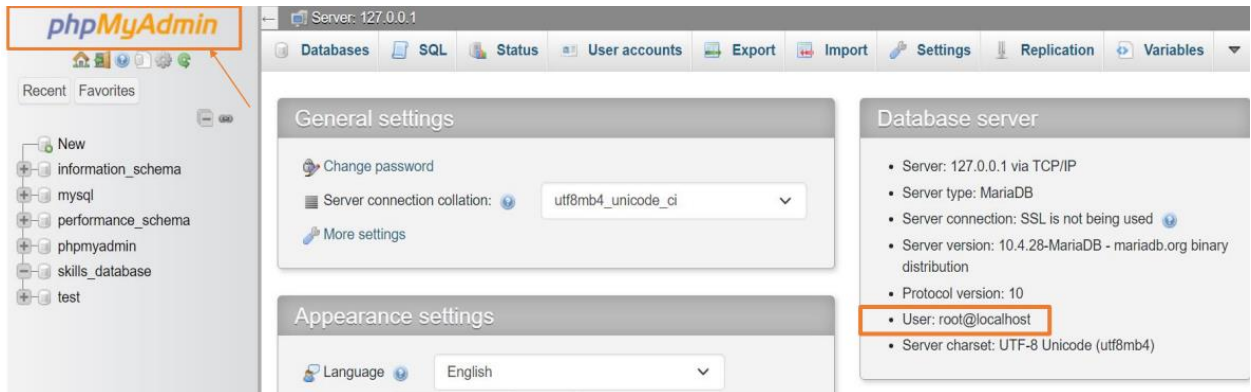


# Developer's Guide

## 1- Application used to create the dataset:

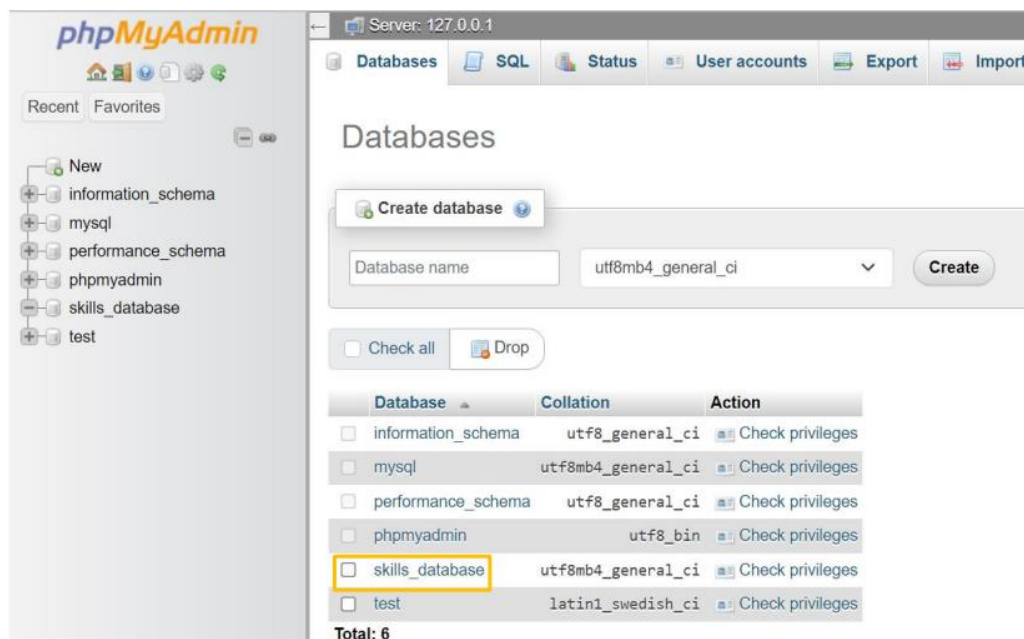
- PHPMyAdmin was used to build the dataset as we were able to create the tables there and connect them together.
- By clicking on the logo of PHPMyAdmin following page appears.



- To determine the username and host name, it appears on the database server box on the user line where the user name is what appears before the @ sign which is root and the host name is what's after the @ sign which is localhost.

## 2- Creating database:

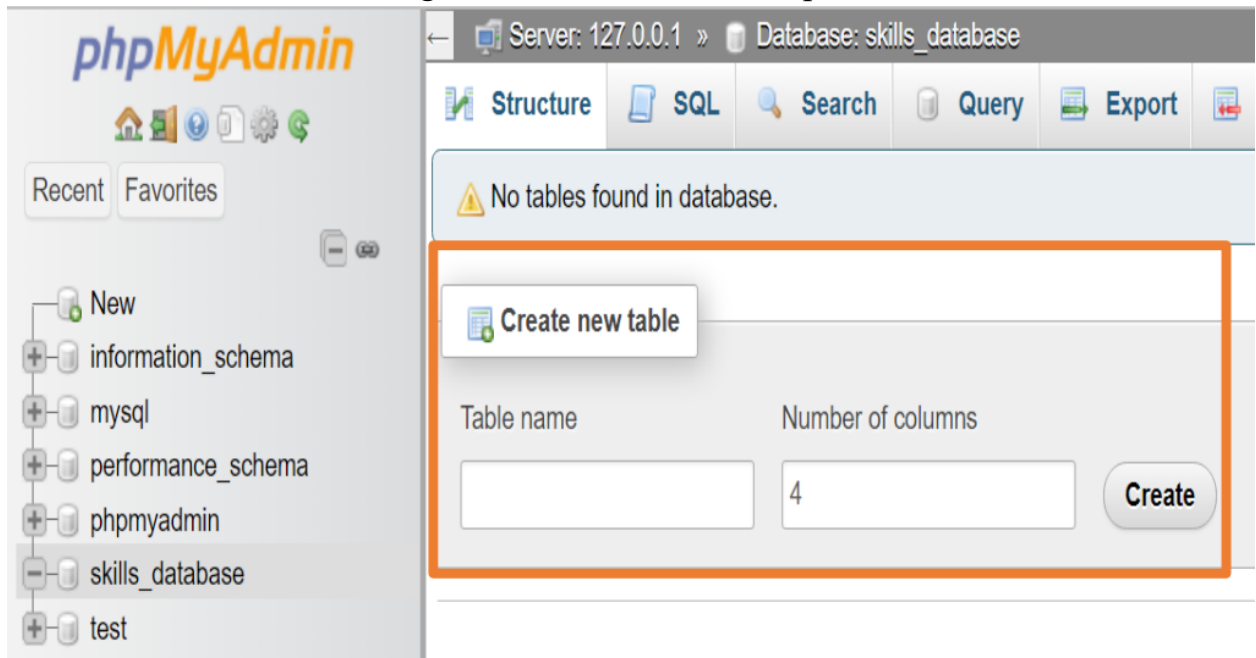
- By clicking on the database tab, we type the database name and click create. Here our created database is called **skills\_database**.



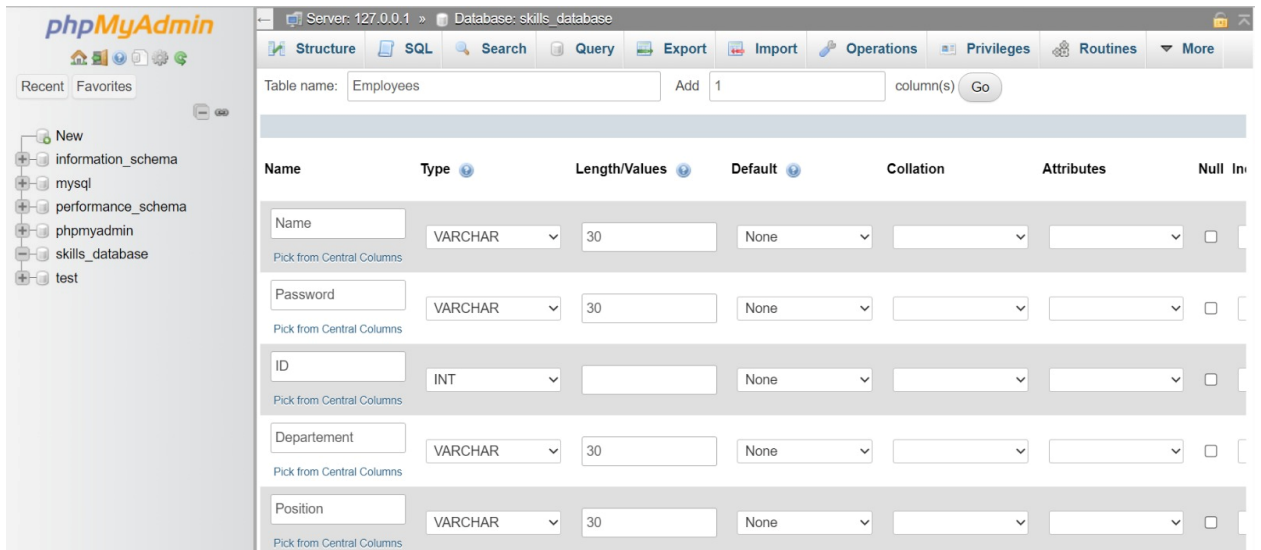
3-

### 3- Creating dataset:

- Here we can create the tables in this dataset by setting the table name and choosing the number of columns per table.

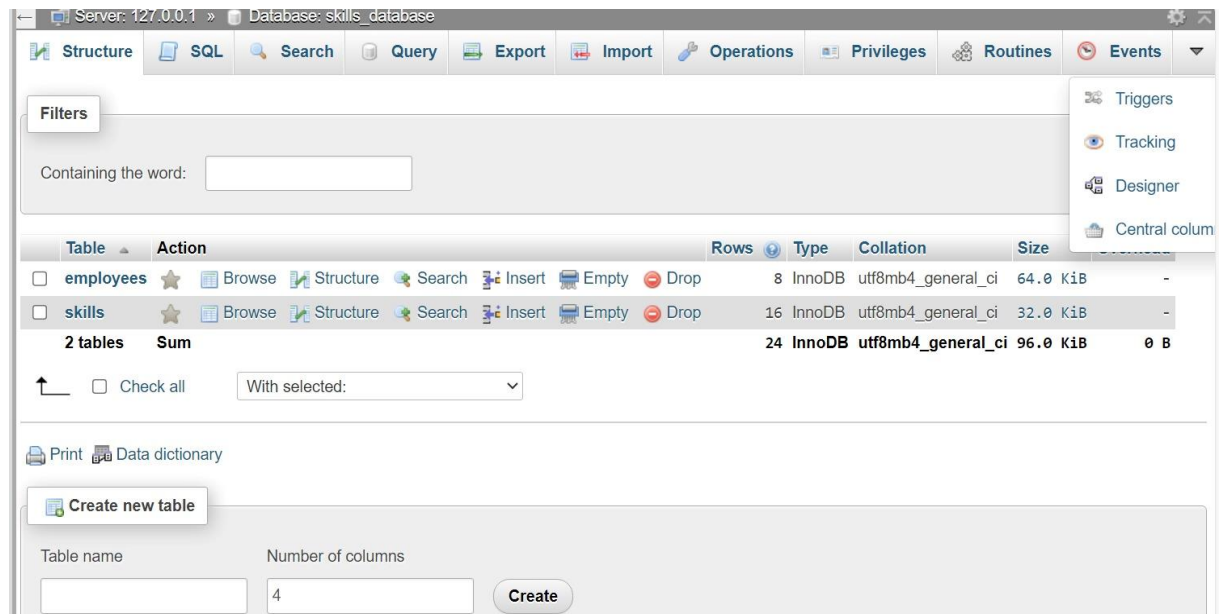


- After pressing create, the following page appears where we write the name of the heading of each column and assign the data type to be inserted in each column.



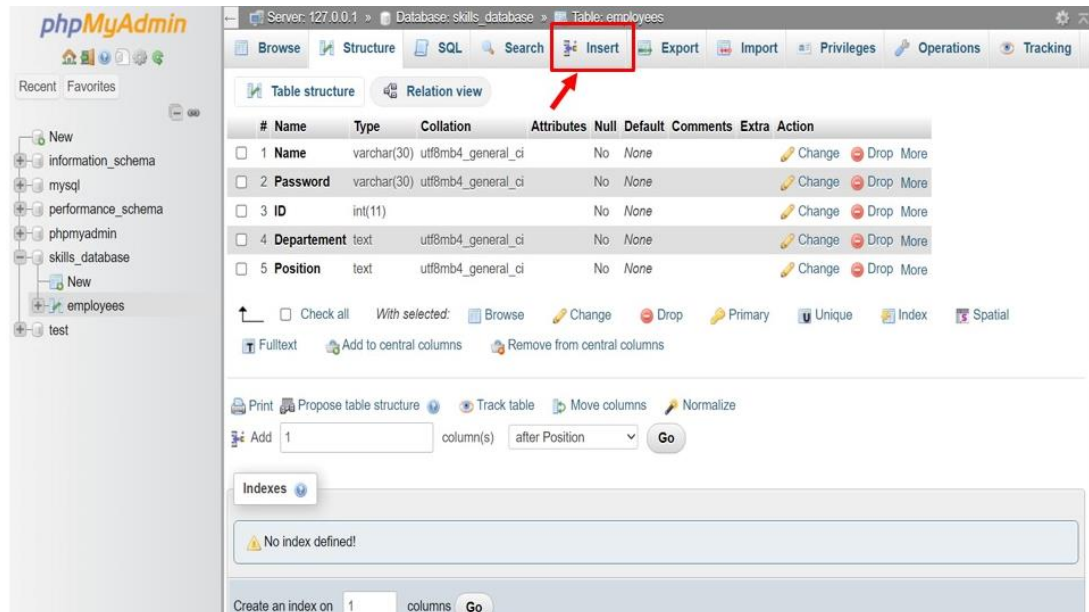
- After finishing we click GO.

- Here we created two tables (2 Entities), and we will connect them together.



#### 4- Inserting data in the dataset:

- Here appears the columns we just created, to fill these columns we click on insert and start filling the data.



- Here's a snippet from inserting the data.

The screenshot shows the phpMyAdmin interface for the 'skills\_database' database, specifically the 'employees' table. The left sidebar shows the database structure with 'skills\_database' expanded and 'employees' selected. The main area displays the table structure with columns: Name (varchar(30)), Password (varchar(30)), ID (int(11)), and Departement (text). Below the structure, there is an 'Insert' form with input fields for each column. The 'Name' field contains 'Maisara Akef', the 'Password' field contains 'maisaraseitech7', and the 'ID' field contains '00012'. The 'Departement' field is empty. A large text area labeled 'Testing' is also visible.

Column	Type	Function	Null	Value
Name	varchar(30)			Maisara Akef
Password	varchar(30)			maisaraseitech7
ID	int(11)			00012
Departement	text			

- After inserting the data, the query appears like this.

The screenshot shows the phpMyAdmin interface after inserting data. A green banner at the top indicates '2 rows inserted.' Below this, the SQL query is displayed in a text area. The query is an INSERT statement into the 'employees' table, specifying columns 'Name', 'Password', 'ID', 'Departement', and 'Position'. The values provided are ('Maisara Akef', 'maisaraseitech7', '00012', 'hela', 'junior') and ('Nardin Radany', 'nardinseitech1', '00014', 'hela', 'team leader'). The interface also shows options to 'Edit inline', 'Edit', or 'Create PHP code' for the query.

```

INSERT INTO `employees` (`Name`, `Password`, `ID`, `Departement`, `Position`) VALUES ('Maisara Akef', 'maisaraseitech7', '00012', 'hela', 'junior'), ('Nardin Radany', 'nardinseitech1', '00014', 'hela', 'team leader');

```

## 5- Creating a Primary key for each entity:

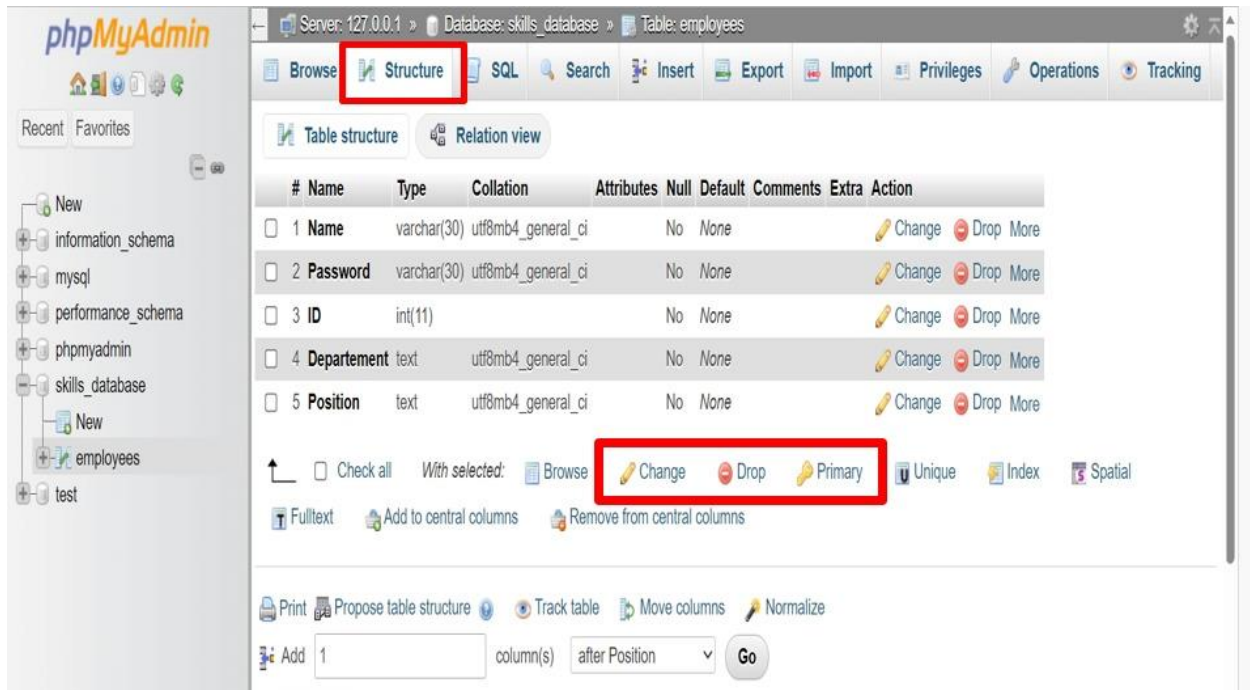
a- Click on our created database which is employee in our case then open the structure tab.

b- 3 main options are helpful which are:

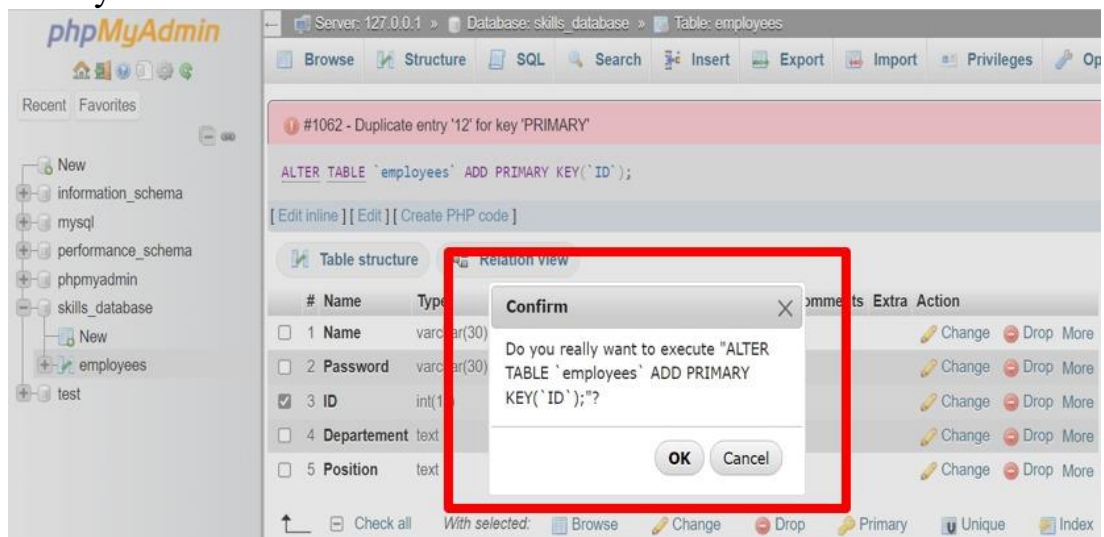
Change: to change in the table.

Drop: to delete data or the whole database.

Primary: to create the primary key.

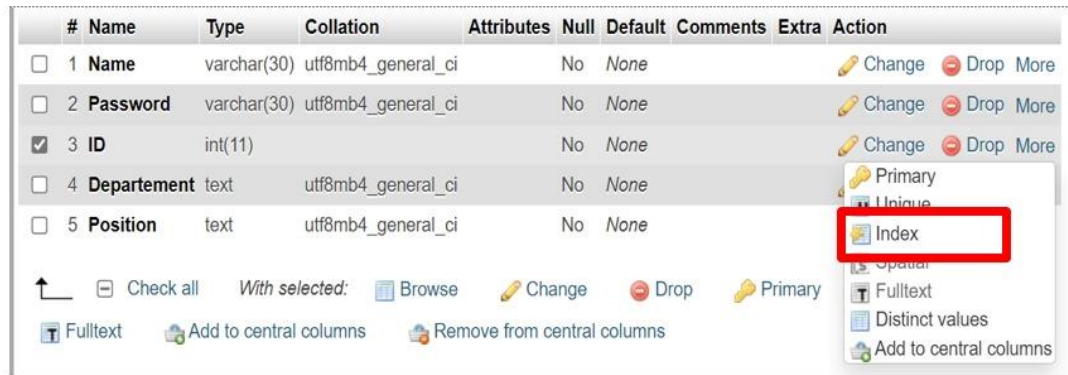


c- The thing that identifies each member is the ID, select it, in the same row select more, click on primary, and then agree that it is the primary key.



## 6- Creating a Foreign key for each entity:

- First we choose an attribute which is one column from the created columns in the database, in our case we choose the **ID** attribute.
- Select it and on the same row click more then choose index.

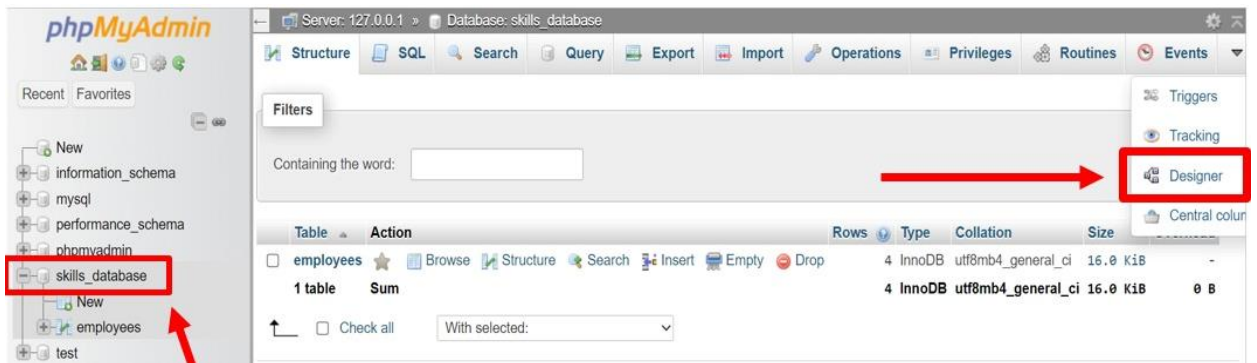


- We need to do the same thing in the other entity which is SKILLS. By repeating the same steps and setting index for the ID column there.

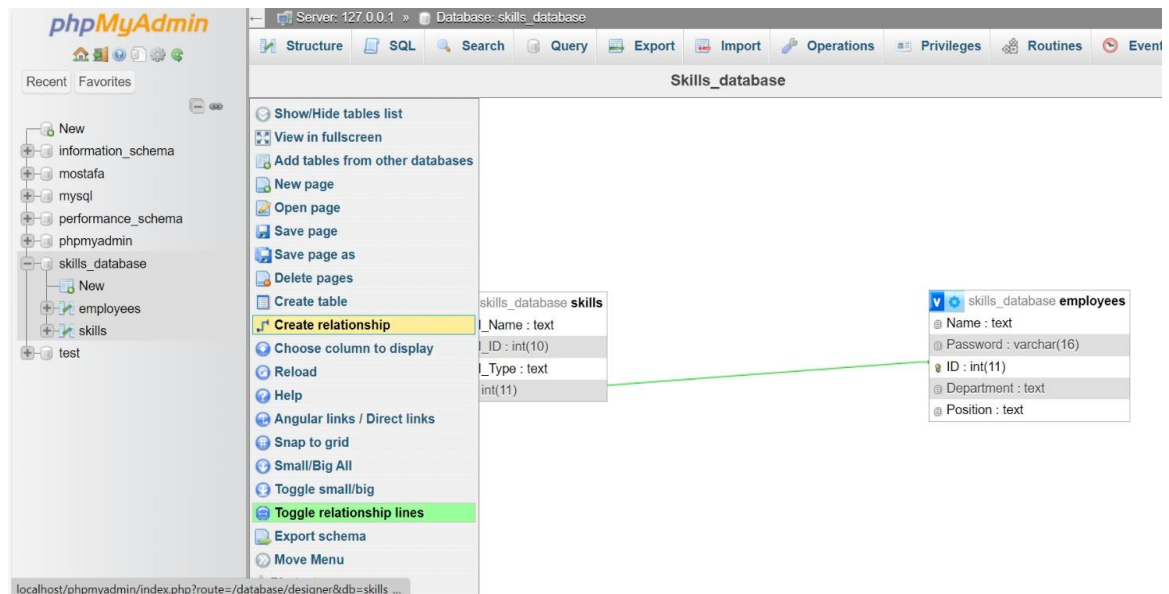


## 7- Creating ERD:

- Getting back to our database, and choosing the designer option from the arrow.



- Then our two entities (two tables) appear.
- Then choose create relationship, then we are asked to select the foreign key which is the **employee ID**.



## 8- Exporting the SQL file:

The screenshot shows the phpMyAdmin interface. The top navigation bar includes links for Browse, Structure, SQL, Search, Insert, **Export** (highlighted with a red box), Import, Privileges, Operations, and Tracking. The left sidebar shows a tree view of databases and tables, with 'employees' selected under the 'skills\_database'.

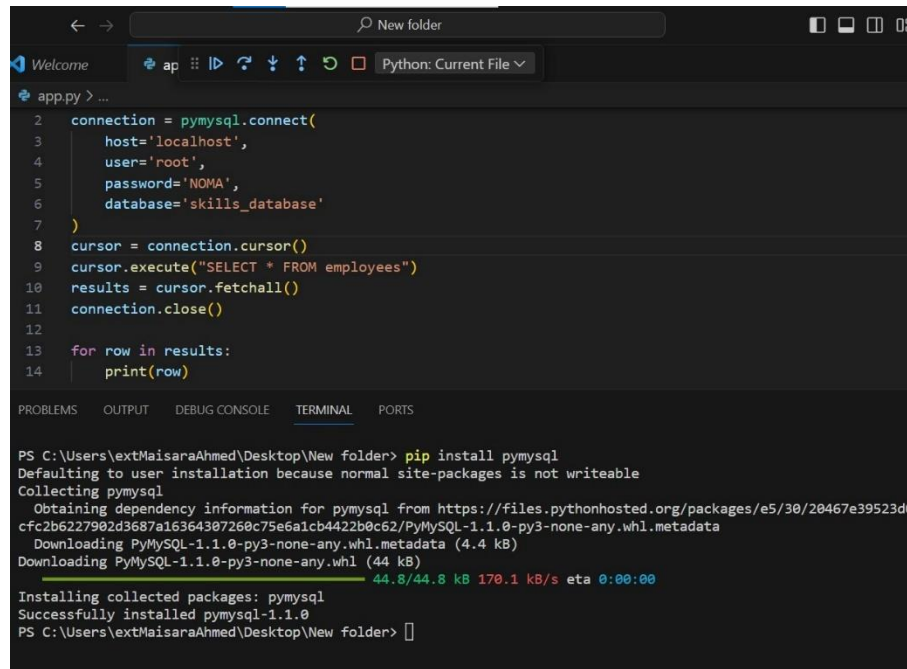
The main content area is titled "Exporting rows from 'employees' table". It contains the following sections:

- Export templates:**
  - New template:** A form with a "Template name" input field, a "Create" button, and a "Template:" dropdown menu.
  - Existing templates:** A section with a dropdown menu labeled "-- Select a template --", and "Update" and "Delete" buttons.
- Export method:**
  - ☒ Quick - display only the minimal options
  - ☐ Custom - display all possible options
- Format:** A dropdown menu showing "SQL".
- Rows:**
  - ☒ Dump all rows
  - ☐ Dump some row(s)
  - Number of rows:
  - Row to begin at:

At the bottom of the form is an "Export" button.



- PHPMyAdmin is used as it can be connected easily with python as the rest of the project is built with python where:
  - 1- First we have to install pymysql using (`pip install pymysql`)

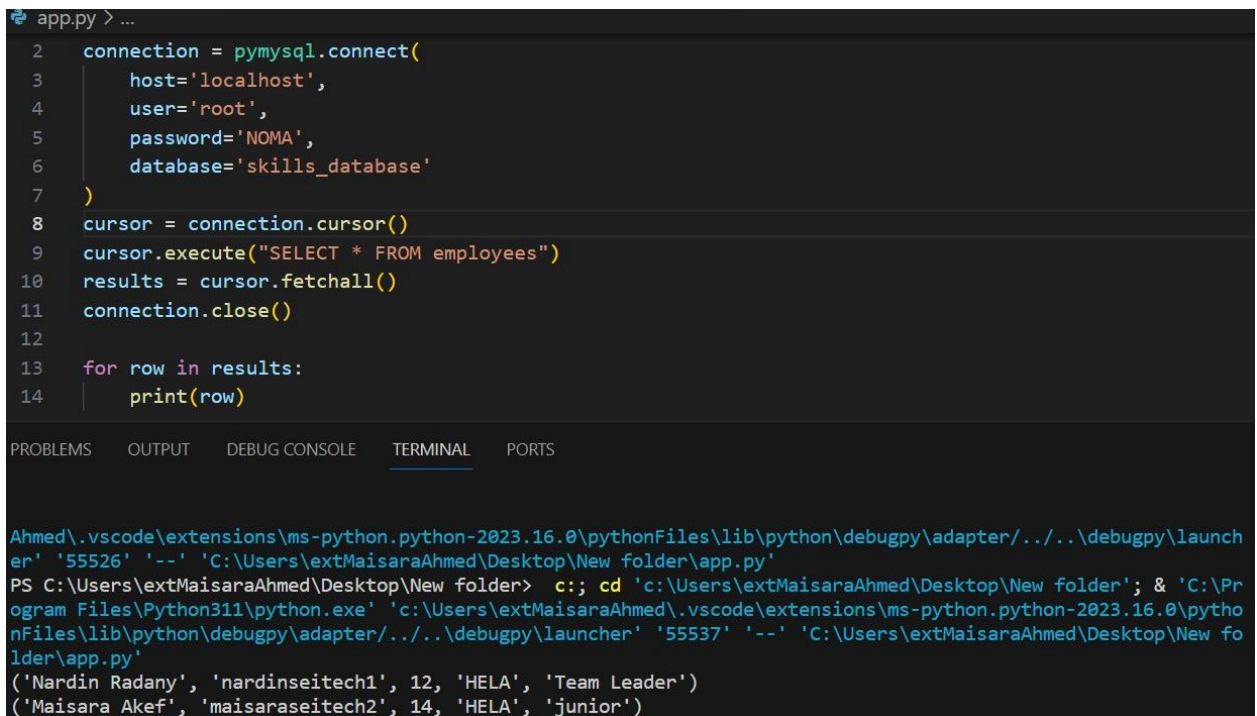


The screenshot shows a VS Code editor with a Python file named `app.py`. The script connects to a MySQL database named `skills_database` on `localhost` using the `root` user and password `NOMA`. It executes a `SELECT * FROM employees` query and prints the results. The terminal output shows the successful installation of `pymysql` using `pip install pymysql`.

```
app.py > ...
2 connection = pymysql.connect(
3     host='localhost',
4     user='root',
5     password='NOMA',
6     database='skills_database'
7 )
8 cursor = connection.cursor()
9 cursor.execute("SELECT * FROM employees")
10 results = cursor.fetchall()
11 connection.close()
12
13 for row in results:
14     print(row)
```

```
PS C:\Users\extMaisaraAhmed\Desktop\New folder> pip install pymysql
Defaulting to user installation because normal site-packages is not writeable
Collecting pymysql
  Obtaining dependency information for pymysql from https://files.pythonhosted.org/packages/e5/30/20467e39523d0cfc2b6227902d3687a16364307260c75e6a1cb4422b0c62/PyMySQL-1.1.0-py3-none-any.whl.metadata
  Downloading PyMySQL-1.1.0-py3-none-any.whl.metadata (4.4 kB)
  Downloading PyMySQL-1.1.0-py3-none-any.whl (44 kB)
    44.8/44.8 kB 170.1 kB/s eta 0:00:00
Installing collected packages: pymysql
Successfully installed pymysql-1.1.0
PS C:\Users\extMaisaraAhmed\Desktop\New folder>
```

- 2- second we print the result which is the content of the table to make sure the connection is done.



The screenshot shows the same VS Code editor with the `app.py` file. The terminal output now shows the execution of the script, which prints the contents of the `employees` table. The output is as follows:

```
app.py > ...
2 connection = pymysql.connect(
3     host='localhost',
4     user='root',
5     password='NOMA',
6     database='skills_database'
7 )
8 cursor = connection.cursor()
9 cursor.execute("SELECT * FROM employees")
10 results = cursor.fetchall()
11 connection.close()
12
13 for row in results:
14     print(row)
```

```
Ahmed\.vscode\extensions\ms-python.python-2023.16.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher
er' '55526' '--' 'C:\Users\extMaisaraAhmed\Desktop\New folder\app.py'
PS C:\Users\extMaisaraAhmed\Desktop\New folder> c:; cd 'c:\Users\extMaisaraAhmed\Desktop\New folder'; & 'C:\Pr
ogram Files\Python311\python.exe' 'c:\Users\extMaisaraAhmed\.vscode\extensions\ms-python.python-2023.16.0\pytho
nFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '55537' '--' 'C:\Users\extMaisaraAhmed\Desktop\New fo
lder\app.py'
('Nardin Radany', 'nardinseitech1', 12, 'HELA', 'Team Leader')
('Maisara Akef', 'maisaraseitech2', 14, 'HELA', 'junior')
```

References: