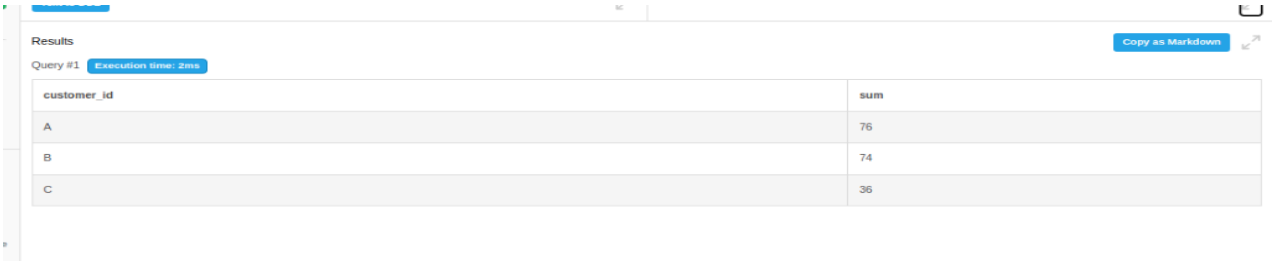
Case study - 1

Wednesday, January 26, 2022 10:40 AM

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

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



Results

Copy as Markdown

Query #1 Execution time: 2ms

customer_id	sum
A	76
B	74
C	36

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

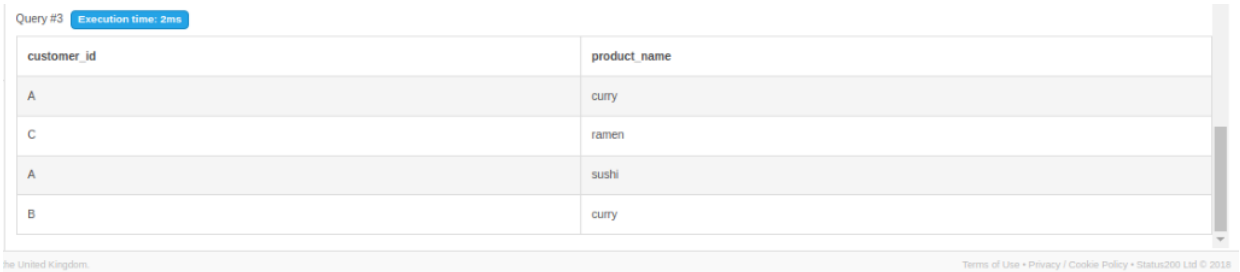
```
select customer_id,
        count(DISTINCT(order_date)) from dannys_diner.sales
        group by customer_id;
```

Query #2 Execution time: 1ms

customer_id	count
A	4
B	6
C	2

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

```
SELECT DISTINCT(customer_id),
     product_name FROM dannys_diner.sales s
JOIN dannys_diner.menu m
ON m.product_id = s.product_id
WHERE s.order_date = ANY
    (
     SELECT MIN(order_date)
     FROM dannys_diner.sales
     GROUP BY customer_id
    )
```

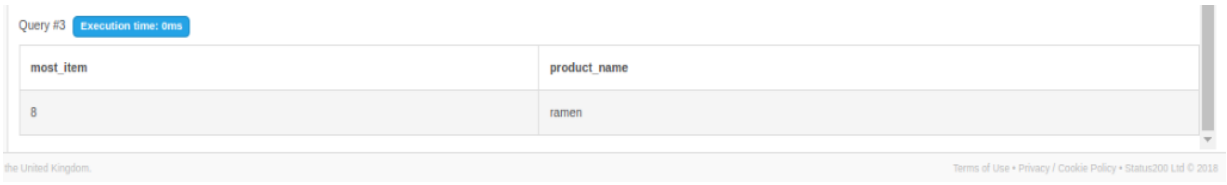


Query #3 Execution time: 2ms

customer_id	product_name
A	curry
C	ramen
A	sushi
B	curry

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

```
select count(m.product_name) as most_item,m.product_name from dannys_diner.sales s join
dannys_diner.menu m on s.product_id=m.product_id group by m.product_name order by most_item
DESC LIMIT 1;
```

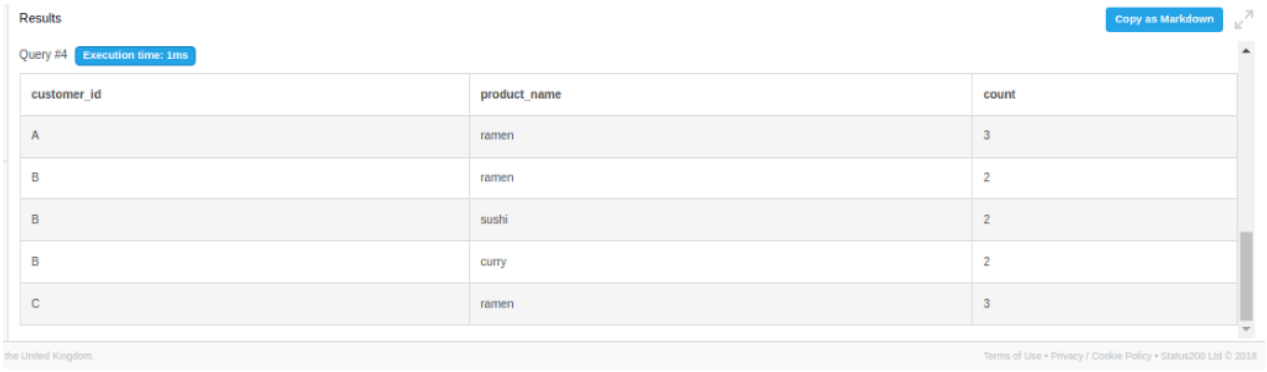


Query #3 Execution time: 0ms

most_item	product_name
8	ramen

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

```
WITH r AS
(
SELECT s.customer_id,m.product_name,
    COUNT(s.product_id) as count,
    DENSE_RANK() OVER (PARTITION BY s.customer_id ORDER BY COUNT(s.product_id) DESC) AS r
FROM dannys_diner.menu m
JOIN dannys_diner.sales s
ON s.product_id = m.product_id
GROUP BY s.customer_id, s.product_id, m.product_name
)
SELECT customer_id, product_name, count
FROM r
WHERE r = 1
```



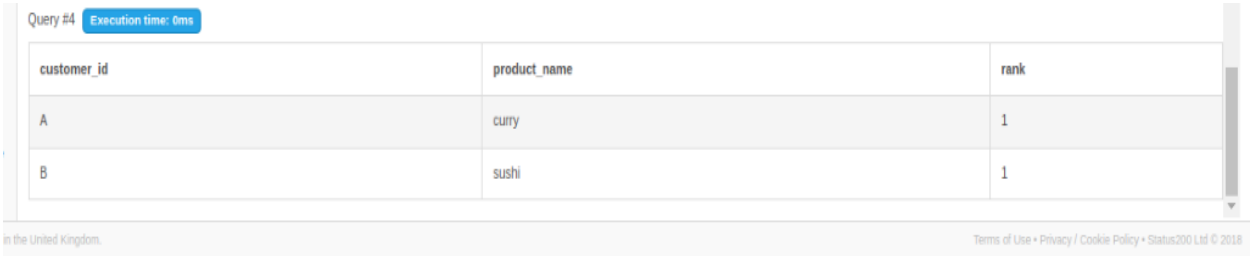
Results — Copy as Markdown

Query #4 Execution time: 1ms

customer_id	product_name	count
A	ramen	3
B	ramen	2
B	sushi	2
B	curry	2
C	ramen	3

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

```
with ranks AS
    (select s.customer_id,
        m.product_name,
        Dense_rank() over (partition by s.customer_id order by s.order_date) AS rank
            from dannys_diner.sales s join dannys_diner.menu m
            on s.product_id=m.product_id join dannys_diner.members ms
            on s.customer_id=ms.customer_id
        where s.order_date>=ms.join_date)
select * from ranks where rank=1;
```

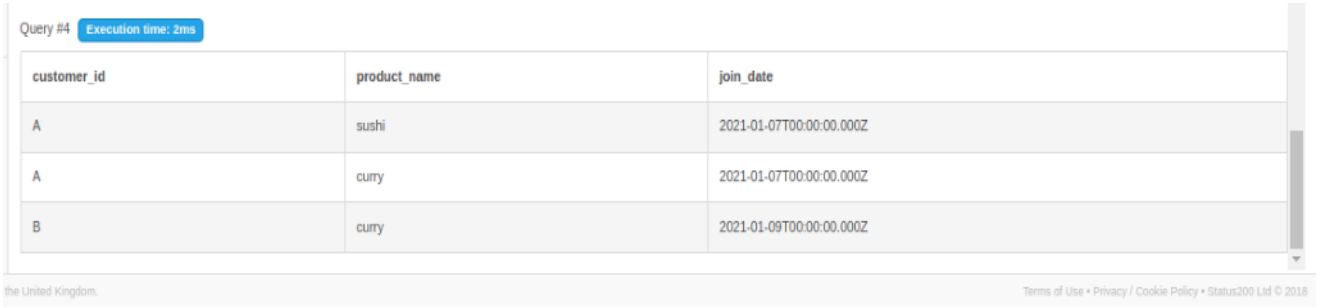


Query #4 Execution time: 0ms

customer_id	product_name	rank
A	curry	1
B	sushi	1

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

```
with ranks AS
        (select s.customer_id,
                m.product_name,
                ms.join_date,
                Dense_rank() over (partition by s.customer_id order by s.order_date) AS rank from
                dannys_diner.sales s join dannys_diner.menu m on s.product_id=m.product_id join
                dannys_diner.members ms on s.customer_id=ms.customer_id where
                s.order_date<ms.join_date)
select customer_id,product_name,join_date from ranks where rank=1;
```



Query #4 Execution time: 2ms

customer_id	product_name	join_date
A	sushi	2021-01-07T00:00:00.000Z
A	curry	2021-01-07T00:00:00.000Z
B	curry	2021-01-09T00:00:00.000Z

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

```
select
        s.customer_id,
        count(m.product_id),
        sum(m.price) from dannys_diner.sales s join dannys_diner.menu m
                on s.product_id=m.product_id join dannys_diner.members ms
                on s.customer_id=ms.customer_id where s.order_date < ms.join_date
        group by s.customer_id;
```

Query #4 Execution time: 1ms

customer_id	count	sum
B	3	40
A	2	25

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

```
With points as (
 select * ,
```

```
case
when m.product_name = 'sushi' then price*20
when m.product_name != 'sushi' then price*10
End as points from dannys_diner.menu m)
select customer_id,sum(points) from dannys_diner.sales s join points p on s.product_id=p.product_id
group by s.customer_id order by s.customer_id;
```

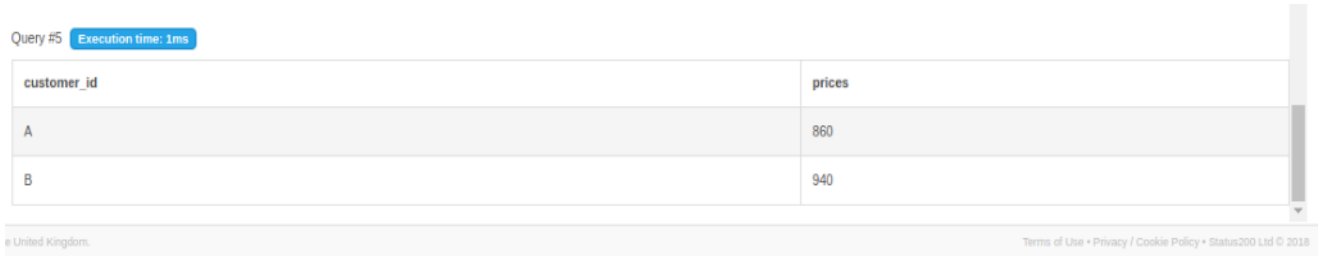
Query #5 Execution time: 0ms

customer_id	sum
A	860
B	940
C	360

DB Fiddle – Crafted with ♥ by Status200 in the United Kingdom. Terms of Use • Privacy / Cookie Policy • Status200 Ltd © 2018

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 date_cte AS(SELECT *,JOIN_DATE + INTERVAL '7 day' AS first_date,date_trunc('month', '2021-01-
31'::date) + interval '1 month' - interval '1 day'
AS end_of_month from dannys_diner.members ms)
select s.customer_id,
sum(CASE
    when m.product_name='sushi' then 2*10*m.price
    when s.order_date between d.first_date and d.end_of_month then 2*10*m.price
    else 10*m.price
    END) AS Prices
    from date_cte as d
    join dannys_diner.sales s on
    s.customer_id=d.customer_id
    join dannys_diner.menu m on
    s.product_id=m.product_id
    join dannys_diner.members ms on
    s.customer_id=ms.customer_id
    where s.order_date<d.end_of_month
    GROUP BY d.customer_id;
```



Query #5 Execution time: 1ms

customer_id	prices
A	860
B	940

e United Kingdom. Terms of Use • Privacy / Cookie Policy • Status200 Ltd © 2018