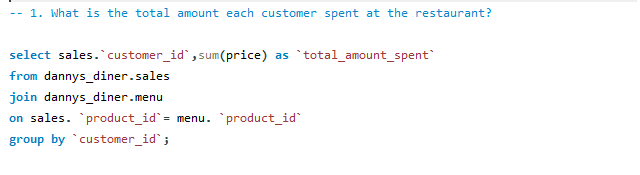
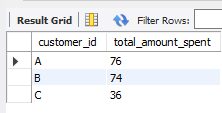
**DOCUMENTATION ON DANNYS DINER**

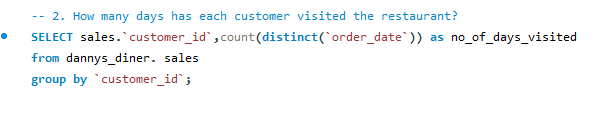
**1. What is the total amount each customer spent at the restaurant?**



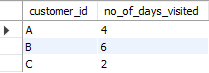
Before moving into coding aspect, let first understand the question. The question says *“what is the total amount each customer spent at the restaurant”***.** that is the total amount each customer spent after buying product in the restaurant. after getting the meaning of the question, now let move into the coding aspect. As we all know in SQL we use ***select*** statement to extract a data from a database, and I used the *sum*(**price**)function to get the **total\_amount \_spent** by each customer at the restaurant. I also use *join* statement to combine sales table and menu table together, the *as* statement in the coding above is an Alias in SQL (i.e. to give the *sum*(**price**) a temporaryname **total\_amount\_spent**) and as well as *group by* **customer\_id** to obtain the **total\_amount \_spent** by each customer at the restaurant and we end our coding with semi-colon (;). Then the SQL will give us the result in the table below for the question the total amount spent by each customer at the restaurant.



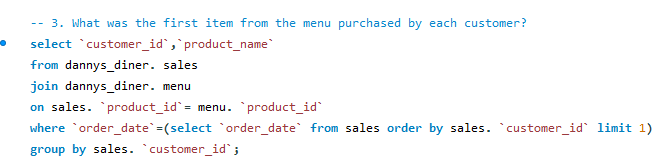
**2. How many days has each customer visited the restaurant?**



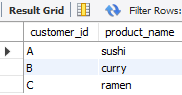
Danny’s want to the number of days each customer come to order for a product in the restaurant, and the we have explained it in the code above, let understand the meaning of the code. We get the column **customer\_id** from **sales** table so as to know the ID of each customer that have visited the restaurant also the **count(distinct** help to count the unique **order\_date** that is to count how many day has each customer come to order in the restaurant. we use ***as*** to give the **order\_date** a name **no\_of\_days\_visited** then group by customer\_id to obtain the number of days each customer visited the restaurant. the table below show us the result of the question.



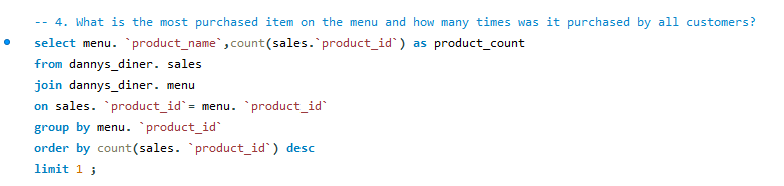
**3. What was the first item from the menu purchased by each customer**



In this kind of question, we can use either use sub query or CTE to solve it. I use sub query to solve the question. Before moving far, sub query is a query used inside another query. So let check out subquery in the coding above. We will select our column **customer\_id** and **product\_name** from sales table joining menu table. Then, we used a sub query to get the first date from sales table and filter the **order\_date by** first row of the sub query result and then we limit by 1. We ***group by* customer\_id** to obtain the first item purchased by each customer from the menu table.



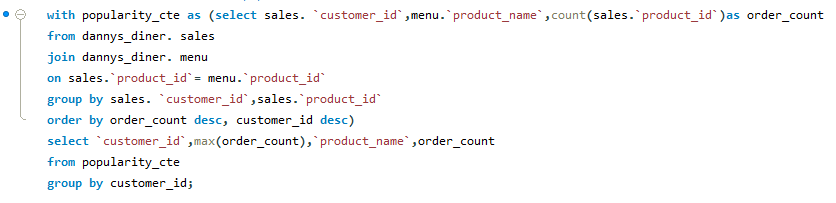
**4. What is the most purchased item on the menu and how many times was it purchased by all customers?**



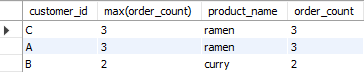
I used the **count(sales.** **`product\_id`)** to count the most purchase **product\_id** on the menu table and I used aliases to the give product\_id as product\_count, also I joined sales table and menu table on **product\_id**. Grouped the output by product\_id, sorted **DESC** and **limit** by 1 row for each **product\_id**.



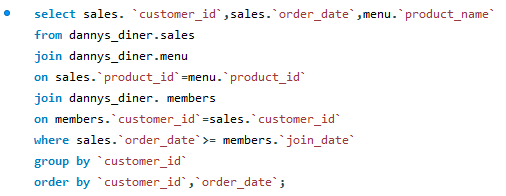
**5. Which item was the most popular for each customer?**



The question says “*which item was the most popular for each customer*” that is the product that was purchased or bought the most by each customer. Let move to the creation of Common Table Expression(CTE), the CTE called popularity\_cte. I used the count() function on product\_id so as to count the products and alias it as order\_count. Grouped the output by customer\_id and product\_id, ordered by order\_count and customer\_id. Then, I select from the popularity\_cte, I used max() function to count the maximum order\_count, group by customer\_id.



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



To answer this question, let first understand what it means. The question says “which item was purchase first by the customer after they become a member”, which simply mean that the item that was order immediately after the customer gain membership. After that, let move into coding aspect, we join sales table and menu table on **product\_id**, also we join member table and sales table on **customer\_id**, to filter the query by the where clause where sales. `order\_id` >= member. `join\_date`, using the operator >=. Grouped the output by **customer\_id**, then ordered by **customer\_id**, **order\_id.**



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

select sales.`customer\_id`,sales.`order\_date`,menu.`product\_name`

from dannys\_diner.sales

join dannys\_diner.menu

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

join dannys\_diner.members

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

where sales.`order\_date`<= members.`join\_date`

group by `customer\_id`

order by `order\_date`;

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

select sales.`customer\_id`,count(sales.`product\_id`)as items,sum(menu.`price`)as amount\_spent

from dannys\_diner.sales

join dannys\_diner.menu

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

join dannys\_diner.members

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

where sales.`order\_date`< members.`join\_date`

group by sales.`customer\_id`;

USE dannys\_diner;

select \* from points\_table;

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

select sales.`customer\_id`,menu.`product\_name`,sum(points\_table.`price`),sum(case when `product\_name`="sushi"

then menu.price\*2\*10

else menu.price\*10 end) as "point"

from dannys\_diner.points\_table

join dannys\_diner.menu

on points\_table.`product\_id`=menu.`product\_id`

join dannys\_diner.sales

on sales.`product\_id`=points\_table.`product\_id`

group by sales.`customer\_id`;