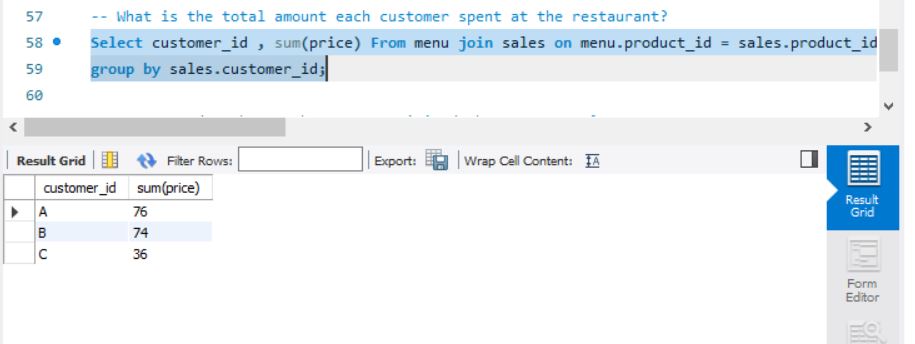
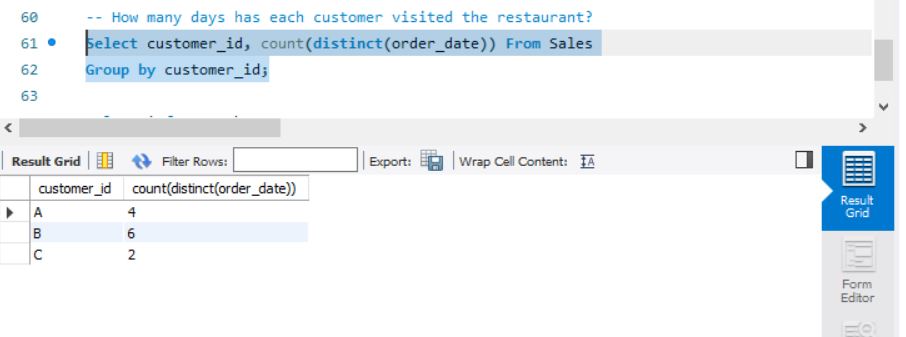
**Case Study – Dannys Dinner**

Danny 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 also which menu items are their favourite. Having this deeper connection with his customers will help him deliver a better and more personalised experience for his loyal customers. He plans on using these insights to help him decide whether he should expand the existing customer loyalty program.  
  
The sample data is been provided sales table, menu table, members table. There is a set of problem statement which need the insights. Using Sql query fetched the insights

1. What is the total amount each customer spent at the restaurant?  
     
   Select customer\_id , sum(price) From menu join sales on menu.product\_id = sales.product\_id group by sales.customer\_id;  
     
   
2. How many days has each customer visited the restaurant?  
     
   Select customer\_id, count(distinct(order\_date)) From Sales Group by customer\_id;  
     
   
3. What was the first item from the menu purchased by each customer?

With cte As

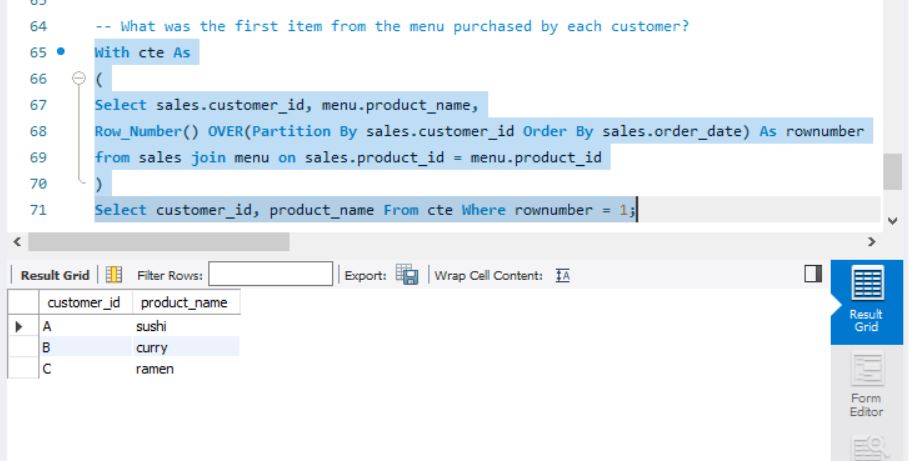
(

Select sales.customer\_id, menu.product\_name,

Row\_Number() OVER(Partition By sales.customer\_id Order By sales.order\_date) As rownumber

from sales join menu on sales.product\_id = menu.product\_id

)

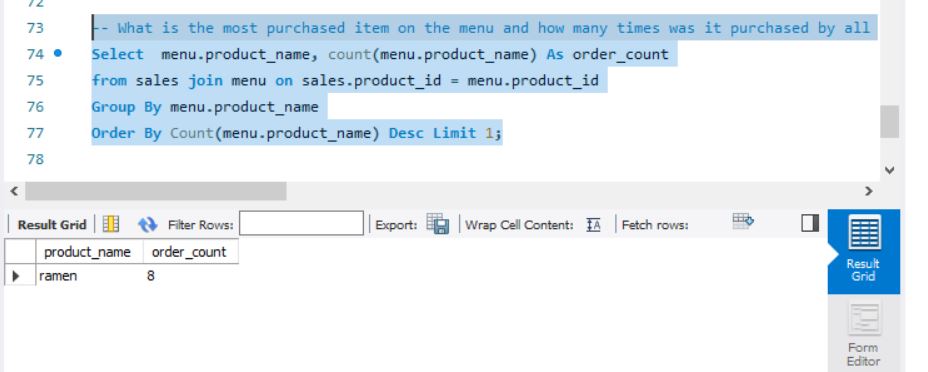
Select customer\_id, product\_name From cte Where rownumber = 1;  
  


1. What is the most purchased item on the menu and how many times was it purchased by all customers?  
     
   Select menu.product\_name, count(menu.product\_name) As order\_count

from sales join menu on sales.product\_id = menu.product\_id

Group By menu.product\_name

Order By Count(menu.product\_name) Desc Limit 1;



1. Which item was the most popular for each customer?  
     
   With popular\_item as (

Select sales.customer\_id, menu.product\_name,

Count(\*) as order\_count,

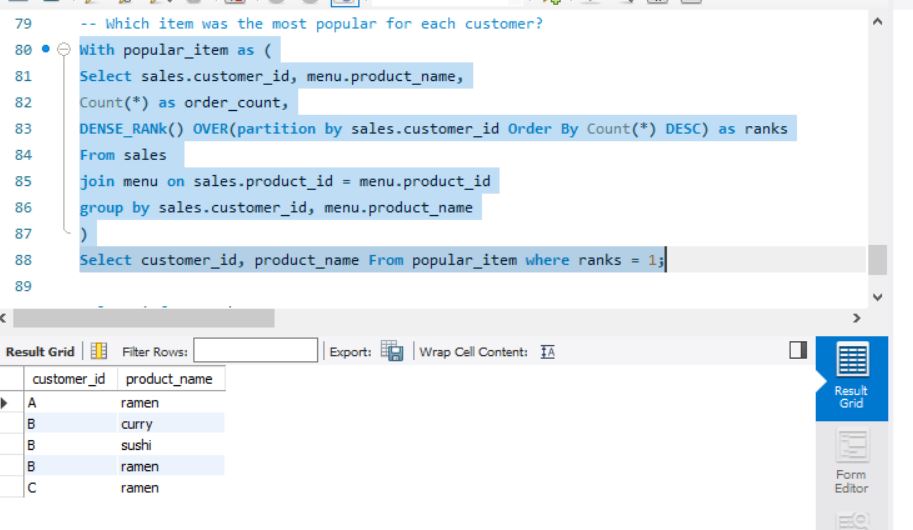
DENSE\_RANk() OVER(partition by sales.customer\_id Order By Count(\*) DESC) as ranks

From sales

join menu on sales.product\_id = menu.product\_id

group by sales.customer\_id, menu.product\_name

)

Select customer\_id, product\_name From popular\_item where ranks = 1;  
  


1. Which item was purchased first by the customer after they became a member?  
     
   With orders As (

Select sales.customer\_id, menu.product\_name, sales.order\_date, members.join\_date,

DENSE\_RANk() OVER(Partition By sales.customer\_id Order By order\_date) As ranks

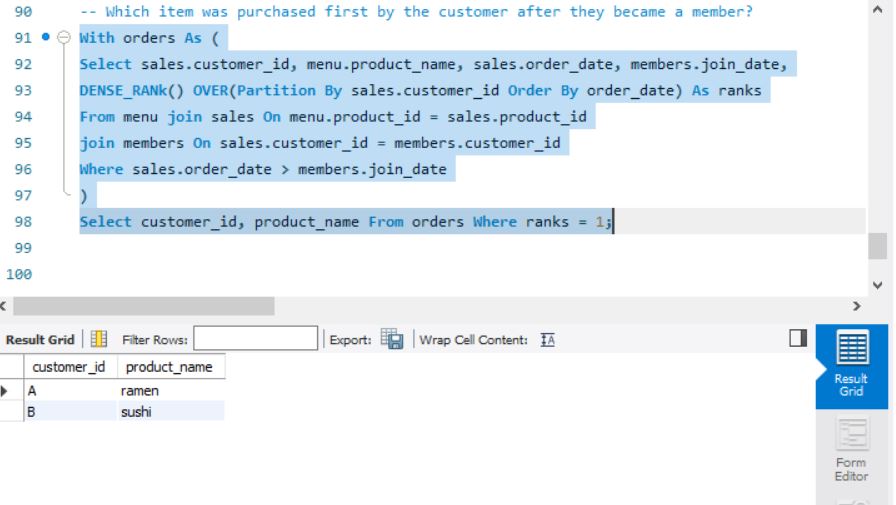
From menu join sales On menu.product\_id = sales.product\_id

join members On sales.customer\_id = members.customer\_id

Where sales.order\_date > members.join\_date

)

Select customer\_id, product\_name From orders Where ranks = 1;



1. Which item was purchased just before the customer became a member?  
     
   With orders As (

Select sales.customer\_id, menu.product\_name, sales.order\_date, members.join\_date,

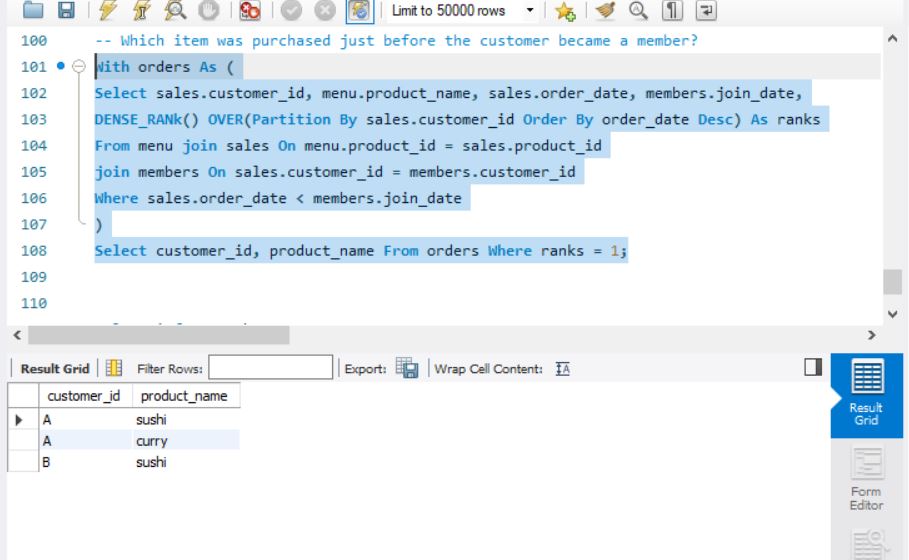
DENSE\_RANk() OVER(Partition By sales.customer\_id Order By order\_date Desc) As ranks

From menu join sales On menu.product\_id = sales.product\_id

join members On sales.customer\_id = members.customer\_id

Where sales.order\_date < members.join\_date

)

Select customer\_id, product\_name From orders Where ranks = 1;  
  


1. What is the total items and amount spent for each member before they became a member?  
     
   Select sales.customer\_id, sales.order\_date, members.join\_date,

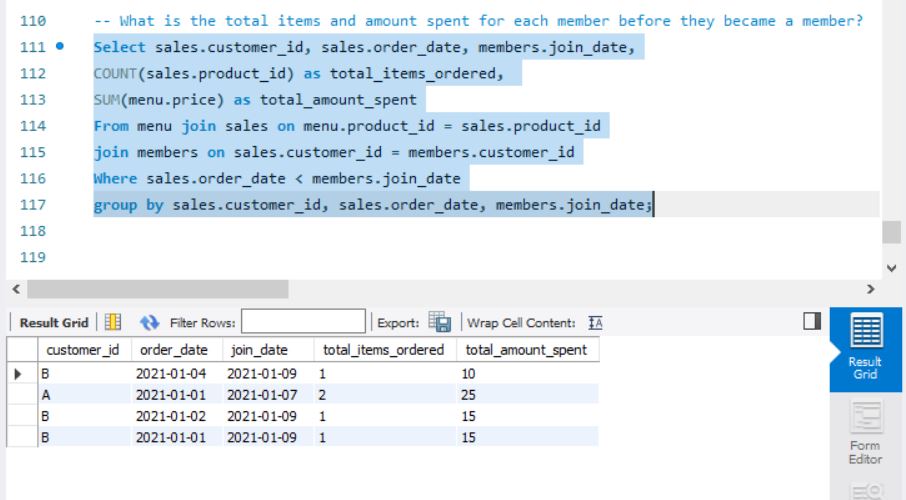
COUNT(sales.product\_id) as total\_items\_ordered,

SUM(menu.price) as total\_amount\_spent

From menu join sales on menu.product\_id = sales.product\_id

join members on sales.customer\_id = members.customer\_id

Where sales.order\_date < members.join\_date

group by sales.customer\_id, sales.order\_date, members.join\_date;  
  


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

Select sales.customer\_id, menu.product\_name, menu.price,

CASE

WHEN menu.product\_name = 'sushi' Then menu.price\*10\*2

ELSE menu.price\*10

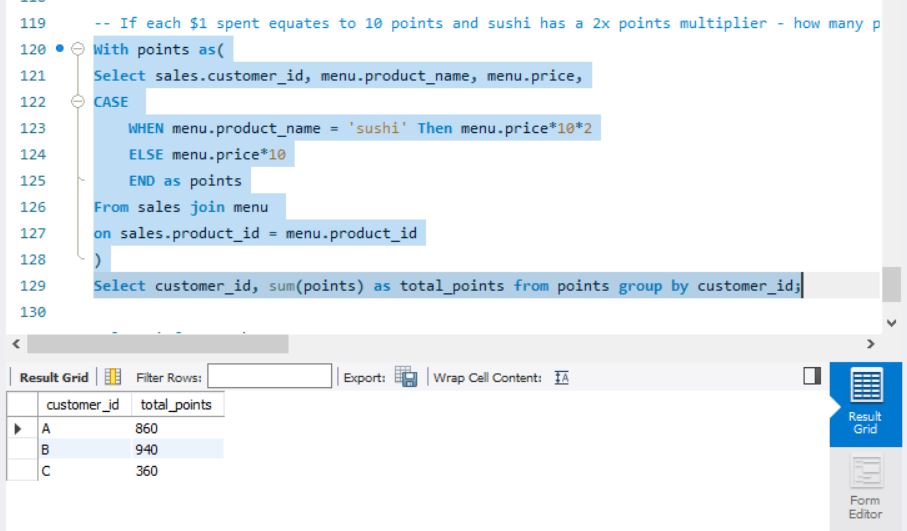
END as points

From sales join menu

on sales.product\_id = menu.product\_id

)

Select customer\_id, sum(points) as total\_points from points group by customer\_id;



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 January?  
  
With points as(

Select sales.customer\_id, menu.product\_name, menu.price, sales.order\_date, members.join\_date,

CASE

WHEN sales.order\_date BETWEEN members.join\_date AND DATE\_ADD(members.join\_date, INTERVAL 7 DAY) THEN menu.price\*10\*2

WHEN menu.product\_name - 'sushi' Then menu.price\*10\*2

ELSE menu.price\*10

END as points

FROM menu

join sales on menu.product\_id = sales.product\_id

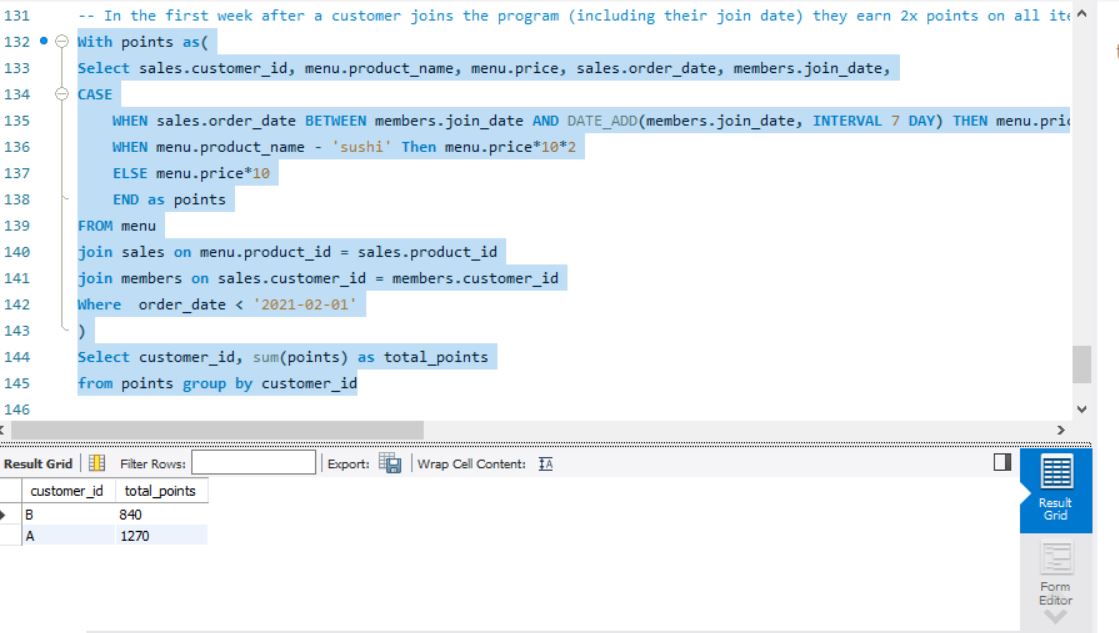
join members on sales.customer\_id = members.customer\_id

Where order\_date < '2021-02-01'

)

Select customer\_id, sum(points) as total\_points

from points group by customer\_id;



11) Determine the name and price of the product ordered by each customer on all order dates & find out whether the customer was a member on the order date or not  
  
Select sales.customer\_id, sales.order\_date, menu.product\_name, menu.price,

CASE

When members.join\_date <= sales.order\_date Then 'Y'

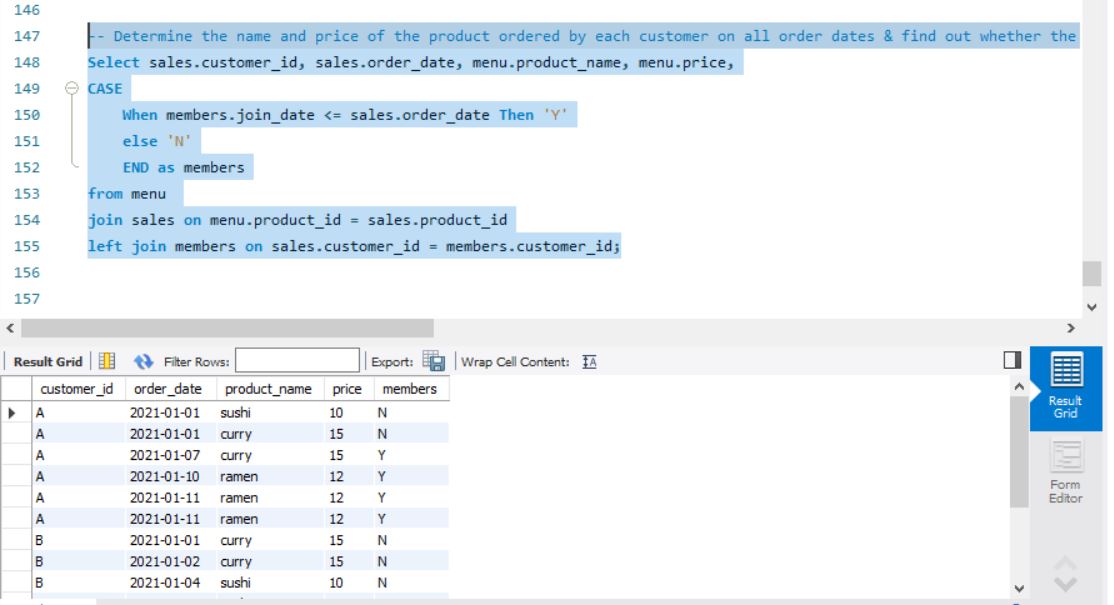
else 'N'

END as members

from menu

join sales on menu.product\_id = sales.product\_id

left join members on sales.customer\_id = members.customer\_id;



12) Rank the previous output based the order\_date for each customer. Display null if customer was not a member when dis was ordered  
  
with ranks as(

Select sales.customer\_id, sales.order\_date, menu.product\_name, menu.price,

CASE

When members.join\_date <= sales.order\_date Then 'Y'

else 'N'

END as member\_status

from menu

join sales on menu.product\_id = sales.product\_id

left join members on sales.customer\_id = members.customer\_id

)

Select \*,

CASE

WHEN ranks.member\_status = 'Y' THEN RANK() OVER(Partition By ranks.customer\_id, ranks.member\_status Order By order\_date)

ELSE NULL

END As ranking

from ranks;  
  
