Exercise 1

The initial table **is already in 1NF** because it has single(atomic) valued attributes, stores values in the same domain and no some repeating in the column's names.

For moving to 2NF I notice that some columns have **Partial Dependency** such as *orderId & date & quant* (info about order) with *customerId & customerName & city* (info about customer) with *itemId & itemName & price* (info about item).

Therefore, I decided to separate initial table into four tables for **2NF**:

IDs *(orderId, customerId, itemId)*

Order *(orderId, date, quant)*

Customer *(customerId, customerName, city)*

Item *(itemId, itemName, price)*

In order for the table to become 3NF, I decide to get rid of **Transitive Dependency**.

I notice this in the *Order* table: *orderId* dependent from *date* and also dependent from *quant*.

Also, I notice this in the *ID* table: *orderId* dependent from *customerId* and also dependent from *itemId*.

Therefore, the initial table int the **3NF** looks like:

IDsO&C *(orderId, customerId)*

IDsC&I (customerId, itemId)

OrderDate *(orderId, date)*

OrderQuant *(date, quant)*

Customer *(customerId, customerName, city)*

Item *(itemId, itemName, price)*

Query for each of the requirements:

1. SELECT count(OQ .quant), sum (I.price * OQ.quant)

   FROM OrderDate OD, OrderQunt OQ, Item I

   GROUP BY orderId

2. SELECT CI.customerId

   FROM IDsC&I CI, Item I

   WHERE CI.itemId = I.itemId AND max(I.price)