

SQL Case Study Challenge-Danny's Dinner

Query

Query History

1

2

3

4

5

6

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

SELECT a.customer_id,SUM(b.price) as total_spent

FROM sales a

JOIN menu b

ON a.product_id=b.product_id

GROUP BY a.customer_id |

Data Output

Messages

Notifications

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

Query

Query History

1

2

3

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

SELECT customer_id,COUNT(DISTINCT(order_date)) FROM sales

GROUP BY customer_id

Data Output

Messages

Notifications

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

Query Query History

```

1  /* What is the most purchased item on the menu and how many times was it purchased by all customers? */
2  SELECT b.product_name,COUNT(*) FROM sales a
3  join menu b
4  ON a.product_id=b.product_id
5  GROUP BY b.product_name

```

Data Output Messages Notifications

	product_name character varying (5)	count bigint
1	ramen	8
2	sushi	3
3	curry	4

Query Query History

```

1  /* What was the first item from the menu purchased by each customer? */
2  WITH final AS(
3  SELECT a.*,b.product_name,
4  RANK() over (partition by customer_id order by order_date) as ranking
5  FROM sales a
6  join menu b
7  on a.product_id=b.product_id)
8  SELECT * FROM final where ranking=1

```

Data Output Messages Notifications

	customer_id character varying (1)	order_date date	product_id integer	product_name character varying (5)	ranking bigint
1	A	2021-01-01	2	curry	1
2	A	2021-01-01	1	sushi	1
3	B	2021-01-01	2	curry	1
4	C	2021-01-01	3	ramen	1
5	C	2021-01-01	3	ramen	1

Query

Query History

```
1  /* Which item was the most popular for each customer? */
2  WITH final AS(
3  SELECT a.customer_id,b.product_name,count(*) as total
4  FROM sales a
5  JOIN menu b
6  ON a.product_id=b.product_id
7  GROUP BY a.customer_id,b.product_name)
8  SELECT customer_id,product_name,total,
9  rank() over (partition by customer_id order by total desc ) as ranking
10 FROM final
```

Data Output

Messages

Notifications

	customer_id character varying (1)	product_name character varying (5)	total bigint	ranking bigint
1	A	ramen	3	1
2	A	curry	2	2
3	A	sushi	1	3
4	B	sushi	2	1
5	B	curry	2	1
6	B	ramen	2	1
7	C	ramen	3	1

Query










Query History





```
1  /* Which item was purchased first by the customer after they became a member? */
2  WITH final AS (
3  SELECT a.customer_id,a.product_id,c.product_name,
4  RANK() OVER (partition by a.customer_id ORDER BY order_date ) as ranking
5  FROM sales AS a
6  JOIN members AS B
7  ON a.customer_id=b.customer_id
8  JOIN menu c
9  ON a.product_id=c.product_id
10 WHERE order_date>=join_date
11 ORDER BY order_date asc)
12 SELECT * FROM final WHERE ranking=1
```

Data Output

Messages

Notifications



	customer_id character varying (1) 	product_id integer 	product_name character varying (5) 	ranking bigint 
1	A	2	curry	1
2	B	1	sushi	1

Query Query History

```

1  /* Which item was purchased just before the customer became a member? */
2  WITH final AS (
3  SELECT a.customer_id,a.product_id,c.product_name,
4  RANK() OVER (partition by a.customer_id ORDER BY order_date ) as ranking
5  FROM sales AS a
6  JOIN members AS b
7  ON a.customer_id=b.customer_id
8  JOIN menu c
9  ON a.product_id=c.product_id
10 WHERE join_date>order_date
11 ORDER BY order_date asc)
12 SELECT * FROM final WHERE ranking=1

```

Data Output Messages Notifications

	customer_id character varying (1)	product_id integer	product_name character varying (5)	ranking bigint
1	A	1	sushi	1
2	A	2	curry	1
3	B	2	curry	1

Query Query History

```

1  /* What is the total items and amount spent for each member before they became a member? */
2  WITH final AS(
3  SELECT a.customer_id,a.product_id,c.product_name,c.price
4  FROM sales AS a
5  JOIN members AS b
6  ON a.customer_id=b.customer_id
7  JOIN menu as c
8  ON c.product_id=a.product_id
9  WHERE join_date>order_date
10 ORDER BY customer_id asc)
11 SELECT customer_id,COUNT(*) AS total_items,SUM(price) AS total_amount from FINAL
12 GROUP BY customer_id
13 |

```

Data Output Messages Notifications

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

Query
Query History

```

1  /* If each $1 spent equates to 10 points and sushi has a 2x points multiplier
2     | how many points would each customer have? */
3  WITH final AS (
4    SELECT a.customer_id,c.product_name,c.price,
5    CASE WHEN product_name='sushi' then 2*c.price
6    else c.price end as new_price
7    FROM sales a
8    JOIN menu c
9    ON c.product_id=a.product_id
10   ORDER BY customer_id asc)
11   SELECT customer_id,sum(new_price)*10 as final_price FROM final
12   GROUP BY customer_id
13

```

Data Output
Messages
Notifications

+

📄

▼

📋

▼

🗑️

🔄

⬇️

📈

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

Query
Query History

```

1  /* In the first week after a customer joins the program
2     (including their join date) they earn 2x points on all
3     items, not just sushi - how many points do customer A
4     and B have at the end of January? */
5  with finalpoints as (
6    select a.customer_id,a.order_date,c.product_name,c.price,
7    case when product_name='sushi' then 2*c.price
8    when a.order_date between b.join_date
9    and (b.join_date+ interval 6 day) then 2*c.price
10   else c.price end as newprice
11   from cricket.sales1 a
12   join cricket.menu1 c
13   on a.product_id=c.product_id
14   join cricket.members1 b
15   on a.customer_id=b.customer_id
16   where a.order_date<='2021-01-31'
17  )
18  select customer_id,sum(newprice)*10 from finalpoints
19  group by customer_id
20

```

Query Query History

```

1  /* JOIN ALL THE TABLES */
2
3  select a.customer_id,a.order_date,c.product_name,c.price,
4  CASE WHEN order_date<join_date THEN 'Y'
5  ELSE 'N' END AS member
6  FROM sales a
7  JOIN menu c
8  ON a.product_id=c.product_id
9  JOIN members b
10 ON a.customer_id=b.customer_id
11

```

Data Output Messages Notifications

	customer_id character varying (1)	order_date date	product_name character varying (5)	price integer	member text
1	A	2021-01-07	curry	15	N
2	A	2021-01-11	ramen	12	N
3	A	2021-01-11	ramen	12	N
4	A	2021-01-10	ramen	12	N
5	A	2021-01-01	sushi	10	Y
6	A	2021-01-01	curry	15	Y
7	B	2021-01-04	sushi	10	Y
8	B	2021-01-11	sushi	10	N
9	B	2021-01-01	curry	15	Y
10	B	2021-01-02	curry	15	Y
11	B	2021-01-16	ramen	12	N
12	B	2021-02-01	ramen	12	N

Total rows: 12 of 12 Query complete 00:00:00.091

Query Query History

```

1  --RANKING ALL THINGS
2  with ranking as (
3  select a.customer_id,a.order_date,c.product_name,c.price,
4  case when a.order_date<b.join_date then 'N'
5  when b.join_date is null then 'N'
6  else 'Y' end as member
7  from sales a
8  join menu c
9  on a.product_id=c.product_id
10 left join members b
11 on a.customer_id=b.customer_id
12
13 select *,
14 case when member='N' then null
15 else
16 rank() over (partition by customer_id,member order by order_date)
17 end as rankcalc
18 from ranking
19
20
21
22
23

```

Data Output Messages Notifications

	customer_id character varying (1)	order_date date	product_name character varying (5)	price integer	member text	rankcalc
1	A	2021-01-01	sushi	10	N	
2	A	2021-01-01	curry	15	N	
3	A	2021-01-07	curry	15	Y	
4	A	2021-01-10	ramen	12	Y	
5	A	2021-01-11	ramen	12	Y	
6	A	2021-01-11	ramen	12	Y	
7	B	2021-01-01	curry	15	N	
8	B	2021-01-02	curry	15	N	
9	B	2021-01-04	sushi	10	N	
10	B	2021-01-11	sushi	10	Y	
11	B	2021-01-16	ramen	12	Y	
12	B	2021-02-01	ramen	12	Y	
13	C	2021-01-01	ramen	12	N	
14	C	2021-01-01	ramen	12	N	
15	C	2021-01-07	ramen	12	N	
