8weekSQLCHALLENGE

Case study #1: Danny's Diner

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

Contents

Introduction

This is first case study out of **'8WeekSQLChallenge'** by Danny Ma and all details for this challenge can be found at https://8weeksqlchallenge.com/case-study-1/ .

Danny is the Chief Data Mentor @ Data With Danny and the Founder & CEO of Sydney Data Science, a boutique data consultancy based out of Sydney, Australia AU. He has tremendous knowledge and experience in SQL and helps SQL lovers to brush up their SQL skills.

This case study is about a new restaurant 'Danny's Diner' where Danny is interested to find out useful insights about Customer visiting patterns, how much money they've spent ,which menu items are their favourite and whether he should expand the existing customer loyalty program or not. Danny has given 10 questions for this 'Case Study to be solved using SQL with 2 bonus questions. Danny has shared 3 datasets for this case study : sales, menu, members. All datasets exist within dannys_diner database schema.

For this case study, SQL Server Management studio is used. All the queries done to solve the questions are result of my SQL knowledge. Reason for documenting this is to save it for my own future use and for all SQL learners out there.

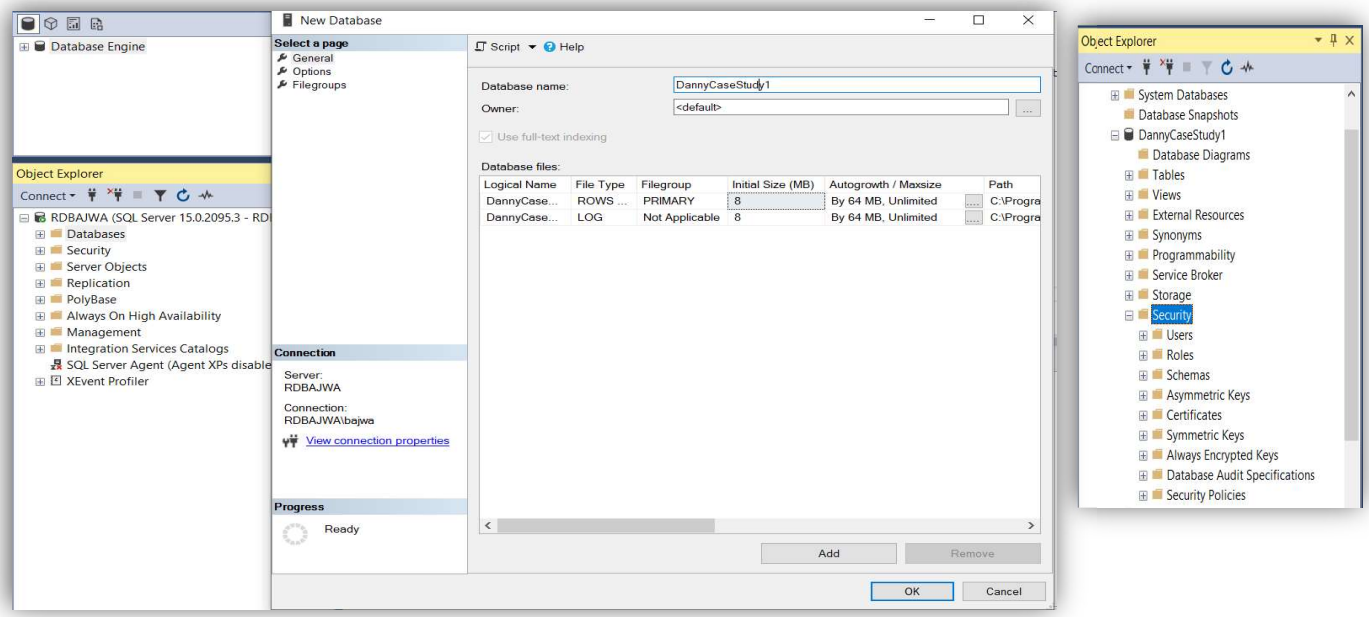
Problem Statement

Danny wants to use the data to answer few questions :

- About his customer visiting pattern
- Favourite menu items
- Expansion of existing customer loyalty program
- Wants to join all three datasets into one and about the ranking of all customers so that his staff don't need to use SQL to fetch important insights

Creating database, tables and schema

For this case study, database name **'DannyCaseStudy1'** and schema '**dannys_diner'** was created. By creating these, we are creating a total separate container for our SQL queries.
We can create these by using menu options after right clicking on database as shown in following image:



Schema can be created by using following path : <database> -> security ->schemas -> New schema. Once database and schema were created, next all three tables were created as per the SQL queries provided by Danny Ma.

```
CREATE TABLE dannys_diner.sale (
  "customer_id" VARCHAR(1),
  "order_date" DATE,
```

```
  "product_id" INTEGER
);



INSERT INTO dannys_diner.sale
  ("customer_id", "order_date", "product_id")
VALUES
  ('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');



 CREATE TABLE dannys_diner.menu (
  "product_id" INTEGER,
  "product_name" VARCHAR(5),
  "price" INTEGER
);

INSERT INTO dannys_diner.menu
  ("product_id", "product_name", "price")
VALUES
  ('1', 'sushi', '10'),
  ('2', 'curry', '15'),
  ('3', 'ramen', '12');


CREATE TABLE dannys_diner.members (
  "customer_id" VARCHAR(1),
  "join_date" DATE
);
```

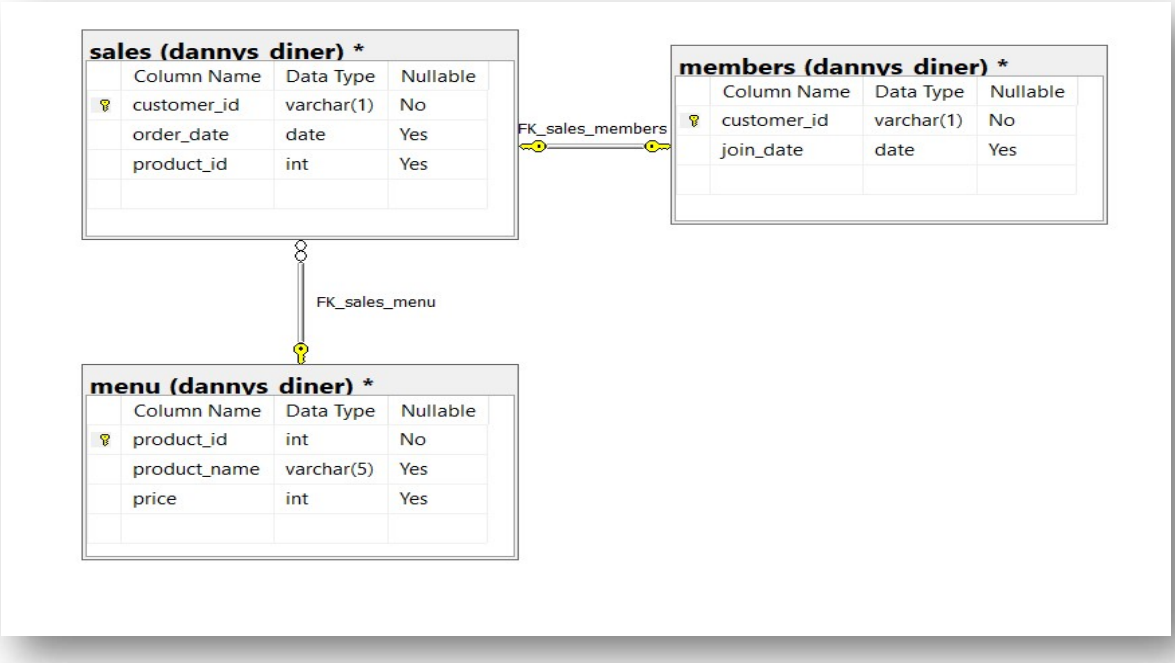
```
INSERT INTO dannys_diner.members
  ("customer_id", "join_date")
VALUES
  ('A', '2021-01-07'),
  ('B', '2021-01-09');
```

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

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

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

Entity Relationship Diagram



Case Study Questions

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

```
select customer_id, sum(price) as total_money_spent
 from
 (
select s.*, m.product_name, m.price
 from dannys_diner.sales as s
 join dannys_diner.menu as m
 on s.product_id=m.product_id) as t
 group by customer_id
```

	customer_id	total_money_spent
1	A	76
2	B	74
3	C	36

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

```
select customer_id, count(order_date) as total_num_of_days
 from dannys_diner.sales
 group by customer_id
```

	customer_id	total_num_of_days
1	A	6
2	B	6
3	C	3

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

```
select r.*,m.product_name from
 (
 select row_number()over(partition by customer_id order by order_date) as first_Day, customer_id,
order_date, product_id
 from DannyCaseStudy1.dannys_diner.sales as s) r
 inner join DannyCaseStudy1.dannys_diner.menu m
 on r.product_id=m.product_id
  where first_Day=1
```

	first_Day	customer_id	order_date	product_id	product_name
1	1	A	2021-01-01	1	sushi
2	1	B	2021-01-01	2	curry
3	1	C	2021-01-01	3	ramen

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

```
select  Top 1 count(s.product_id) as  most_purchased, m.product_name
 from DannyCaseStudy1.dannys_diner.sales as s
 join DannyCaseStudy1.dannys_diner.menu as m
 on s.product_id=m.product_id
 group by s.product_id, product_name
 order by count(s.product_id) desc
```

	most_purchased	product_name
1	8	ramen

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

```
select *
 from
 (
 select r.*, m.product_name ,
 DENSE_RANK() over(partition by r.customer_id order by total_purchased desc) as rank1
from
 (
 select s.customer_id, count(s.product_id) as total_purchased, product_id
 from DannyCaseStudy1.dannys_diner.sales s
 group by customer_id, product_id) r
 join DannyCaseStudy1.dannys_diner.menu m
 on r.product_id=m.product_id) t
where rank1=1
```

	customer_id	total_purchased	product_id	product_name	rank1
1	A	3	3	ramen	1
2	B	2	1	sushi	1
3	B	2	2	curry	1
4	B	2	3	ramen	1
5	C	3	3	ramen	1

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

```
select r.* , m2.product_name from
 (select row_number()over(partition by s.customer_id order by order_date) as row_num
,s.customer_id, s.product_id,s.order_date, m.join_datefrom DannyCaseStudy1.dannys_diner.sales s
 join DannyCaseStudy1.dannys_diner.members m
```

```
on s.customer_id=m.customer_id
where s.order_date>=m.join_date) r
join DannyCaseStudy1.dannys_diner.menu m2
on r.product_id=m2.product_id
where row_num=1
```

	row_num	customer_id	product_id	order_date	join_date	product_name
1	1	A	2	2021-01-07	2021-01-07	curry
2	1	B	1	2021-01-11	2021-01-09	sushi

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

```
select r.* , m2.product_name from
 (select rank()over(partition by s.customer_id order by order_date desc) as rank_num ,s.customer_id,
s.product_id,s.order_date, m.join_date
 from DannyCaseStudy1.dannys_diner.sales s
 join DannyCaseStudy1.dannys_diner.members m
 on s.customer_id=m.customer_id
 where s.order_date < m.join_date) r
 join DannyCaseStudy1.dannys_diner.menu m2
 on r.product_id=m2.product_id
 where rank_num=1
```

	rank_num	customer_id	product_id	order_date	join_date	product_name
1	1	A	1	2021-01-01	2021-01-07	sushi
2	1	A	2	2021-01-01	2021-01-07	curry
3	1	B	1	2021-01-04	2021-01-09	sushi

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

```
select s.customer_id, count(s.order_date) as total_items, sum(m.price) as total_amount
 from DannyCaseStudy1.dannys_diner.sales s
 join DannyCaseStudy1.dannys_diner.menu m
 on s.product_id=m.product_id
 join DannyCaseStudy1.dannys_diner.members m2
 on s.customer_id=m2.customer_id
where s.order_date < m2.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?

```
select s.customer_id,
sum
(
case when s.product_id=1 then price*20
else price*10
end
) as total_points
from DannyCaseStudy1.dannys_diner.sales s
join DannyCaseStudy1.dannys_diner.menu m
on s.product_id=m.product_id
group by s.customer_id
```

	customer_id	total_points
1	A	860
2	B	940
3	C	360

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?

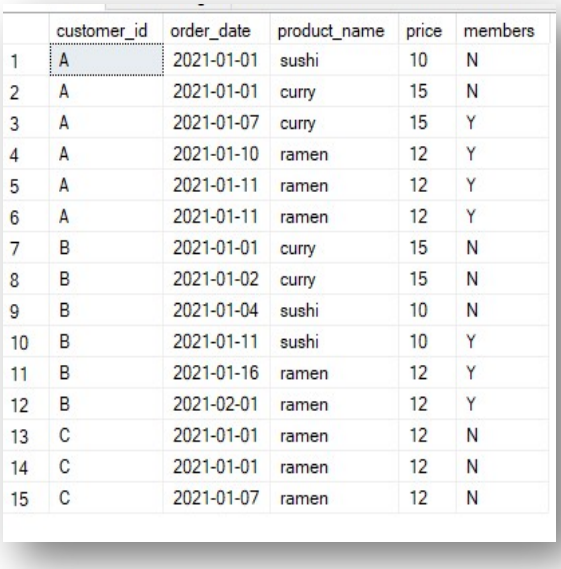
```
select s.customer_id,
sum
(
case when s.order_date >= m2.join_date and  s.order_date <=dateadd(day,6,m2.join_date) then
m.price*20
else
( case when s.product_id=1 then m.price*20
else m.price*10
end
)end
) as total_points
from DannyCaseStudy1.dannys_diner.sales s
join DannyCaseStudy1.dannys_diner.menu m
on s.product_id=m.product_id
join DannyCaseStudy1.dannys_diner.members m2
on s.customer_id=m2.customer_id
where s.order_date between '2021-01-01' and '2021-01-31'
group by s.customer_id
```

	customer_id	total_points
1	A	1370
2	B	820

Bonus questions

Joining all tables together so that Danny team can derive quick insights

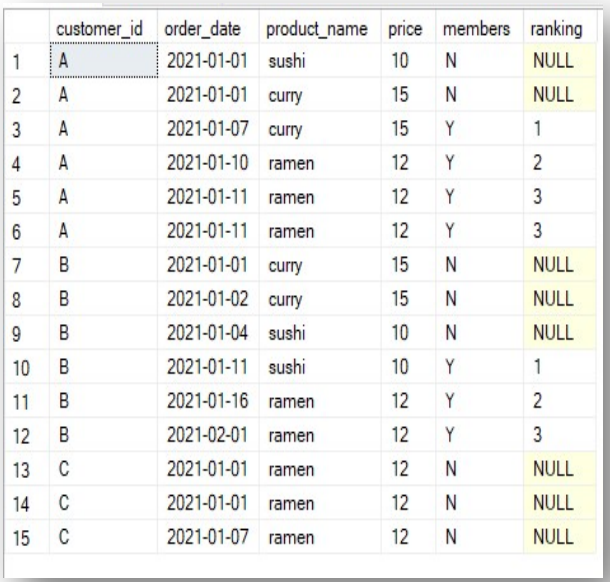
```
select s.customer_id, s.order_date, m.product_name,
 m.price,
(case when s.order_date>=m2.join_date then 'Y'
else 'N'
end)as members
from DannyCaseStudy1.dannys_diner.sales s
join DannyCaseStudy1.dannys_diner.menu m
on s.product_id=m.product_id
left join DannyCaseStudy1.dannys_diner.members m2
        on s.customer_id=m2.customer_id
```



	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	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

Rank all things in Table

```
with ranking_table as
(
select s.customer_id, s.order_date, m.product_name,  m.price,
(case when s.order_date>=m2.join_date then 'Y'
else 'N'
end)as members
from DannyCaseStudy1.dannys_diner.sales s
join DannyCaseStudy1.dannys_diner.menu m
on s.product_id=m.product_id
left join DannyCaseStudy1.dannys_diner.members m2
on s.customer_id=m2.customer_id
)
Select *, Case
when members='N' then null
else
 dense_rank()over(partition by customer_id,members
order by order_date) end as ranking
 from ranking_table
```



	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	Y	1
4	A	2021-01-10	ramen	12	Y	2
5	A	2021-01-11	ramen	12	Y	3
6	A	2021-01-11	ramen	12	Y	3
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
12	B	2021-02-01	ramen	12	Y	3
13	C	2021-01-01	ramen	12	N	NULL
14	C	2021-01-01	ramen	12	N	NULL
15	C	2021-01-07	ramen	12	N	NULL

Interesting Insights

- Customer A spends most followed b y customer B
- Customer A visited 'Danny's Diner' 6 times which is same as customer B but customer C visited only 3 times
- 'Ramen' is the most popular item out of three ands was purchased 8 times which is 53% of whole.
- Ramen is favourite item for customer A and C whereas B likes all three items equally as per the data.
- Customer A is the first 'Loyal Customer' followed by B
- Even though Ramen is popular but before joining 'Customer loyalty' program A ordered 'sushi' and 'curry' and B ordered 'sushi'.
- Customer C has purchased the lowest out of all three customer and also, he is not a member of 'loyalty program'. Danny team can request all customer to fill up survey to get feedback specially from customer C.

Conclusion

This case study was very interesting and helped me to gain more confidence in using SQL queries to solve problems. I used various SQL functions to solved the questions : Aggregate functions(COUNT, SUM), Windows functions (ROW_NUMBER, RANK,DENSE_RANK), filtering and sorting functions (WHERE, ORDER BY, GROUP BY), date functions (DATEADD) and used CTE with complex sub queries. SQL Server Management Studio was used for solving this case study.

I am more confident in using windows function and writing complex queries. After finishing this case study, I am excited for Next case study by Danny Ma.