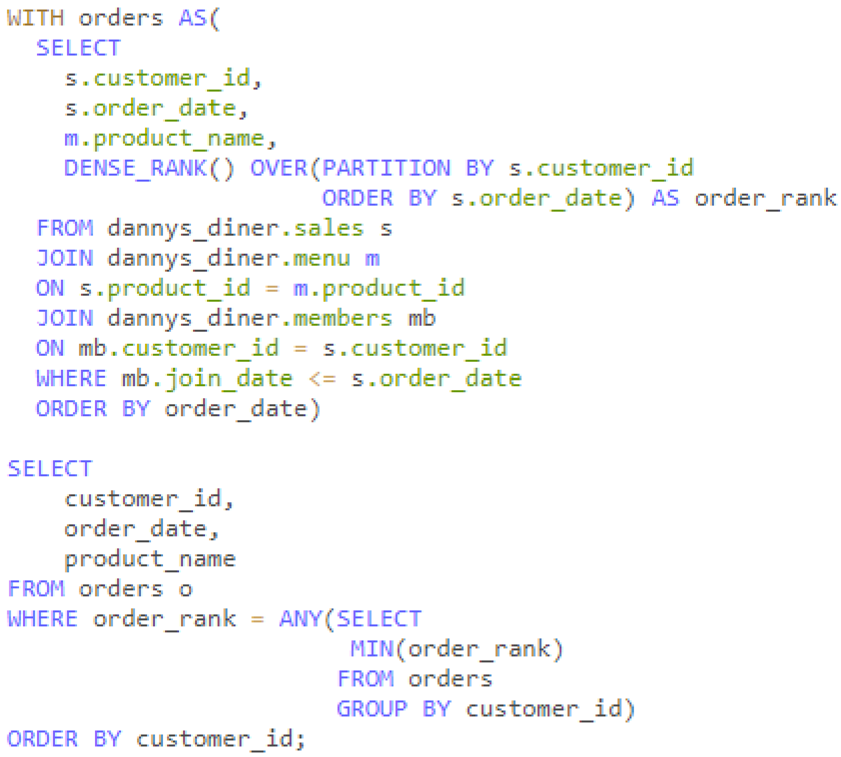


```
WITH order_ranks AS(
  SELECT
      s.customer_id,
      m.product_name,
      COUNT(s.product_id) as order_count,
      DENSE_RANK() OVER(PARTITION BY s.customer_id
                        ORDER BY COUNT(s.product_id) DESC) as ranks
  FROM dannys_diner.menu m
  JOIN dannys_diner.sales s
  ON m.product_id = s.product_id
  GROUP BY s.customer_id, s.product_id, m.product_name
  ORDER BY order_count DESC)

SELECT
    customer_id,
    product_name,
    order_count
FROM order_ranks
WHERE ranks = 1
ORDER BY customer_id;
```

customer_id	product_name	order_count
A	ramen	3
B	sushi	2
B	curry	2
B	ramen	2
C	ramen	3

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

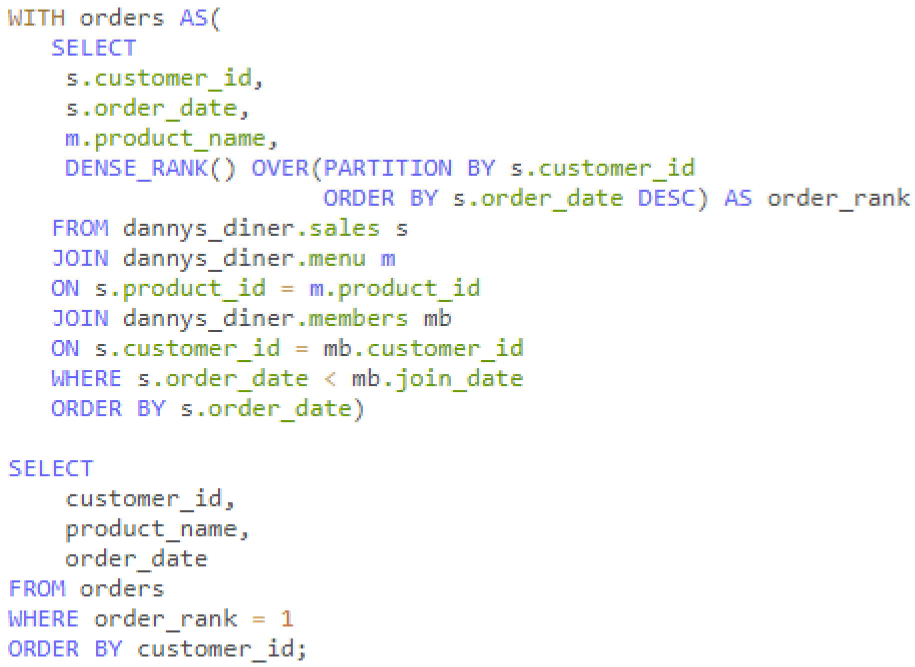


```
WITH orders AS(
  SELECT
    s.customer_id,
    s.order_date,
    m.product_name,
    DENSE_RANK() OVER(PARTITION BY s.customer_id
                      ORDER BY s.order_date) AS order_rank
  FROM dannys_diner.sales s
  JOIN dannys_diner.menu m
  ON s.product_id = m.product_id
  JOIN dannys_diner.members mb
  ON mb.customer_id = s.customer_id
  WHERE mb.join_date <= s.order_date
  ORDER BY order_date)

SELECT
    customer_id,
    order_date,
    product_name
FROM orders o
WHERE order_rank = ANY(SELECT
                        MIN(order_rank)
                       FROM orders
                       GROUP BY customer_id)
ORDER BY customer_id;
```

customer_id	order_date	product_name
A	2021-01-07T00:00:00.000Z	curry
B	2021-01-11T00:00:00.000Z	sushi

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

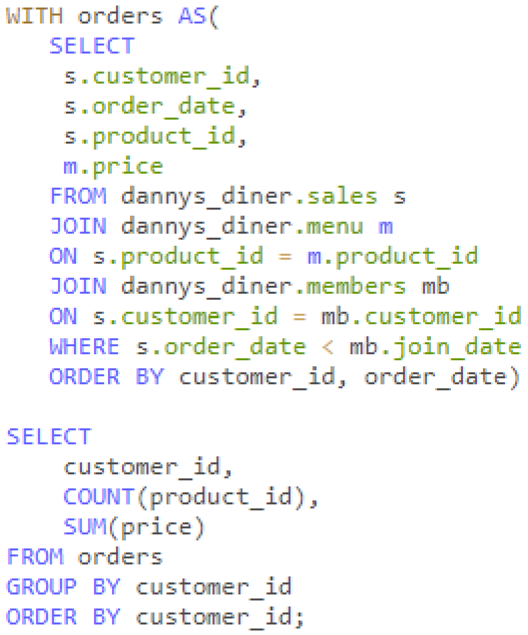


```
WITH orders AS(
   SELECT
    s.customer_id,
    s.order_date,
    m.product_name,
    DENSE_RANK() OVER(PARTITION BY s.customer_id
                      ORDER BY s.order_date DESC) AS order_rank
   FROM dannys_diner.sales s
   JOIN dannys_diner.menu m
   ON s.product_id = m.product_id
   JOIN dannys_diner.members mb
   ON s.customer_id = mb.customer_id
   WHERE s.order_date < mb.join_date
   ORDER BY s.order_date)

SELECT
    customer_id,
    product_name,
    order_date
FROM orders
WHERE order_rank = 1
ORDER BY customer_id;
```

customer_id	product_name	order_date
A	sushi	2021-01-01T00:00:00.000Z
A	curry	2021-01-01T00:00:00.000Z
B	sushi	2021-01-04T00:00:00.000Z

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

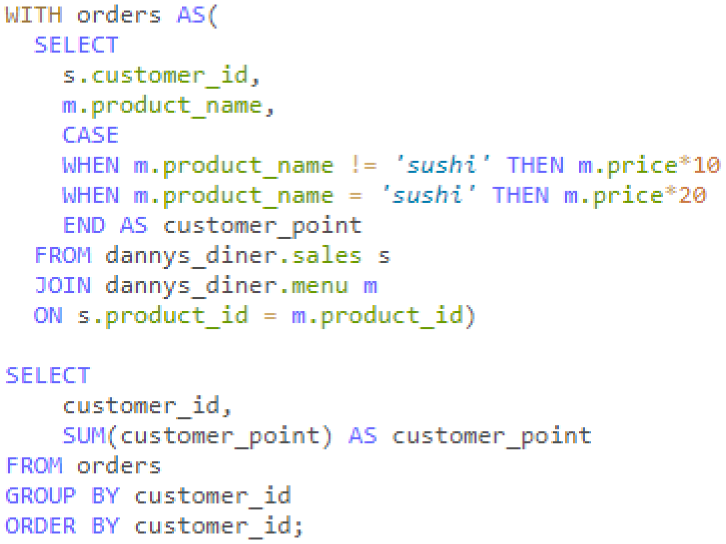


```
WITH orders AS(
   SELECT
    s.customer_id,
    s.order_date,
    s.product_id,
    m.price
   FROM dannys_diner.sales s
   JOIN dannys_diner.menu m
   ON s.product_id = m.product_id
   JOIN dannys_diner.members mb
   ON s.customer_id = mb.customer_id
   WHERE s.order_date < mb.join_date
   ORDER BY customer_id, order_date)

SELECT
    customer_id,
    COUNT(product_id),
    SUM(price)
FROM orders
GROUP BY customer_id
ORDER BY customer_id;
```

customer_id	count	sum
A	2	25
B	3	40

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

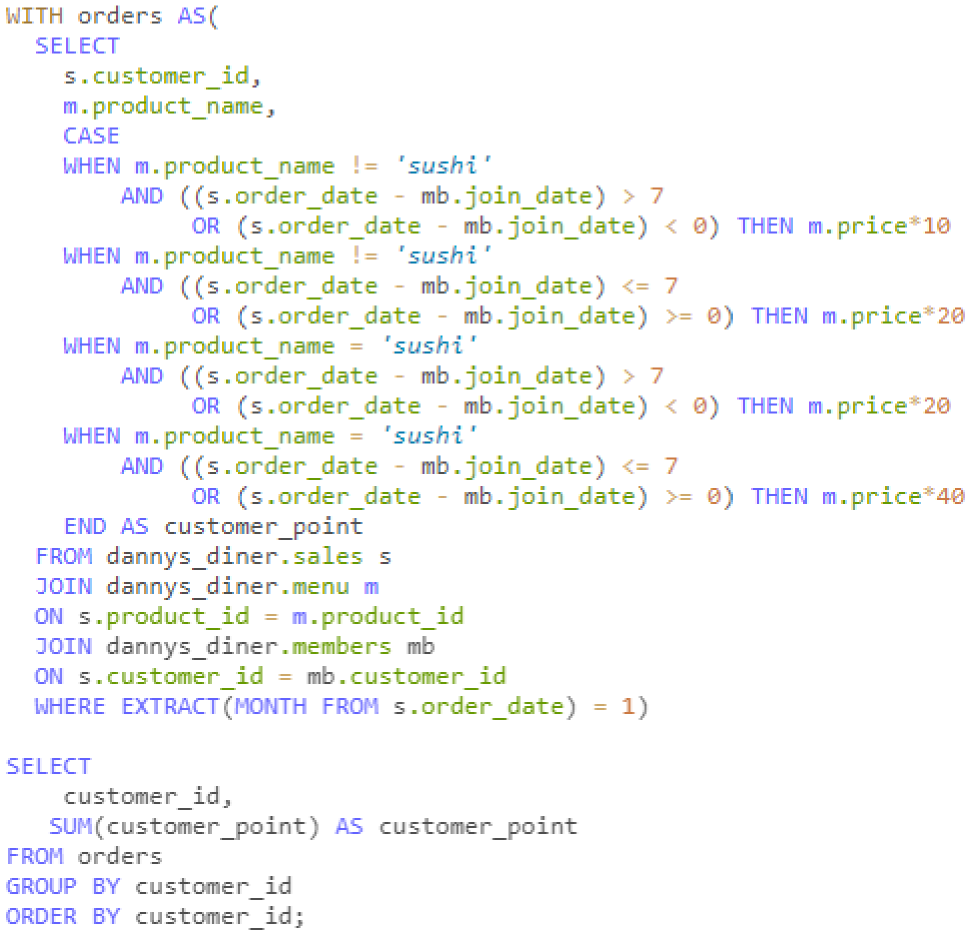


```
WITH orders AS(
  SELECT
    s.customer_id,
    m.product_name,
    CASE
    WHEN m.product_name != ’sushi’ THEN m.price*10
    WHEN m.product_name = ’sushi’ THEN m.price*20
    END AS customer_point
  FROM dannys_diner.sales s
  JOIN dannys_diner.menu m
  ON s.product_id = m.product_id)

SELECT
    customer_id,
    SUM(customer_point) AS customer_point
FROM orders
GROUP BY customer_id
ORDER BY customer_id;
```

customer_id	customer_point
A	860
B	940
C	360

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 orders AS(
  SELECT
    s.customer_id,
    m.product_name,
    CASE
    WHEN m.product_name != 'sushi'
        AND ((s.order_date - mb.join_date) > 7
             OR (s.order_date - mb.join_date) < 0) THEN m.price*10
    WHEN m.product_name != 'sushi'
        AND ((s.order_date - mb.join_date) <= 7
             OR (s.order_date - mb.join_date) >= 0) THEN m.price*20
    WHEN m.product_name = 'sushi'
        AND ((s.order_date - mb.join_date) > 7
             OR (s.order_date - mb.join_date) < 0) THEN m.price*20
    WHEN m.product_name = 'sushi'
        AND ((s.order_date - mb.join_date) <= 7
             OR (s.order_date - mb.join_date) >= 0) THEN m.price*40
    END AS customer_point
  FROM dannys_diner.sales s
  JOIN dannys_diner.menu m
  ON s.product_id = m.product_id
  JOIN dannys_diner.members mb
  ON s.customer_id = mb.customer_id
  WHERE EXTRACT(MONTH FROM s.order_date) = 1)

SELECT
    customer_id,
   SUM(customer_point) AS customer_point
FROM orders
GROUP BY customer_id
ORDER BY customer_id;
```

customer_id	customer_point
A	1370
B	1140