

1. Write a query which select all female customers

```
SELECT *  
from customer  
WHERE gender = "Female"
```

Output:

id	first_name	last_name	gender	date_of_birth	country
abc Filter...	abc Filter...	abc Filter...	abc Filter...	abc Filter...	abc Filter...
4	Christin	Dawn	female	1978-08-02	USA
5	Angela	Gutierrez	female	1986-01-16	Spain

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:

first_name	last_name	number_of_orders
abc Filter...	abc Filter...	abc Filter...
Max	Mustermann	2
Will	Myer	1
Angela	Gutierrez	1
Jason	Smith	0
Christin	Dawn	0
Peter	Jackson	0

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

first_name	last_name	SUM(orders.sum)
abc Filter...	abc Filter...	abc Filter...
Max	Mustermann	134.40
Will	Myer	77.34
Angela	Gutierrez	30.99

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:

order_nr	number_of_items
abc Filter...	abc Filter...
2783292423	3
3783292423	2

## TASK #2

I don't have any answer for this task as I don't have any background in fetching data using python. But I have research it during the technical exam.

## Task #3

From the data that you got from task#2, imagine reading millions or billions of rows from it. Describe a way on 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 be breaking down the data or partitioning it per month or week so that processing and querying the data into table

will be optimized. However, if the data does not have a large gap, I would just be breaking down the data per gb or whatever the storage can handle.

#### Task #4

I don't have any answer for this task as I don't have any background in fetching data using python. But I have research it during the technical exam.