EXERCISE LIST N°2:
SQL – Data Definition Language and Data Manipulation Language

I. Data Definition Language

The relational model that we are going to use for this exercise list corresponds to the database of an enterprise fixing air-conditioners. It is the same as in exercise list number one.

CUSTOMER (**cust_no**, name, surname, address, city, ZIP, tel)
PRODUCT (**reference**, designation, unit_price, stock_quantity, min_quantity)
INVOICE (**inv_no**, date, state, *customer#*)
REPLACEMENT (***product#, intervention#***, qtity)
INTERVENTION (**interv_no**, date, *technician#*, duration, *invoice#*)
TECHNICIAN (**tech_no**, name, hour_rate)

1. When you will create the tables in the dataset, in which order should you create them? Write them from the first to the last. Multiple answers are possible, so for each table which you **cannot** create at any point, write some comment, *e.g.* "SCORE_PER_SUBJECT (must be created after STUDENT and SUBJECT)".

2. If you need to drop all the tables in the dataset, in which order should you drop them? Write them from the first to the last. Multiple answers are possible, so for each table which you **cannot** drop at any point, write some comment, *e.g.* "SCORE_PER_SUBJECT (must be dropped before STUDENT and SUBJECT)".

3. Write the SQL code needed to create the table REPLACEMENT. You can pick the attribute types you find most relevant among the types given in the lecture "Chapter 4 – part 1", and add additional constraints / NOT NULL / default values whenever you think it's relevant.

4. Write **ALL** the SQL code that is needed to change the type of the attribute "reference" from the table PRODUCT (it used to be VARCHAR(7), but now you change it to VARCHAR(10)), in the whole database.

5. Your boss just informed you that the attribute "state" for the table INVOICE should always be either 'R' (regulated, paid) or 'O' (not paid yet, ongoing). Write the SQL code needed to add that constraint.

II. Data Manipulation Language

6. Write the SQL code needed to populate the tables with the data below. If the value of a given attribute is not given to you below, and this attribute is not a primary key, then you can consider that it is not a NOT NULL attribute, and just put "NULL" instead. All products cited below have already been inserted in the database (you don't need to insert the products).

Customer 10, named “NGUYEN Dô”, had two interventions in his house: intervention number 1001 on the 3rd of July, 2020 and intervention number 1002, on the 4th of July, 2020. During intervention 1001, the product DT101 was replaced (1 piece) and DT 103 was replaced (2 pieces). During intervention 1002, the product DT105 was replaced (3 pieces). These two interventions resulted in a single invoice (invoice “I122”), which was issued on the 5th of July, 2020, and regulated (paid) on the same day. Intervention number 1001 was done by the technician T2, “Dupont Jean”, which hour rate is 15 euros, while intervention 1002 was done by the technician T1, “Hunter Steve”, which hour rate is 18 euros. Intervention number 1001 lasted 1 hour and a half, while intervention 1002 lasted 1 hour.

7. You realize now that there was a mistake in your data. Intervention 1002 lasted 2h, and not 1h. Write the SQL code needed to change the corresponding value in the database.