**Case Study #1 - Danny's Diner**

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

**SELECT**

s.customer\_id ,

**SUM**( m.price ) **AS** total\_amount

**FROM** sales s

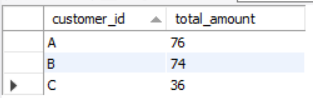
**JOIN**

menu m

**ON**

s.product\_id = m.product\_id

**GROUP BY** 1 ;



1. **How many days has each customer visited the restaurant?**

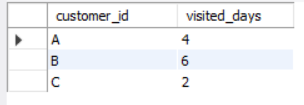
**SELECT**

customer\_id ,

**COUNT**( **DISTINCT** order\_date ) **AS** visited\_days

**FROM** sales

**GROUP BY** 1 ;



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

**WITH** result **AS**

(

**SELECT**

s.customer\_id ,

s.order\_date ,

m.product\_name ,

**RANK**() **OVER** (**PARTITION BY** customer\_id **ORDER BY** order\_date **ASC** ) **AS** num\_purchase

**FROM** sales s

**LEFT JOIN**

menu m

**ON**

s.product\_id = m.product\_id

)

**SELECT**

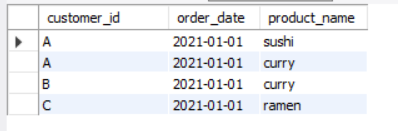
**DISTINCT** customer\_id ,

order\_date ,

product\_name

**FROM** result

**WHERE** num\_purchase = 1 ;



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

**SELECT**

m.product\_name ,

**COUNT**(s.product\_id) **AS** num\_purchased

**FROM**  menu m

**JOIN**

sales s

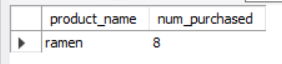
**ON**

m.product\_id = s.product\_id

**GROUP BY** 1

**ORDER BY** 2 DESC

**LIMIT** 1 ;



1. **Which item was the most popular for each customer?**

**WITH** semi\_result **AS**

**(**

**SELECT** s.customer\_id , m.product\_name , **COUNT**(s.order\_date) **AS** num\_purchase

**FROM** sales s

**JOIN**

menu m

**ON**

s.product\_id = m.product\_id

**GROUP BY** 1,2

**) ,**

result **AS**

**(**

**SELECT**

\* ,

**RANK() OVER (PARTITION BY** customer\_id **ORDER BY** num\_purchase desc**) AS** num

**FROM** semi\_result

**)**

**SELECT** customer\_id , product\_name

**FROM** result

**WHERE** num = 1 ;



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

**WITH** result **AS**

**(**

**SELECT**

s.customer\_id ,

s.order\_date ,

mem.join\_date ,

s.product\_id,

**RANK() OVER (PARTITION BY** customer\_id **ORDER BY** order\_date**) AS** ranks

**FROM** sales s

**JOIN**

members mem

**ON**

s.customer\_id = mem.customer\_id

**where** s.order\_date >= mem.join\_date

**)**

**SELECT**

r.customer\_id ,

m.product\_name ,

r.order\_date

**FROM** result r

**JOIN**

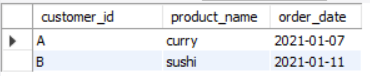
menu m

**ON**

r.product\_id = m.product\_id

**WHERE** ranks = 1

**ORDER BY** 1 ;



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

**WITH** result **AS**

**(**

**SELECT**

s.customer\_id ,

s.order\_date ,

mem.join\_date ,

m.product\_name ,

**RANK() OVER (PARTITION BY** customer\_id **ORDER BY** s.order\_date **DESC) AS** num

**FROM** sales s

**JOIN**

menu m

**ON**

s.product\_id = m.product\_id

**JOIN**

members mem

**ON**

s.customer\_id = mem.customer\_id

**where** s.order\_date < mem.join\_date

**)**

**SELECT**

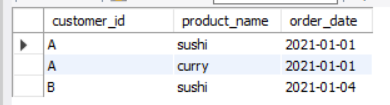
customer\_id ,

product\_name ,

order\_date

**FROM** result

**WHERE** num = 1 ;



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

**SELECT**

s.customer\_id ,

**COUNT**(s.product\_id) as total\_items ,

**SUM**(m.price) total\_amount

**FROM** sales s

**JOIN**

menu m

**ON**

s.product\_id = m.product\_id

**RIGHT JOIN**

members mem

**ON**

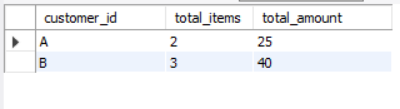
s.customer\_id = mem.customer\_id

**WHERE**

s.order\_date < mem.join\_date

**GROUP BY** 1

**ORDER BY** 1 ;



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

**WITH** result **AS**

**(**

**SELECT**

s.customer\_id ,

s.product\_id ,

m.product\_name ,

m.price ,

**CASE WHEN** m.product\_name = 'sushi' **THEN m.price\*2\*10 ELSE** **m.price\*10** **END AS** points

**FROM** sales s

**JOIN**

menu m

**ON**

s.product\_id = m.product\_id

**)**

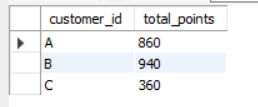
**SELECT**

customer\_id ,

**SUM**(points) as total\_points

**FROM** result

**GROUP BY** 1 ;



1. **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** result **AS**

**(**

**SELECT**

s.customer\_id ,

s.order\_date ,

mem.join\_date ,

m.product\_name ,

m.price ,

**CASE**

**WHEN** **m.product\_name = 'sushi'** **THEN** **m.price\*2\*10**

**WHEN** **s.order\_date >= mem.join\_date**

**AND**

**s.order\_date < date\_add(mem.join\_date , interval 1 WEEK)** **THEN** **m.price\*2\*10**

**ELSE** m.price\*10 **END AS** point

**FROM** sales s

**JOIN**

menu m

**ON**

s.product\_id = m.product\_id

**JOIN**

members mem

**ON**

s.customer\_id = mem.customer\_id

s.order\_date <= '2021-01-31'

**)**

**SELECT**

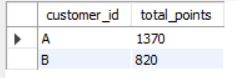
customer\_id ,

SUM(point) as total\_points

**FROM** result

**GROUP BY** 1

**ORDER BY** 1;



BONUS QUESTIONS

Join All The Table - Recreate the table : customer\_id,order\_date ,product\_name, price,member (Y / N)

**SELECT**

s.customer\_id ,

s.order\_date ,

m.product\_name ,

m.price ,

**CASE**

**WHEN** mem.join\_date is null **THEN** 'N'

**WHEN** mem.join\_date is not null **AND** s.order\_date < mem.join\_date **THEN** 'N'

**ELSE** 'Y' **END** **AS** member

**FROM** sales s

**LEFT JOIN**

menu m

**ON**

s.product\_id = m.product\_id

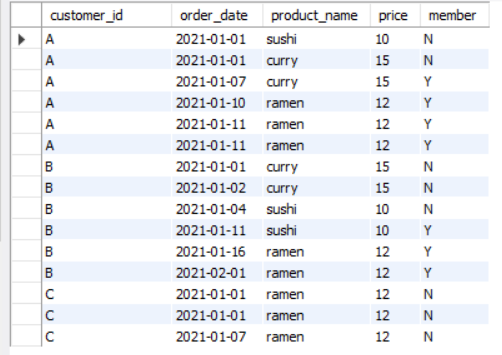
**LEFT JOIN**

members mem

**ON**

s.customer\_id = mem.customer\_id

**ORDER BY** 1 ;



**Rank All The Things**

Danny also requires further information about the ranking of customer products, but he purposely does not need the ranking for non-member purchases so he expects null ranking values for the records when customers are not yet part of the loyalty program.

**WITH** result **AS**

**(**

**SELECT**

s.customer\_id ,

s.order\_date ,

m.product\_name ,

m.price ,

**CASE**

**WHEN** mem.join\_date is null **THEN** 'N'

**WHEN** mem.join\_date is not null **AND** s.order\_date < mem.join\_date **THEN** 'N'

**ELSE** 'Y' **END** **AS** member

**FROM** sales s

**LEFT JOIN**

menu m

**ON**

s.product\_id = m.product\_id

**LEFT JOIN**

members mem

**ON**

s.customer\_id = mem.customer\_id

**ORDER BY** 1 ;

**)**

**SELECT**

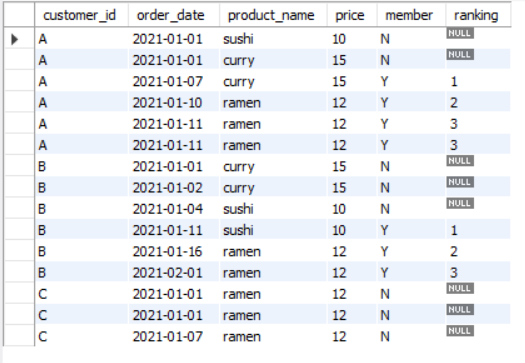
\* ,

**CASE**

**WHEN** member = 'Y' **THEN RANK() OVER (PARTITION BY** customer\_id ,member **ORDER BY** order\_date**)**

**ELSE** NULL **END AS** ranking

**FROM** result ;



Name - Sudhanshu Tomar

Linkedin - <https://www.linkedin.com/in/sudhanshu-tomar-493633255/>

Source - <https://8weeksqlchallenge.com/case-study-1/>

