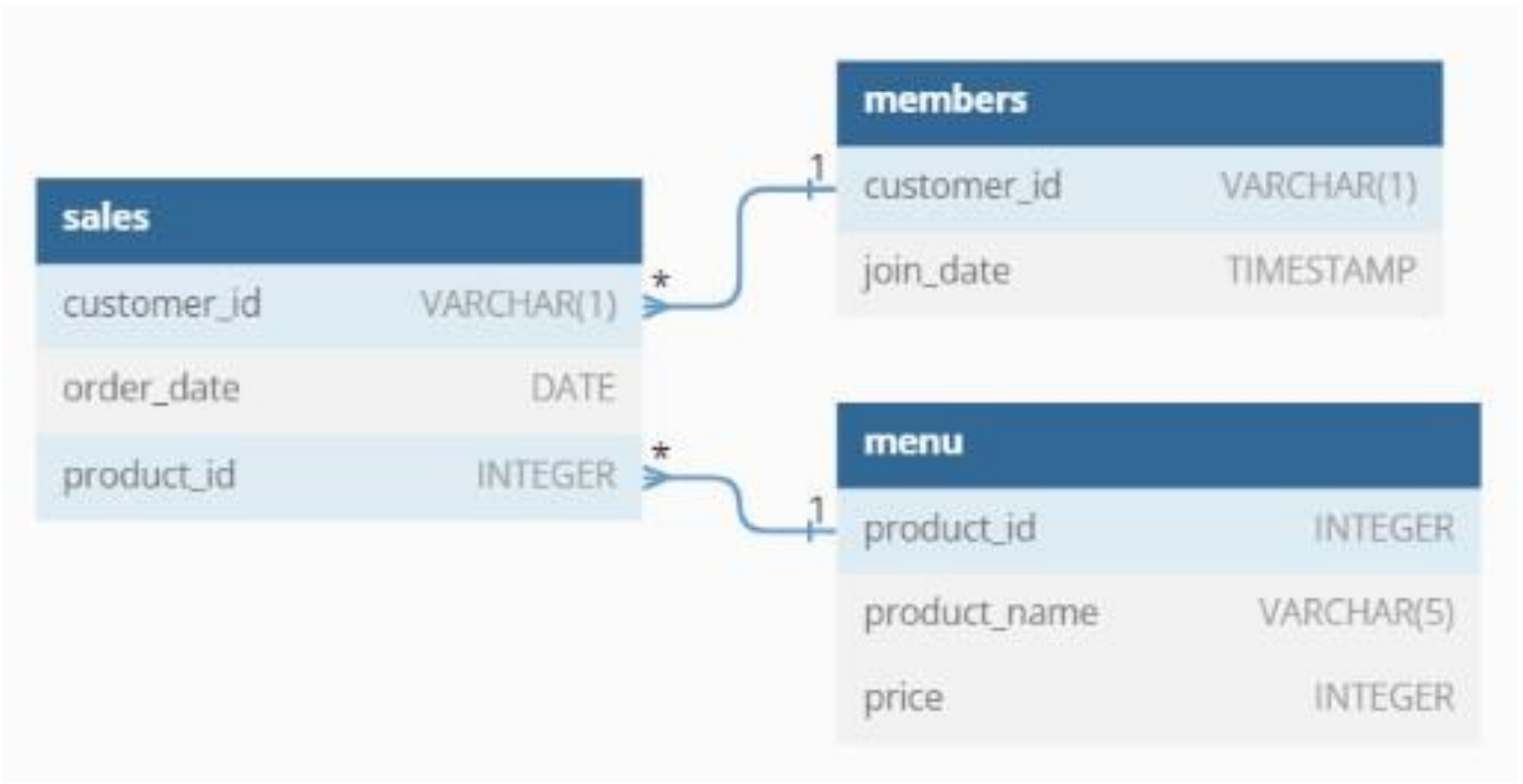
Challenge:

- Client aims to understand customer behavior, focusing on visiting patterns, expenditure, and favorite menu items to enhance personalized experiences.
- The insights will guide decisions on expanding the customer loyalty program, and Danny seeks basic datasets for easy team inspection without SQL.
- A sample of customer data is provided for privacy reasons, expecting fully functioning SQL queries to address his questions.

Tables:

- Sales
- Menu
- Members

ER Diagram:



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

Solution:

```sql
SELECT
    S.customer_id, SUM(M.price) AS Total_amount
FROM
    Sales S
        JOIN
    Menu M ON S.product_id = M.product_id
GROUP BY S.customer_id;
```

Customer_id	Total_amount
A	76
B	74
C	36

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

Solution:

```sql
SELECT
    Customer_id
    , COUNT(DISTINCT order_date) AS NO_of_days
FROM
    Sales S
GROUP BY S.customer_id;
```

Customer_id	No_of_Days
A	4
B	6
C	2

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

Solution:

```sql
select Customer_id
        ,Product_name from
(
select  S.customer_id, M.product_name
,row_number()
  over(partition by customer_id order by order_date ) as rnk
FROM Sales S
JOIN Menu M
ON S.product_id = M.product_id
) X
where rnk = 1;
```

Customer_id	Product_Name
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?

Solution:

```sql
SELECT
M.Product_name, count(order_date) AS NO_of_times
FROM Sales S
JOIN Menu M
ON S.product_id = M.product_id
group by M.product_name
order by count(order_date) desc
```

Product_name	No_of_times
sushi	8

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

Solution:

```sql
Select customer_id,product_name,nooftimes from
(
Select customer_id,M.product_name, count(order_date) AS nooftimes,
row_number( )
  over( partition by customer_id order by count(order_date) desc) as rnk
FROM Sales S
JOIN Menu M
  ON S.product_id = M.product_id
group by customer_id,M.product_name
) X
where rnk =1;
```

Customer_id	Product_name	No_of_times
A	ramen	3
B	curry	2
C	ramen	3

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

Solution:

```sql
With cte1 as(
  select S.customer_id,order_date,join_date,M.product_name,
rank( ) over(partition by customer_id order by order_date) as rnk
FROM Sales S
JOIN Menu M ON S.product_id = M.product_id
Join members P on S.customer_id = P.customer_id
where order_date >= join_date
)
select customer_id
     ,product_name
from cte1
where rnk = 1;
```

Customer_id	Product_name
A	curry
B	sushi

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

Solution:

```sql
With cte1 as(
select S.customer_id,order_date,join_date,M.product_name,
rank()
  over(partition by customer_id order by order_date desc) as rnk
FROM Sales S
JOIN Menu M ON S.product_id = M.product_id
Join members P on S.customer_id = P.customer_id
where order_date < join_date
)
select customer_id
     ,product_name
from cte1
where rnk = 1;
```

Customer_id	Product_name
A	sushi
A	curry
B	sushi

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

Solution:

```sql
SELECT
    S.customer_id,
    COUNT(S.product_id) AS Products,
    SUM(M.price) AS Price
FROM
    Sales S
        JOIN
    Menu M ON S.product_id = M.product_id
        JOIN
    members P ON S.customer_id = P.customer_id
WHERE
    order_date < join_date
GROUP BY customer_id
ORDER BY customer_id;
```

Customer_id	Product_count	Price
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?

Solution:

```sql
SELECT
    S.customer_id,
    SUM(CASE
        WHEN product_name = 'sushi' THEN price * 10 * 2
        ELSE price * 10
    END) AS points
FROM
    Sales S
        JOIN
    Menu M ON S.product_id = M.product_id
GROUP BY customer_id;
```

Customer_id	Points
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?

Solution:

```sql
WITH CTE1 AS (
  SELECT
      S.customer_id,
      S.order_date,
      M.product_name,
      M.price,
      CASE
          WHEN product_name = 'sushi' THEN 2 * M.price
          WHEN order_date BETWEEN P.join_date AND DATE_ADD(P.join_date, INTERVAL 6 DAY) THEN 2 *
M.price
          ELSE M.price
      END AS points
  FROM
      Sales S
      JOIN Menu M ON S.product_id = M.product_id
      JOIN Members P ON S.customer_id = P.customer_id
  WHERE
      DATE_FORMAT(order_date, '%Y-%m-01') = '2021-01-01'
)
SELECT
  customer_id,
  SUM(points) * 10 AS total_points
FROM
  CTE1
GROUP BY
  customer_id
order by customer_id;
```

Customer_id	Total_points
A	1370
B	820

Q11. Join All The Things

Solution:

```sql
SELECT
    S.customer_id,
    S.order_date,
    M.product_name,
    M.price,
    CASE
        WHEN order_date >= join_date THEN 'Y'
        ELSE 'N'
    END AS Member
FROM
    Sales S
        JOIN
    Menu M ON S.product_id = M.product_id
        LEFT JOIN
    Members P ON S.customer_id = P.customer_id
ORDER BY S.customer_id , S.order_date , M.product_name;
```

Output:

Customer_id	Order_date	Product_name	Price	Member
A	2021-01-01	curry	15	N
A	2021-01-01	sushi	10	N
A	2021-01-07	curry	15	Y
A	2021-01-10	ramen	12	Y
A	2021-01-11	ramen	12	Y
B	2021-01-01	curry	15	N
B	2021-01-02	curry	15	N
B	2021-01-04	sushi	10	N
B	2021-01-11	sushi	10	Y
B	2021-01-16	ramen	12	Y
B	2021-02-01	ramen	12	Y
C	2021-01-01	ramen	12	N
C	2021-01-01	ramen	12	N
C	2021-01-07	ramen	12	N

Q12. Rank all Things

Solution:

```sql
with CTE1 as (
SELECT
    S.customer_id,
    S.order_date,
    M.product_name,
    M.price,
    CASE
        WHEN order_date >= join_date THEN 'Y'
        ELSE 'N'
    END AS Member
FROM
    Sales S
        JOIN
    Menu M ON S.product_id = M.product_id
        LEFT JOIN
    Members P ON S.customer_id = P.customer_id
ORDER BY S.customer_id , S.order_date , M.product_name
)
select * ,
    CASE

        WHEN Member = 'N' then NULL
        ELSE
        rank( )
        over(
          PARTITION BY customer_id
          ,Member ORDER BY order_date) end AS RNK
from CTE1;
```

Customer_id	Order_date	Product_name	Price	Member	Rank
A	2021-01-01	curry	15	N	null
A	2021-01-01	sushi	10	N	null
A	2021-01-07	curry	15	Y	1
A	2021-01-10	ramen	12	Y	2
A	2021-01-11	ramen	12	Y	3
A	2021-01-11	ramen	12	Y	3
B	2021-01-01	curry	15	N	null
B	2021-01-02	curry	15	N	null
B	2021-01-04	sushi	10	N	null
B	2021-01-11	sushi	10	Y	1
B	2021-01-16	ramen	12	Y	2
B	2021-02-01	ramen	12	Y	3
C	2021-01-01	ramen	12	N	null
C	2021-01-01	ramen	12	N	null
C	2021-01-07	ramen	12	N	null