--case study 1 questions:

/\*

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?

(Note: you can choose to return all items for their first order or pick 1 of the items from their first order, I'll accept either)

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?

(Note: you can choose to return all or 1 of the items that meet this condition, I'll accept either)

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

(Note: you can choose to return all or 1 of the items that meet this condition, I'll accept either)

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 customers A and B have at the end of January?

\*/

----------------------

--all queries combined

----------------------

select

sum(price) as total\_spent,

sales.customer\_id,

count(sales.order\_date) as visits,

max(first\_order\_ever) as first\_order\_ever,

max(first\_order\_date) as first\_order\_date,

max(most\_purchased\_product) as most\_purchased\_product,

max(quantity\_of\_most\_purchased\_product) as quantity\_of\_most\_purchased\_product,

max(quantity\_of\_most\_pop\_product\_per\_cus) as quantity\_of\_most\_pop\_product\_per\_cus,

max(most\_pop\_product\_per\_cus) as most\_pop\_product\_per\_cus,

max(first\_order\_of\_membership) as first\_order\_of\_membership,

max(first\_order\_date\_of\_membership) as first\_order\_date\_of\_membership,

max(orderdate\_before\_membership) as orderdate\_before\_membership,

max(product\_before\_membership) as product\_before\_membership,

max(total\_items\_before\_membership) as total\_items\_before\_membership,

max(total\_price\_before\_membership)as total\_price\_before\_membership,

max(points) as points,

max(sum\_a\_b) as sum\_of\_points

from TIL\_PLAYGROUND.CS1\_DANNYS\_DINER.sales

join TIL\_PLAYGROUND.CS1\_DANNYS\_DINER.menu on TIL\_PLAYGROUND.CS1\_DANNYS\_DINER.sales.product\_id = menu.product\_id

join

(

select product\_name as first\_order\_ever, customer\_id, order\_date as first\_order\_date

from(

select customer\_id,

product\_name,

order\_date,

--zeilenzahl erzeugen mit row\_number() over(-mit partition by column wird bei einer neuen gruppe der column wieder neu mit der zählung angefangen;

-- mit order by column werden alle zeilen durchgezählt )

row\_number() over (partition by customer\_id order by customer\_id, order\_date asc) as row\_number

from TIL\_PLAYGROUND.CS1\_DANNYS\_DINER.sales

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

order by customer\_id, order\_date

)

where row\_number = 1

)

as t on sales.customer\_id = t.customer\_id

join

(

select count(menu.product\_name) as quantity\_of\_most\_purchased\_product, product\_name as most\_purchased\_product

from menu

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

group by product\_name

order by quantity\_of\_most\_purchased\_product desc

limit 1

)

as t2 on menu.product\_name = t2.most\_purchased\_product

join(

select customer\_id, quantity as quantity\_of\_most\_pop\_product\_per\_cus, product\_name as most\_pop\_product\_per\_cus

from (

select count(product\_name) as quantity,

product\_name,

customer\_id,

row\_number() over (partition by customer\_id order by customer\_id, quantity desc) as row\_number

from menu

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

group by customer\_id, product\_name

order by customer\_id, quantity desc

)

where row\_number = 1

)

as t3 on sales.customer\_id = t3.customer\_id

left join(

select Customer\_id, product\_name as first\_order\_of\_membership, order\_date as first\_order\_date\_of\_membership

from

(

select

product\_id,

sales.customer\_id,

order\_date,

row\_number() over( partition by sales.customer\_id order by sales.customer\_id, order\_date asc) as row\_number

from sales

join members on sales.customer\_id=members.customer\_id

where order\_date >= join\_date

order by customer\_id, order\_date asc

) as temp

join menu on temp.product\_id = menu.product\_id

where row\_number = 1

)

as t4 on sales.customer\_id = t4.customer\_id

--7.

left join (

select Customer\_id, product\_name as product\_before\_membership, order\_date as orderdate\_before\_membership

from

(

select

product\_id,

sales.customer\_id,

order\_date,

row\_number() over( partition by sales.customer\_id order by sales.customer\_id, order\_date asc) as row\_number

from sales

join members on sales.customer\_id=members.customer\_id

where order\_date < join\_date

order by customer\_id, order\_date desc

) as temp

join menu on temp.product\_id = menu.product\_id

where row\_number = 1

) as t5 on sales.customer\_id = t5.customer\_id

--8.

left join(

select count(sales.customer\_id) as total\_items\_before\_membership, sum(price) as total\_price\_before\_membership, members.customer\_id

from sales

join members

on sales.customer\_id = members.customer\_id

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

where

order\_date < members.join\_date

group by members.customer\_id

) as t6 on sales.customer\_id =t6.customer\_id

--9.

join(

select customer\_id, sum(case when product\_name = 'sushi' then 2\*price else 1\*price end)\*10 as points

from sales

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

group by customer\_id

) as t7 on sales.customer\_id = t7.customer\_id

--1.0

left join(

select sum(sum\_a\_b) as sum\_a\_b

from(

select

members.customer\_id,

sum(

case when order\_date < DATEADD(day, 7, join\_date)

and order\_date >= join\_date

then 2\*price else 1\*price end)\*10 as sum\_a\_b

from sales

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

join members on sales.customer\_id = members.customer\_id

group by members.customer\_id)

)

group by sales.customer\_id;

----------------------------------------------------------------------

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

----------------------------------------------------------------------

select product\_name as first\_order\_ever, customer\_id, order\_date as first\_order\_date

from(

select customer\_id,

product\_name,

order\_date,

row\_number() over (partition by customer\_id order by customer\_id, order\_date asc) as row\_number

from sales

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

order by customer\_id, order\_date

)

where row\_number = 1

;

------------------------------------------------------------------------------------------------------

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

------------------------------------------------------------------------------------------------------

select count(menu.product\_name) as quantity\_of\_most\_purchased\_product, product\_name as most\_purchased\_product

from menu

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

group by product\_name

order by quantity\_of\_most\_purchased\_product desc

limit 1;

-------------------------------------------------------

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

-------------------------------------------------------

--use row number() to add a row number to each row

--use over partition to start numbering at each partition/group

select customer\_id, quantity as quantity\_of\_most\_pop\_product\_per\_cus, product\_name as most\_pop\_product\_per\_cus

from (

select count(product\_name) as quantity,

product\_name,

customer\_id,

row\_number() over (partition by customer\_id order by customer\_id, quantity desc) as row\_number

--add row\_number so that first row can be selected

from menu

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

group by customer\_id, product\_name

order by customer\_id, quantity desc

)

where row\_number = 1;

-------------------------------------------------------------------------------

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

-------------------------------------------------------------------------------

select Customer\_id, product\_name as first\_order\_of\_membership, order\_date as first\_order\_date\_of\_membership

from

(

select

product\_id,

sales.customer\_id,

order\_date,

row\_number() over( partition by sales.customer\_id order by sales.customer\_id, order\_date asc) as row\_number

from sales

join members on sales.customer\_id=members.customer\_id

where order\_date >= join\_date

order by customer\_id, order\_date asc

) as temp

join menu on temp.product\_id = menu.product\_id

where row\_number = 1;

-----------------------------------------------------------------------

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

-----------------------------------------------------------------------

select Customer\_id, product\_name, order\_date

from

(

select

product\_id,

sales.customer\_id,

order\_date,

row\_number() over( partition by sales.customer\_id order by sales.customer\_id, order\_date asc) as row\_number

from sales

join members on sales.customer\_id=members.customer\_id

where order\_date < join\_date

order by customer\_id, order\_date desc

) as temp

join menu on temp.product\_id = menu.product\_id

where row\_number = 1;

------------------------------------------------------------------------------------------

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

------------------------------------------------------------------------------------------

select count(sales.customer\_id) as total\_items\_before\_membership, sum(price) as total\_price\_before membership, members.customer\_id

from sales

join members

on sales.customer\_id = members.customer\_id

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

where

order\_date < members.join\_date

group by members.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?

---------------------------------------------------------------------------------------------------------------------------

select customer\_id, sum(case when product\_name = 'sushi' then 2\*price else 1\*price end)\*10

from sales

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

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 customers A and B have at the end of January?

------------------------------------------------------------------------------------------------------------------------

select sum(sum\_a\_b) as sum\_a\_b

from(

select members.customer\_id,

sum(

case when order\_date < DATEADD(day, 7, join\_date)

and order\_date >= join\_date

then 2\*price else 1\*price end

)\*10 as sum\_a\_b

from sales

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

join members on sales.customer\_id = members.customer\_id

group by members.customer\_id

);