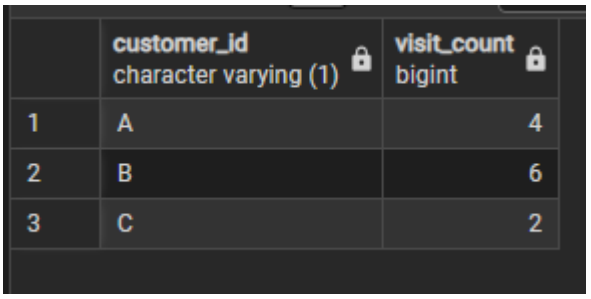
Q1. What is the total amount each customer spent at the restaurant?

```
SELECT customer_id, SUM(price) AS total_sales
FROM sales
JOIN menu using (product_id)
GROUP BY customer_id
ORDER BY customer_id ASC;
```

	customer_id character varying (1)	total_sales bigint
1	A	76
2	B	74
3	C	36

Q2. How many days has each customer visited the restaurant?

```
SELECT customer_id, COUNT(DISTINCT order_date) AS visit_count
FROM sales
GROUP BY customer_id;
```

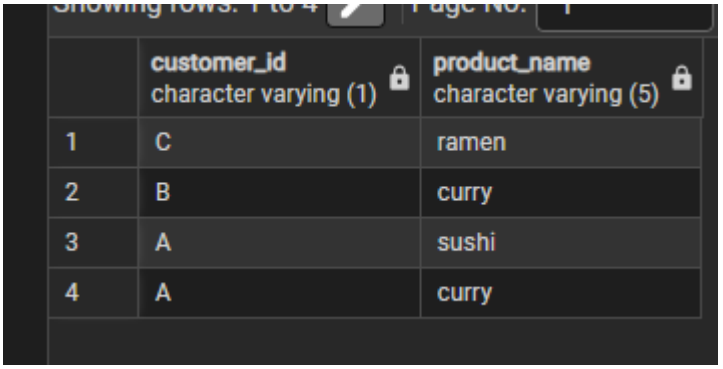


	customer_id character varying (1)	visit_count bigint
1	A	4
2	B	6
3	C	2

Q3. What was the first item from the menu purchased by each customer?

```
SELECT customer_id, product_name
FROM sales
JOIN menu USING(product_id)
WHERE (customer_id, order_date) IN
```

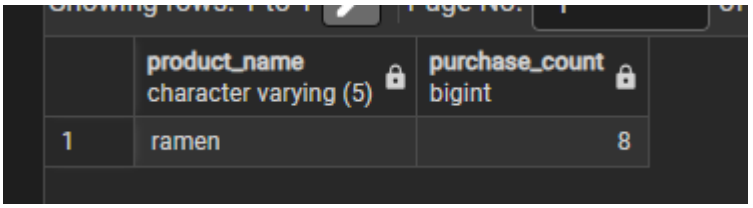
```
(
SELECT customer_id, MIN(order_date)
FROM sales
GROUP BY customer_id
);
```



	customer_id character varying (1)	product_name character varying (5)
1	C	ramen
2	B	curry
3	A	sushi
4	A	curry

Q4. What is the most purchased item on the menu and how many times was it purchased?

```
SELECT product_name, COUNT(*) AS purchase_count
FROM sales
JOIN menu USING(product_id)
GROUP BY product_name
ORDER BY purchase_count DESC
LIMIT 1;
```



	product_name character varying (5)	purchase_count bigint
1	ramen	8

Q5. Which item was the most popular for each customer?

```
WITH most_popular AS (
  SELECT distinct customer_id, product_name,
```

```
    COUNT(product_id) AS order_count,
    DENSE_RANK() OVER (
      PARTITION BY customer_id
      ORDER BY COUNT(customer_id) DESC) AS rank
  FROM menu
  JOIN sales using (product_id)
  GROUP BY customer_id,product_name
)

SELECT customer_id, product_name, order_count
FROM most_popular
WHERE rank = 1;
```

	customer_id character varying (1)	product_name character varying (5)	order_count bigint
1	A	ramen	3
2	B	curry	2
3	B	ramen	2
4	B	sushi	2
5	C	ramen	3

Q6. Which item was purchased first by the customer after they became a member?

```
WITH joined_as_member AS (
  SELECT
      s.customer_id,
      s.product_id,
      ROW_NUMBER() OVER (
          PARTITION BY s.customer_id
          ORDER BY s.order_date
      ) AS row_num
  FROM members m
  JOIN sales s
      ON s.customer_id = m.customer_id
     AND s.order_date > m.join_date
```

```
)

SELECT
    j.customer_id,
    mn.product_name
FROM joined_as_member j
JOIN menu mn
    ON j.product_id = mn.product_id
WHERE j.row_num = 1
ORDER BY j.customer_id ASC;
```

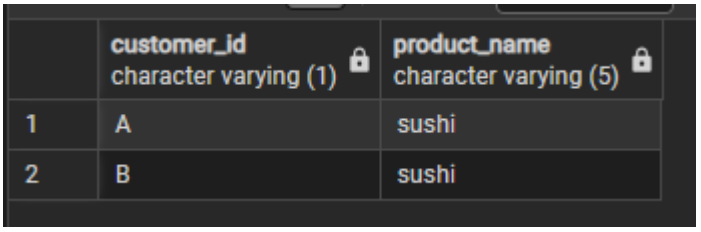
	customer_id character varying (1)	product_name character varying (5)
1	A	ramen
2	B	sushi

Q7. Which item was purchased just before the customer became a member?

```
WITH p_member AS (
  SELECT
      s.customer_id,
      s.product_id,
      ROW_NUMBER() OVER (
          PARTITION BY s.customer_id
          ORDER BY s.order_date DESC
      ) AS rank
  FROM members m
  JOIN sales s
      ON s.customer_id = m.customer_id
     AND s.order_date < m.join_date
)

SELECT
    p.customer_id,
    mn.product_name
FROM p_member p
JOIN menu mn
```

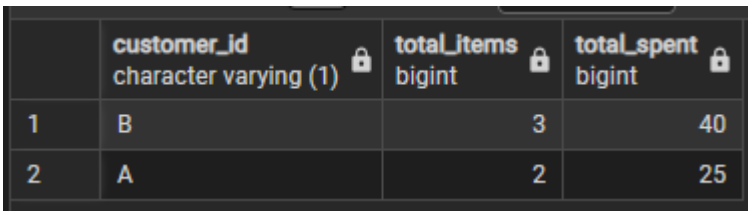
```
    ON p.product_id = mn.product_id
WHERE p.rank = 1
ORDER BY p.customer_id ASC;
```



	customer_id character varying (1)	product_name character varying (5)
1	A	sushi
2	B	sushi

Q8. What is the total items and amount spent for each member before they became a member?

```
SELECT customer_id,
COUNT(*) AS total_items,
SUM(price) AS total_spent
FROM sales
JOIN members USING(customer_id)
JOIN menu USING(product_id)
WHERE order_date < join_date
GROUP BY customer_id;
```



	customer_id character varying (1)	total_items bigint	total_spent bigint
1	B	3	40
2	A	2	25

Q9. If each $1 spent = 10 points and sushi has 2x points, how many points does each customer have?

```
SELECT customer_id,
SUM(
CASE
```

```
WHEN product_name = 'sushi' THEN price * 20
ELSE price * 10
END
) AS points
FROM sales
JOIN menu USING(product_id)
GROUP BY customer_id;
```

	customer_id character varying (1)	points bigint
1	B	940
2	C	360
3	A	860

Q10. In the first week after joining, customers earn 2x points on all items. How many points do customers A and B have at the end of January?

```
SELECT s.customer_id,
    SUM(
        CASE
            WHEN s.order_date BETWEEN m.join_date
            AND m.join_date + INTERVAL '6 days'
            THEN me.price * 20

            WHEN me.product_name = 'sushi'
            THEN me.price * 20

            ELSE me.price * 10
        END
    ) AS points
FROM sales s
JOIN members m
ON s.customer_id = m.customer_id
JOIN menu me
ON s.product_id = me.product_id
WHERE s.order_date <= DATE '2021-01-31'
```

```
GROUP BY s.customer_id
ORDER BY s.customer_id;
```

	customer_id character varying (1)	points bigint
1	A	1370
2	B	820