**Questions**

**Case Study Questions**

**Each of the following case study questions can be answered using a single SQL statement:**

1. **What is the total amount each customer spent at the restaurant?**
2. **How many days has each customer visited the restaurant?**
3. **What was the first item from the menu purchased by each customer?**
4. **What is the most purchased item on the menu and how many times was it purchased by all customers?**
5. **Which item was the most popular for each customer?**
6. **Which item was purchased first by the customer after they became a member?**
7. **Which item was purchased just before the customer became a member?**
8. **What are the total items and amount spent for each member before they became a member?**
9. **If each $1 spent equates to 10 points and sushi has a 2x points multiplier - how many points would each customer have?**
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?**

**Schema and tables**

CREATE SCHEMA dannys\_diner;

SET search\_path = dannys\_diner;

CREATE TABLE sales (

"customer\_id" VARCHAR(1),

"order\_date" DATE,

"product\_id" INTEGER

);

INSERT INTO sales

("customer\_id", "order\_date", "product\_id")

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

);

INSERT INTO members

("customer\_id", "join\_date")

VALUES

**('A', '2021-01-07'),**

**('B', '2021-01-09');**

**Solution**

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

select s.customer\_id,sum(m.price) as Amount from sales s

left join menu m on

m.product\_id = s.product\_id

group by s.customer\_id;

***/\* How many days has each customer visited the restaurant? \*/***

select customer\_id,count(distinct order\_date) as visit\_days from sales

group by customer\_id;

***/\* What was the first item from the menu purchased by each customer?\*/***

SELECT s.customer\_id, m.product\_name, min(s.order\_date) order\_date

FROM sales s

JOIN menu m ON s.product\_id = m.product\_id

GROUP BY customer\_id;

-- WITH ranked\_orders AS (

-- SELECT s.customer\_id, m.product\_name, s.order\_date,

-- ROW\_NUMBER() OVER (PARTITION BY s.customer\_id ORDER BY s.order\_date) AS rn

-- FROM sales s

-- JOIN menu m ON s.product\_id = m.product\_id

-- )

-- SELECT customer\_id, product\_name

-- FROM ranked\_orders

-- WHERE rn = 1;

***/\* What is the most purchased item on the menu and how many times was it purchased by each customers? \*/***

with rankorder AS (

select s.product\_id,count(s.product\_id) as purchase\_count,m.product\_name

from sales s

join menu m on s.product\_id = m.product\_id

group by s.product\_id

order by purchase\_count desc

limit 1

)

select s.customer\_id,count(r.product\_id) as products\_purchased,r.product\_name from rankorder r

join sales s on r.product\_id = s.product\_id

group by s.customer\_id;

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

select s.product\_id,count(s.product\_id) as purchase\_count,m.product\_name

from sales s

join menu m on s.product\_id = m.product\_id

group by s.product\_id

order by purchase\_count desc

limit 1;

***/\* Which item was the most popular for each customer? \*/***

WITH most\_popular AS (

SELECT

sales.customer\_id,

menu.product\_name,

COUNT(menu.product\_id) AS order\_count,

DENSE\_RANK() OVER (

PARTITION BY sales.customer\_id

ORDER BY COUNT(sales.customer\_id) DESC) AS rank1

FROM dannys\_diner.menu

INNER JOIN dannys\_diner.sales

ON menu.product\_id = sales.product\_id

GROUP BY sales.customer\_id, menu.product\_name

)SELECT

customer\_id,

product\_name,

order\_count

FROM most\_popular

WHERE rank1 = 1;

***/\*Which item was purchased first by the customer after they became a member? \*/***

WITH first\_purchase AS (

SELECT s.customer\_id, m.product\_name, s.order\_date,

ROW\_NUMBER() OVER (PARTITION BY s.customer\_id ORDER BY s.order\_date) AS rn

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 first\_purchase

WHERE rn = 1;

***/\*Which item was purchased just before the customer became a member? \*/***

with before\_member AS (

select s.customer\_id, s.order\_date,m.product\_id,ma.join\_date, m.product\_name,

row\_number() OVER (PARTITION BY s.customer\_id order by s.order\_date desc) ok

from sales s

join members ma on s.customer\_id = ma.customer\_id

join menu m on s.product\_id = m.product\_id

where ma.join\_date > s.order\_date

order by s.customer\_id,s.order\_date

)select \* from before\_member where ok = 1;

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

with before\_member AS(

select s.customer\_id, s.order\_date,m.product\_id,ma.join\_date, m.product\_name, m.price

from sales s

join members ma on s.customer\_id = ma.customer\_id

join menu m on s.product\_id = m.product\_id

where ma.join\_date > s.order\_date

order by s.customer\_id,s.order\_date

)select \*,sum(price),count(product\_id) from before\_member

group by customer\_id

order by customer\_id;

/\* Below is better solution \*/

SELECT

sales.customer\_id,

COUNT(sales.product\_id) AS total\_items,

SUM(menu.price) AS total\_sales

FROM dannys\_diner.sales

INNER JOIN dannys\_diner.members

ON sales.customer\_id = members.customer\_id

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

INNER JOIN dannys\_diner.menu

ON sales.product\_id = menu.product\_id

GROUP BY sales.customer\_id

ORDER BY sales.customer\_id;

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

select s.customer\_id, sum(if(m.product\_name = 'sushi',m.price \* 2 \* 10,m.price \* 10)) as total\_points from sales s

join menu m on m.product\_id = s.product\_id

group by customer\_id

order by customer\_id;

***/\*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 dates\_cte AS (

SELECT

customer\_id,

join\_date,

join\_date + 6 AS valid\_date,

'2021-01-31' AS last\_date

FROM dannys\_diner.members

)SELECT

sales.customer\_id,

SUM(CASE

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

WHEN sales.order\_date BETWEEN dates.join\_date AND dates.valid\_date THEN 2 \* 10 \* menu.price

ELSE 10 \* menu.price END) AS points

FROM dannys\_diner.sales

INNER JOIN dates\_cte AS dates

ON sales.customer\_id = dates.customer\_id

AND dates.join\_date <= sales.order\_date

AND sales.order\_date <= last\_date

INNER JOIN dannys\_diner.menu

ON sales.product\_id = menu.product\_id

GROUP BY sales.customer\_id;