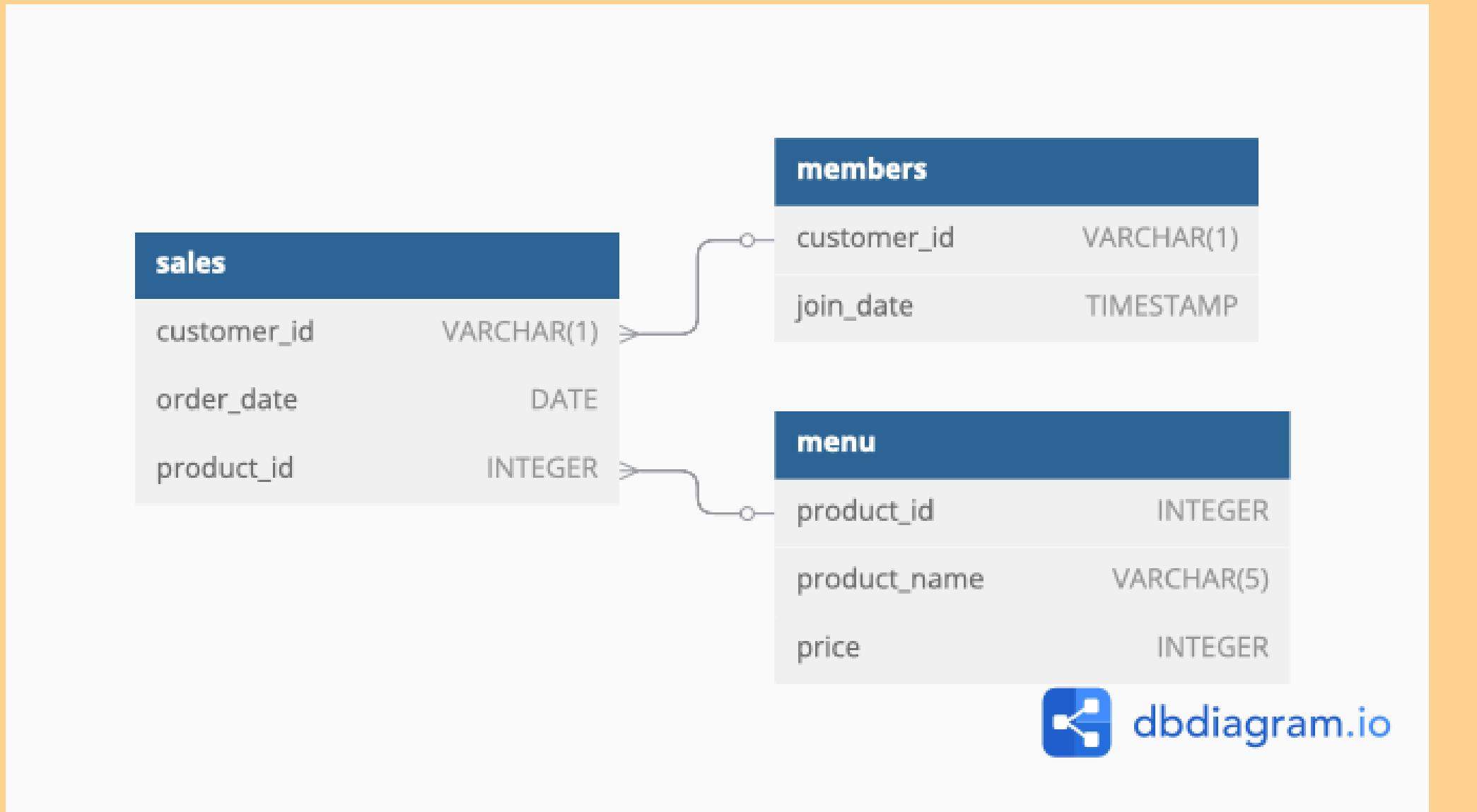


Case Study

#1 -

Danny's Diner

Aaryash Singh



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

```
1 #aaryashsingh
2 • SELECT customer_id, SUM(price) FROM sales
3 JOIN menu
4 ON sales.product_id = menu.product_id
5 GROUP BY customer_id;
6
```

	customer_id	sum(price)
	A	76
	B	74
	C	36

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

```
-- 2. How many days has each customer visited the restaurant?  
SELECT customer_id, count(DISTINCT order_date) AS 'Days Visited' FROM sales  
GROUP BY customer_id;
```

customer_id	Days Visited
A	4
B	6
C	2

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

```
-- 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 rn  
    FROM dannys_diner.sales  
    INNER JOIN dannys_diner.menu  
        ON sales.product_id = menu.product_id  
)  
SELECT  
    customer_id,  
    product_name  
FROM ordered_sales  
WHERE rn = 1  
GROUP BY customer_id, product_name;
```

customer_id	product_na...
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?

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

product_id	count(*)	product_na...
3	8	ramen

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

```
-- 5. Which item was the most popular for each customer?  
With fav_order as (Select  
    customer_id,  
    menu.product_name,  
    count(sales.product_id) as order_count,  
    dense_rank() over(partition by customer_id order by count(sales.product_id) desc) AS ranking  
from sales join menu  
ON sales.product_id = menu.product_id  
group by customer_id,menu.product_name  
order by customer_id asc, count(sales.product_id) desc)  
Select customer_id,  
    product_name,  
    order_count  
from fav_order  
where ranking = 1;
```

customer_id	product_na...	order_count
A	ramen	3
B	curry	2
B	sushi	2
B	ramen	2
C	ramen	3

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

```
-- 6. Which item was purchased first by the customer after they became a member?  
with member_first_order as(  
    select  
        members.customer_id,  
        product_name,  
        order_date,  
        dense_rank() over(partition by members.customer_id order by order_date) AS ranking  
    from members JOIN sales  
    on members.customer_id = sales.customer_id  
    JOIN menu  
    on sales.product_id = menu.product_id  
    where sales.order_date > members.join_date)  
SELECT customer_id,  
    product_name  
from member_first_order  
where ranking = 1;
```

customer_id	product_na...
A	ramen
B	sushi



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

```
-- 7. Which item was purchased just before the customer became a member?  
WITH orders_befor_membership AS (  
    SELECT  
        members.customer_id,  
        menu.product_name,  
        dense_rank() over(PARTITION BY members.customer_id ORDER BY order_date) AS ranking  
    FROM members JOIN sales  
    ON members.customer_id = sales.customer_id  
    JOIN menu  
    ON sales.product_id = menu.product_id  
    WHERE order_date < join_date)  
SELECT  
    customer_id,  
    product_name  
FROM orders_befor_membership  
WHERE ranking = 1;
```

customer_id	product_na...
A	sushi
A	curry
B	curry



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

```
-- 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),  
    sum(price)  
FROM menu JOIN sales  
ON sales.product_id = menu.product_id  
JOIN members  
ON members.customer_id = sales.customer_id  
where order_date < join_date  
group by sales.customer_id  
order by sales.customer_id;
```

customer_id	count(sales.product_id)	sum(price)
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?

```
-- 9. If each $1 spent equates to 10 points and sushi has a 2x points multiplier

SELECT
    sales.customer_id,
    SUM(CASE
        WHEN sales.product_id = 1 THEN price*20
        ELSE price*10
    END) AS points
FROM sales JOIN menu
ON sales.product_id = menu.product_id
GROUP BY sales.customer_id;
```

customer_id	points
A	860
B	940
C	360



ThankYou

-Aaryash Singh