**QUERYING DATA WITH SQL**

**DANNY MA’S DINNER**

Mr Danny, a restaurant owner is interested in having information about his customers, especially about their visiting patterns, how much money they’ve spent and also which menu items are their favorite. Having this deeper connection with his customers will help him deliver a better and more personalized experience for his loyal customers. He also needs these insights to help him decide whether he should expand the existing customer loyalty program or not.

3 key datasets were presented for analysis:

sales

menu

members

These datasets were gotten from the Danny Ma’s 8 weeks SQL challenge.

The Schema SQL code for the creation of permanent tables used for this project is attached below

CREATE SCHEMA dannys\_diner;

SET search\_path = dannys\_diner;

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');

CREATE TABLE menu (

"product\_id" INTEGER,

"product\_name" VARCHAR(5),

"price" INTEGER

);

INSERT INTO menu

("product\_id", "product\_name", "price")

VALUES

('1', 'sushi', '10'),

('2', 'curry', '15'),

('3', 'ramen', '12');

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');

Case Study Questions

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

select customer\_id, sum(price)as total\_amount from sales

left join menu

on sales.product\_id = menu.product\_id

group by customer\_id

order by sum(price)desc

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

select customer\_id, count(order\_date)as num\_of\_times\_visited

from

(select order\_date, customer\_id, count(order\_date) as num\_times\_visited

from sales

group by customer\_id, order\_date

order by count(order\_date)desc) date

group by customer\_id

order by 2 desc

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

select distinct customer\_id, min(order\_date), product\_name

from

(select s.customer\_id, min(s.order\_date) as order\_date, product\_name,

RANK() OVER(PARTITION BY s.customer\_id ORDER BY order\_date) as rank

from sales s

left join menu m

on s.product\_id = m.product\_id

group by s.customer\_id, product\_name, s.order\_date

order by s.order\_date

)sales

where rank = 1

group by customer\_id, product\_name

order by min(order\_date)

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

select product\_name, count(sales.product\_id) as No\_of\_time\_purchased

from menu

join sales

on menu.product\_id = sales.product\_id

group by product\_name

order by count(sales.product\_id) desc

limit 1

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

select customer\_id, product\_name,product\_count

from (select s.customer\_id, m.product\_name, count(m.product\_name)as product\_count,

RANK() OVER(PARTITION BY s.customer\_id ORDER BY product\_name) as rank

from sales s

join menu m

on s.product\_id = m.product\_id

group by s.customer\_id, m.product\_name

order by count(m.product\_name)desc)order\_count

where rank = 1

group by customer\_id, product\_name, product\_count

order by product\_count desc

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

**select customer\_id, product\_name**

from

(select s.customer\_id, s.order\_date, m.product\_name, me.join\_date,

Rank()over (partition by s.customer\_id order by s.order\_date) as rank

from sales s

left join menu m

on s.product\_id = m.product\_id

left join members me

on s.customer\_id = me.customer\_id

where

s.order\_date > me.join\_date) join\_date

where rank = 1

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

select customer\_id, product\_name

from

(select s.customer\_id, s.order\_date, m.product\_name, me.join\_date,

Rank()over (partition by s.customer\_id order by s.order\_date) as rank

from sales s

left join menu m

on s.product\_id = m.product\_id

left join members me

on s.customer\_id = me.customer\_id

where

s.order\_date < me.join\_date) join\_date

where rank = 1

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

select distinct s.customer\_id, m.product\_name,s.order\_date,count(s.product\_id), sum(m.price),me.join\_date

from sales s

left join menu m

on s.product\_id = m.product\_id

left join members me

on s.customer\_id = me.customer\_id

where s.order\_date < me.join\_date

group by 1,2,3,me.join\_date

order by 1

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

select customer\_id, sum(points)as total\_point

from

(SELECT s.customer\_id, m.product\_name, COUNT(s.product\_id), SUM(m.price) as total\_spent,

CASE WHEN m.product\_name= 'sushi' THEN 2 \* 10 \* SUM(m.price)

ELSE 10 \* SUM(m.price) end as points

FROM dannys\_diner.sales s

LEFT JOIN dannys\_diner.menu m

ON s.product\_id = m.product\_id

LEFT JOIN dannys\_diner.members mb

ON s.customer\_id = mb.customer\_id

GROUP BY 1,2) point

group by 1

order by 1

**--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?**

With CTE AS (

SELECT s.customer\_id, m.product\_name,s.order\_date, mb.join\_date,

COUNT(s.product\_id), SUM(m.price) AS total\_spent,

CASE WHEN s.order\_date BETWEEN mb.join\_date - 1 AND mb.join\_date + 7

OR m.product\_name= 'sushi' THEN 2 \* 10 \* SUM(m.price)

ELSE 10 \* SUM(m.price) END AS points

FROM dannys\_diner.sales s

LEFT JOIN dannys\_diner.menu m

ON s.product\_id = m.product\_id

LEFT JOIN dannys\_diner.members mb

ON s.customer\_id = mb.customer\_id

GROUP BY 1,2,3,4

)

SELECT customer\_id, SUM(points) as total\_points

FROM CTE

GROUP BY 1

ORDER BY 1

INSIGHTS

Customer A has spent the most money in the restaurant, and this is closely followed by customer B with a difference of just $2.

Customer B however, has visited the store more than any other customer has.

Curry was most ordered item on the menu for first time customers.

Overall, Ramen was the most purchased item on the menu.

Data trends amongst the individual purchases showed ramen as the most frequent purchase amongst all customers as well.

Customers spent more money when the became members, compared to when they were not members of the loyalty program

The point-based reward system can drive purchase of a particular product on the menu if its points are more rewarding.

RECOMMENDATIONS

The natural endearment towards ramen shows that it is a stellar product and this should be maximized. Ramen should be highlighted and prominently displayed on the menu with a compelling and mouth-watering description.

Special deals or promotions that focus on ramen, such as a discount for ordering it, more reward points, or a complimentary drink or dessert should be considered

More data that can encourage targeted serving of customers should be collected e.g. addresses, opinion on delivery services, customer feedback etc.

A positive experience in the restaurant will definitely increase sales and drive customer repeat purchases. This refers to how customers are treated by staff.

Members are observed to spend more money, so strategies must be put in place to convert customers to members

The point-based reward system works and should be encouraged. This system does not have to be fixed but can be flexible to ensure that different products can be promoted at different times.