```
/* --------------------
   Case Study-1  Questions and solutions using sqlite
   --------------------*/
```

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

```
SELECT Distinct customer_id, sum(price)
FROM dannys_diner.sales
Join dannys_diner.menu
Using(product_id)
Group By customer_id;
```

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

```
SELECT  customer_id, count(distinct order_date) as days_count
FROM dannys_diner.sales
Group By customer_id;
```

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

```
With sales_customer As(
SELECT customer_id,product_name,order_date,
Dense_Rank() Over(Partition By customer_id Order By order_date) As Rank
FROM dannys_diner.sales
Join dannys_diner.menu
Using(product_id)
)

Select customer_id,product_name
From sales_customer
Where Rank =1
Group By customer_id,product_name
```

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

```
SELECT  s.product_id, product_name, count(s.product_id) as total_purchased
FROM dannys_diner.menu as m
join dannys_diner.sales as s
Using(product_id)
Group By s.product_id, product_name
Order by total_purchased desc
```

```
Limit 1;
```

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

```
With most_popular As(
SELECT  s.customer_id, product_name, count(s.product_id) as total_purchased,
Dense_Rank() Over(Partition By s.customer_id Order By count(s.product_id)desc ) As Rank
FROM dannys_diner.menu as m
join dannys_diner.sales as s
Using(product_id)
Group By s.customer_id,product_name
Order by s.customer_id,total_purchased desc
)

Select
customer_id,product_name,total_purchased
From most_popular
Where Rank =1
```

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

```
WITH item As (select product_id,product_name from dannys_diner.menu),

when_member As(
Select join_date,m.customer_id,product_name,
DENSE_RANK() OVER(PARTITION BY s.customer_id
     ORDER BY s.order_date) AS rank
From dannys_diner.members as m
Join dannys_diner.sales as s
using(customer_id)
join item
Using(product_id)
Order by join_date

)

select join_date, customer_id, product_name
From when_member
Where Rank =1
```

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

```
SELECT s.customer_id,product_name,order_date,s.product_id
```

```
FROM dannys_diner.sales as s
JOIN dannys_diner.members as m
Using(customer_id)
Join dannys_diner.menu as m1
On s.product_id = m1.product_id
Where order_date < join_date
```

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

```
Select
s.customer_id,count(distinct m.product_id) as tot_item ,sum(m.price) as total_price
From dannys_diner.sales as s
Join dannys_diner.menu as m
Using(product_id)
Join dannys_diner.members as m1
Using(customer_id)
Where order_date < join_date
Group by s.customer_id
```

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

```
With n As(
Select m.product_id,order_date,s.customer_id,m.product_name,sum(m.product_id) as tot_item
,sum(price) as total_price
From dannys_diner.sales as s
Join dannys_diner.menu as m
Using(product_id)
  Group by m.product_id,order_date,s.customer_id,m.product_name
),

points As(
Select customer_id,product_name,tot_item,total_price,
Case When product_name ='sushi' then total_price*20 Else total_price*10 End As rewards
From n)

Select customer_id,sum(rewards) as tot_rewards
From points
group by customer_id
```

-- 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 n As(
Select order_date,s.customer_id,m.product_name,sum(m.product_id) as tot_item ,sum(price) as total_price
From dannys_diner.sales as s
Join dannys_diner.menu as m
Using(product_id)
group by s.customer_id,m.product_name,order_date
  ),

points As(
Select join_date, n.customer_id,product_name,tot_item,total_price,Case When n.customer_id in('A','B') And join_date between DATE_TRUNC('month', '2021-01-01'::date) and DATE_TRUNC('month', '2021-02-01'::date) then 'January' Else'*' End As month,

Case When product_name ='sushi' then total_price*20
          When Order_date between join_date and  DATE_TRUNC('week', '2021-01-14 '::date) then total_price*20 Else total_price*10 End As rewards

From dannys_diner.members as m
Join n
On  m.customer_id= n.customer_id
)


Select customer_id, sum(rewards)
from points
group by customer_id
--*/


/*
With n As(
Select order_date,s.customer_id,m.product_name,sum(m.product_id) as tot_item ,sum(price) as total_price
From dannys_diner.sales as s
Join dannys_diner.menu as m
Using(product_id)
```

```
group by s.customer_id,m.product_name,order_date
  )

Select join_date, n.customer_id,product_name,tot_item,total_price,
Case When n.customer_id in('A','B') And join_date between DATE_TRUNC('month',
'2021-01-01'::date) and DATE_TRUNC('month', '2021-02-01'::date) then 'January' Else'*' End As
month,

Case When product_name ='sushi' then total_price*20
          When Order_date between join_date and  DATE_TRUNC('week', '2021-01-14 '::date)
then total_price*20 Else total_price*10 End As rewards

From dannys_diner.members as m
Join n
On  m.customer_id= n.customer_id
Order by customer_id desc
*/



Bonus #1

Select s.customer_id,order_date,product_name,price,
Case When m.join_date > s.order_date Then 'N'
When m.join_date <= s.order_date Then 'Y'
Else 'N' End As member

From dannys_diner.members as m
Left Join dannys_diner.sales as s
On m.customer_id = s.customer_id
Left Join dannys_diner.menu as m1
Using(product_id)

#2

With dinner As (
Select s.customer_id,order_date,product_name,price,
Case When m.join_date > s.order_date Then 'N'
When m.join_date <= s.order_date Then 'Y'
Else 'N' End As member

From dannys_diner.members as m
Left Join dannys_diner.sales as s
On m.customer_id = s.customer_id
```

```
Left Join dannys_diner.menu as m1
Using(product_id)
)

Select *,
Case When member ='N' Then Null
Else Rank() Over(Partition By customer_id,member Order By order_date) End As ranking

From dinner
```