

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?

```
select customer_id, sum(price) as total_spent
from sales s
join menu m
on s.product_id = m.product_id
group by customer_id
order by customer_id
```

customer_id	total_spent
A	76
B	74
C	36

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

```
select customer_id, count(order_date) as visit_count
from sales
group by customer_id
```

customer_id	visit_count
A	4
B	6
C	2

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

```
select customer_id, product_name
from
(select s.customer_id, m.product_name, min(order_date) min_order,
dense_rank()over(partition by customer_id order by min(order_date)) as rnk
from sales s
join menu m
on m.product_id = s.product_id
group by customer_id, product_name) x
where rnk =1;
```

customer_id	product_name
A	sushi
A	curry
B	curry
C	ramen

4. 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 order_count
from menu m
join sales s
```

```

on m.product_id = s.product_id
group by m.product_name
order by order_count desc
limit 1;

```

product_name	order_count
ramen	8

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

select customer_id, product_name, cnt
from
(select customer_id, product_name, count(s.product_id) as cnt,
dense_rank()over(partition by customer_id order by count(s.product_id) desc)
as rnk
from sales s
join menu m
on s.product_id = m.product_id
group by customer_id, product_name) x
where rnk =1;

```

|  | customer_id | product_name | cnt |
|--|-------------|--------------|-----|
|  | A           | ramen        | 3   |
|  | B           | curry        | 2   |
|  | B           | sushi        | 2   |
|  | B           | ramen        | 2   |
|  | C           | ramen        | 3   |

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

```

with cte_1 as (select m.customer_id, product_name, order_date,
dense_rank()over(partition by customer_id order by order_date) as rnk
from members m
join sales s
on s.customer_id = m.customer_id
join menu p
on p.product_id = s.product_id

where join_date <= order_date)

select customer_id, product_name, order_date
from cte_1
where rnk =1

```

(OR)

```

with cte_1 as (select s.customer_id, product_id, join_date, order_date
from members m
join sales s
on s.customer_id = m.customer_id
),
cte_2 as (select customer_id, product_name, order_date,
dense_rank()over(partition by customer_id order by order_date) as rnk
from cte_1 c1
join menu m
on c1.product_id = m.product_id
where join_date <= order_date)

```

```

select customer_id, product_name, order_date
from cte_2
where rnk =1;

```

| customer_id | product_name | order_date |
|-------------|--------------|------------|
| A           | curry        | 2021-01-07 |
| B           | sushi        | 2021-01-11 |

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

```

with cte_1 as (select customer_id, join_date
from members),
cte_2 as
(select customer_id, product_name, min(order_date) as ordered_on
from sales s
join menu m
on s.product_id = m.product_id
group by customer_id, product_name),

```

```

cte_3 as (select c2.customer_id, product_name, ordered_on,
dense_rank()over(partition by customer_id order by ordered_on desc) as rnk
from cte_1 c1
right join cte_2 c2
on c1.customer_id = c2.customer_id
where ordered_on < join_date)

```

```

select customer_id, product_name, ordered_on
from cte_3
where rnk =1

```

| customer_id | product_name | ordered_on |
|-------------|--------------|------------|
| A           | sushi        | 2021-01-01 |
| A           | curry        | 2021-01-01 |
| B           | sushi        | 2021-01-04 |

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

```

with cte_1 as (select *
 from members
),
cte_2 as (select customer_id, order_date, s.product_id , price
from sales s
join menu m
on s.product_id = m.product_id)

select c2.customer_id, count(product_id) as item_count, sum(price) as
total_spent
from cte_1 c1
join cte_2 c2
on c1.customer_id = c2.customer_id
where order_date < join_date
group by c2.customer_id
order by customer_id

```

| customer_id | item_count | total_spent |
|-------------|------------|-------------|
| A           | 2          | 25          |
| 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?

```

with cte_1 as (select product_id, product_name, (price*10) new_price
from menu
where product_name <> 'sushi'),

```

```

cte_2 as (select product_id, product_name, (price*20) new_price
from menu
where product_name = 'sushi'),
cte_3 as (select product_id, product_name, new_price
from cte_1
union all
select product_id, product_name, new_price
from cte_2)

```

```

select customer_id, sum(new_price) as points
from sales s
join cte_3 c3
on s.product_id = c3.product_id
group by customer_id

```

| customer_id | points |
|-------------|--------|
| A           | 860    |
| B           | 940    |
| 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?

```
with cte_1 as (select customer_id, join_date,
date_add(join_date, interval 6 day) as first_week, '2021-01-31' as last_date
from members),

cte_2 as (select s.customer_id, order_date, product_name, price
from sales s
join menu p
on p.product_id = s.product_id)
select c2.customer_id, sum(case when product_name = "Sushi" then
price*20
 when order_date between join_date and first_week then
price*20
 else price*10
 end) as points
from cte_1 c1
join cte_2 c2
on c1.customer_id = c2.customer_id
where order_date <last_date
group by customer_id
order by customer_id
```

| customer_id | points |
|-------------|--------|
| A           | 1370   |
| B           | 820    |