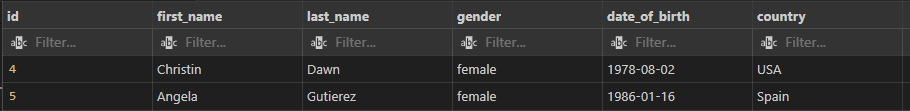
1. Write a query which select all female customers

SELECT \*

from customer

WHERE gender = "Female"

Output:



2. Write a query which prints out all customer names with the number of orders they did

SELECT customer.first\_name, customer.last\_name,

Count(orders.fk\_customer) as number\_of\_orders

from customer

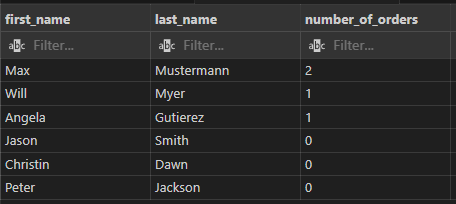
LEFT JOIN orders

On customer.id = orders.fk\_customer

GROUP BY customer.first\_name, customer.last\_name

ORDER BY number\_of\_orders DESC

Output:



3. Write a query which prints out customers with the money they spend excluding customers without any orders

SELECT customer.first\_name, customer.last\_name,

SUM(orders.sum)

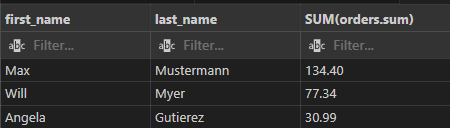
from customer

RIGHT JOIN orders

ON customer.id = orders.fk\_customer

GROUP BY customer.first\_name, customer.last\_name

Output



4. Write a query which prints out the order nr of all orders with at least 2 items

SELECT orders.order\_nr, COUNT(order\_item.fk\_order)

AS number\_of\_items

from orders

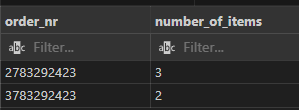
LEFT JOIN order\_item

ON order\_item.fk\_order = orders.id

GROUP BY order\_nr

HAVING number\_of\_items >= 2

Output:



TASK #2

- In fetching the tweets by using a Python script, I would be using a Python library such as tweepy, config parser, JSON, and google cloud. Tweepy is a Python Library that can help in the process of fetching the data such as tweets through the use of an API. config parser as this will be the file of the API Key, API Secret Key, Access Token, and Access Secret Token. This will be the connection of the Twitter account to the program. I will be using this so that the initialization of the said Keys and Tokens will be in a different file called "config.ini" in order for the code to be clean and reusable. Json because the said tweets that were gathered will be saved as a JSON file. Lastly, google cloud as this will be the cloud storage of the data that was gathered. These libraries will help the script run smoothly and will make the data-gathering process less hassle.

Task #3

From the data that you got from task#2, imagine reading millions or billions of rows from it. Describe how you will design the table so that processing or querying the table will be optimized.

- It depends on the start and end date. If the data has a large gap between its start and end date, I would break down the data or partition it per month or week so that processing and querying the data into a table will be optimized. However, if the data does not have a large gap, I would just break down the data per gb or whatever the storage can handle.

Task#4

- First, we need to handle the data-gathering process and where it will be stored. This is a need because if the storage or cloud can’t handle the data that it was gathering, there will be an error. As there are some cloud storages that offer free on their platforms, I would suggest buying or subscribing to the package that the cloud storage offers aligned to our needs so that every time it fetches data it can manage it. Another point would be its Authentication or its security. Lastly, the performance itself. As this will be crucial when gathering loads of data and storing it at some point. When you deploy the script in some cheap server, it will become more hassle. I think it's good to spend more money so that the process will not have bottleneck issues or anything that would become an error than to not spend some money but should have loads of errors.