

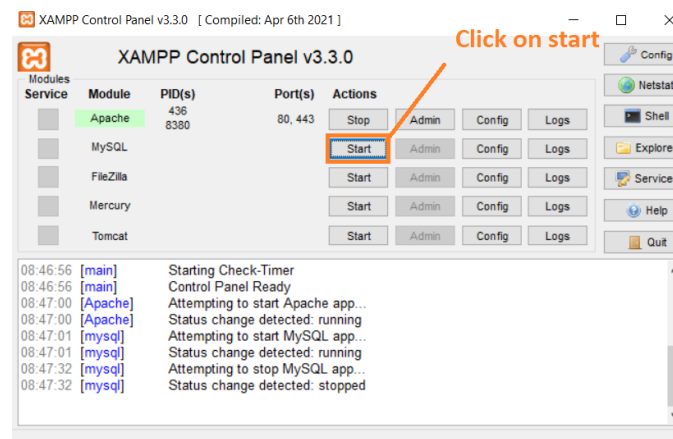
NLIDB Project Documentation

I. Project Installation Requirements

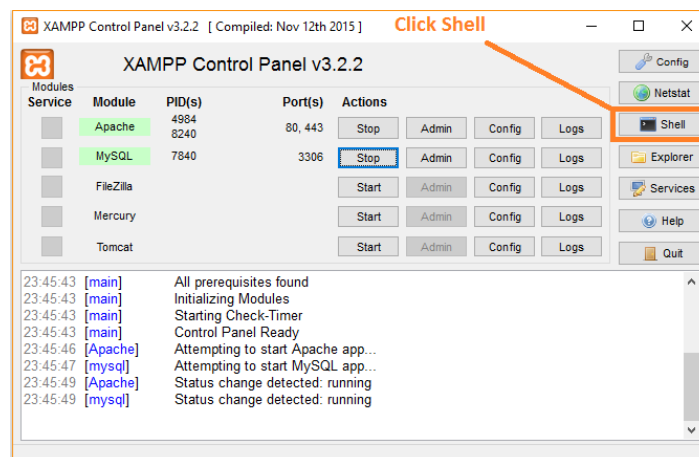
- **XAMPP** : for Apache and MySQL database hosting
- **Eclipse** : Development environment for compiling Java application

II. Configuring the project environment

- 1) Start the XAMPP database server and the Apache server :



- 2) Open the shell command line interface on XAMPP :



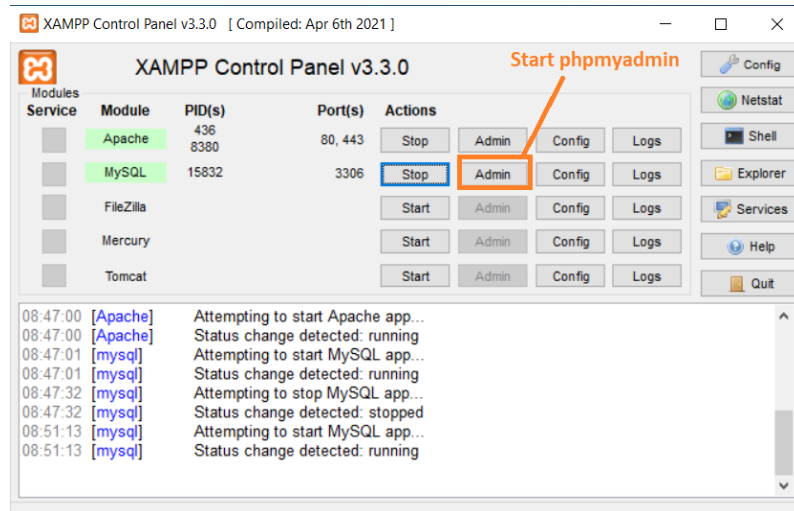
- 3) Create a new database named « mas » on MySQL using the following command :

```
mysql --user=root --password="" -e "CREATE DATABASE mas"
```

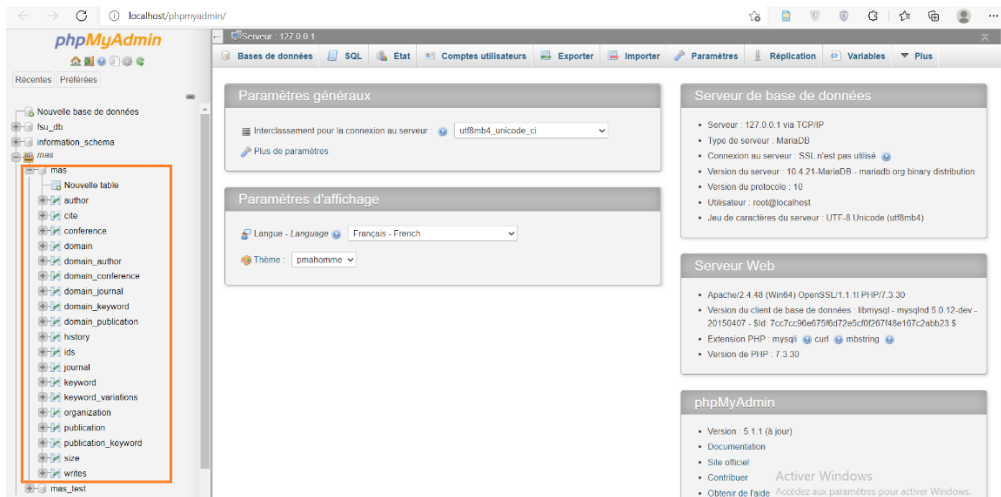
- 4) Then enter the following command to import the MAS database dataset (file : mas.sql, size : 3.3 Giga) to MySQL database with the path of the « mas.sql » file (loading the database in this step can take up to 25-30 minutes) :

```
mysql -u root -p mas < "/path/mas.sql"
```

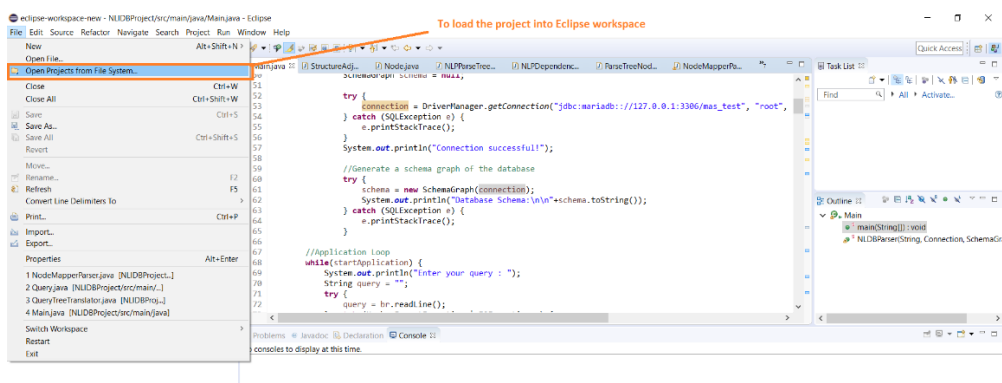
- 5) Verify if the database has been successfully imported by starting « phpmyadmin » :



- 6) Once the « phpmyadmin » interface opened, the « mas » database should contains these tables :

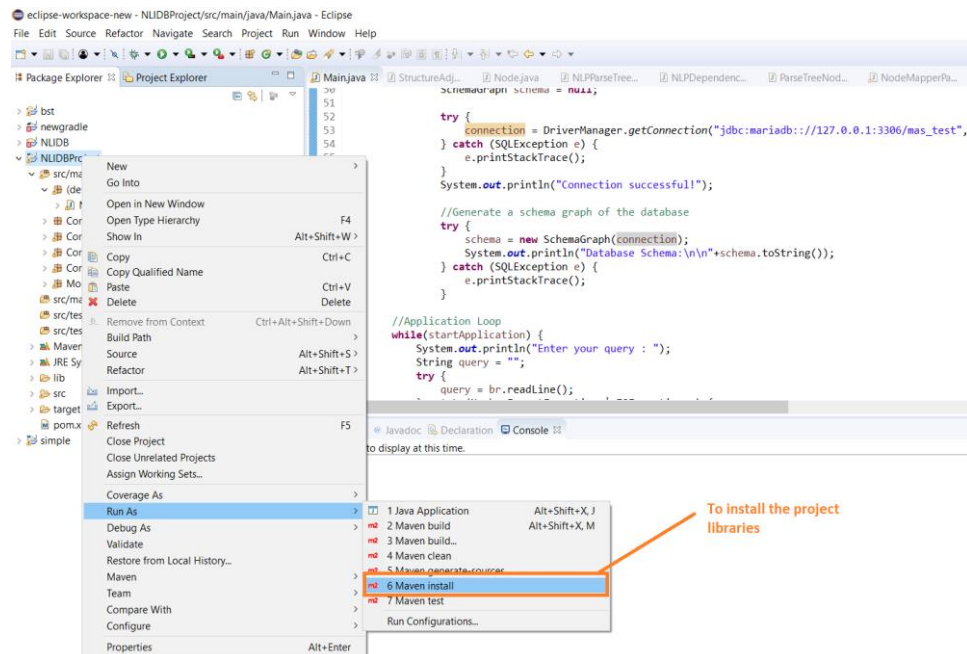


- 7) Then to import the project source code into « Eclipse », click on « File » and « Open Projects from File systems » and select the project folder called “NLIDBProject” :



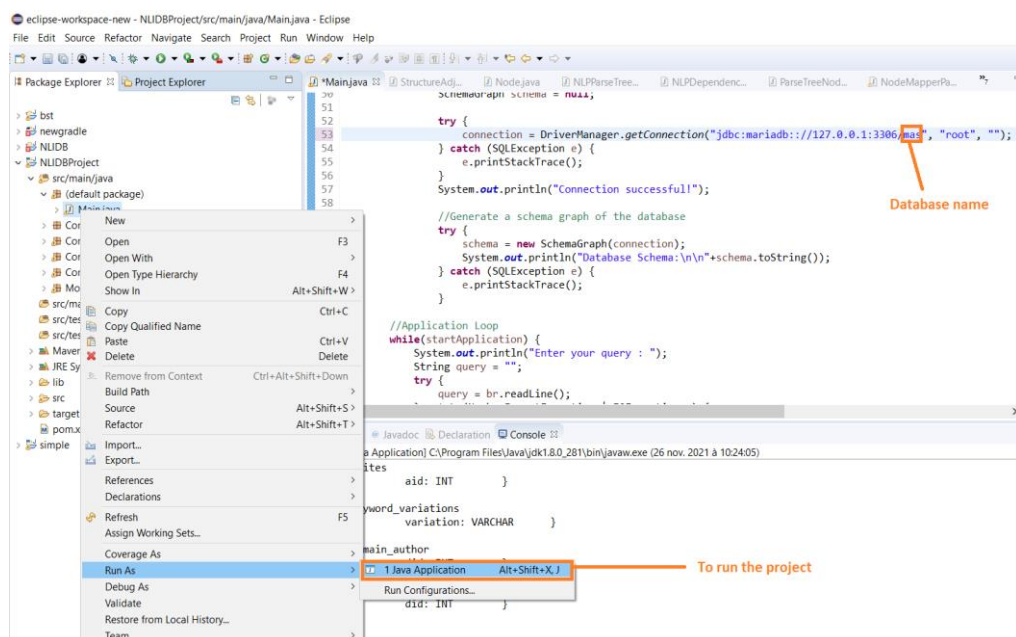
III. Running the NLIDB project application

- 1) Before running the project, verify if the project libraries have been properly imported in « Maven dependencies », if not imported right click on the project and select « Run as » and « Maven Install » :

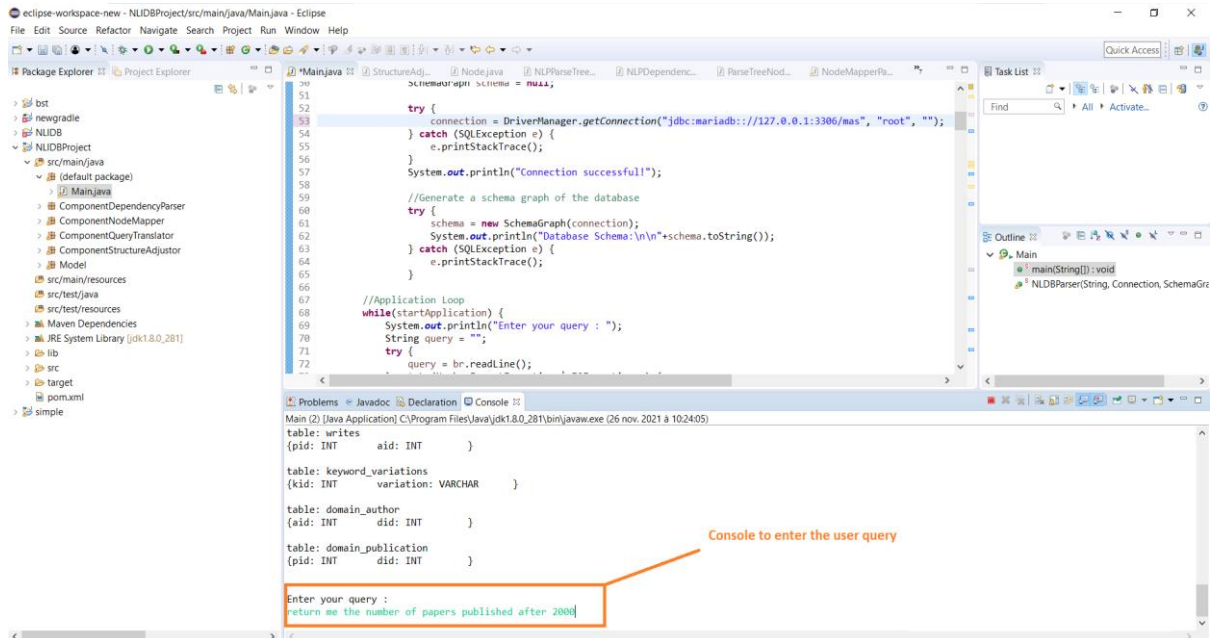


Note : Make sure before running the project that the JDK 1.8 is installed and configured in eclipse.

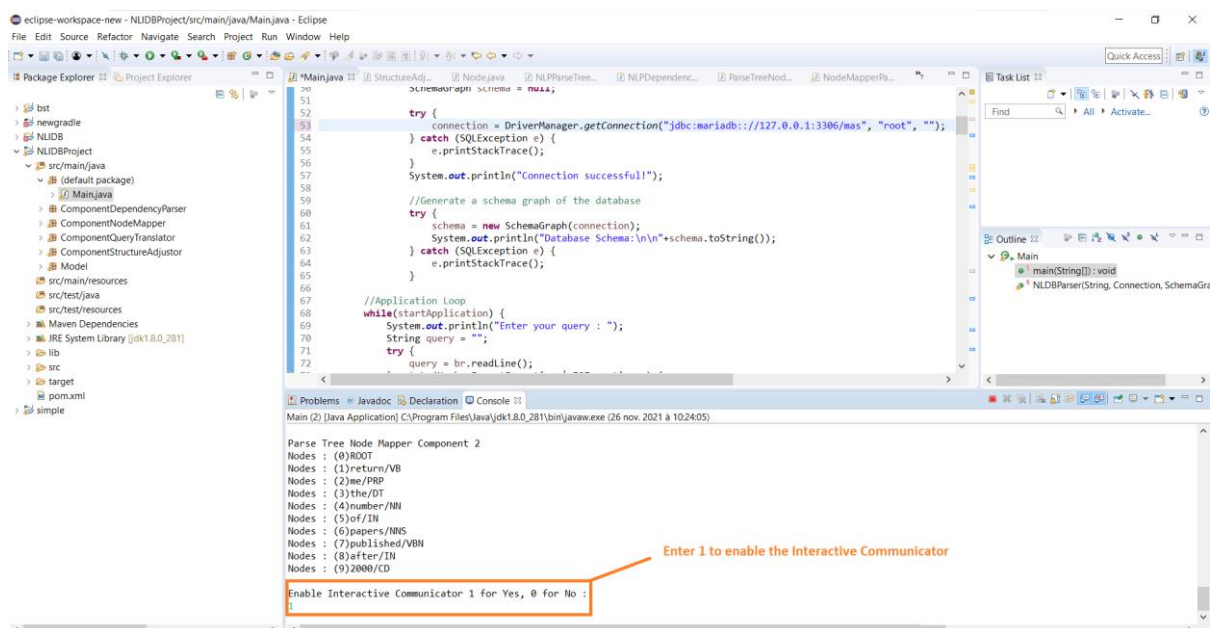
- 2) To run the NLIDB program, go to the project folder and open the « default package » and right click on the class « Main.java » (note : the database name can be changed on line 53 of the Main.java class, « jdbc:mariadb://127.0.0.1:3306/database_name »)



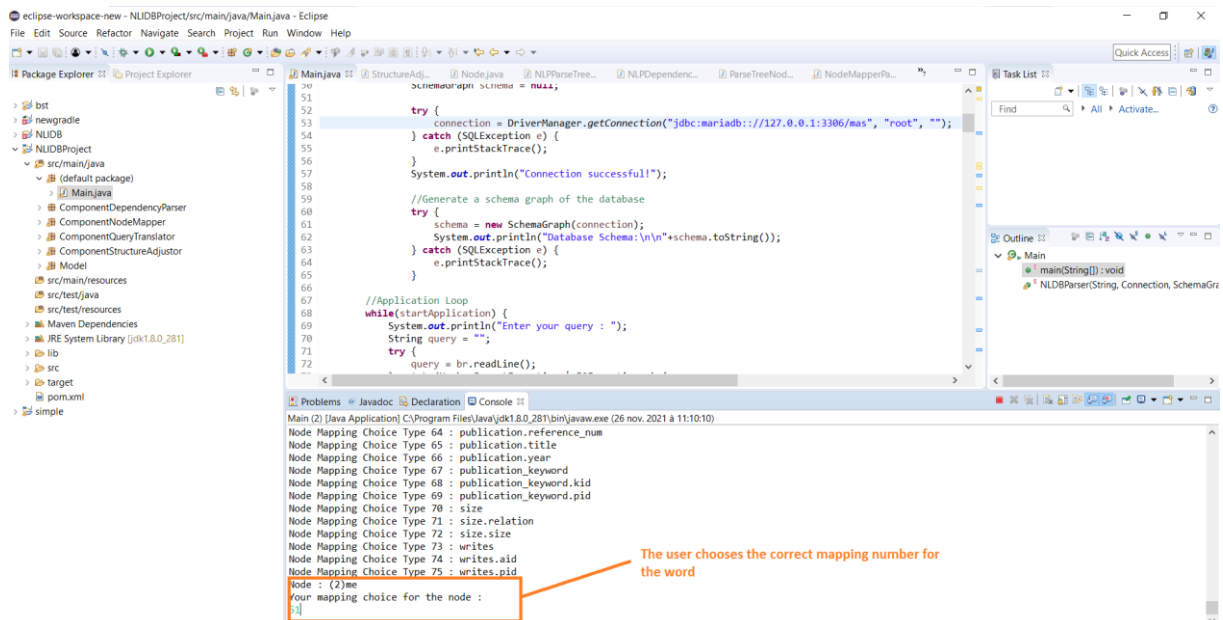
- 3) Once the database schema loaded (can time up to 10 minutes since the database size is approximately 3.3 Giga), the user enter the query in natural language :



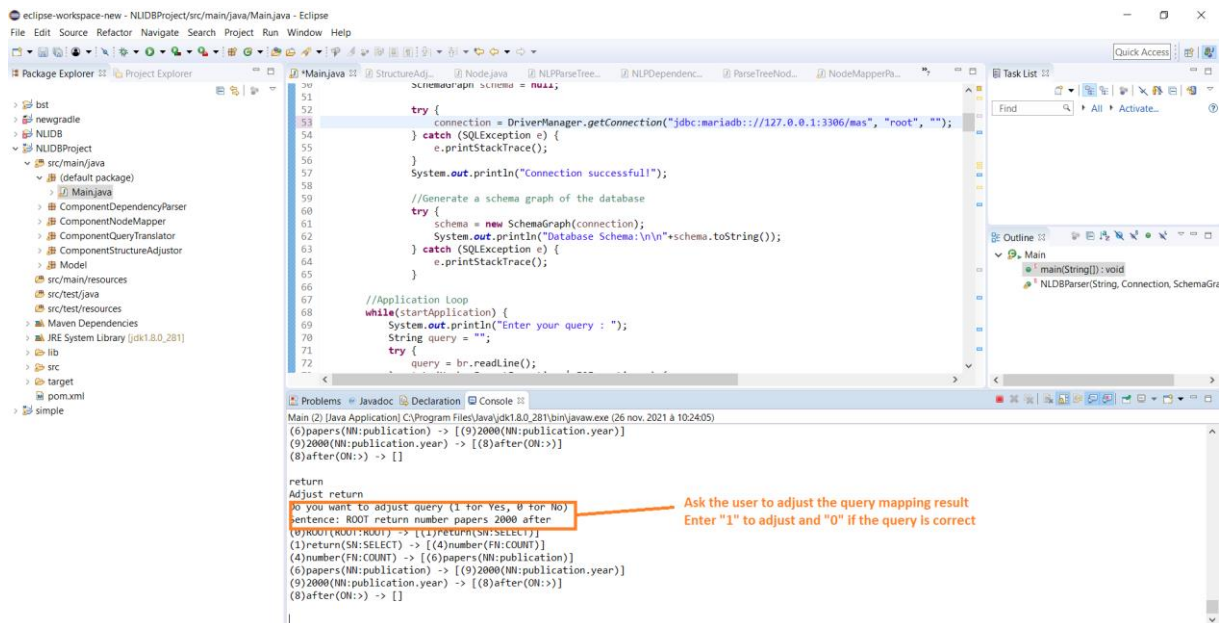
- 4) Then the user chooses if he wants to enable the « Interactive Communicator » (which is recommended based on the project architecture) by entering « 1 » to enable it :



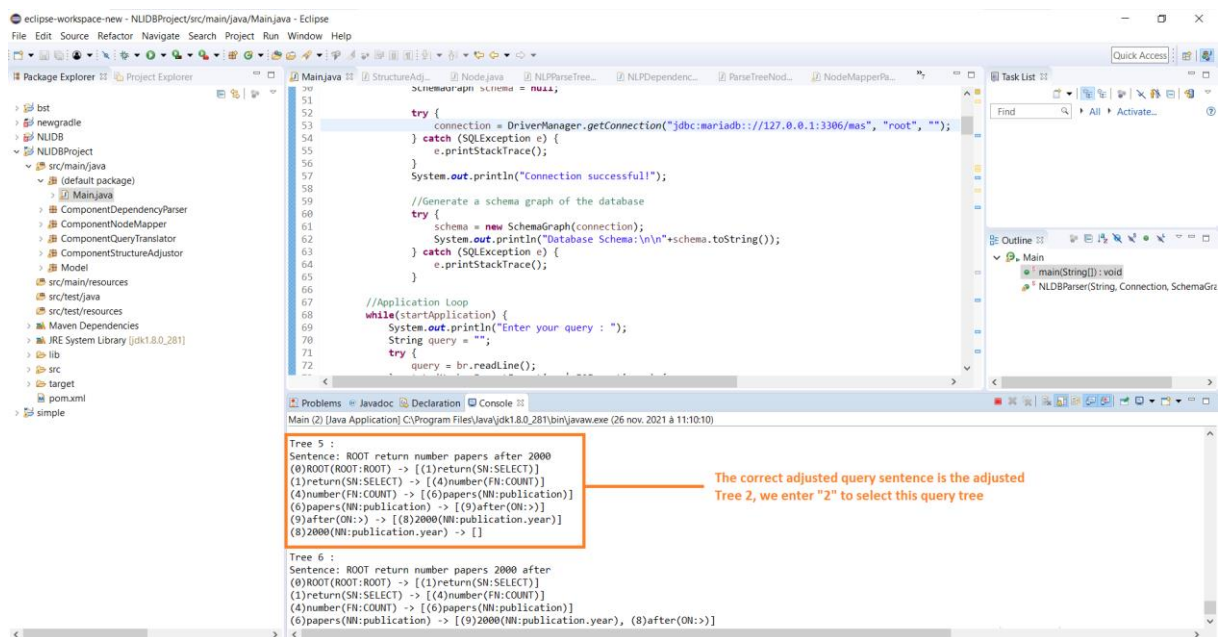
- 5) The user then chooses the mapping of each word with that require mapping with the database schema objects (table name or attribute), for exemple here the word « **me** » is a meaningless node, the correct mapping for this word is the choice « meaningless » which refers to number « **51** » in the list of mapping choices :



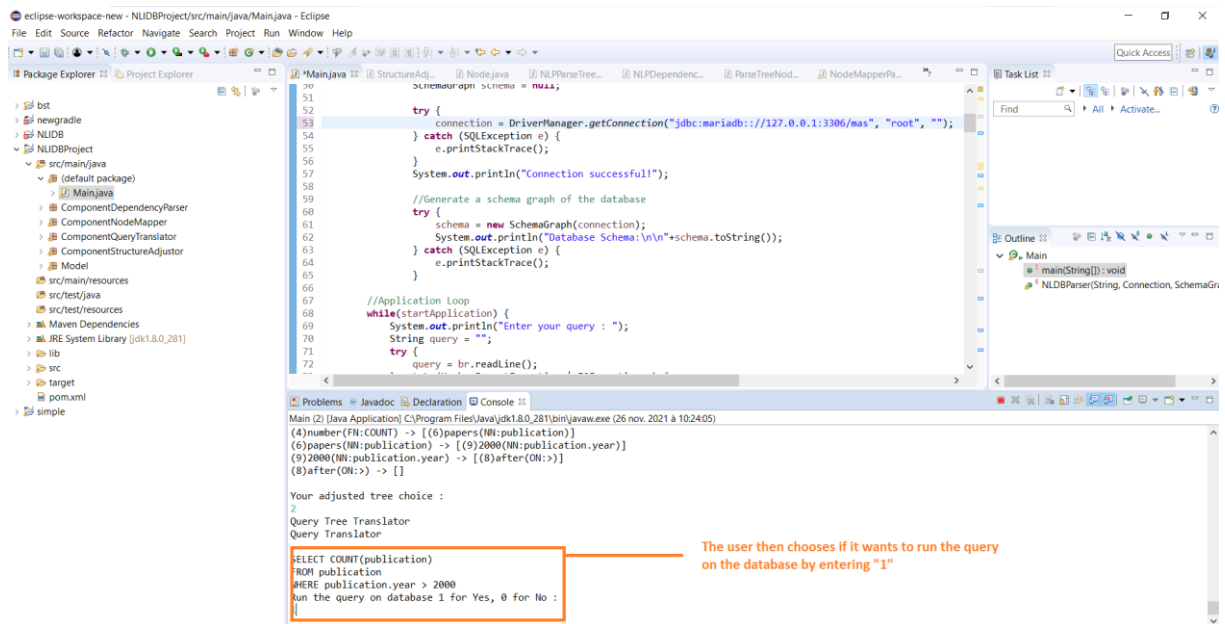
- 6) The same process is then repeated for each word (words « **the** », « **of** » are mapped with the number « **51** », the word « **papers** » is the number « **57** » which refer to « **publication** », the word « **published** » correspond to the number « **51** » and finally the word « **2000** » correspond to the « **publication.year** » which is the choice number « **66** »).
- 7) The parse tree mapping result sentence is then returned, the user chooses if it wants to adjust the query mapping result, here we enter « **1** » since the sentence is partially incorrect :



8) A list of adjusted sentences are returned, the user selects the correct adjusted query sentence number, here we selected the adjusted « Tree 5 » by entering the number « 5 » :



9) Then the user chooses if it wants to run the query on the database and return the results, by entering « 1 » :



10) The generated SQL query and the query results are then returned to the user.

