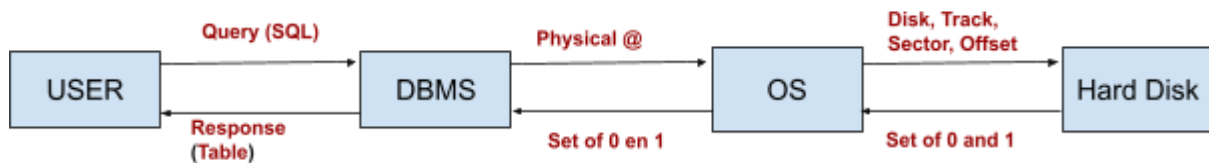


## Correction of the MidTerm (Course)

- The following diagram shows the exchange of information between the user and the database. Complete the diagram with the missing information **(0.25 by item: 1.5 all)**.



- Consider the following database containing two tables  
 Employee (NSS, Last name, First name, Date of birth, Dep)  
 Department (DepID, Name, Manager)

- What does the metabase represent for the DBMS **(0.5)**? Can users modify it **(0.5)**? Why **(0.5)**?

The metabase represents the DBMS memory. All information on tables, constraints, users, resources, etc. are stored in the meta-base.

Users cannot directly modify the metabase. Any modifications are made by the DBMS to update the information.

If users make direct modifications to the meta-base, there will be a risk to the correct operation of the DBMS, as all functionalities are linked to the meta-base.

- What effect do the following two commands have on the DB and metabase?
  - Alter Table Employee Add constraint fk\_Emp\_Dep FOREIGN KEY (Dep) REFERENCES Department(DepID)

Effect on the DB	Add a foreign key constraint on the Dep attribute of the table Employee. The values of this attribute are linked with the values of the attribute DepID on the Department table. <b>(0.5)</b>
Effect on the MB	Add a row on the CONSTRAINT TABLE with all information about the foreign key constraint added : the source table and attribute (Employee, Dep) and the Target table and attribute (Department, DepId), the type of the constraint (FK). <b>(0.5)</b>

- Insert into Department VALUES (14, "Physics", "Pr. Ouaddah")

Effect on the DB	After checking all the values with attribute types, one row is added to the table Department on the disk. <b>(0.5)</b>
Effect on the MB	Increment the number of tuples of the table Department. <b>(0.5)</b>

- Propose an algorithm to check the foreign key constraint when adding a tuple to the Employee table **(1.5)**.

## Correction of the MidTerm (Course)

### Check FK Algorithm

**Input** : Insert into Query

**Output** : Success or Exception

**Begin**

Extract from the query and the metabase all information about the FK constraint:

The Source Table : ST

The source Attribute : SA

The Target Table : TT

The Target Attribute : TA

Extract from the query the value of SA : **VSA**

Execute the query : **select TA from TT where TA=VSA**

If (Response null) then

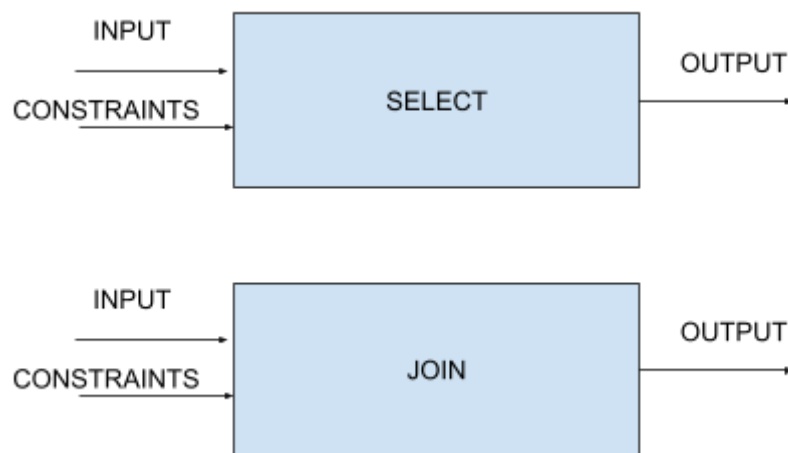
show message (" Impossible to add the tuple, the value"+VSA+"of attribute  
SA, doesn't exist on the table"+TT+ "Please check your Data!");

Generate exception to cancel the insert operation.

Else Add the tuple on the disk.

**End**

3. Let's formalize the two relational operators SELECT and JOIN.



1. Give the inputs, outputs and constraints for each operator

Operator	Input	Output	Constraint
SELECT (0.75)	Table, Condition Attribute theta Value	Subset of tuples of the source Table that satisfy the condition	the condition is defined on an attribute of the source table
JOIN (0.75)	2 Table, Join Condition	New Table with all attributes of the two input tables	The condition is defined on common attributes

## Correction of the MidTerm (Course)

2. Propose an algorithm to implement the selection operator (2).

### **Check SELECT Algorithm**

**Input** : Select Operator

**Output** : Set of tuples of the source table

**Begin**

    Extract from the algebra query :

        The Source Table : ST

        The source Attribute used by the condition : SA

    Extract from the metabase the physical @ of the first tuple of ST

    Send the @ to OS

    //OS send the data to DBMS then to the RAM

    For all tuple returned by the OS do

        If the value of SA satisfy the condition (True) then Add the tuple in the  
        **RESULT**

    If (RESULT not null) then Display the RESULT as a table

    Else show message ("No selected lines").

**End**