



SQL PROJECT



# DANNY'S DINER

presentation by  
Sumanth Thiruvedula







# 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.







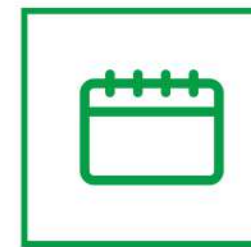
# TABLES



## SALES



## MENU

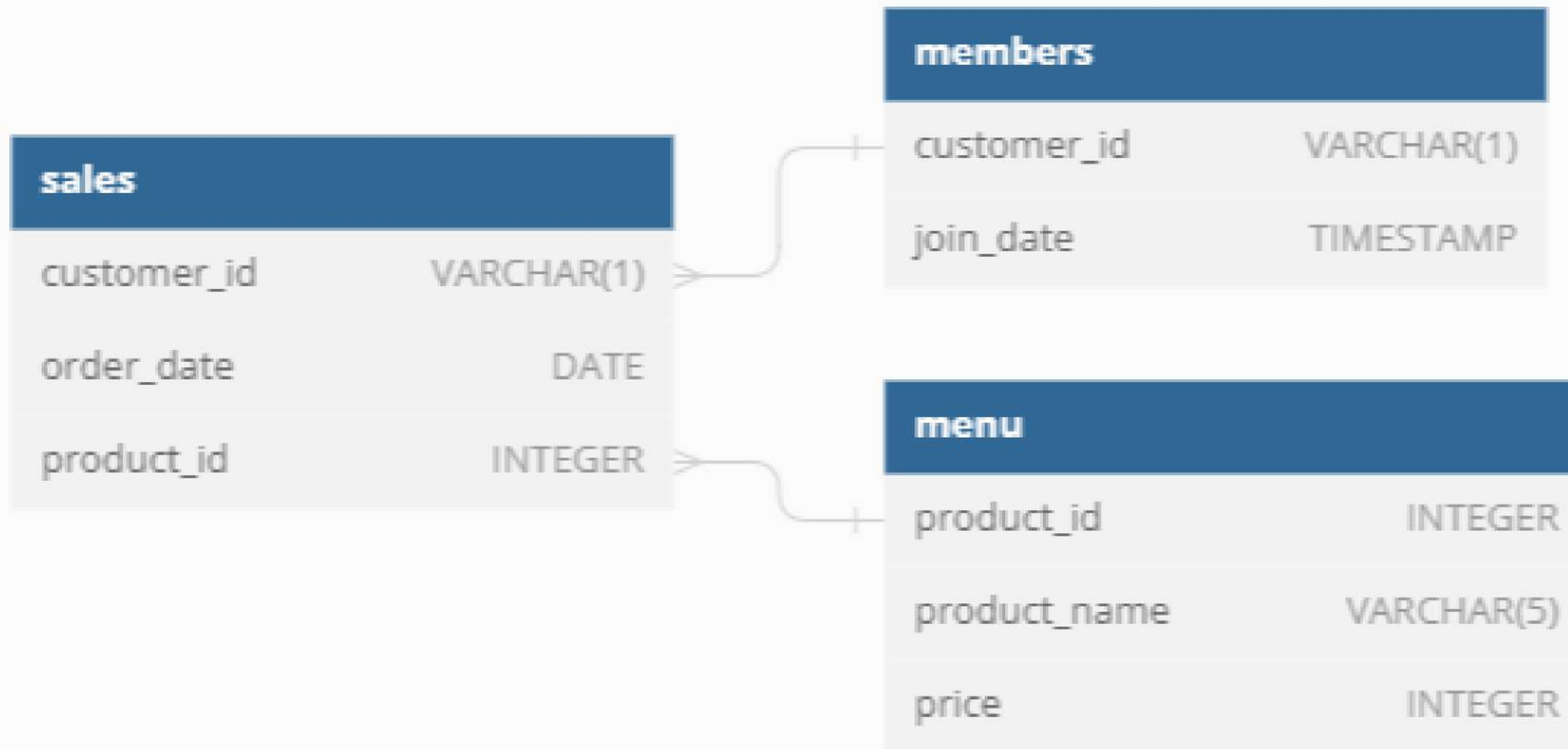


## MEMBERS





# SCHEMA





# CASE STUDY QUESTIONS

## 1. WHAT IS THE TOTAL AMOUNT EACH CUSTOMER SPENT AT THE RESTAURANT?

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

	customer_id	Total_Amount
1	A	76
2	B	74
3	C	36





## 2. HOW MANY DAYS HAS EACH CUSTOMER VISITED THE RESTAURANT?

```
select customer_id, count(Distinct(order_date)) as Visited_days  
from sales  
Group by customer_id  
Order by Visited_days desc
```

	customer_id	Visited_days
1	B	6
2	A	4
3	C	2







### 3. WHAT WAS THE FIRST ITEM FROM THE MENU PURCHASED BY EACH CUSTOMER?

```
with first_item_purchase as
(
    select s.customer_id,s.order_date,m.product_id,m.product_name,
    DENSE_RANK() over(partition by s.customer_id order by s.order_date) as Item_Ranks
    from sales s
    join menu m
    on s.product_id = m.product_id
)
select customer_id,product_name
from first_item_purchase
where Item_Ranks = 1
Group by customer_id,product_name
order by customer_id,product_name
```

	customer_id	product_name
1	A	curry
2	A	sushi
3	B	curry
4	C	ramen





#### 4. WHAT IS THE MOST PURCHASED ITEM ON THE MENU AND HOW MANY TIMES WAS IT PURCHASED BY ALL CUSTOMERS?

```
select TOP 1 m.product_name, COUNT(s.order_date) as orders  
from sales s  
join menu m  
on s.product_id = m.product_id  
Group by m.product_name  
order by 2 desc
```

	product_name	orders
1	ramen	8







## 5. WHICH ITEM WAS THE MOST POPULAR FOR EACH CUSTOMER?

```
with most_popular_item as
(
    select s.customer_id, COUNT(s.order_date) as order_count, m.product_name,
    DENSE_RANK() over(partition by s.customer_id order by count(s.customer_id) desc) as odr_rnk
    from sales s
    join menu m
    on s.product_id = m.product_id
    Group by s.customer_id, m.product_name
)
select customer_id, product_name, order_count
from most_popular_item
where odr_rnk = 1
```

	customer_id	product_name	order_count
1	A	ramen	3
2	B	sushi	2
3	B	curry	2
4	B	ramen	2
5	C	ramen	3





## 6. WHICH ITEM WAS PURCHASED FIRST BY THE CUSTOMER AFTER THEY BECAME A MEMBER?

```
with joined_member as
(
    select s.customer_id,m.product_name,
    DENSE_RANK() over(partition by s.customer_id order by s.order_date) as rnk_id
    from menu m
    join sales s
    on m.product_id = s.product_id
    join members ms
    on ms.customer_id = s.customer_id
    where s.order_date > ms.join_date
)
select customer_id,product_name
from joined_member
where rnk_id = 1
```

	customer_id	product_name
1	A	ramen
2	B	sushi







## 7. WHICH ITEM WAS PURCHASED JUST BEFORE THE CUSTOMER BECAME A MEMBER?

```
with joined_member as
(
    select s.customer_id,m.product_name,
    DENSE_RANK() over(partition by s.customer_id order by s.order_date) as rnk_id
    from menu m
    join sales s
    on m.product_id = s.product_id
    join members ms
    on ms.customer_id = s.customer_id
    where s.order_date < ms.join_date
)
select customer_id,product_name
from joined_member
where rnk_id = 1
```

	customer_id	product_name
1	A	sushi
2	A	cumy
3	B	cumy





## 8. WHAT IS THE TOTAL ITEMS AND AMOUNT SPENT FOR EACH MEMBER BEFORE THEY BECAME A MEMBER?

```
select s.customer_id, count(s.product_id) as total_items, sum(m.price) as total_amount
from sales s
join members me
on s.customer_id = me.customer_id
join menu m
on m.product_id = s.product_id
where s.order_date < me.join_date
Group by s.customer_id
```

	customer_id	total_items	total_amount
1	A	2	25
2	B	3	40







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 product_name = 'sushi' THEN price * 20
        ELSE price * 10
        END as points
    from menu
)
select s.customer_id,sum(p.points) as Total_points
from sales s
join points p
on p.product_id = s.product_id
Group by s.customer_id
```

	customer_id	Total_points
1	A	860
2	B	940
3	C	360





JOIN ALL THE THINGS

THE FOLLOWING QUESTIONS ARE RELATED CREATING BASIC DATA TABLES THAT DANNY AND HIS TEAM CAN USE TO QUICKLY DERIVE INSIGHTS WITHOUT NEEDING TO JOIN THE UNDERLYING TABLES USING SQL

```
with members_joined as
(
    select s.customer_id,s.order_date,m.product_name,m.price,me.join_date,
    case when order_date > join_date then 'Y'
    Else 'N'
    End as Members
    from sales s
    join members me
    on me.customer_id = s.customer_id
    join menu m
    on m.product_id = s.product_id
)
select customer_id,order_date,
product_name,price,Members
from members_joined
```

	customer_id	order_date	product_name	price	Members
1	A	2021-01-01	sushi	10	N
2	A	2021-01-01	curry	15	N
3	A	2021-01-07	curry	15	N
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







## RANK ALL THE THINGS

DANNY ALSO REQUIRES FURTHER INFORMATION ABOUT THE RANKING OF CUSTOMER PRODUCTS, BUT HE PURPOSELY DOES NOT NEED THE RANKING FOR NON-MEMBER PURCHASES SO HE EXPECTS NULL RANKING VALUES FOR THE RECORDS WHEN CUSTOMERS ARE NOT YET PART OF THE LOYALTY PROGRAM.

```
with members_joined as
(
    select s.customer_id,s.order_date,m.product_name,m.price,me.join_date,
    case when order_date > join_date then 'Y'
    Else 'N'
    End as Members
    from sales s
    join members me
    on me.customer_id = s.customer_id
    join menu m
    on m.product_id = s.product_id
)
select customer_id,order_date,product_name,price,Members,
CASE when Members = 'N' Then Null
ELSE RANK() over(partition by customer_id,Members order by order_date)
END as Ranking
from members_joined
```

	customer_id	order_date	product_name	price	Members	Ranking
1	A	2021-01-01	sushi	10	N	NULL
2	A	2021-01-01	curry	15	N	NULL
3	A	2021-01-07	curry	15	N	NULL
4	A	2021-01-10	ramen	12	Y	1
5	A	2021-01-11	ramen	12	Y	2
6	A	2021-01-11	ramen	12	Y	2
7	B	2021-01-01	curry	15	N	NULL
8	B	2021-01-02	curry	15	N	NULL
9	B	2021-01-04	sushi	10	N	NULL
10	B	2021-01-11	sushi	10	Y	1
11	B	2021-01-16	ramen	12	Y	2







Thanks You

Sumanth Thiruvudula

