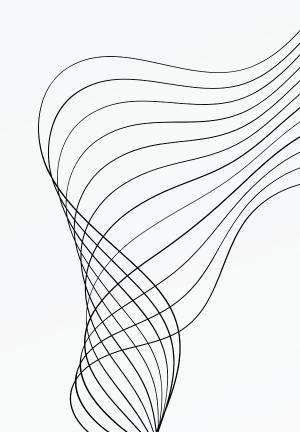


# SQL PROJECT



#### **Probelm 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 3 key datasets for this case study:

# INSERT INTO sales ("customer\_id", "order\_date", "product\_id")

CREATE DATABASE dannys\_diner; use dannys\_diner;

```
CREATE TABLE sales (
"customer_id" VARCHAR(1),
    "order_date" DATE,
    "product_id" INTEGER
    ):
```

```
VALUES
('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');
```

```
CREATE TABLE menu (
    "product_id" INTEGER,
"product_name" VARCHAR(5),
       "price" INTEGER
     INSERT INTO menu
("product_id", "product_name",
           "price")
           VALUES
       ('1', 'sushi', '10'),
       (2', curry', '15'),
       ('3', 'ramen', '12');
```

```
CREATE TABLE members (
"customer_id" VARCHAR(1),

"join_date" DATE

);
```

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

select s.customer\_id, sum(m.price) 'total amount spent' from sales s inner join menu m on m.product\_id=s.product\_id group by customer\_id

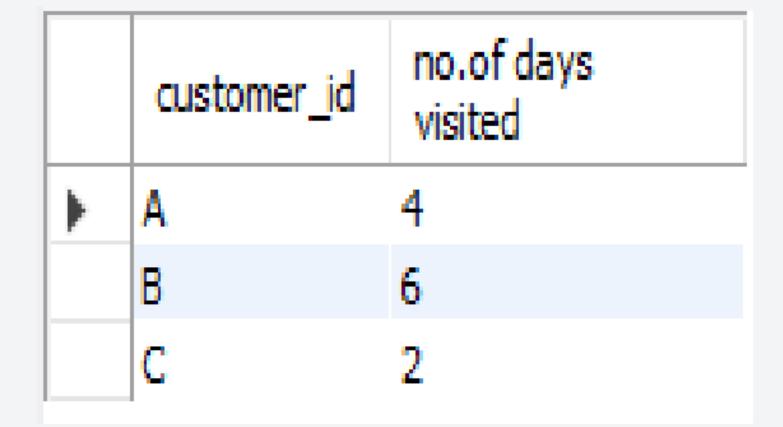
O	l l'	T	n		T	•
	U	L	$\mathbf{P}$	U	L	•

	customer_id	total amount spent
•	Α	76
	В	74
	С	36

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

select customer\_id,count(distinct(order\_date)) 'no.of days visited' from sales group by customer\_id

# output:



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

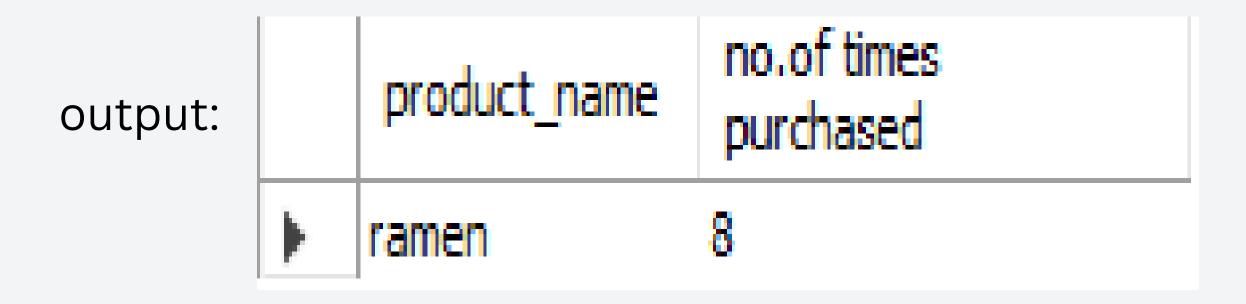
select customer\_id,product\_name from (select \*,rank() over(partition by customer\_id order by order\_date asc) rnk from (select s.customer\_id,s.order\_date,m.product\_name from sales s inner join menu m on s.product\_id=m.product\_id)a)b where rnk=1 group by customer\_id,product\_name

output:

	customer_id	product_name
•	A	sushi
	Α	curry
	В	curry
	С	ramen

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

select product\_name,count(\*) 'no.of times purchased' from sales s inner join menu m on s.product\_id=m.product\_id group by product\_name order by count(\*) desc limit 1

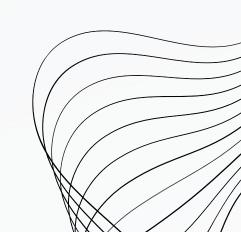


# 5. Which item was the most popular for each customer?

select customer\_id,product\_name 'popular item' from (select \*,rank() over(partition by customer\_id order by count desc) rnk from( select customer\_id,product\_name,count(\*) count from sales s inner join menu m on s.product\_id=m.product\_id group by customer\_id,product\_name)a)b where rnk=1;

output:

	customer_id	popular item
-	A	ramen
	В	curry
	В	sushi
	В	ramen
	C	ramen



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

select customer\_id,order\_date,product\_name from (select \*,rank() over(partition by customer\_id order by order\_date asc) rnk from (select s.customer\_id,s.order\_date,m.product\_name from sales s inner join menu m on s.product\_id=m.product\_id inner join members mem on s.customer\_id=mem.customer\_id where s.order\_date>=mem.join\_date)a)b where rnk=1;

output:		customer_id	order_date	product_name
-	•	A	2021-01-07	curry
		В	2021-01-11	sushi

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

select customer\_id,order\_date,product\_name from (select \*,rank() over(partition by customer\_id order by order\_date desc) rnk from (select s.customer\_id,s.order\_date,m.product\_name from sales s inner join menu m on s.product\_id=m.product\_id inner join members mem on s.customer\_id=mem.customer\_id where s.order\_date<mem.join\_date)a)b where rnk=1;

	_			
	t	n	П	t:
u			<b>U</b>	C.

	customer_id	order_date	product_name
•	A	2021-01-01	sushi
	A	2021-01-01	curry
	В	2021-01-04	sushi

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

select s.customer\_id,sum(m.price) 'amount spent',count(\*)'total items purchased' from sales s inner join menu m on s.product\_id=m.product\_id inner join members mem on s.customer\_id=mem.customer\_id where s.order\_date<mem.join\_date group by customer\_id;

$\bigcirc$	U	+			+	•
U	u	L	$oldsymbol{oldsymbol{\sqcup}}$	u	L	_
	•	•		•	•	•

	customer_id	amount spent	total items purchased
•	В	40	3
	A	25	2



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

create temporary table temp1
select customer\_id,product\_name,price,
case when product\_name='sushi' then price\*20
else price\*10
end as 'points' from sales s inner join menu m on s.product\_id=m.product\_id

select customer\_id,sum(points) 'points' from temp1 group by customer\_id order 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?

select customer\_id,sum(points) from (select customer\_id,point+point1 as points from (select \*, case when product\_name='sushi' then price\*20 else price\*10 end as 'point1' from (select \*, case when day(order\_date)<=day(join\_date)+6 then price\*20 else price\*10 end as 'point' from

(select s.customer\_id,order\_date,product\_name,join\_date,price from sales s inner join menu m on s.product\_id=m.product\_id inner join members mem on s.customer\_id=mem.customer\_id where s.order\_date>=mem.join\_date and s.order\_date<='2021-01-31')a)b)c)d group by customer\_id

		customer_id	sum(points)
output:	•	В	640
		A	1530

#### Insights/Recommendations:

- -> Danny can know his earnings from the restaurant and also how frequently the customers are visiting, so that he can give discounts to the loyal customers.
- -> First purchased item by the customers, this gives a simple indication that the customers are getting attracted to that particular product and because of that they are coming to restaurant.
- -> Though people are getting atracted and they are coming to the restaurant but they are not interested in buying it regularly, they are finding 'ramen' is much more better. And it is best to prepare an item similar to 'ramen' but with new addings, so that there is a chance of attracting customers.
- -> By knowing the specific product which is being liked by each of the customer. Danny can definitely run offers for that particular product so that sales get increased.
- -> May be the customers find that the items are good and want to experience completely the restaurant's items by becoming a member. By improving the quality and service of that particular items they can make customers to become the members.
- After spending such amount they are becoming a member so this is the very good indication that restaurant is going on in a good way.
- Having the information of points acquired by each of the customer, they can run offers for the customers who are having a specific number of points, in this way customers tries to acquire that number of points, so that sales get increased.