THE TASTE OF SUCCESS

SRINIBAS RANA

INTRODUCTION

Danny seriously loves Japanese food so in the beginning of 2021, he decides to embark upon a risky venture and opens a cute little restaurant that sells his 3 favorite foods: sushi, curry and ramen.

Danny's Diner needs your assistance to help the restaurant stay afloat - the restaurant has captured some very basic data from their few months of operation but have no idea how to use their data to help them run the business.

SRINIBAS RANA

PROBLEM 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 which menu items are their favorite. Having this deeper connection with his customers will help him deliver a better and more personalized 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!

8 WEEKSQLCHALLENGE #1

SRINIBAS RANA

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

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

OUTPUT:

customer_id	total_amount
B	74
C	36
A	76

in SRINIBAS RANA

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

```
select customer_id, count(1) as no_of_customers
from sales
group by customer_id;
```

OUTPUT:

customer_id	no_of_customers
B	6
C	3
A	6

in SRINIBAS RANA

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

```
WITH cte AS (
  SELECT
    s.customer_id,
    m.product_name AS first_purchased_item,
    s.order_date,
    ROW_NUMBER() OVER (PARTITION BY s.customer_id ORDER BY
s.order_date) AS rnk
  FROM
    sales s
  JOIN menu m ON s.product_id = m.product_id
)

SELECT
  customer_id,
  first_purchased_item,
  order_date
FROM
cte
 where rnk = 1;
```

OUTPUT:

customer_id	first_purchased_item	order_date
A	curry	01-01-2021
B	curry	01-01-2021
C	ramen	01-01-2021

SRINIBAS RANA

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

```
select m.product_name as most_purchased_item, count(1)
as no_of_time
from sales s
join menu m
on s.product_id = m.product_id
group by 1
order by 2 desc
limit 1
```

OUTPUT:

most_purchased_item	no_of_time
ramen	8

in SRINIBAS RANA

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

```
with cte as (
select s.customer_id, m.product_name as
most_popular_item, count(1) as no_of_items
            from sales s
            join menu m
            on s.product_id = m.product_id
            group by 1,2
            ),
rrnk as(
select customer_id, most_popular_item, no_of_items,
dense_rank() over(partition by customer_id order by
no_of_items desc) as rnk
            from cte
)
select customer_id, most_popular_item
from rrnk
where rnk = 1;
```

OUTPUT:

customer_id	most_popular_item
A	ramen
B	sushi
B	curry
B	ramen
C	ramen

SRINIBAS RANA

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

```
WITH MemberFirstPurchase AS (
  SELECT
    s.customer_id,
    m.product_name AS first_purchase_after_membership,
    s.order_date,
    ROW_NUMBER() OVER (PARTITION BY s.customer_id ORDER BY
s.order_date) AS row_num
  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
  mfp.customer_id,
  mfp.first_purchase_after_membership,
  mfp.order_date
FROM
  MemberFirstPurchase mfp
WHERE
  mfp.row_num = 1;
```

OUTPUT:

customer_id	first_purchase_after_membership	order_date
A	curry	07-01-2021
B	sushi	11-01-2021

in SRINIBAS RANA

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

```
WITH MemberFirstPurchase AS (
  SELECT
    s.customer_id,
    m.product_name AS first_purchase_after_membership,
    s.order_date,
    ROW_NUMBER() OVER (PARTITION BY s.customer_id ORDER BY
s.order_date) AS row_num
  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
  mfp.customer_id,
  mfp.first_purchase_after_membership,
  mfp.order_date
FROM
  MemberFirstPurchase mfp
WHERE
  mfp.row_num = 1;
```

OUTPUT:

customer_id	first_purchase_after_membership	order_date
A	curry	07-01-2021
B	sushi	11-01-2021

SRINIBAS RANA

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

```
WITH MemberPurchases AS (
  SELECT
    s.customer_id,
    m.product_name,
    m.price,
    s.order_date
  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
  mp.customer_id,
  COUNT(mp.product_name) AS total_items,
  SUM(mp.price) AS total_amount_spent
FROM
  MemberPurchases mp
GROUP BY
  mp.customer_id;
```

OUTPUT:

customer_id	total_items	total_amount_spent
B	3	40
A	2	25

SRINIBAS RANA

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 as(
        select customer_id, product_name, price,
        case when product_name = 'sushi' then price*2*10
        else price*10
        end as Points
        from sales s
        join menu m
        on s.product_id = m.product_id
        )
        select customer_id,
        sum(points) as total_points
        from cte
        group by customer_id;
```

OUTPUT:

customer_id	total_points
B	940
C	360
A	860

SRINIBAS RANA

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 PointsCalculation AS (
  SELECT
    s.customer_id,
    m.product_name,
    m.price,
    s.order_date,
    mem.join_date,
    CASE
      WHEN s.order_date <= mem.join_date + INTERVAL '7 days' THEN m.price * 2 * 10
      ELSE
        CASE
          WHEN m.product_name = 'sushi' THEN m.price * 2 * 10
          ELSE m.price * 10
        END
    END AS points
  FROM
    sales s
  JOIN menu m ON s.product_id = m.product_id
  JOIN members mem ON s.customer_id = mem.customer_id
)

SELECT
  pc.customer_id,
  SUM(pc.points) AS total_points_at_end_of_january
FROM
  PointsCalculation pc
WHERE
  pc.order_date <= '2021-01-31'
GROUP BY
  pc.customer_id;
```

OUTPUT:

customer_id	total_points_at_end_of_january
A	1520
B	1240

SRINIBAS RANA

THANK YOU

THE TASTE OF SUCCESS