DATA ANALYSIS

SQL PROJECT

Probelm 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.

He plans on using these insights to help him decide whether he should expand the existing customer loyalty program, additionally he needs help to generate some basic datasets so his team can easily inspect the data without needing to use SQL.

Danny has provided you with a sample of his overall customer data due to privacy issues, but he hopes that these examples are enough for you to write fully functioning SQL queries to help him answer his questions!

Danny has shared 3 key datasets for this case study:

```
CREATE DATABASE dannys_diner;
use 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');
```

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

```
select s.customer_id, sum(m.price) 'total amount spent'
from sales s
inner join menu m on m.product_id=s.product_id
group by customer_id
```

output:

customer_id	total amount spent
A	76
B	74
C	36

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

```
select customer_id,count(distinct(order_date)) 'no.of days visited'
from sales
group by customer_id
```

output:

customer_id	no.of days visited
A	4
B	6
C	2

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

```
select customer_id,product_name from
(select *,rank() over(partition by customer_id order by order_date asc) rnk from
(select s.customer_id,s.order_date,m.product_name from sales s inner join menu m
on s.product_id=m.product_id)a)b
where rnk=1 group by customer_id,product_name
```

output:

customer_id	product_name
A	sushi
A	curry
B	curry
C	ramen

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

```
select product_name,count(*) 'no.of times purchased'
from sales s
inner join menu m on s.product_id=m.product_id
group by product_name
order by count(*) desc limit 1
```

output:

	product_name	no.of times purchased
▶	ramen	8

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

```
select customer_id,product_name 'popular item' from
(select *,rank() over(partition by customer_id order by count desc) rnk from(
select customer_id,product_name,count(*) count from
sales s inner join menu m on
s.product_id=m.product_id
group by customer_id,product_name)a)b
where rnk=1;
```

output:

customer_id	popular item
A	ramen
B	curry
B	sushi
B	ramen
C	ramen

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

```
select customer_id,order_date,product_name from
(select *,rank() over(partition by customer_id order by order_date asc) rnk from
(select s.customer_id,s.order_date,m.product_name from
 sales s inner join menu m
on s.product_id=m.product_id inner join members mem
on s.customer_id=mem.customer_id where s.order_date>=mem.join_date)a)b
where rnk=1;
```

output:

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

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

```
select customer_id,order_date,product_name from
(select *,rank() over(partition by customer_id order by order_date desc) rnk from
(select s.customer_id,s.order_date,m.product_name from
sales s inner join menu m on
s.product_id=m.product_id inner join members mem
on s.customer_id=mem.customer_id where s.order_date<mem.join_date)a)b
where rnk=1;
```

output:

customer_id	order_date	product_name
A	2021-01-01	sushi
A	2021-01-01	curry
B	2021-01-04	sushi

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

```
select s.customer_id,sum(m.price) ‘amount spent’,count(*)'total items purchased'
from sales s inner join menu m on s.product_id=m.product_id
inner join members mem on s.customer_id=mem.customer_id
where s.order_date<mem.join_date
group by customer_id;
```

output:

	customer_id	amount spent	total items purchased
▶	B	40	3
	A	25	2

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

```
create temporary table temp1
select customer_id,product_name,price,
case when product_name='sushi' then price*20
else price*10
end as 'points' from sales s inner join menu m on s.product_id=m.product_id

select customer_id,sum(points) ‘points’ from temp1 group by customer_id order by customer_id
```

output:

	customer_id	points
▶	A	860
	B	940
	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 customer_id,sum(points) from (select customer_id,point+point1 as points from
(select *, case when product_name='sushi' then price*20 else price*10 end as 'point1'
from (select *, case when day(order_date)<=day(join_date)+6 then price*20 else
price*10 end as 'point' from
(select s.customer_id,order_date,product_name,join_date,price from
sales s inner join menu m on s.product_id=m.product_id inner join members mem
on s.customer_id=mem.customer_id where s.order_date>=mem.join_date
 and s.order_date<='2021-01-31')a)b)c)d group by customer_id
```

output:

customer_id	sum(points)
B	640
A	1530

Insights/Recommendations:

-> Danny can know his earnings from the restaurant and also how frequently the customers are visiting, so that he can give discounts to the loyal customers.

-> First purchased item by the customers, this gives a simple indication that the customers are getting attracted to that particular product and because of that they are coming to restaurant.

-> Though people are getting atracted and they are coming to the restaurant but they are not interested in buying it regularly, they are finding ‘ramen’ is much more better. And it is best to prepare an item similar to 'ramen' but with new addings, so that there is a chance of attracting customers.

-> By knowing the specific product which is being liked by each of the customer. Danny can definitely run offers for that particular product so that sales get increased.

-> May be the customers find that the items are good and want to experience completely the restaurant's items by becoming a member. By improving the quality and service of that particular items they can make customers to become the members.

-> After spending such amount they are becoming a member so this is the very good indication that restaurant is going on in a good way.

-> Having the information of points acquired by each of the customer, they can run offers for the customers who are having a specific number of points, in this way customers tries to acquire that number of points, so that sales get increased.