

Capstone 1

Relevel
by Unacademy



Assignment Discussion- Previous Class

Assignment Question

Use mode.com:

Tables:

tutorial.sf_crime_incidents_cleandate.

Write a query that counts the number of incidents reported by week. Cast the week as a date to get rid of the hours/minutes/seconds.

Write a query that shows exactly how long ago each indicent was reported. Assume that the dataset is in Pacific Standard Time (UTC - 8).

Assignment Solution [Link](#)

Introduction to Case

Ginny is a big fan of Japanese food, so she decided to start a restaurant at the beginning of 2021 that sells his three favorite foods: sushi, curry, and ramen.

Ginny's Diner needs your help to stay afloat – the restaurant has collected some fundamental data from their few months of operation but has no idea how to use it to help them operate the business.



Problem Statement

Ginny wants to use the data to answer a few simple questions about his customers, especially about their:

- visiting patterns,
- how much money they've spent, and
- which menu items are their favorite.

This deeper connection with her customers will help her deliver a better and more personalized experience for her loyal customers.

She plans on using these insights to help him decide whether she should expand the existing customer loyalty program. Additionally, she needs help to generate some essential datasets so her team can quickly inspect the data without needing to use SQL.

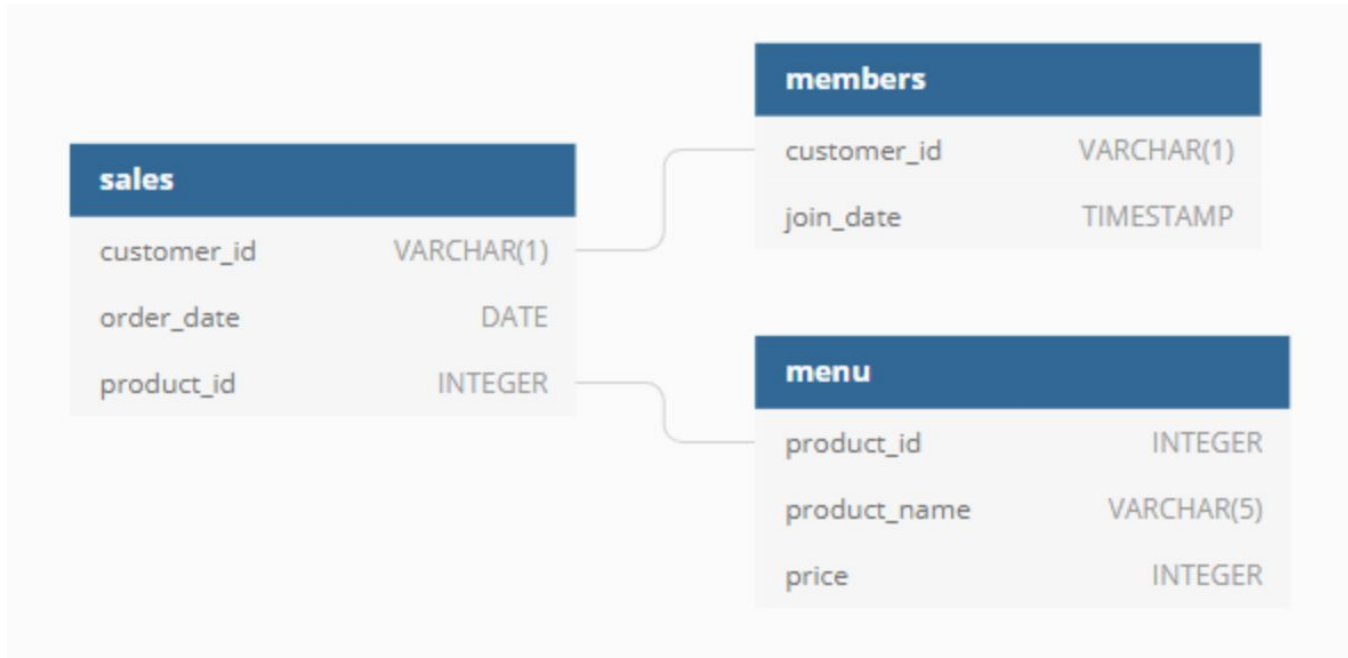
Problem Statement (contd.)

The data set contains the following three tables, which you may refer to in the relationship diagram below to understand the connection:

- sales
- members
- menu



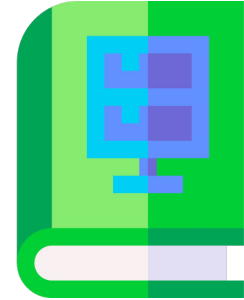
Table Relationship



Database

The database can be accessed here: <https://www.db-fiddle.com/f/wFVPv4s89DLRB2fP5tfX8c/8>

This tool will also be used to query.



Problem Statement - 1

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



Solution - 1

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



Problem Statement - 2

How many days has each customer visited the restaurant?



Solution - 2

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



Problem Statement - 3

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



Solution - 3

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



Problem Statement - 4

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



Solution - 4

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



Problem Statement - 5

Which item was the most popular one for each customer?



Solution - 5

```
WITH fav_item_cte AS
(
    SELECT s.customer_id, m.product_name,
           COUNT(m.product_id) AS order_count,
           DENSE_RANK() OVER(PARTITION BY s.customer_id
                              ORDER BY COUNT(m.product_id) DESC) AS rank
    FROM dbo.menu AS m
    JOIN dbo.sales AS s
         ON m.product_id = s.product_id
    GROUP BY s.customer_id, m.product_name
)
SELECT customer_id, product_name, order_count
FROM fav_item_cte
WHERE rank = 1;
```



Problem Statement - 6

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



Solution - 6

```
WITH member_sales_cte AS
(
    SELECT s.customer_id, m.join_date, s.order_date, s.product_id,
           DENSE_RANK() OVER(PARTITION BY s.customer_id
                              ORDER BY s.order_date) AS rank
    FROM dbo.sales AS s
    JOIN dbo.members AS m
    ON s.customer_id = m.customer_id
    WHERE s.order_date = m.join_date
)
SELECT s.customer_id, s.order_date, m2.product_name
FROM member_sales_cte AS s
JOIN dbo.menu AS m2
ON s.product_id = m2.product_id
```



Problem Statement - 7

Which item was purchased right before the customer became a member?



Solution - 7

```
WITH prior_member_purchased_cte AS
(
    SELECT s.customer_id, m.join_date, s.order_date, s.product_id,
           DENSE_RANK() OVER(PARTITION BY s.customer_id
                             ORDER BY s.order_date DESC) AS rank
    FROM dbo.sales AS s
    JOIN dbo.members AS m
      ON s.customer_id = m.customer_id
    WHERE s.order_date < m.join_date
)
SELECT s.customer_id, s.order_date, m2.product_name
FROM prior_member_purchased_cte AS s
JOIN dbo.menu AS m2
  ON s.product_id = m2.product_id
WHERE rank = 1;
```



Problem Statement - 8

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



Solution - 8

```
SELECT
    s.customer_id,
    COUNT(DISTINCT s.product_id) AS unique_menu_item,
    SUM(mm.price) AS total_sales
FROM
    dbo.sales AS s
JOIN
    dbo.members AS m
    ON s.customer_id = m.customer_id
JOIN
    dbo.menu AS mm
    ON s.product_id = mm.product_id
WHERE s.order_date < m.join_date
GROUP BY s.customer_id;
```



Problem Statement - 9

If each customers' \$1 spent equates to 10 points and sushi has a 2x points multiplier — how many points would each customer have?



Solution - 9

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



Solution - 9

```
SELECT
    s.customer_id,
    SUM(p.points) AS total_points
FROM
    price_points AS p
JOIN
    dbo.sales AS s
ON p.product_id = s.product_id
GROUP BY
    s.customer_id
ORDER BY
    customer_id
```



Problem Statement - 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 Jan 21?

Note :here you can use a concept of **interval function** which returns the index of the argument that is more than the first argument meaning It returns 0 if 1st number is less than the 2nd number and 1 if 1st number is less than the 3rd number and so on or -1 if 1st number is null



Solution - 10

```
WITH dates_cte AS
(
SELECT *,
    join_date + INTERVAL'6 day' AS valid_date,
    DATE('2021-01-31') AS last_date
FROM
    dbo.members AS m
),
points_cte AS (
SELECT d.customer_id, s.order_date, d.join_date,
    d.valid_date, d.last_date, m.product_name, m.price,
    SUM(CASE
        WHEN m.product_name = 'sushi' THEN 2 * 10 * m.price
        WHEN s.order_date BETWEEN d.join_date AND d.valid_date THEN 2 * 10 * m.price
        ELSE 10 * m.price
    END) AS points
```



Solution - 10

```

FROM dates_cte AS d
JOIN dbo.sales AS s
    ON d.customer_id = s.customer_id
JOIN dbo.menu AS m
    ON s.product_id = m.product_id
WHERE s.order_date < d.last_date
GROUP BY d.customer_id, s.order_date, d.join_date, d.valid_date, d.last_date, m.product_name, m.price
)
SELECT
    customer_id,
    SUM(points) AS total_points
FROM
    points_cte
GROUP BY
    customer_id
    
```



Thank You