Restaurant Database Analysis with SQL

1. Project Background

Danny's Diner is in need of your assistance to use their data to help them run the business, a little restaurant which has just open and launched its 3 favourite foods : sushi, curry and ramen.

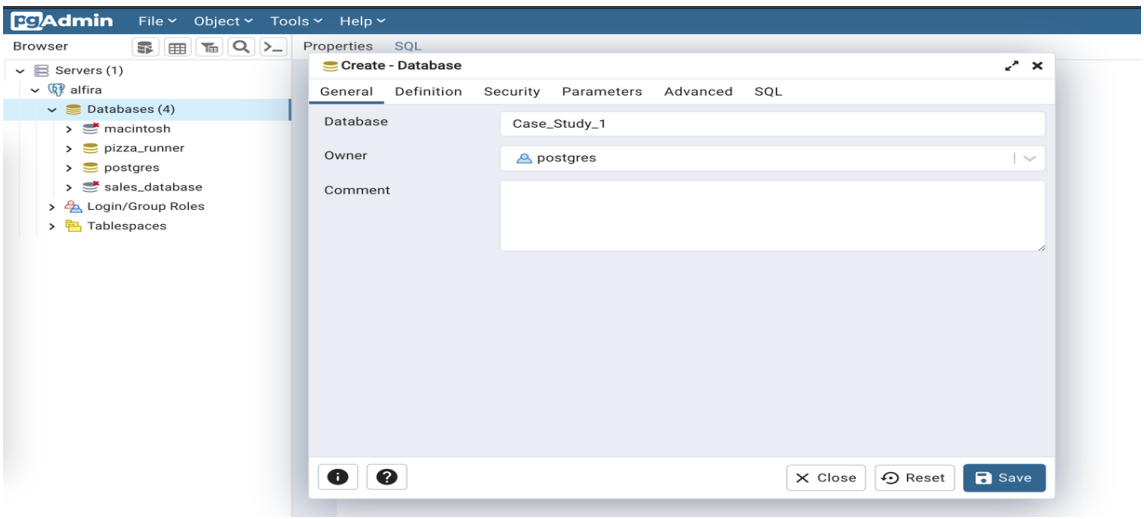
Danny as the owner, opens up a cute little restaurant in the beginning of 2021. He has captured some very basic data from few months of operation but have no idea how to use the data to run the business.

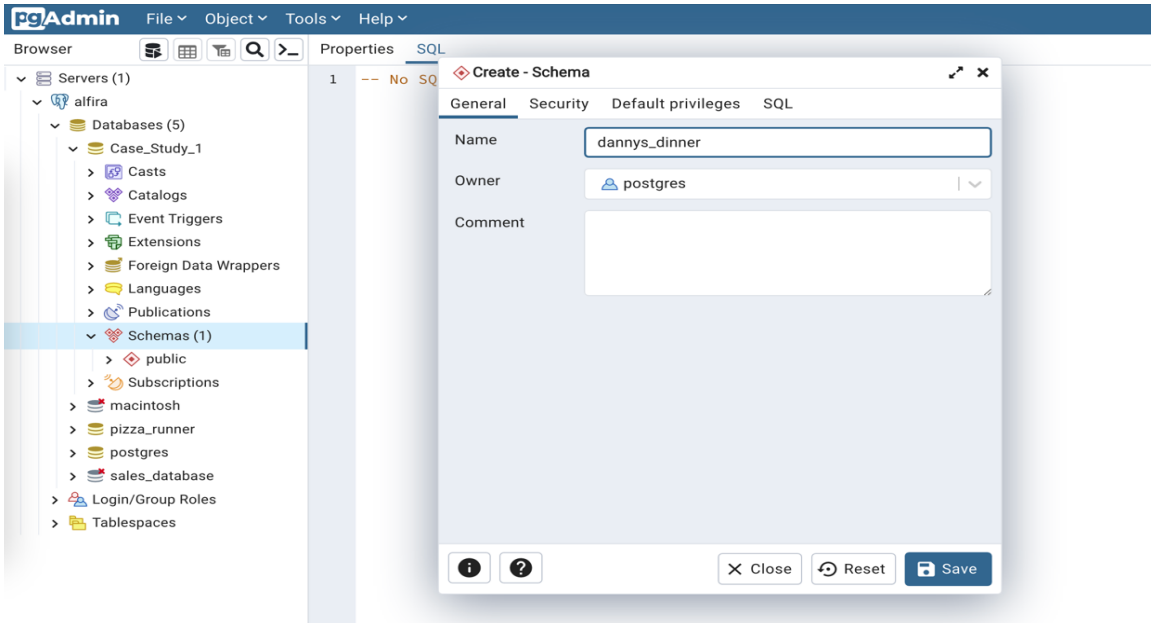
You will extract and analyze performance data from Danny's Diner to quantify the customer's overview and tell the story of how you have been able to generate that growth.

2. Data Preparation

2.1 Create database, schema and table

For this case study, I used PostgreSQL then I create the database, schema and table by using menu options after right clicking on database as shown in following image:





2.2 Tables

Once the database and schema are created, then the three tables are created according to the following SQL query :

a. Sales

Records consist of customer's orders with customer id, date when the order is ordered, product id of item ordered. There are 15 rows in this table.

```sql
CREATE TABLE sales (
    "customer_id" VARCHAR(1),
    "order_date" DATE,
    "product_id" INTEGER
);
INSERT INTO sales ("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');
```

	customer_id character varying (1)	order_date date	product_id integer
1	A	2021-01-01	1
2	A	2021-01-01	2
3	A	2021-01-07	2
4	A	2021-01-10	3
5	A	2021-01-11	3

b. Menu

Records consist of product id, name of the product and price of the product. There are 3 rows in this table.

```
24  CREATE TABLE menu (
25      "product_id" INTEGER,
26      "product_name" VARCHAR(5),
27      "price" INTEGER
28  );
29  INSERT INTO menu ("product_id","product_name","price")
30  VALUES
31   ('1', 'sushi', '10'),
32    ('2', 'curry', '15'),
33    ('3', 'ramen', '12');
```

	product_id integer	product_name character varying (5)	price integer
1	1	sushi	10
2	2	curry	15
3	3	ramen	12

c. Members

Records consist of customer id and date when customer is registered as member. There are 2 rows in this table.

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

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

	customer_id character varying (1)	join_date date
1	A	2021-01-07
2	B	2021-01-09

3. Project Goal

Danny needs your help gaining insights to provide a better and more personalized experience for his loyal customers.

3.1 Objectives

- How is customer visiting pattern ?
- How much money customers have spent and also which menu items are their favourite ?
- Expansion of existing customer loyalty program
- Join all tables and about the ranking of customer products so Danny and his team can quickly derive insights.

3.2 Problems Question

1. What is the total amount each customer spent at the restaurant?
2. How many days has each customer visited the restaurant?
3. What was the first item from the menu purchased by each customer?
4. What is the most purchased item on the menu and how many times was it purchased by all customers?
5. Which item was the most popular for each customer?
6. Which item was purchased first by the customer after they became a member?
7. Which item was purchased just before the customer became a member?
8. What is the total items and amount spent for each member before they became a member?
9. If each $1 spent equates to 10 points and sushi has a 2x points multiplier - how many points would each customer have?
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?
11. Join all the things
12. Rank all the things

4. Analysis and Visualization

Q1. What is the total amount each customer spent at the restaurant?

```
SELECT customer_id,
        SUM(price)as total_amount
FROM dannys_dinner.sales t1
    LEFT JOIN dannys_dinner.menu t2
        ON t1.product_id = t2.product_id
GROUP BY 1
ORDER BY 1;
```

Result :

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

Q2. How many days has each customer visited the restaurant?

```
11  SELECT customer_id,
12         COUNT(DISTINCT order_date)as number_days
13  FROM dannys_dinner.sales
14  GROUP BY 1
15  ORDER BY 1
```

Result :

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

Insight :

	Customer_id	Total_Spent	Number_Visit
1.	A	76	4
2.	B	74	6
3.	C	36	2

Even though it is not customer B who spend the most in total, customer B is the ones who visit the restaurant the most.

Meanwhile, Customer A generates the highest total spend and this can be an initial diagnosis of a customer being interested or satisfied with the menu.

Q3. What was the first item from the menu purchased by each customer?

To get the first item purchased, we can use window function rank () to rank the items ordered by each customer in CTE. Then once we get the rank, we can filter rows using WHERE clause.

```
18  WITH item_rank AS(
19      SELECT customer_id,
20              order_date,
21              product_name,
22              ROW_NUMBER() OVER(PARTITION BY customer_id ORDER BY order_date)as ranking
23      FROM dannys_dinner.sales t1
24          LEFT JOIN dannys_dinner.menu t2
25              ON t1.product_id = t2.product_id
26  )
27      SELECT customer_id,
28              order_date as first_date,
29              product_name as first_item
30      FROM item_rank
31      WHERE ranking = 1;
```

Result :

	customer_id character varying (1)	first_date date	first_item character varying (5)
1	A	2021-01-01	curry
2	B	2021-01-01	curry
3	C	2021-01-01	ramen

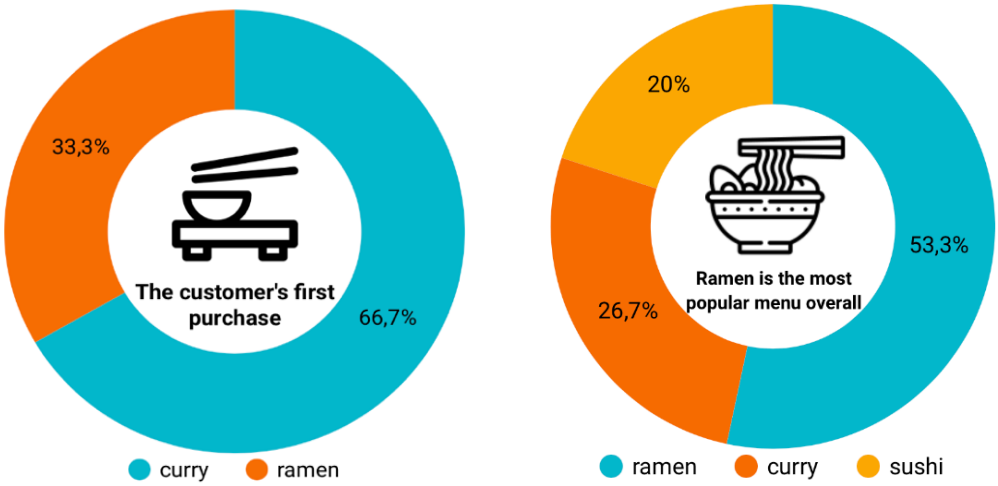
Q4. What is the most purchased item on the menu and how many times was it purchased by all customers?

```
34  SELECT product_name,
35          COUNT(*) as number_items,
36          ROW_NUMBER() OVER(ORDER BY COUNT(*) desc)as ranking
37  FROM dannys_dinner.sales t1
38          LEFT JOIN dannys_dinner.menu t2
39              ON t1.product_id = t2.product_id
40  GROUP BY 1
41  ORDER BY 2 DESC;
```

Result :

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

Insight :



Even though 66.7% of customers chose Curry as the menu they ordered for the first time. Apparently, the overall favorite menu is ramen, as much as 53.3%.

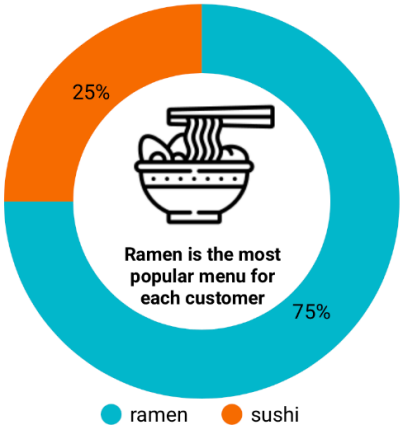
Q5. Which item was the most popular for each customer?

```
WITH item_rank AS(
    SELECT customer_id,
            product_name,
            COUNT(*) as number_items,
            ROW_NUMBER() OVER(PARTITION BY customer_id ORDER BY COUNT(*) desc)as ranking
    FROM dannys_dinner.sales t1
            LEFT JOIN dannys_dinner.menu t2
                ON t1.product_id = t2.product_id
    GROUP BY 1,2
    ORDER BY 1,3 DESC
)
    SELECT customer_id,
            product_name,
            number_items
    FROM item_rank
    WHERE ranking = 1;
```

Result :

	customer_id character varying (1)	product_name character varying (5)	number_items bigint
1	A	ramen	3
2	B	sushi	2
3	C	ramen	3

Insight :



75% of customers choose the ramen menu as their favorite menu.

Q6. Which item was purchased first by the customer after they became a member?

Let's consider that if the order_date matches the membership date then the purchase made on this date, was the first customer's purchase as a member. It means we need to include this date in the WHERE clause.

```
62  WITH item_rank AS(
63      SELECT t1.customer_id,
64              order_date,
65              join_date,
66              product_name,
67              ROW_NUMBER() OVER(PARTITION BY t1.customer_id ORDER BY order_date)as ranking
68      FROM dannys_dinner.sales t1
69          LEFT JOIN dannys_dinner.menu t2 ON t1.product_id = t2.product_id
70          LEFT JOIN dannys_dinner.members t3 ON t1.customer_id = t3.customer_id
71      WHERE order_date >= join_date
72  )
73      SELECT customer_id,
74              order_date as first_date,
75              product_name as first_item
76      FROM item_rank
77      WHERE ranking = 1;
```

Result :

	customer_id character varying (1)	first_date date	first_item character varying (5)
1	A	2021-01-07	curry
2	B	2021-01-11	sushi

Q7. Which item was purchased just before the customer became a member?

```
WITH item_rank AS(
    SELECT t1.customer_id,
            order_date,
            join_date,
            product_name,
            ROW_NUMBER() OVER(PARTITION BY t1.customer_id ORDER BY order_date)as ranking
    FROM dannys_dinner.sales t1
        LEFT JOIN dannys_dinner.menu t2 ON t1.product_id = t2.product_id
        LEFT JOIN dannys_dinner.members t3 ON t1.customer_id = t3.customer_id
    WHERE order_date < join_date
)
    SELECT customer_id,
            order_date as date_before_member,
            product_name as first_item
    FROM item_rank
    WHERE ranking = 1;
```

Result :

	customer_id character varying (1)	date_before_member date	first_item character varying (5)
1	A	2021-01-01	sushi
2	B	2021-01-01	curry

Insight :

As we can see, before and after joining ‘Customer loyalty’ program both customer ordered ‘sushi’ and ‘curry’.

Q8. What is the total items and amount spent for each member before they became a member?

```
98     SELECT t1.customer_id,
99            COUNT(t1.product_id)as total_items,
100           SUM(price)as total_spent
101    FROM dannys_dinner.sales t1
102        LEFT JOIN dannys_dinner.menu t2 ON t1.product_id = t2.product_id
103        LEFT JOIN dannys_dinner.members t3 ON t1.customer_id = t3.customer_id
104    WHERE order_date < join_date
105    GROUP BY 1
106    ORDER BY 1;
```

Result :

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

If we compare this table with table Q1 (which is the total expenses of each customer), the total expenses of each member increase before and after joining.

For customer A's total spending, it increased 67% from $25 to $76
For customer B's total spending, it increased 46% from $40 to $74

Q9. If each $1 spent equates to 10 points and sushi has a 2x points multiplier - how many points would each customer have?

Since points value only for members so we need to specify for each customer after became a member.

```
112 SELECT t1.customer_id,
113            SUM(points)as total_points
114 FROM dannys_dinner.sales t1
115     JOIN (
116            SELECT product_id,
117                   CASE WHEN product_name = 'sushi' THEN price * 20
118                       ELSE price * 10
119                   END as points
120            FROM dannys_dinner.menu)t2
121         ON t1.product_id = t2.product_id
122     JOIN dannys_dinner.members t3 ON t1.customer_id = t3.customer_id
123 WHERE order_date >= join_date
124 GROUP BY 1
125 ORDER BY 1;
```

Result :

	customer_id character varying (1)	total_points numeric
1	A	510
2	B	440

Q10. 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?

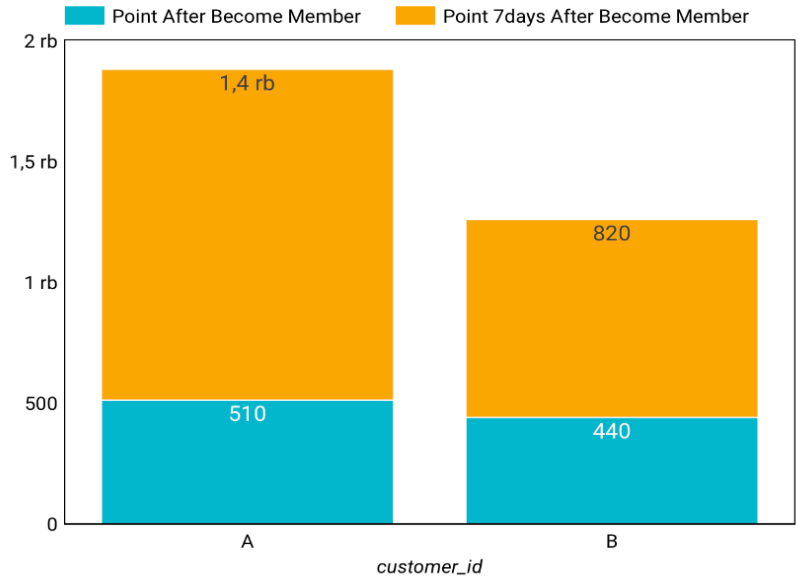
- First Step : calculate the total points for each customer as in number 9
- Once we get total point, we can get total points for each customer at the end of January
- Since the hint is end of January so we can use CASE WHEN function and Interval function to retrieve data.

```
130 WITH count_total_points AS(
131     SELECT t1.customer_id,
132             order_date,
133             join_date,
134             product_name,
135             SUM(points)as total_points
136     FROM dannys_dinner.sales t1
137         JOIN (SELECT product_id,
138                         product_name,
139                         CASE WHEN product_name = 'sushi' THEN price * 20
140                             ELSE price * 10
141                         END as points
142                 FROM dannys_dinner.menu)t2
143             ON t1.product_id = t2.product_id
144         JOIN dannys_dinner.members t3 ON t1.customer_id = t3.customer_id
145     GROUP BY 1,2,3,4
146     ORDER BY 1
147 )
148     SELECT customer_id,
149             SUM(CASE WHEN order_date >= join_date
150                         AND order_date < join_date + (7*INTERVAL '1 day')
151                             AND product_name != 'sushi' THEN total_points *2
152                     ELSE total_points
153                 END)as new_total_points
154     FROM count_total_points
155     WHERE DATE_PART('month', order_date) = 1
156     GROUP BY 1
157     ORDER BY 1;
```

Result :

	customer_id character varying (1)	new_total_points numeric
1	A	1370
2	B	820

Insight :



Both customers experienced an increase in points after they became members. But customer A who showed a significant increase in points after 7 days became a member from 510 increased 62.7% to 1370 during January.

5. Insight and Recommendation

5.1 Insight :

- Customer patterns improved during January. For customer A's total spend, that increased 67% from $25 to $76. And for customer B's total spend, this increased 46% from $40 to $74.

- Customer B was the person who frequents the restaurant the most. Meanwhile, Customer A generated the highest total spending and it could be an initial diagnosis of a customer being interested or satisfied with the menu.
- Ramen was favourite menu for customer A and C whereas B likes all three items equally as per the data.
- Even though Ramen was popular but before and after joining 'Customer loyalty' program, both customer ordered 'sushi' and 'curry'.
- Customer A was the first 'Loyal Customer' followed by B .
- Customer C has purchased the lowest out of all three customer and also, he is not a member of 'loyalty program'.

5.2 Recommendation :

- Find out what makes sushi their favorite menu for customers made first order and apply the same strategy in other customer cities.
- With the majority of revenue coming from members, Danny and team can focus campaigns and budgets on loyal and potential customers.
- The restaurant should utilize customer and product information for marketing strategies that will help in get loyal customer.