DANNY'S DINNER

DATA ANALYSIS

to help the restaurant stay afloat

PRESENTATION

Febi Febriyanti Salina Berutu~

Introduction

Danny's Diner is a small restaurant founded by Danny in early 2021. It serves three of his favorite foods: sushi, curry, and ramen. Despite Danny's enthusiasm for Japanese food, the diner **needed help** managing the basic data that had been collected over several months of operation **to keep the business afloat**.

Project Description

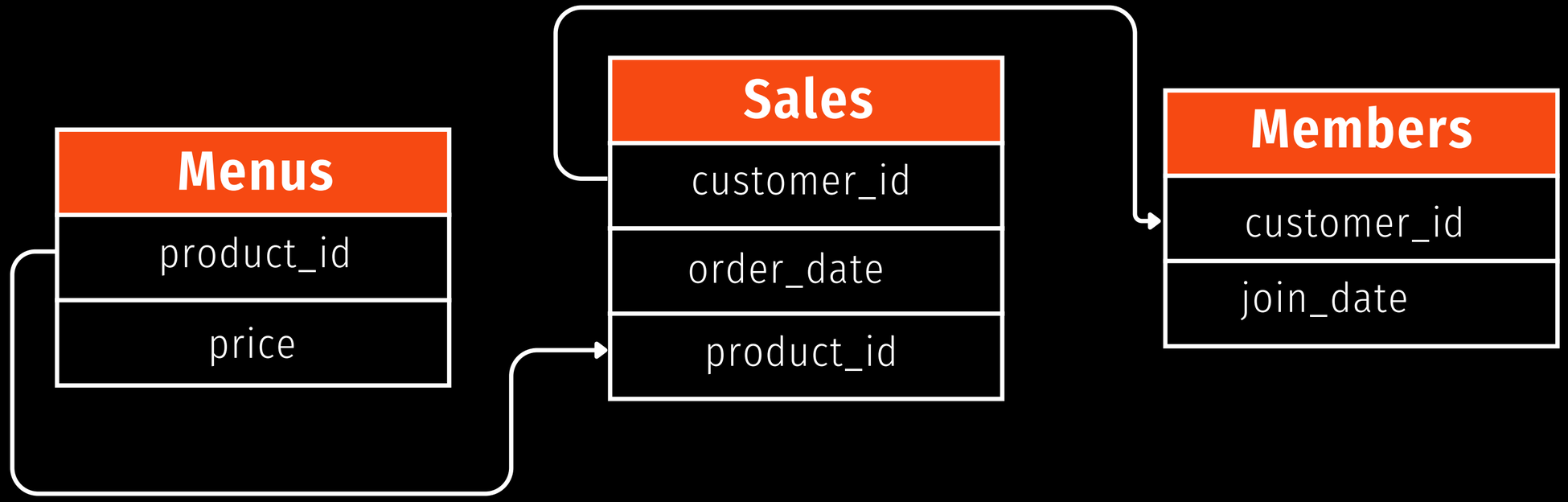
This project **aims to help Danny's Diner use the data it has collected to optimize its restaurant operations and improve its business performance.**

This case study **focuses on analyzing customer visit patterns, analyzing customer favorite menus, and evaluating loyalty programs**.

This project uses **MySQL** as a database management system (DBMS) to store and analyze restaurant data.

BORCELLE *RESTAURANT*

The Database Schema



Dataset

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

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

Query

```
select customer_id, sum(price) as total_price
from Sales
inner join menu on sales.product_id=menu.product_id
group by customer_id;
```

Output

customer_id	total_price
A	76
B	74
C	36

The total amount spent by customers A, B, C is $76, $74, $36 respectively

Actionable Insight

Here are the actions Danny can consider based on the analysis of customer spending at his restaurant

- Danny can give points to customers who have a total purchase of at least $50.
- Giving special discounts or gifts to customers who successfully invite others to buy.

2 How many days has each customer visited the restaurant?

Query

```
select customer_id,
count(distinct order_date) as total_visited
from sales
group by customer_id;
```

Output

	customer_id	total_visited
▶	A	4
	B	6
	C	2

The total number of days customer A, B, C visit danny's dinner are 4,6,2 respectively. This means customer B makes the most visits.

Actionable Insight

Here are the actions Danny can consider based on the analysis of the total visits (days) of each customer

- Danny can ask customer B what makes him frequent the restaurant.
- Danny can reward customers with a total of 10 visits (for example) in a month.

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

Query

```
select s.customer_id, min(s.order_date) as first_date_order, m.product_name
from sales s
inner join menu m on s.product_id=m.product_id
group by s.customer_id, m.product_name
having min(s.order_date) =
    (Select s2.order_date
    from sales s2
    where s2.customer_id=s.customer_id
    order by order_date
    limit 1);
```

Output

customer_id	first_date_order	product_name
A	2021-01-01	sushi
A	2021-01-01	curry
B	2021-01-01	curry
C	2021-01-01	ramen

- Assume that the first menu item is the menu ordered by the customer on the first visit. It is obtained that A bought sushi and curry, B bought curry, and C bought ramen.

Actionable Insight

Here are the actions Danny can consider based on the analysis of the customer's first purchase item

It can be considered that curry is one of the must-order Japanese foods for customers when visiting a Japanese restaurant. Therefore, it is very important for Danny's Restaurant to ensure that their curry flavor image is superior, so that it can capture the hearts of customers from their first visit.

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

Query

```
select m.product_name, count(s.product_id) as times_purchased
from sales s
inner join menu m on s.product_id=m.product_id
group by s.product_id
order by times_purchased desc
limit 1;
```

Output

	product_name	times_purchased
▶	ramen	8

It can be seen that ramen is the most ordered menu, with a total of 8 orders.

Actionable Insight

Here are the actions Danny can consider based on the analysis of the most purchased menu items.

- if it is assumed that the most purchased menu is the most favorite menu, then danny can emphasize ramen products are the superior menu of his restaurant and provide attractive offers for every ramen purchase.

5 Which item was the most popular for each customer?

Query

```
select s.customer_id, m.product_name as popular_item, count(s.product_id) as times_purchased
from sales s
inner join menu m on s.product_id=m.product_id
group by s.customer_id, s.product_id
having count(s.product_id) = (
select count(s2.product_id) from sales s2
where s2.customer_id=s.customer_id
group by s2.product_id
order by count(s2.product_id) desc
limit 1);
```

Output

customer_id	popular_item	times_purchased
A	ramen	3
B	curry	2
B	sushi	2
B	ramen	2
C	ramen	3

Customer A and C have the same popular menu, ramen, with a total of 3 purchases. Meanwhile, customer B buys an equal amount of each menu, so there is no specific popular menu for him.

Actionable Insight

Here are the actions Danny can consider based on the analysis of each customer's popular menu items

- if it is assumed that the most purchased menu is the most favorite menu, then danny can emphasize ramen products are the superior menu of his restaurant and provide attractive offers for every ramen purchase.

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

Query

```
select s.customer_id , mem.join_date, m.product_name, s.order_date
from menu m
inner join sales s on m.product_id=s.product_id
inner join members mem on s.customer_id=mem.customer_id
where s.order_date =
    (select
    max(s2.order_date) from sales s2
    where s2.customer_id=s.customer_id and s2.order_date <mem.join_date
    group by s.customer_id limit 1 );
```

Output

	customer_id	join_date	product_name	order_date
▶	A	2021-01-07	curry	2021-01-01
	A	2021-01-07	sushi	2021-01-01
	B	2021-01-09	sushi	2021-01-04

Insight

It is assumed that customer A becomes a member first before making a purchase on 2021-01-07.

- Customer A last bought curry and sushi before becoming a member, while customer B only bought sushi.
- Customer C is not shown because he is not a member yet.

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

Query

```
select s.customer_id, mem.join_date, m.product_name, s.order_date
from menu m
inner join sales s on m.product_id=s.product_id
inner join members mem on s.customer_id=mem.customer_id
where s.order_date =
    (select
    min(s2.order_date) from sales s2
    where s2.customer_id=s.customer_id and s2.order_date>=mem.join_date
    group by s2.customer_id limit 1);
```

Output

	customer_id	join_date	product_name	order_date
▶	A	2021-01-07	curry	2021-01-07
	B	2021-01-09	sushi	2021-01-11

Insight

It is assumed that customer A becomes a member first before making a purchase on 2021-01-07.

- After becoming a member, the first menus purchased by customers A and B are curry and sushi, respectively.
- The first menu ordered by customers A and B is the same as the last menu before they became members, indicating that it is their favorite menu.

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

Query

```
select s.customer_id,
count(s.product_id) as total_item_before_member,
sum(m.price) as total_price_before_member
from menu m
inner join sales s on m.product_id  = s.product_id
inner join members mem on s.customer_id=mem.customer_id
where s.order_date <=
    (select
    max(s2.order_date) from sales s2
    where s2.customer_id=s.customer_id and s2.order_date < mem.join_date
    group by s.customer_id limit 1 )
group by s.customer_id;
```

Output

	customer_id	total_item_before_member	total_price_before_member
▶	A	2	25
	B	3	40

Insight

Here are the actions Danny can consider based on the analysis of each customer's total order before becoming a member.

- Customer A spent $25 before becoming a member, showing an increase in transactions when becoming a member of $51. Meanwhile, B's total spending before and after becoming a member can be assumed to be comparable at $40 and $34 respectively.

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

Query

```
select s.customer_id,
sum(case
        when m.product_name = 'sushi' then m.price*20
        else m.price*10
    end) as total_point
from sales s
inner join menu m on s.product_id=m.product_id
group by s.customer_id;
```

Output

customer_id	total_point
A	860
B	940
C	360

The total points owned by each customer A, B, and C are 860, 940, 360 respectively.

Actionable Insight

Here are the actions Danny can consider based on the analysis of each customer's point total

- Danny can improve the point awarding system by considering the type of customer (member/non-member), total purchases, and the restaurant's featured menus.
- Then, reward customers who reach a certain point, e.g. 1000

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?

Query

```
select s.customer_id, s.order_date ,
sum(case
        when s.order_date between mem.join_date and DATE_ADD(mem.join_date, INTERVAL 7 DAY) then m.price*20
        when m.product_name = 'sushi' then m.price*20
        else m.price*10
    end) as total_point
from menu m
inner join sales s on s.product_id=m.product_id
inner join members mem on s.customer_id=mem.customer_id
where month(s.order_date) = 1
group by s.customer_id;
```

Output

	customer_id	order_date	total_point
▶	A	2021-01-11	1370
	B	2021-01-16	940

The total points owned by each customer A and B are 1370, 940 respectively.

Actionable Insight

Here are the actions Danny can consider based on the analysis of each customer's point total

- Danny can improve the point awarding system by considering the type of customer (member/non-member), total purchases, and the restaurant's featured menus.
- Then, reward customers who reach a certain point, e.g. 1000

Get In Touch

+6281276358224

febifebriyantisb@gmail.com

github.com/febifebriyantisb

linkedin.com/in/febifebriyantisb/