8-WEEK

SQL-CHALLENGE

WEEK-1

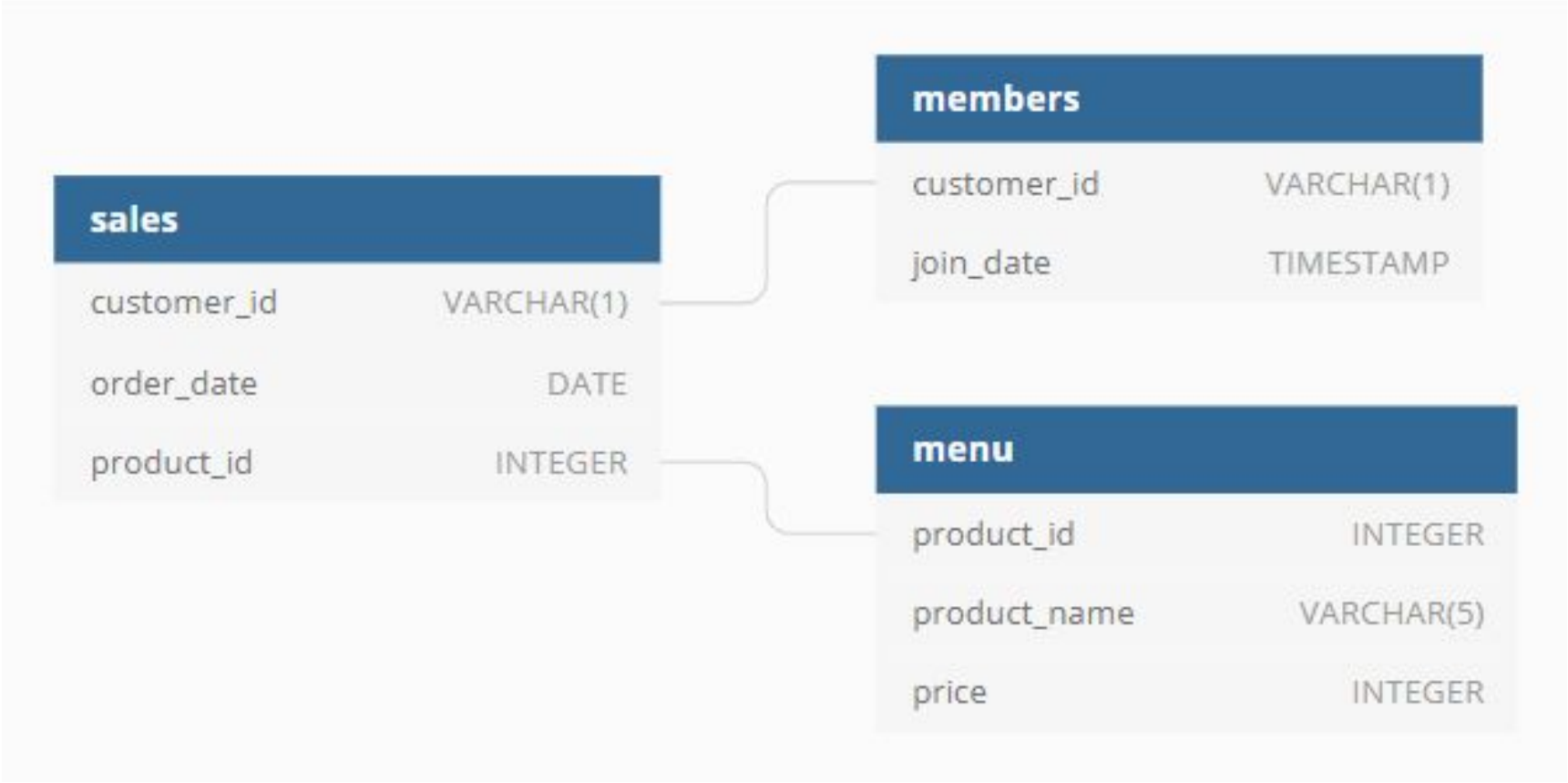
Case Study#1

Danny's Diner

OVERVIEW

1. Entity Relationship Diagram
2. Question and Solution

Entity Relationship Diagram



Question and Solution

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

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

Answer

	customer_id	total_sales
1	A	76
2	B	74
3	C	36

Question and Solution

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

```
SELECT
  sales.customer_id,
  COUNT(DISTINCT order_date) AS visit_count
FROM sales
GROUP BY customer_id;
```

Answer

	customer_id	visit_count
1	A	4
2	B	6
3	C	2

Question and Solution

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

```
 WITH ordered_sales AS (
SELECT
  sales.customer_id,
  sales.order_date,
  menu.product_name,
  DENSE_RANK() OVER (
    PARTITION BY sales.customer_id
    ORDER BY sales.order_date) AS rank
FROM sales AS sales
INNER JOIN menu AS menu
  ON sales.product_id = menu.product_id
)
SELECT
  customer_id,
  product_name
FROM ordered_sales
WHERE rank = 1
GROUP BY customer_id, product_name;
```

Answer

	customer_id	product_name
1	A	curry
2	A	sushi
3	B	curry
4	C	ramen

Question and Solution

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

```
SELECT TOP 1
  menu.product_name,
  COUNT(sales.product_id) AS most_purchased_item
FROM sales AS sales
INNER JOIN menu AS menu
  ON sales.product_id = menu.product_id
GROUP BY menu.product_name
ORDER BY most_purchased_item DESC;
```

Answer

	product_name	most_purchased_item
1	ramen	8

Question and Solution

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

```
WITH most_popular AS (
  SELECT
    sales.customer_id,
    menu.product_name,
    COUNT(menu.product_id) AS order_count,
    DENSE_RANK() OVER (
      PARTITION BY sales.customer_id
      ORDER BY COUNT(sales.customer_id) DESC) AS rank
  FROM menu
  INNER JOIN sales
    ON menu.product_id = sales.product_id
  GROUP BY sales.customer_id, menu.product_name
)

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

Answer

	customer_id	product_name	order_count
1	A	ramen	3
2	B	sushi	2
3	B	curry	2
4	B	ramen	2
5	C	ramen	3

Question and Solution

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

```
WITH joined_as_member AS (
  SELECT
    members.customer_id,
    sales.product_id,
    ROW_NUMBER() OVER (
      PARTITION BY members.customer_id
      ORDER BY sales.order_date) AS row_num
  FROM members
  INNER JOIN sales
    ON members.customer_id = sales.customer_id
    AND sales.order_date > members.join_date
)

SELECT
  customer_id,
  product_name
FROM joined_as_member
INNER JOIN menu
  ON joined_as_member.product_id = menu.product_id
WHERE row_num = 1
ORDER BY customer_id ASC;
```

Answer

	customer_id	product_name
1	A	ramen
2	B	sushi
3	C	ramen

Question and Solution

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

```
WITH purchased_prior_member AS (
  SELECT
    members.customer_id,
    sales.product_id,
    ROW_NUMBER() OVER (
      PARTITION BY members.customer_id
      ORDER BY sales.order_date DESC) AS rank
  FROM members
  INNER JOIN sales
    ON members.customer_id = sales.customer_id
    AND sales.order_date < members.join_date
)
SELECT
  p_member.customer_id,
  menu.product_name
FROM purchased_prior_member AS p_member
INNER JOIN menu
  ON p_member.product_id = menu.product_id
WHERE rank = 1
ORDER BY p_member.customer_id ASC;
```

Answer

	customer_id	product_name
1	A	sushi
2	B	sushi

Question and Solution

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

```
SELECT
  sales.customer_id,
  COUNT(sales.product_id) AS total_items,
  SUM(menu.price) AS total_sales
FROM sales
INNER JOIN members
  ON sales.customer_id = members.customer_id
  AND sales.order_date < members.join_date
INNER JOIN menu
  ON sales.product_id = menu.product_id
GROUP BY sales.customer_id;
```

Answer

	customer_id	total_items	total_sales
1	A	2	25
2	B	3	40

Question and Solution

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

```
WITH points_cte AS (
  SELECT
    menu.product_id,
    CASE
      WHEN product_id = 1 THEN price * 20
      ELSE price * 10 END AS points
  FROM menu
)

SELECT
  sales.customer_id,
  SUM(points_cte.points) AS total_points
FROM sales
INNER JOIN points_cte
  ON sales.product_id = points_cte.product_id
GROUP BY sales.customer_id;
```

Answer

	customer_id	total_points
1	A	860
2	B	940
3	C	360

--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?

```
WITH customer_spending AS (
  SELECT
     s.customer_id,
     m.product_name,
     s.order_date,
     m.price,
     CASE
     -- Double points if within the first week of joining for all items
     WHEN s.order_date BETWEEN mb.join_date
                    AND DATEADD(DAY, 6, mb.join_date)
                    THEN (m.price * 10 * 2)
     -- Double points if sushi, outside of the first week of joining
     WHEN m.product_name = 'sushi' THEN (m.price * 10 * 2)
     ELSE (m.price * 10)
     END AS points
  FROM sales s
  JOIN menu m ON s.product_id = m.product_id
  LEFT JOIN members mb ON s.customer_id = mb.customer_id
)
SELECT customer_id,
      SUM(points) AS total_points
FROM customer_spending
WHERE order_date BETWEEN '2021-01-01' AND '2021-01-31'
  AND customer_id IN ('A', 'B')
GROUP BY customer_id;
```

Answer

	customer_id	total_points
1	A	1370
2	B	820