# Computer Science Exam
# 2nd year - November 2019-2020

INSA

**Duration :** *60mn*
**Authorised documents:** *None.*

— The grading scale is over 20 marks and is given on an indicative basis.
— The exam subject comprises 3 pages.

## Exercise 1 SQL Queries (13 pts)

We consider a database (DB) of a company that exports French products (wine, cheese etc.) to the USA. In this database, we would like to store the sold products, the warehouse where the products are stored and the transporters.

In the corresponding relational schema, the attributes that are keys are underlined. We assume that there are no undefined values ($NULL$). We further consider that the DB is consistent and automatically updated with the position of the products.

— `products(idProduct int(11), name varchar(20), type varchar(20), weight int(10))`
A product is uniquely identified by `idProduct`. We store its Name, for example "comté 18 months AOC", its type, for example wine, cheese, pastry etc., and its weight per unit.
— `warehouses(idWarehouse varchar(20), address varchar(40), telephone varchar(20))`
A company can have several warehouses. If a product is not in transit it is stored in a warehouse. Every warehouse is identified uniquely by `idWarehouse`. Also its address and telephone number is stored.
— `stock(idProduct int(11), idWarehouse varchar(20), numberUnits int(10))`
This relation memorises in which warehouses are stored which products. The attribute `idProduct` is a foreign key referring to the `products` relation. `idWarehouse` is a foreign key referring to the `warehouses` relation. We also store the number of product units that are present in the warehouse.
— `transporter(idTransporter varchar(20), type varchar(40), country varchar(20))`
When a product is in transit, it is with a transporter and identified by `idTransporteur`. We store the transportation type (air, rail, ship etc.) and the country of the headquarters.
— `transit(idTransit int(11), idProduct int(11), idTransporter varchar(20), idFrom varchar(20), idTo varchar(20), sendDate date, numberUnits int(10))`
This relation stores which products are in transit. A transit is uniquely identified by `idTransit`. The attribute `idProduct` is a foreign key referring to the `products` relation. `idTransporteur`, indicating the transporter who is in charge, is a foreign key referring to the `transporter` relation. The attributes `idFrom` and `idTo` are foreign keys referring to the `warehouses` relation and denote the warehouses where the transiting product is send from and to. Also the send date is memorised and the number of units of the product that are in transit. When a product arrives at its destination, it is no longer in transit and the corresponding line is removed from the table.

**Question :** Write the corresponding SQL queries for the following questions. Take care of removing duplicates when there may be any. **The questions are not ordered by difficulty level.**

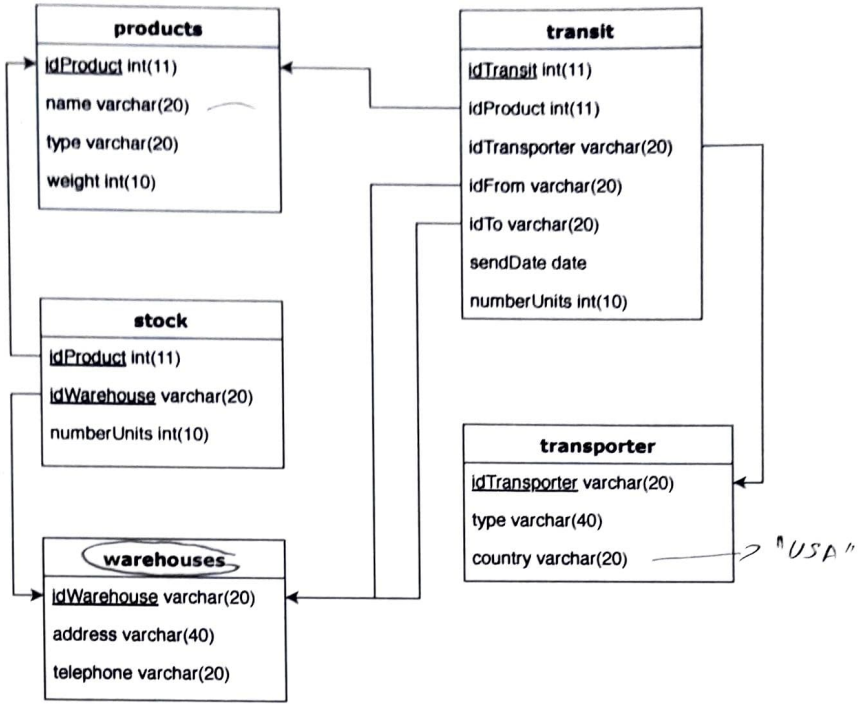


FIGURE 1 – Graphical view of the relational schema of the DB.

1. Which warehouses have an address containing the postal code `69100`? Display all the attributes. *Hint :* Use a condition with the keyword `like`, where % denotes any string of characters.

2. How many American transporters are doing at least one transport? A transporter is considered American if the country of its headquarters is *"USA"*.

3. Give the total number of units of each product stored in the warehouse with the identifier `"MARSEILLE13"`.

4. What are the names of the products of type `"wine"` stored in the warehouse `"CHICAGO11"`?

5. What is the total weight of all products names `'Camembert fermier'` stored in the warehouse `'DALLAS02'`? *Hint :* use the weight of one unit and the number of units in stock.

6. What are the products that are only transported by `"air"` (transporters of type `"air"`)? Display all the attributes of the products.

7. For each product, give the mean number of units in stock.

8. What are the products of type `"fromage"` (cheese) whose number of units in stock is smaller than the mean number of stored units of `"fromage"`? For each cheese, the mean is computed over all warehouses.

9. Which cheese have started their transit at the same date as a cheese named `"Brebis basque"`? *Hint :* This is an self-join (*fr :* "auto-jointure").

10. How many transits does each transporter for each product? Display the identifier of the transporter, the product and the corresponding number of transits.

## Exercise 2 Conceptual and relational model (7 pts)

*(Q2.1) Reverse design (retro-conception)*

**Question :** Using the UML syntax, propose an entity-relationship model (conceptual schema) that produces the preceding relational model, by applying the rules from the lecture and TP sessions.

*(Q2.2) Extension of the entity-relationship (conceptual) model - part 1*

We would like to extend the DB to memorise the organisations that certify the products and which organisation has certified which product. The company stores all certifying organisations, even if they have not certified any product of the company. The stored information is : the identifier (ID) of the organisation, its name, the address of its headquarters and its nationality. The company's policy is to store products only if they have been certified by at least two different organisations.

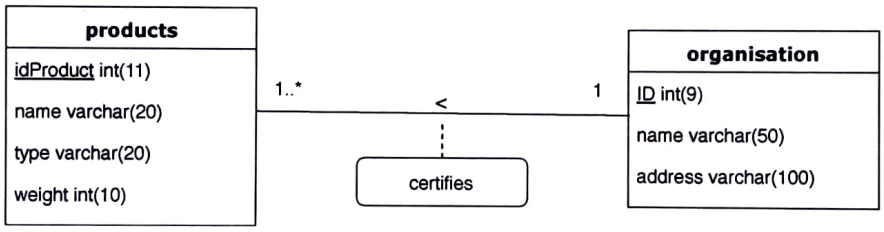


FIGURE 2 – Entity-relationship (conceptual) model in UML syntax ; part on the certification

**Question :** Do you think the schema of Figure 2 is consistent with the preceding information ? If this is not the case, explain why and propose a modification.

*(Q2.3) Extension of the entity-relationship (conceptual) model - part 2*

We would like to extend the DB to store also the company's product purchases. Certain employees of the company are specialised in supply and are called "buyers". A purchase is identified by a purchase number. It is performed by an single buyer at a given date. A buyer can do several purchases. A purchase is performed from a single supplier, but the company can buy from it several times. For each purchase, the supplier and the buyer agree on the bought products, their price per unit and the number of units, for each bought product. Thus, a purchase can be composed of several different products.

**Question :** Propose an entity-relationship model (UML syntax), that takes into account this information. Explain briefly the choice of cardinalities and the underlying hypotheses. **Note : it is not necessary to copy the whole model of the previous question. Only the `"products"` entity is necessary.**

*(Q2.4) Relational model and SQL queries*

**Question : Transform the conceptual model of the previous question (Q2.3) into a relational model. Do not transform the parts related to certification.**