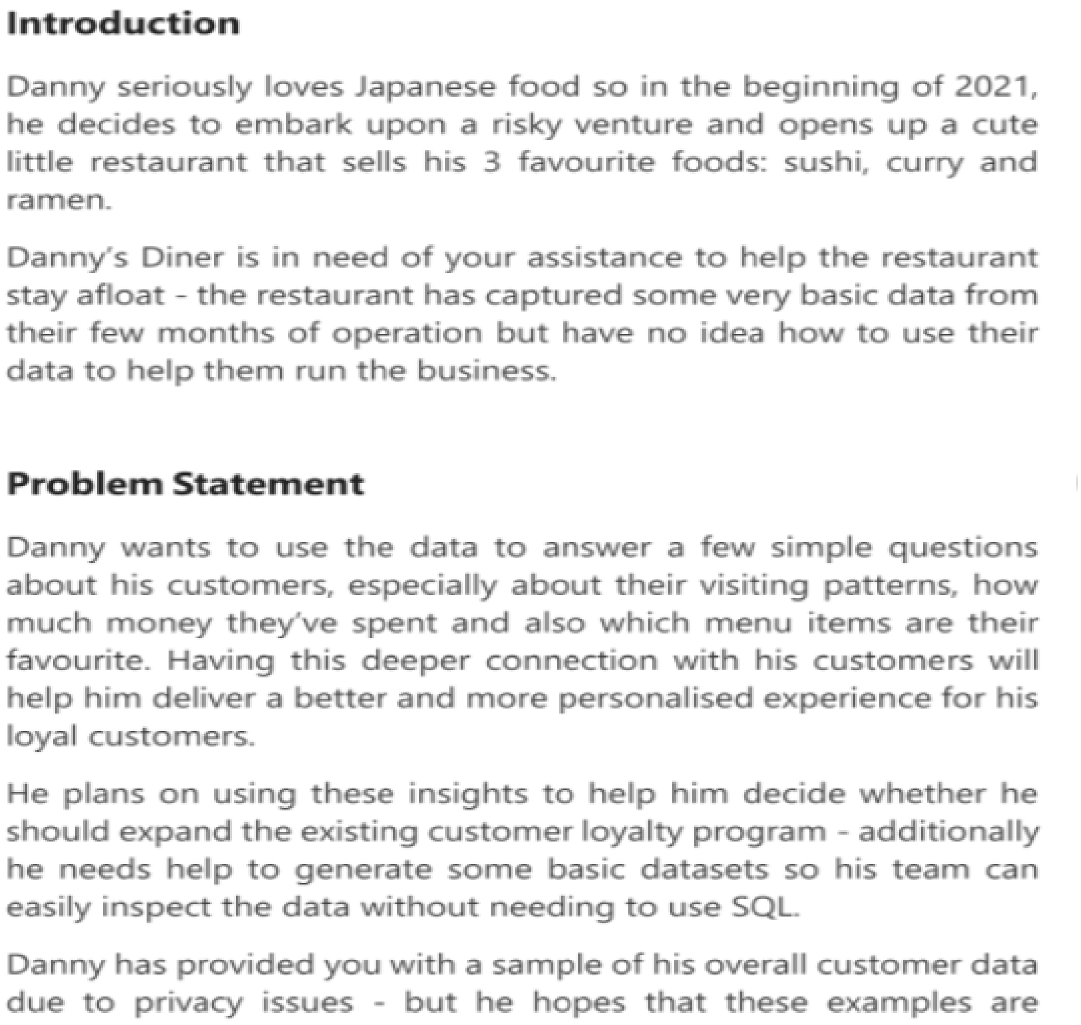
8WEEKSQLCHALLENGE.COM

CASE STUDY #1

THE TASTE OF SUCCESS

DATAWITHDANNY.COM

Case Study #1 of 8 Week SQL Challenge



Introduction

Danny seriously loves Japanese food so in the beginning of 2021, he decides to embark upon a risky venture and opens up a cute little restaurant that sells his 3 favourite foods: sushi, curry and ramen.

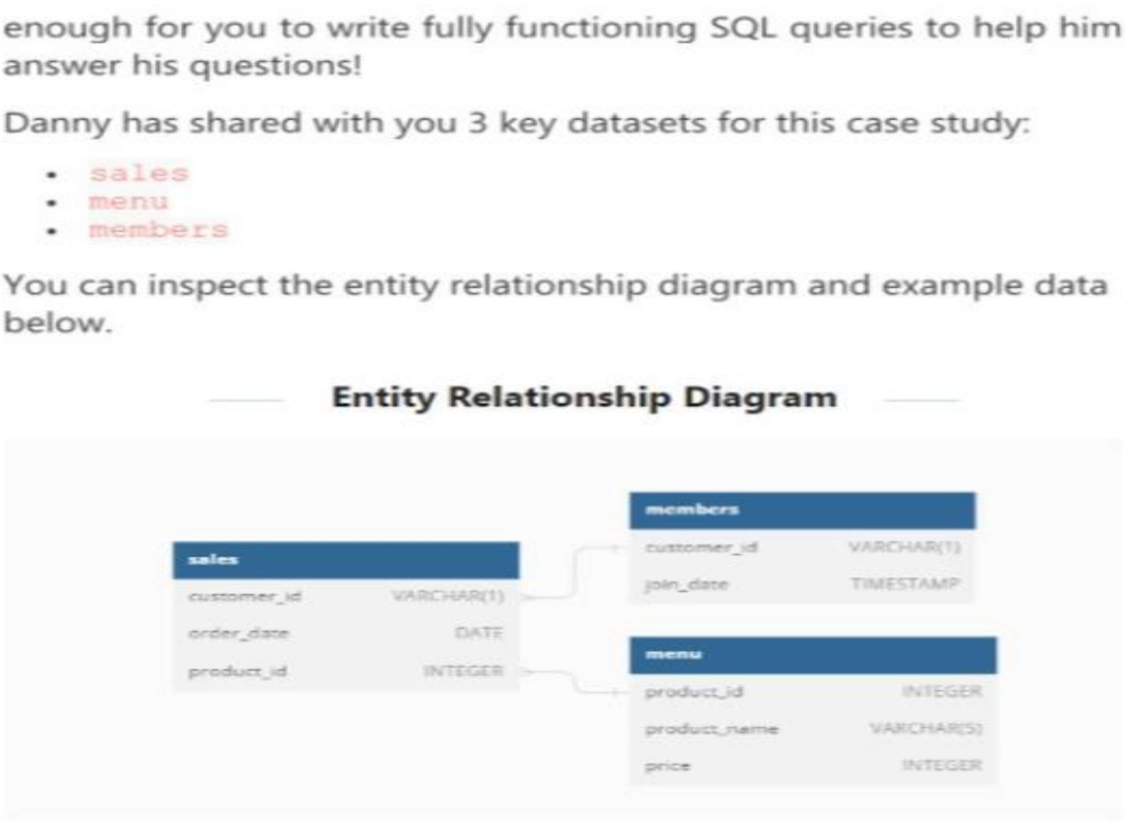
Danny's Diner is in need of your assistance to help the restaurant stay afloat - the restaurant has captured some very basic data from their few months of operation but have no idea how to use their data to help them run the business.

Problem Statement

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 - additionally he needs help to generate some basic datasets so his team can easily inspect the data without needing to use SQL.

Danny has provided you with a sample of his overall customer data due to privacy issues - but he hopes that these examples are



enough for you to write fully functioning SQL queries to help him answer his questions!

Danny has shared with you 3 key datasets for this case study:

- `sales`
- `menu`
- `members`

You can inspect the entity relationship diagram and example data below.

Entity Relationship Diagram

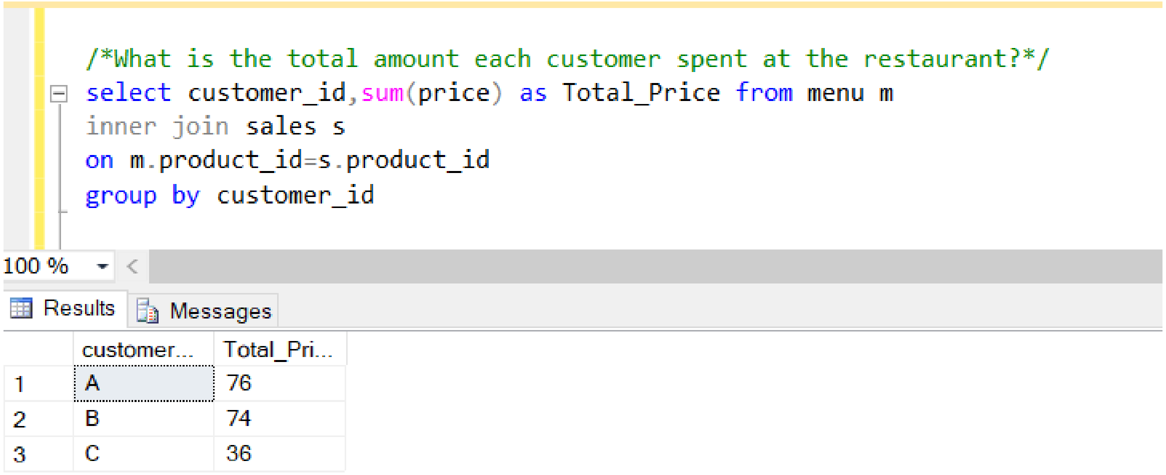
sales	
customer_id	VARCHAR(1)
order_date	DATE
product_id	INTEGER

members	
customer_id	VARCHAR(1)
join_date	TIMESTAMP

menu	
product_id	INTEGER
product_name	VARCHAR(5)
price	INTEGER

Case Study Questions.

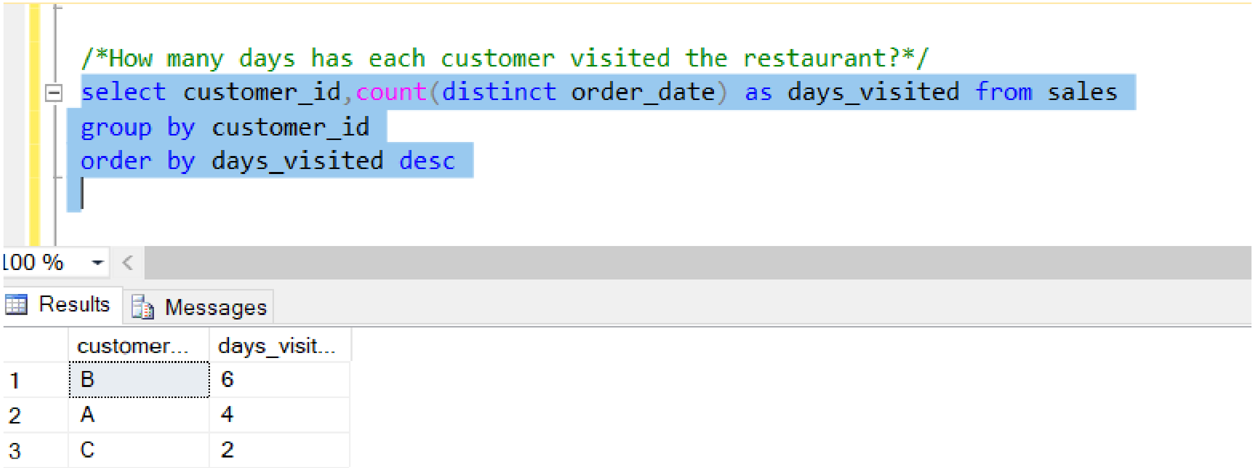
1-What is the total amount each customer spent at the restaurant?



```sql
/*What is the total amount each customer spent at the restaurant?*/
select customer_id,sum(price) as Total_Price from menu m
inner join sales s
on m.product_id=s.product_id
group by customer_id
```

	customer...	Total_Pri...
1	A	76
2	B	74
3	C	36

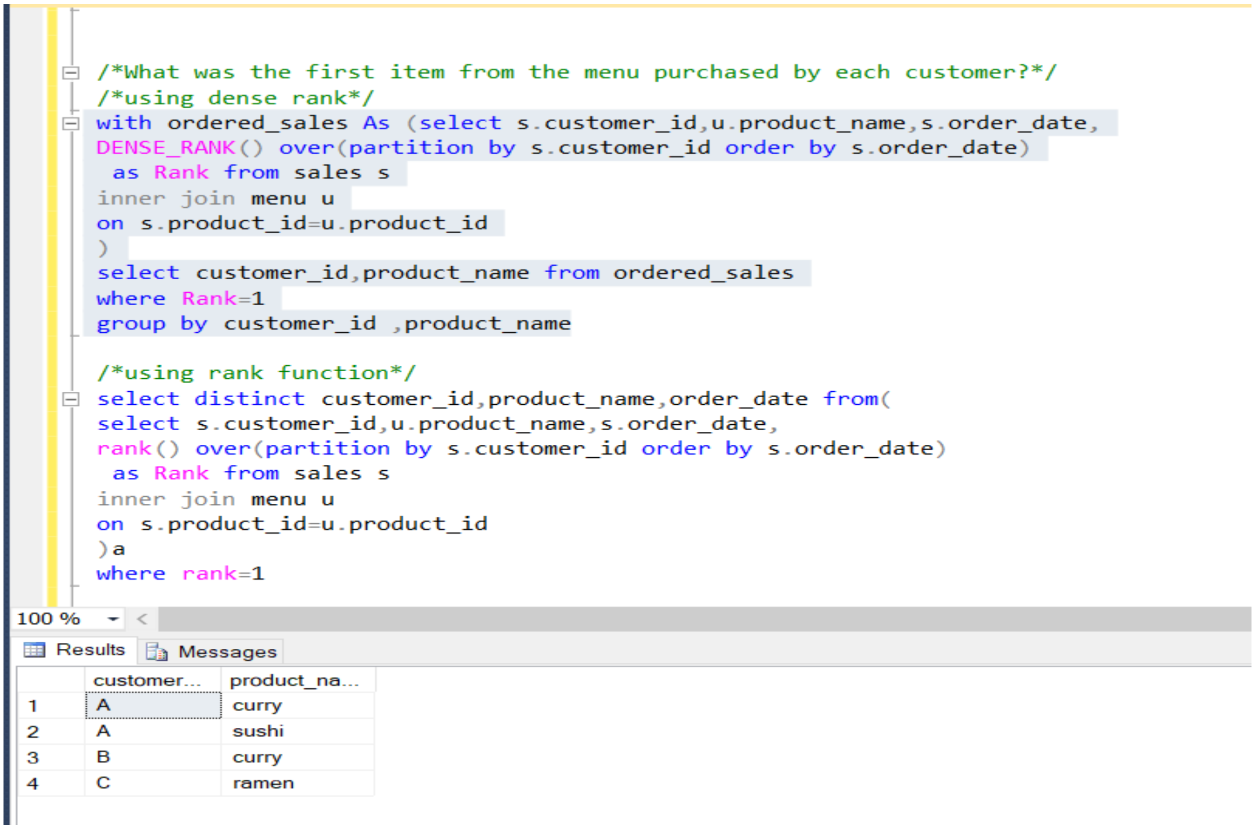
2-How many days has each customer visited the restaurant?



```
/*How many days has each customer visited the restaurant?*/
select customer_id,count(distinct order_date) as days_visited from sales
group by customer_id
order by days_visited desc
```

	customer...	days_visit...
1	B	6
2	A	4
3	C	2

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

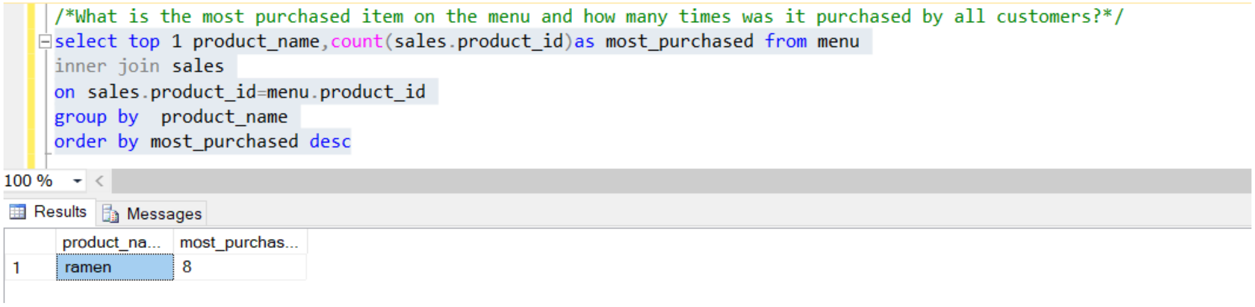


```
/*What was the first item from the menu purchased by each customer?*/
/*using dense rank*/
with ordered_sales As (select s.customer_id,u.product_name,s.order_date,
DENSE_RANK() over(partition by s.customer_id order by s.order_date)
 as Rank from sales s
inner join menu u
on s.product_id=u.product_id
)
select customer_id,product_name from ordered_sales
where Rank=1
group by customer_id ,product_name

/*using rank function*/
select distinct customer_id,product_name,order_date from(
select s.customer_id,u.product_name,s.order_date,
rank() over(partition by s.customer_id order by s.order_date)
 as Rank from sales s
inner join menu u
on s.product_id=u.product_id
)a
where rank=1
```

	customer...	product_na...
1	A	curry
2	A	sushi
3	B	curry
4	C	ramen

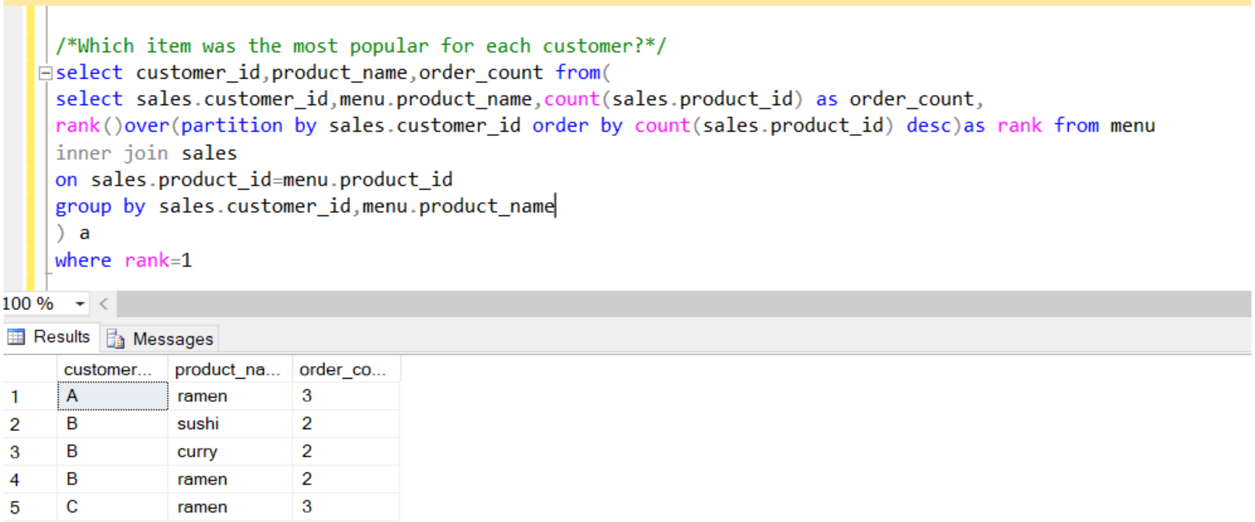
4-What is the most purchased item on the menu and how many times was it purchased by all customers?



```
/*What is the most purchased item on the menu and how many times was it purchased by all customers?*/
select top 1 product_name,count(sales.product_id)as most_purchased from menu
inner join sales
on sales.product_id=menu.product_id
group by  product_name
order by most_purchased desc
```

	product_na...	most_purchas...
1	ramen	8

5-Which item was the most popular for each customer?

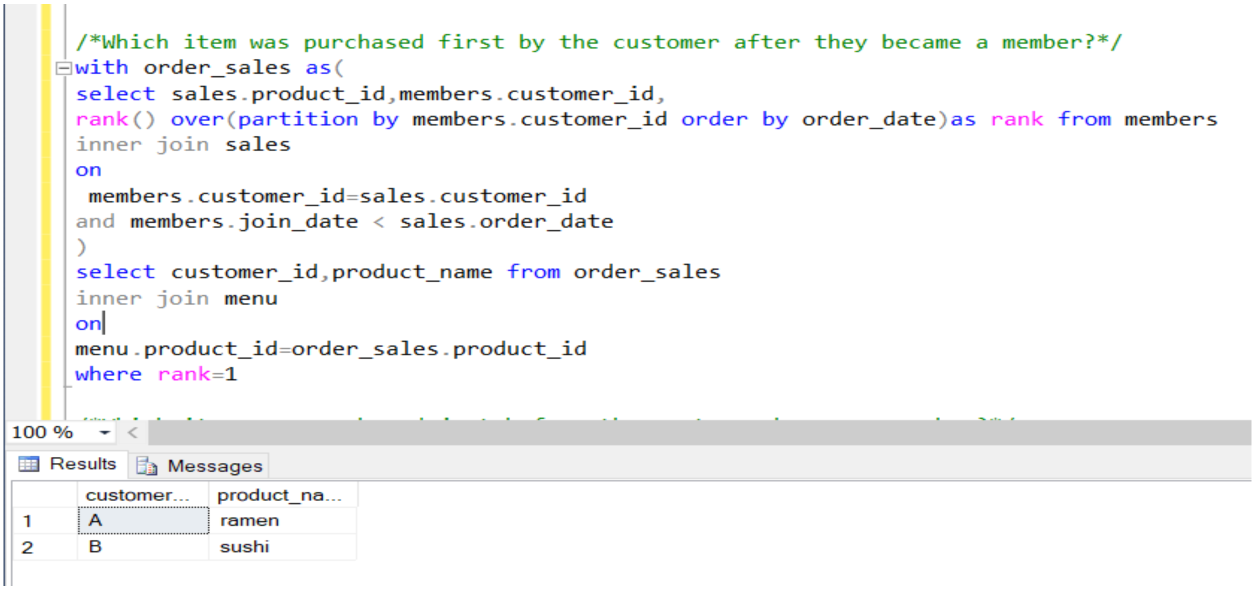


```

/*Which item was the most popular for each customer?*/
select customer_id,product_name,order_count from(
select sales.customer_id,menu.product_name,count(sales.product_id) as order_count,
rank()over(partition by sales.customer_id order by count(sales.product_id) desc)as rank from menu
inner join sales
on sales.product_id=menu.product_id
group by sales.customer_id,menu.product_name
) a
where rank=1
```

	customer...	product_na...	order_co...
1	A	ramen	3
2	B	sushi	2
3	B	curry	2
4	B	ramen	2
5	C	ramen	3

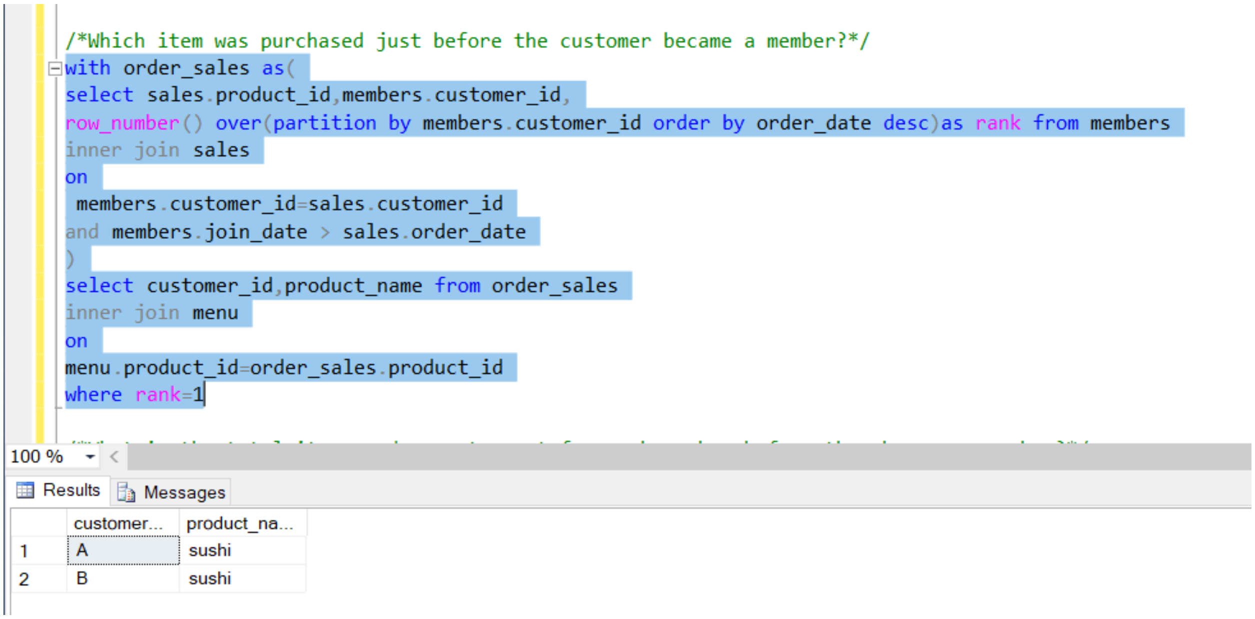
6-Which item was purchased first by the customer after they became a member?



```sql
/*Which item was purchased first by the customer after they became a member?*/
with order_sales as(
select sales.product_id,members.customer_id,
rank() over(partition by members.customer_id order by order_date)as rank from members
inner join sales
on
 members.customer_id=sales.customer_id
and members.join_date < sales.order_date
)
select customer_id,product_name from order_sales
inner join menu
on
menu.product_id=order_sales.product_id
where rank=1
```

	customer...	product_na...
1	A	ramen
2	B	sushi

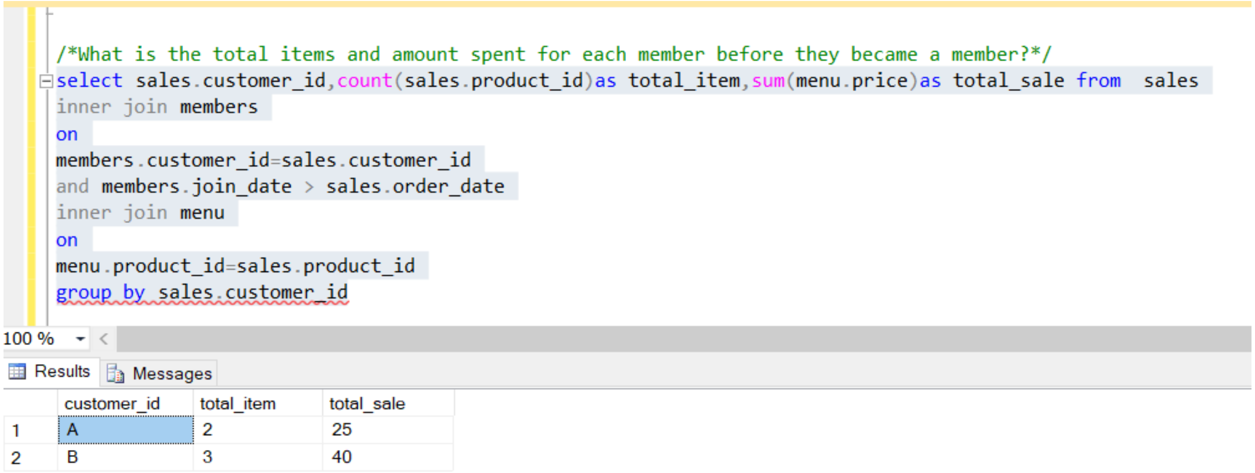
7-Which item was purchased just before the customer became a member?



```sql
/*Which item was purchased just before the customer became a member?*/
with order_sales as(
select sales.product_id,members.customer_id,
row_number() over(partition by members.customer_id order by order_date desc)as rank from members
inner join sales
on
 members.customer_id=sales.customer_id
and members.join_date > sales.order_date
)
select customer_id,product_name from order_sales
inner join menu
on
menu.product_id=order_sales.product_id
where rank=1
```

	customer...	product_na...
1	A	sushi
2	B	sushi

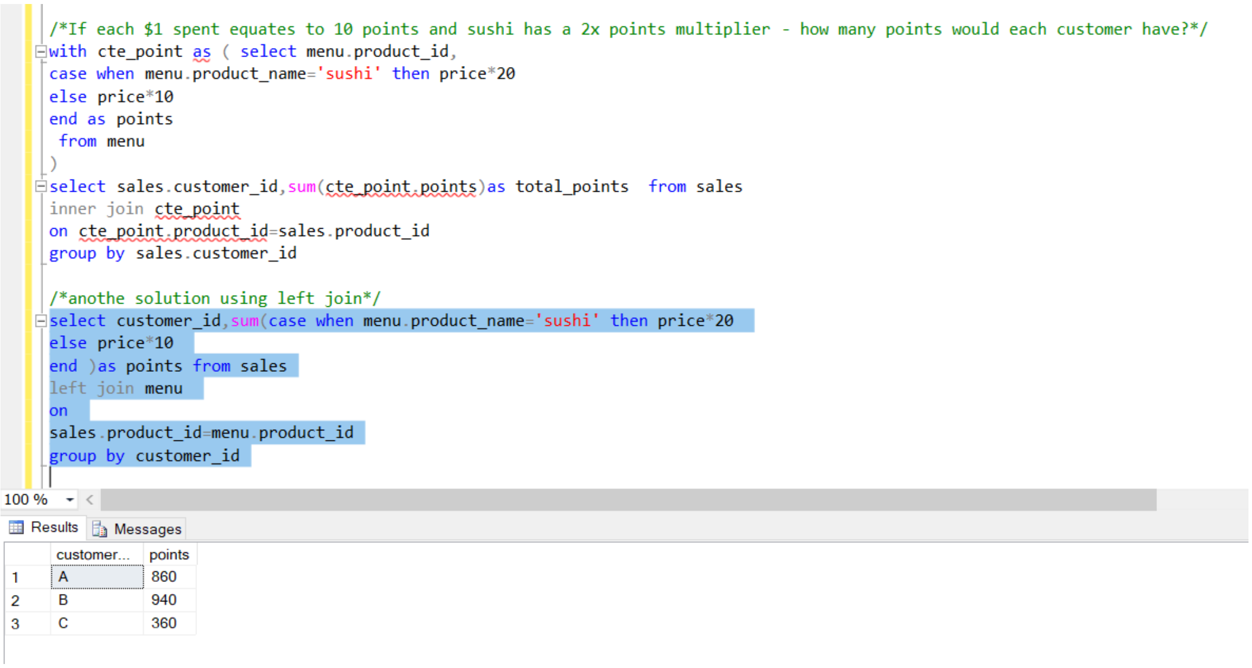
8-What is the total items and amount spent for each member before they became a member?



```
/*What is the total items and amount spent for each member before they became a member?*/
select sales.customer_id,count(sales.product_id)as total_item,sum(menu.price)as total_sale from  sales
inner join members
on
members.customer_id=sales.customer_id
and members.join_date > sales.order_date
inner join menu
on
menu.product_id=sales.product_id
group by sales.customer_id
```

	customer_id	total_item	total_sale
1	A	2	25
2	B	3	40

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



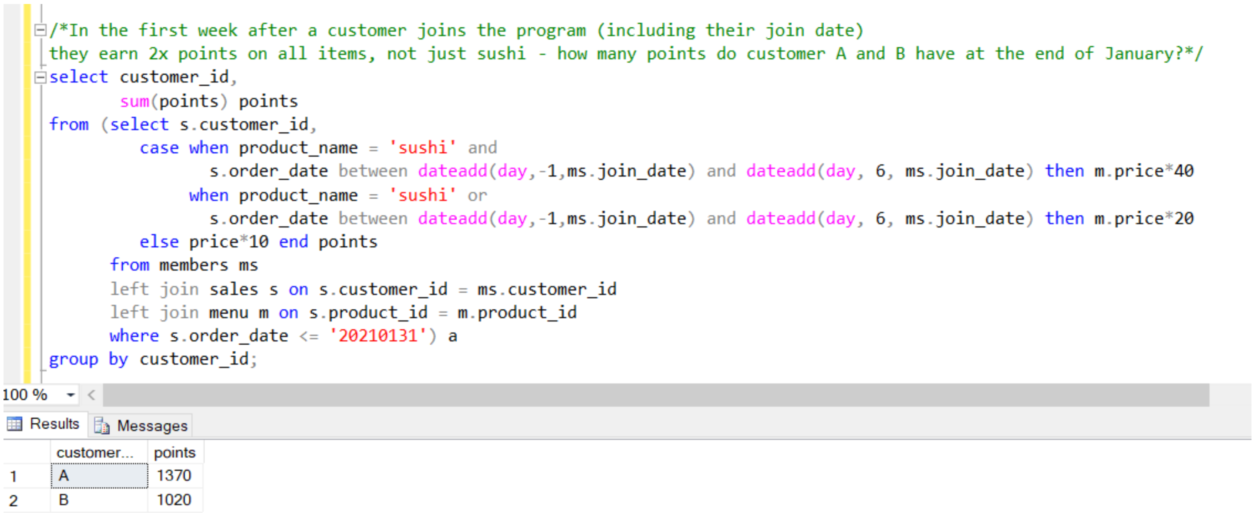
```
/*If each $1 spent equates to 10 points and sushi has a 2x points multiplier - how many points would each customer have?*/
with cte_point as ( select menu.product_id,
case when menu.product_name='sushi' then price*20
else price*10
end as points
 from menu
)
select sales.customer_id,sum(cte_point.points)as total_points  from sales
inner join cte_point
on cte_point.product_id=sales.product_id
group by sales.customer_id

/*anothe solution using left join*/
select customer_id,sum(case when menu.product_name='sushi' then price*20
else price*10
end )as points from sales
left join menu
on
sales.product_id=menu.product_id
group by customer_id
```

	customer...	points
1	A	860
2	B	940
3	C	360

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?



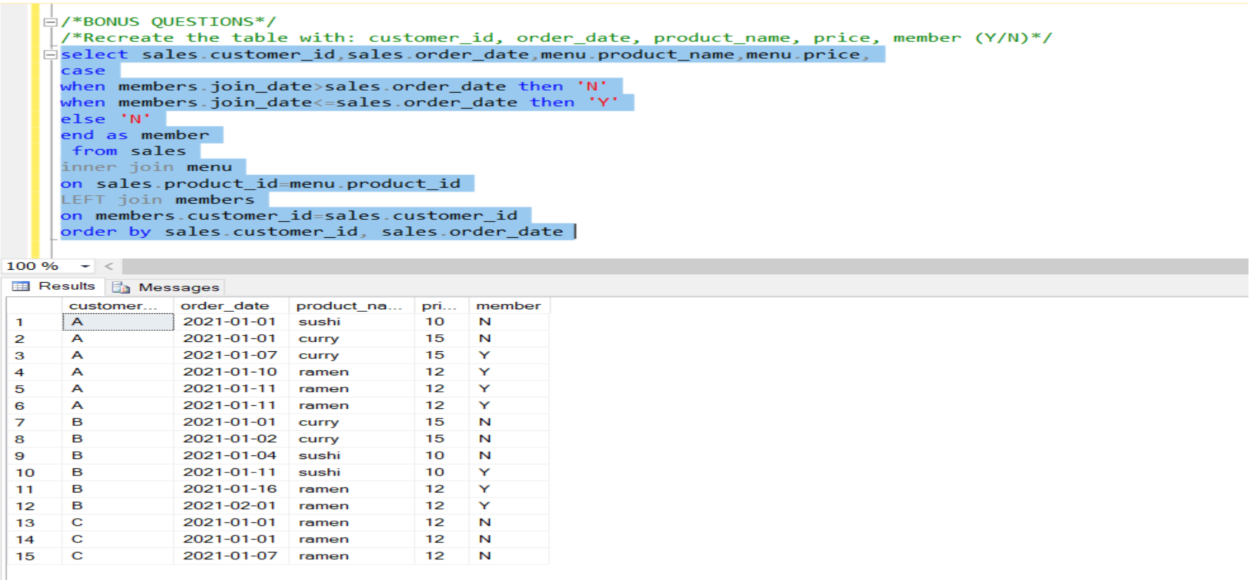
```
/*In the first week after a customer joins the program (including their join date)
they earn 2x points on all items, not just sushi - how many points do customer A and B have at the end of January?*/
select customer_id,
       sum(points) points
from (select s.customer_id,
         case when product_name = 'sushi' and
               s.order_date between dateadd(day,-1,ms.join_date) and dateadd(day, 6, ms.join_date) then m.price*40
              when product_name = 'sushi' or
               s.order_date between dateadd(day,-1,ms.join_date) and dateadd(day, 6, ms.join_date) then m.price*20
         else price*10 end points
      from members ms
      left join sales s on s.customer_id = ms.customer_id
      left join menu m on s.product_id = m.product_id
      where s.order_date <= '20210131') a
group by customer_id;
```

100 %

Results | Messages

	customer...	points
1	A	1370
2	B	1020

11-Recreate the table with: customer_id, order_date, product_name, price, member (Y/N)



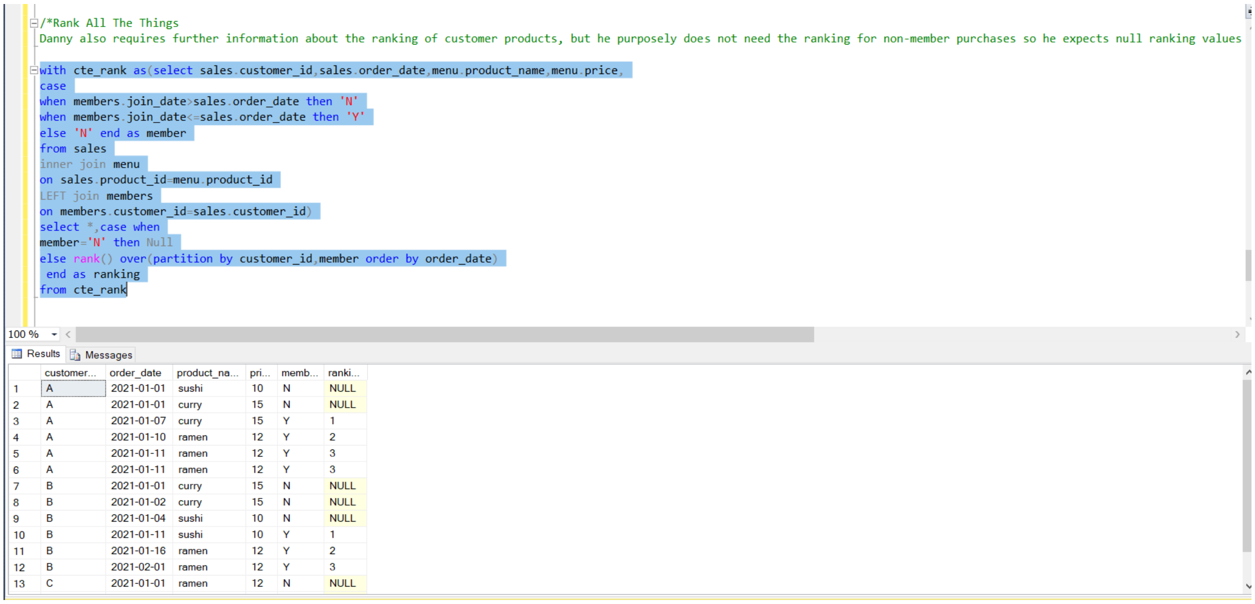
```
/*BONUS QUESTIONS*/
/*Recreate the table with: customer_id, order_date, product_name, price, member (Y/N)*/
select sales.customer_id,sales.order_date,menu.product_name,menu.price,
case
when members.join_date>sales.order_date then 'N'
when members.join_date<=sales.order_date then 'Y'
else 'N'
end as member
 from sales
inner join menu
on sales.product_id=menu.product_id
LEFT join members
on members.customer_id=sales.customer_id
order by sales.customer_id, sales.order_date
```

100 %

Results | Messages

	customer...	order_date	product_na...	pri...	member
1	A	2021-01-01	sushi	10	N
2	A	2021-01-01	curry	15	N
3	A	2021-01-07	curry	15	Y
4	A	2021-01-10	ramen	12	Y
5	A	2021-01-11	ramen	12	Y
6	A	2021-01-11	ramen	12	Y
7	B	2021-01-01	curry	15	N
8	B	2021-01-02	curry	15	N
9	B	2021-01-04	sushi	10	N
10	B	2021-01-11	sushi	10	Y
11	B	2021-01-16	ramen	12	Y
12	B	2021-02-01	ramen	12	Y
13	C	2021-01-01	ramen	12	N
14	C	2021-01-01	ramen	12	N
15	C	2021-01-07	ramen	12	N

12-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



```sql
/*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

with cte_rank as(select sales.customer_id,sales.order_date,menu.product_name,menu.price,
case
when members.join_date>sales.order_date then 'N'
when members.join_date<=sales.order_date then 'Y'
else 'N' end as member
from sales
inner join menu
on sales.product_id=menu.product_id
LEFT join members
on members.customer_id=sales.customer_id)
select *,case when
member='N' then Null
else rank() over(partition by customer_id,member order by order_date)
 end as ranking
from cte_rank
```

	customer...	order_date	product_na...	pri...	memb...	ranki...
1	A	2021-01-01	sushi	10	N	NULL
2	A	2021-01-01	curry	15	N	NULL
3	A	2021-01-07	curry	15	Y	1
4	A	2021-01-10	ramen	12	Y	2
5	A	2021-01-11	ramen	12	Y	3
6	A	2021-01-11	ramen	12	Y	3
7	B	2021-01-01	curry	15	N	NULL
8	B	2021-01-02	curry	15	N	NULL
9	B	2021-01-04	sushi	10	N	NULL
10	B	2021-01-11	sushi	10	Y	1
11	B	2021-01-16	ramen	12	Y	2
12	B	2021-02-01	ramen	12	Y	3
13	C	2021-01-01	ramen	12	N	NULL