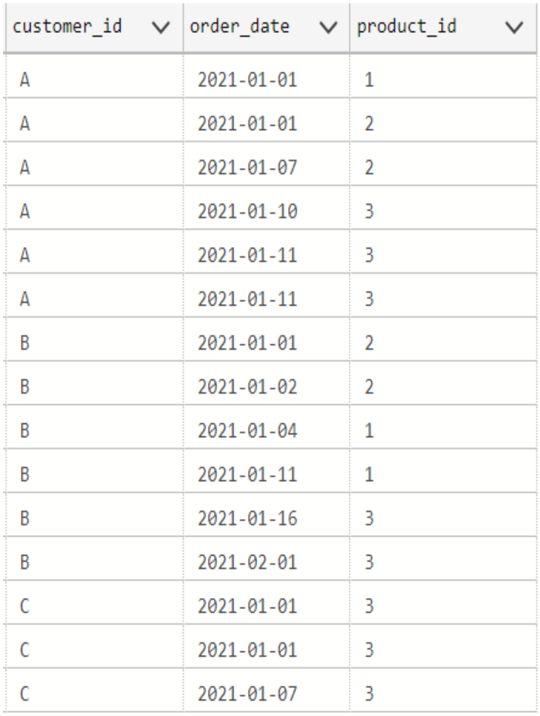
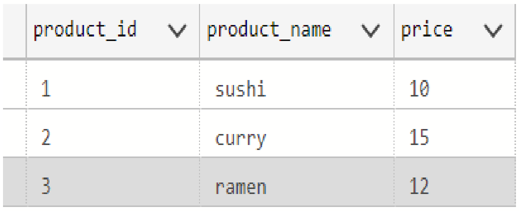
```sql
--#Verifying Entities---
```

```sql
SELECT * FROM sales;
```



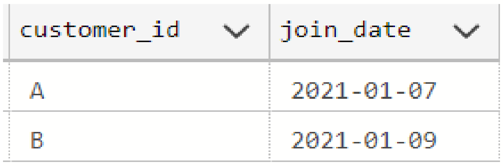
customer_id	order_date	product_id
A	2021-01-01	1
A	2021-01-01	2
A	2021-01-07	2
A	2021-01-10	3
A	2021-01-11	3
A	2021-01-11	3
B	2021-01-01	2
B	2021-01-02	2
B	2021-01-04	1
B	2021-01-11	1
B	2021-01-16	3
B	2021-02-01	3
C	2021-01-01	3
C	2021-01-01	3
C	2021-01-07	3

```sql
SELECT * FROM Menu;
```



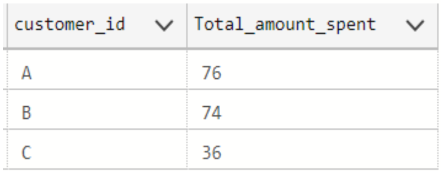
product_id	product_name	price
1	sushi	10
2	curry	15
3	ramen	12

```sql
SELECT * FROM members;
```



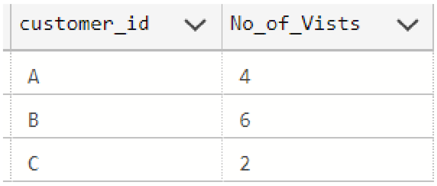
customer_id	join_date
A	2021-01-07
B	2021-01-09

```
--1--Total amount each customer spent at the restaurant--
    SELECT customer_id, SUM(price) Total_amount_spent
    FROM sales s
    JOIN menu m ON s.product_id = m.product_id
    GROUP BY customer_id
```



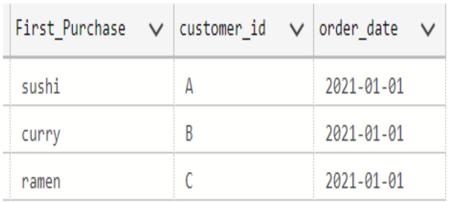
customer_id	Total_amount_spent
A	76
B	74
C	36

```
--2-- Number of days has each customer visited the restaurant---
    SELECT customer_id, COUNT(DISTINCT(order_date)) No_of_Vists
    FROM sales
    GROUP BY customer_id
```



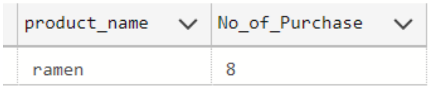
customer_id	No_of_Vists
A	4
B	6
C	2

```
--3-- First item from the menu purchased by each customer---
    SELECT DISTINCT product_name as First_Purchase, customer_id,
order_date FROM
    (SELECT *,
    ROW_NUMBER() OVER (
    PARTITION BY customer_id
    ORDER BY order_date)
    AS Ranking FROM
    (SELECT customer_id, product_name, order_date
    FROM sales s
    JOIN menu m ON s.product_id = m.product_id)TF) F1
    WHERE Ranking = 1
    ORDER BY Customer_id, order_date
```



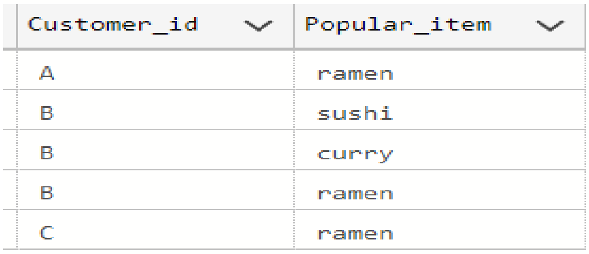
First_Purchase	customer_id	order_date
sushi	A	2021-01-01
curry	B	2021-01-01
ramen	C	2021-01-01

```
--4--The most purchased item on the menu and how many times was it purchased by all
customers----
    SELECT TOP 1 product_name, COUNT(product_name) No_of_Purchase
    FROM sales s
    JOIN menu m ON s.product_id = m.product_id
    GROUP BY product_name
    ORDER BY COUNT(product_name) DESC
```



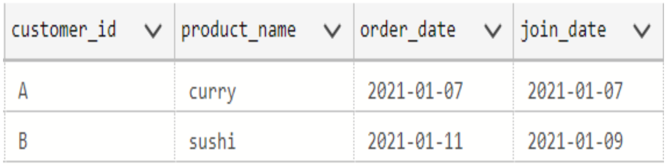
product_name	No_of_Purchase
ramen	8

```
--5--The most popular item for each customer--
    SELECT Customer_id, product_name AS Popular_item FROM
    (SELECT Customer_id, product_name, COUNT(product_name)
No_of_Purchase,
    RANK() OVER (PARTITION BY s.customer_id ORDER BY
COUNT(product_name) DESC) AS rk
    FROM sales s
    JOIN menu m ON s.product_id = m.product_id
    GROUP BY Customer_id, product_name) TY
    WHERE rk = 1
```



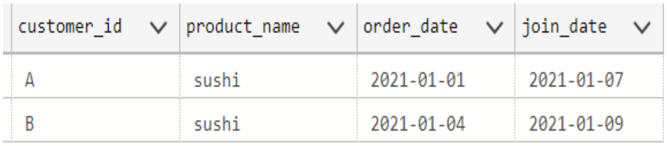
Customer_id	Popular_item
A	ramen
B	sushi
B	curry
B	ramen
C	ramen

```
 --6--First item purchased by the customer after they became a member--
     SELECT customer_id, product_name, order_date, join_date FROM
     (SELECT s.customer_id, m.product_name, s.order_date, mb.join_date,
     RANK() OVER (PARTITION BY s.customer_id ORDER BY order_date) AS rk
     FROM sales s
     FULL OUTER JOIN menu m ON s.product_id = m.product_id
     FULL OUTER JOIN members mb ON s.customer_id = mb.customer_id
     WHERE order_date >= join_date)T1
     WHERE rk = 1
```



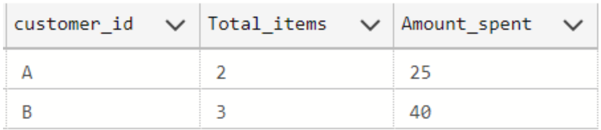
customer_id	product_name	order_date	join_date
A	curry	2021-01-07	2021-01-07
B	sushi	2021-01-11	2021-01-09

```
--7--The last item purchased just before the customer became a member---
  SELECT customer_id, product_name, order_date, join_date FROM
  (SELECT s.customer_id,  m.product_name, s.order_date, mb.join_date,
  ROW_NUMBER() OVER (PARTITION BY s.customer_id ORDER BY order_date DESC) AS rk
  FROM sales s
  FULL OUTER JOIN menu m ON s.product_id = m.product_id
  FULL OUTER JOIN members mb ON s.customer_id = mb.customer_id
  WHERE order_date < join_date)T1
  WHERE rk = 1
```



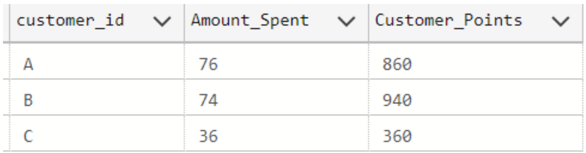
customer_id	product_name	order_date	join_date
A	sushi	2021-01-01	2021-01-07
B	sushi	2021-01-04	2021-01-09

```sql
--8--The total items and amount spent for each member before they became a member---
  SELECT s.customer_id, COUNT(product_name) AS Total_items, SUM(price) AS Amount_spent
  FROM sales s
  FULL OUTER JOIN menu m ON s.product_id = m.product_id
  FULL OUTER JOIN members mb ON s.customer_id = mb.customer_id
  WHERE order_date < join_date
  GROUP BY s.customer_id
```



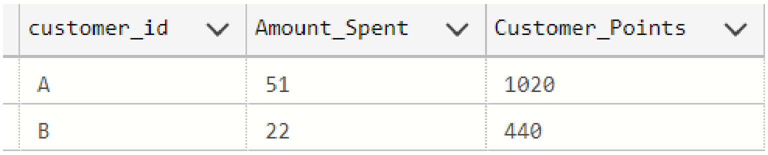
customer_id	Total_items	Amount_spent
A	2	25
B	3	40

```sql
--9--If each $1 spent equates to 10 points and sushi has a 2x points multiplier - how many
points would each customer have?
  SELECT customer_id, SUM(price) Amount_Spent, SUM(price*10*Multiplier) AS Customer_Points
FROM
  (SELECT s.customer_id, m.price, m.product_name,
  CASE WHEN m.product_name = 'sushi' THEN 2
  ELSE 1 END AS Multiplier
  FROM sales s
  LEFT JOIN menu m ON s.product_id = m.product_id
  LEFT JOIN members mb ON s.customer_id = mb.customer_id) TB
  GROUP BY customer_id
```



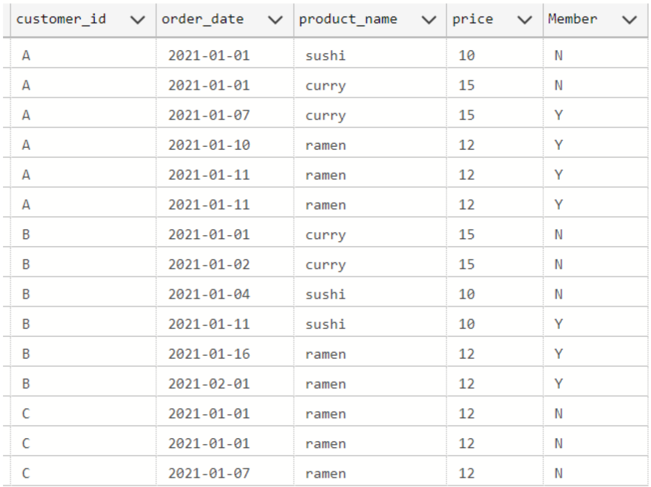
customer_id	Amount_Spent	Customer_Points
A	76	860
B	74	940
C	36	360

```sql
--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 does customer A and B have at the end of January?
  SELECT s.customer_id, SUM(price*10*2) AS Points_Earned
  FROM sales s
  LEFT JOIN menu m ON s.product_id = m.product_id
  LEFT JOIN members mb ON s.customer_id = mb.customer_id
  WHERE order_date>=join_date
  AND order_date BETWEEN '2021-01-01' AND '2021-01-31'
  GROUP BY s.customer_id

  OR

  SELECT customer_id, SUM(price) Amount_Spent, SUM(price*10*2) AS Customer_Points FROM
  (SELECT s.customer_id, m.price, m.product_name
    FROM sales s
  LEFT JOIN menu m ON s.product_id = m.product_id
  LEFT JOIN members mb ON s.customer_id = mb.customer_id
  WHERE order_date>=join_date
  AND order_date BETWEEN '2021-01-01' AND '2021-01-31') TB
  GROUP BY customer_id
```



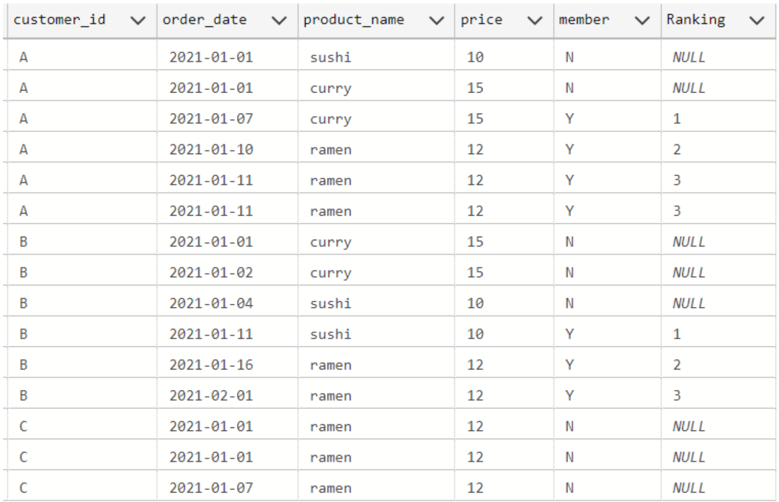
customer_id	Amount_Spent	Customer_Points
A	51	1020
B	22	440

```
  --Bonus--Join All The Things
    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
    FULL OUTER JOIN menu m ON s.product_id = m.product_id
    FULL OUTER JOIN members mb ON s.customer_id = mb.customer_id
```



customer_id	order_date	product_name	price	Member
A	2021-01-01	sushi	10	N
A	2021-01-01	curry	15	N
A	2021-01-07	curry	15	Y
A	2021-01-10	ramen	12	Y
A	2021-01-11	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

```
---Rank All The Things---CTE--
WITH sq AS (
 SELECT s.customer_id, s.order_date, mn.product_name, mn.price,
   CASE
    WHEN s.order_date >= me.join_date THEN 'Y'
    ELSE 'N'
   END AS [member]
  FROM sales s
  LEFT JOIN menu mn
  ON mn.product_id = s.product_id
  LEFT JOIN members me
  ON me.customer_id = s.customer_id)


 SELECT *,  CASE
    WHEN sq.member = 'N' THEN NULL
    ELSE RANK() OVER(PARTITION BY customer_id, member ORDER BY  order_date)
    END AS Ranking
    FROM sq
```



customer_id	order_date	product_name	price	member	Ranking
A	2021-01-01	sushi	10	N	*NULL*
A	2021-01-01	curry	15	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*