8WEEKSQLCHALLENGE.COM

CASE STUDY #1

THE TASTE OF SUCCESS

DATAWITHDANNY.COM

https://8weeksqlchallenge.com/case-study-1/

Introduction

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

Danny's Diner is in need of 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.

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 also which menu items are their favourite. Having this deeper connection with his customers will help him deliver a better and more personalised 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!

Danny has shared with you 3 key datasets for this case study:

- sales
- menu
- members

Entity-Relationship Diagram

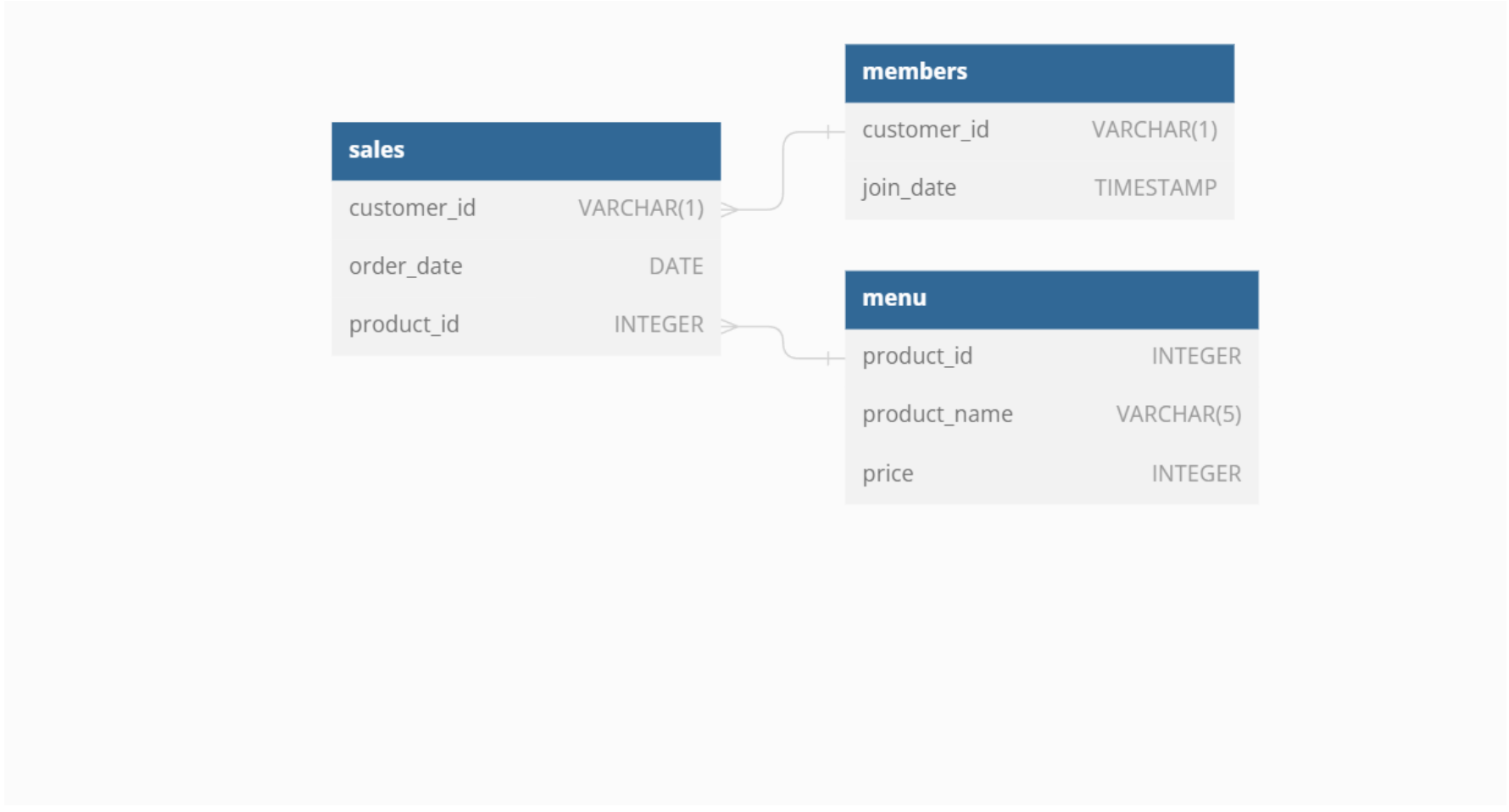


Table 1: Sales

The sales table captures all customer_id level purchases with an corresponding order_date and product_id information for when and what menu items were ordered.

customer_id	order_date	product_id
A	2021-01-01	1
A	2021-01-01	2
A	2021-01-07	2
A	2021-01-10	3
A	2021-01-11	3
A	2021-01-11	3
B	2021-01-01	2
B	2021-01-02	2
B	2021-01-04	1
B	2021-01-11	1
B	2021-01-16	3
B	2021-02-01	3
C	2021-01-01	3
C	2021-01-01	3
C	2021-01-07	3

Table 2: Menu

The menu table maps the product_id to the actual product_name and price of each menu item

product_id	product_name	price
1	sushi	10
2	curry	15
3	ramen	12

Table 3: Members

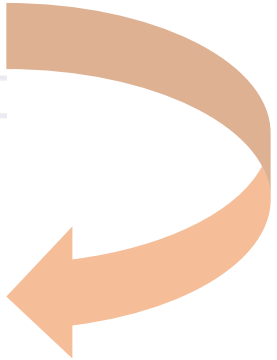
The final members table captures the join_date when a customer_id joined the beta version of the Danny's Diner loyalty program

customer_id	join_date
A	2021-01-07
B	2021-01-09

Case Study Questions

Ques1. What is the total amount each customer spent at the restaurant?

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



	customer_id	total_amount_spent_by_each_customer
1	A	76
2	B	74
3	C	36

Ques2. How many days has each customer visited the restaurant?

```sql
select customer_id,count(distinct order_date) as days
from sales s
join menu m
on s.product_id=m.product_id
group by customer_id
```

	customer_id	days
1	A	4
2	B	6
3	C	2

Ques3. What was the first item from the menu purchased by each customer?

```sql
with cte as (
select customer_id,product_name,
ROW_NUMBER() over (partition by customer_id order by order_date) as rn
from sales s
join menu m
on s.product_id=m.product_id
)
select customer_id,product_name
from cte where rn=1
```

	customer_id	product_name
1	A	sushi
2	B	curry
3	C	ramen

Ques4. What is the most purchased item on the menu and how many times was it purchased by all customers?

```sql
select top 1 count(*) as most_purchased,product_name
from sales s
join menu m
on s.product_id=m.product_id
group by product_name
order by most_purchased desc
```

	most_purchased	product_name
1	8	ramen

Ques5. Which item was the most popular for each customer?

```sql
with cte as (
select count(*) as most_purchased,customer_id,product_name as most_popular_item,
rank() over (partition by customer_id order by count(*) desc) rankk
from sales s
join menu m
on s.product_id=m.product_id
group by customer_id,product_name)
select customer_id,most_popular_item
from cte where rankk=1
```

	customer_id	most_popular_item	rankk
1	A	ramen	1
2	B	sushi	1
3	B	curry	1
4	B	ramen	1
5	C	ramen	1

Ques6. Which item was purchased first by the customer after they became a member?

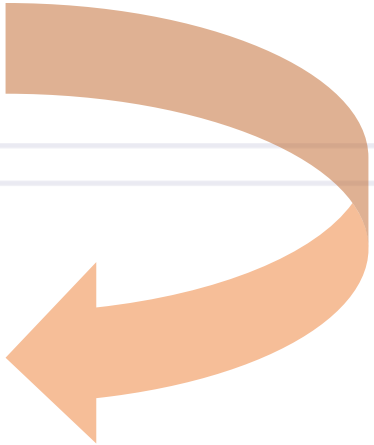
```sql
with cte as (
select s.customer_id as customer,order_date,product_name,
rank() over (partition by s.customer_id order by order_date) rankk
from sales s
join menu m
on s.product_id=m.product_id
join members on members.customer_id=s.customer_id
where order_date >= (select min(join_date) from members)
)
select customer,product_name from cte
where rankk=1
```

customer	product_name
A	curry
B	sushi

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

```sql
with cte as
(select s.customer_id as customer_id, product_id, order_date,
    rank() over(partition by s.customer_id order by order_date desc) as rn
from sales s
join members m on s.customer_id=m.customer_id
where order_date<join_date)

select cte.customer_id,product_name, order_date
from cte
join menu m on cte.product_id=m.product_id
where rn=1
order by cte.customer_id;
```



customer_id	product_name	order_date
A	sushi	2021-01-01
A	curry	2021-01-01
B	sushi	2021-01-04

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

```sql
select s.customer_id as customer_id,
sum(price) as total_amount
from sales s
join members on s.customer_id=members.customer_id
join menu on menu.product_id=s.product_id
where order_date<(select min(join_date) from members)
group by s.customer_id
```

customer_id	total_amount
A	25
B	40

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

```sql
select s.customer_id,
sum(case when product_name='sushi' then price*20 else price*10 end ) points
from sales s left
join members on s.customer_id=members.customer_id
join menu on menu.product_id=s.product_id
group by s.customer_id
```

customer_id	points
A	860
B	940
C	360

Ques10. 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?

```sql
select s.customer_id,sum(price)*20 as points
from sales s
join members on s.customer_id=members.customer_id
join menu on menu.product_id=s.product_id
where order_date between join_date and dateadd(day,7,join_date)
group by s.customer_id
```

customer_id	points
A	1020
B	440