Deliverable 2

*Team 19 :*

*Théo Dallet—Thuillier*

*Tom Patrzynski*

*Tom Laine*

*Thibault Yaeche*

Table of contents

[I - Creation of the database and the different tables: 1](#_Toc134964072)

[II - Insertion of the various data in the database: 2](#_Toc134964073)

[III - Creating queries: 2](#_Toc134964074)

[IV - Creation of procedures: 6](#_Toc134964075)

[V - User creation: 7](#_Toc134964076)

# I - Creation of the database and the different tables:

Firstly before creating our database we make a test to check if it is already existing or not and thus to remove it if it already exists, to do this we use the command " DROP IF EXIST ". Then we have to create the database with the command "CREATE DATABASE" and then the command "USE" so that our future requests are made in the right database.

Une image contenant texte, Police, capture d’écran, blanc

Description générée automatiquement

Then we create our tables in the same way as for the database we check if the tables are already existing, then we create the tables with the command "CREATE TABLE" with the name of the table and then the attributes of this table in our case the table "Agencies" is composed by Id\_Age which is an integer then Name\_Age which is a character string of a length of fifty and Id\_Cit which is an integer. Then we have to put the constraint of primary key and foreign key in our case Id\_Age is our primary key because it is unique and will allow us to find all the data that have a relationship with the identifier. For the foreign key it is a constraint that guarantees the referential integrity between two tables.

Une image contenant texte, Police, capture d’écran, nombre

Description générée automatiquement

# II - Insertion of the various data in the database:

After creating the different tables we need to add all the values of the database for this we use the command INSERT INTO followed by the attributes that make up the table to fill, then we insert the value through the command VALUES followed by the data that will be entered between brackets for each line.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

# III - Creating queries:

1)

For the first query we had to list all the agencies to do this we select all the information from the agencies table

Une image contenant texte, Police, capture d’écran, blanc

Description générée automatiquement

2)

For the second one we had to list all the technical staff of the agency of Bordeaux to do this we will have to make a restriction on the name of the cities and on the work of the agents but before we must join the personal table with agencies thanks to the Id\_Age and then cities and the previous join with the Id\_Cit.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

3)

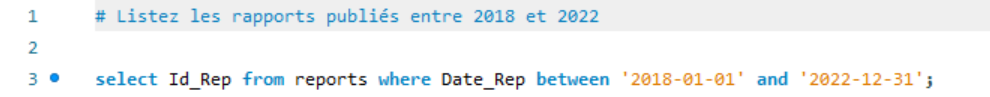
For the third we had to give the total number of sensors deployed to do this we use the "count()" function.

Une image contenant texte, Police, ligne, capture d’écran

Description générée automatiquement

4)

For the fourth one we had to list the reports published between 2018 and 2022 to do this we used the "between" function to check if the date is between 2018 and 2022 :



5)

For the fifth one we had to calculate the total emission of greenhouse gases by region in 2020, we used the function "sum()" that makes us a sum of several things.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

6)

For the sixth we have displayed the most polluting sector of activity in Ile de France to do this we used the function "max()" which as its name indicates gives us the maximum of several values.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

7)

For the seventh we had to classify the reports concerning NH3 emissions by chronological order we used the function "order by asc" what puts us in order our dates in our case by chronological order thanks to the "asc" if we wanted to have it in the other direction it would have been necessary to put "desc" instead of "asc".

Une image contenant texte, Police, capture d’écran, algèbre

Description générée automatiquement

8)

For the eighth we had to give the names of the technical agents now of the sensors concerning the acidifying pollutants for this we will use the table gases to restrict to the acidifying pollutants which are written GRA on our table, we have another restriction to make on the technical agents.

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

9)

For the ninth we must for each gas, give the sum of its emissions (in tons) in the region Ile-de-France in 2020 so we take that the year 2020 with the function "year()" then make the sum of emissions.

Une image contenant texte, Police, nombre, capture d’écran

Description générée automatiquement

10)

For the tenth we have to give the productivity rate of the administrative agents of the agency of Toulouse (according to the number of written reports and their seniority in the position) to do it we are going to take the maximum of report written by an administrative agent then to divide it by the number of year since he was hired.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

11)

For the eleventh for a given gas, list the reports that contain data about it (we must be able to give the name of the gas as a parameter) to do this we use a procedure with a parameter that will be the requested gas.

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

12)

For the twelfth we had to list the regions in which there are less sensors than agencies to do this we just compare the number of agencies and the number of sensors per region using the "count ()" function :

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

# IV - Creation of procedures:

Here we have the creation of procedures that will allow to add, modify and delete information from several tables. These procedures are sets of precompiled SQL statements, stored in a database and executed, they look like functions that can be used in a main program.

The first procedure allows you to add an agent to the personal table, the procedures are created with the CREATE PROCEDURE command and then with the BEGIN statement, it will end with the END command. Within this we can add the desired instructions as in the queries. To finish, you can use the CALL statement to call the procedure.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Here is a procedure to modify an agent in the personal table:

Une image contenant texte, capture d’écran, Police, algèbre

Description générée automatiquement

Finally a procedure to delete an agent in the personal table:

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

# V - User creation:

In order to secure the database we decided to create a user who could not modify the table but only look at the data, to do this we start by creating our user with the command "create user" and then put the where it will connect then with the command "identified by" then the password to define it. Then we have to give the permissions to do this we use the command "grant select on" the 'name of the database.the name of the table' "to" the user. Finally we have to use the command "flush privileges" to validate definitively the previous command :

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

We proceed in the same way to give privileges to another person but this time we only give permission to access the report and collect table.

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement