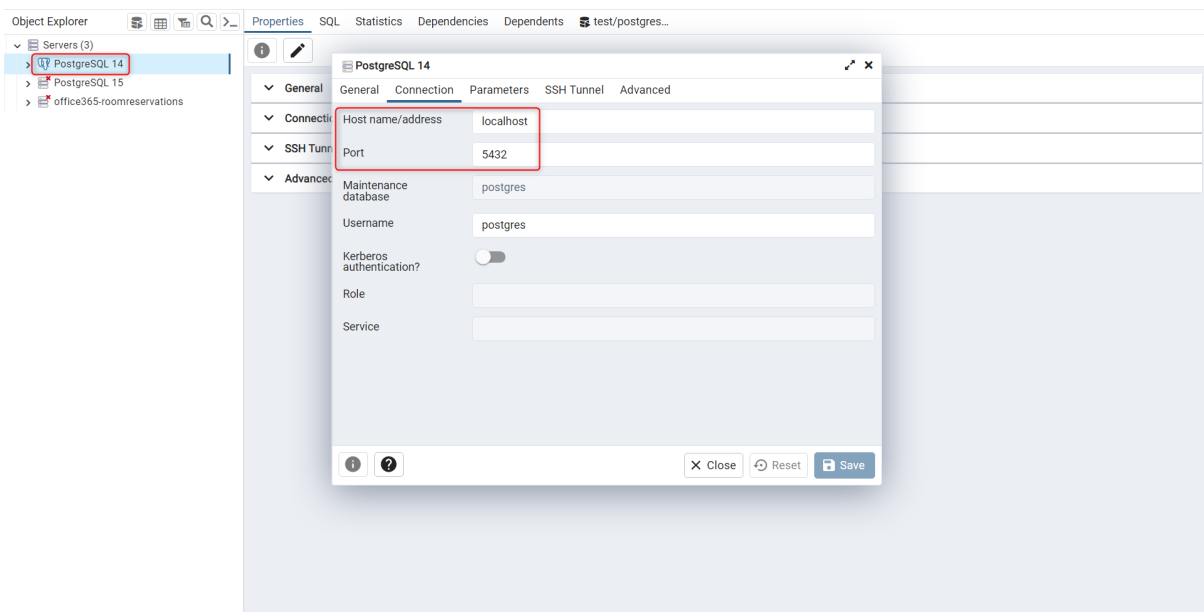


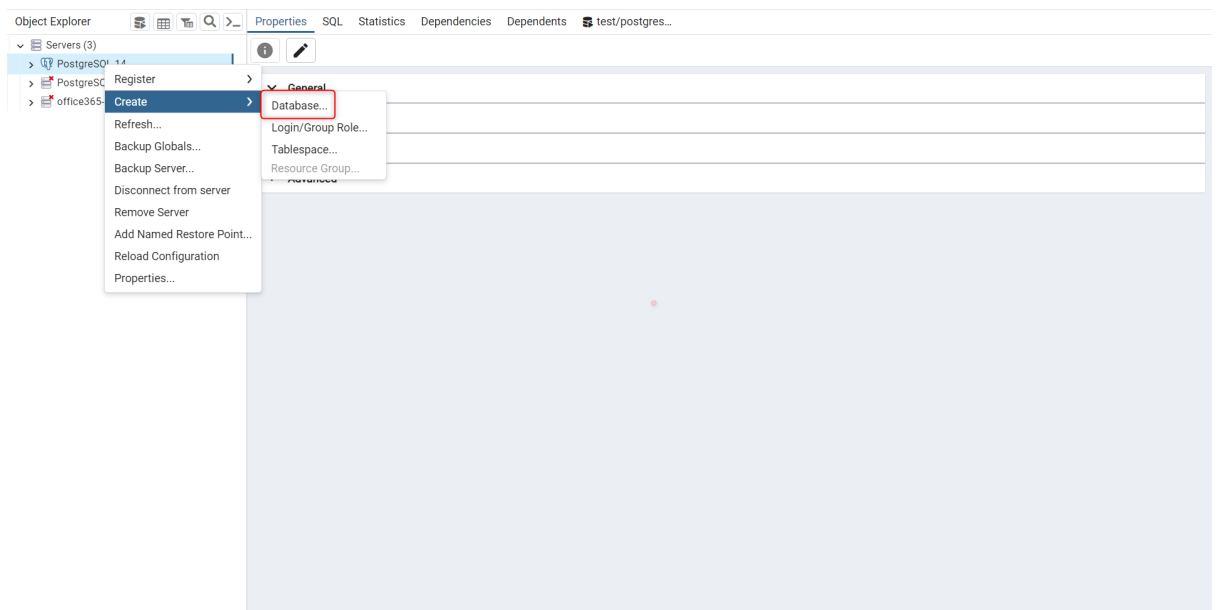
# Creating a new PostgreSQL database:

Please follow the below steps:

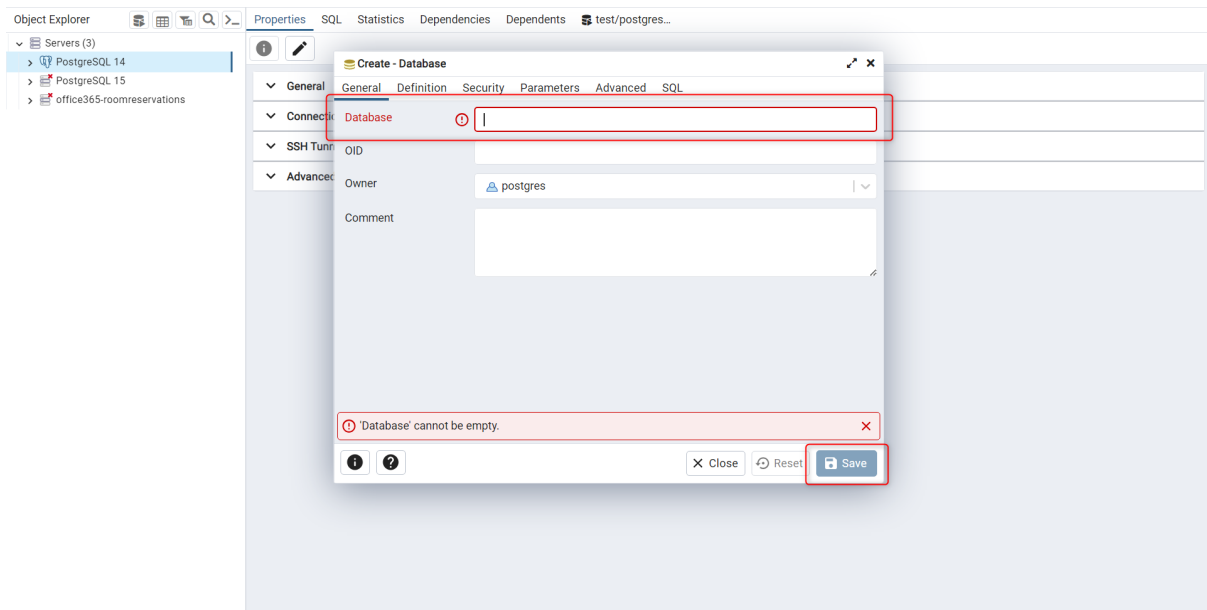
- Open the Pg Admin app on your machine and sign in with your master password.
- Right-click on **PostgreSQL server > Properties**
  - Verify that the hostname and port are **localhost** and **5432** respectively.
  - If the **Port** differs note it down



- Right-click on **PostgreSQL > Create > Database**



- Enter a name for this database and click on Save. Note down the database name.



With this, your PostgreSQL database is successfully created.

## Connecting PostgreSQL database with Django and creating required tables:

- Clone <https://github.com/SKWCE2023/backend-flask> repository and open it in VS Code.
- Open the **app.py** file, and look for the following configurations.

```
app.config['SQLALCHEMY_DATABASE_URI'] = 'postgresql://postgres:root@localhost:5432/shubs'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
```

- Update the **USER, PASSWORD, HOST, PORT, and DATABASE NAME** fields to match your Postgres server and database configurations that we did in the previous section.
  - In the attached image, my user is 'postgres', my password is 'root', my host is 'localhost', my port is '5432' and my database name is 'shubs'.
  - Note **postgresql://** and the **other characters** should remain intact. Please take care of that and update only your values respectively.
- Once done, open the Flask shell by running the following command in the terminal of your "flask backend" directory
  - flask shell
- A Python interactive shell will be opened. This special shell runs commands in the context of your Flask application.

```
PS C:\Users\shubham.kshatriya\Desktop\flask backend> flask shell
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
App: app
Instance: C:\Users\shubham.kshatriya\Desktop\flask backend\instance
>>> █
```

- To create the tables that are associated with your models, run the following commands in the flask shell
  - from app import db, press enter after inserting this command
  - db.create\_all(), once again press enter
  - you can check the following image for reference

```
PS C:\Users\shubham.kshatriya\Desktop\flask backend> flask shell
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
App: app
Instance: C:\Users\shubham.kshatriya\Desktop\flask backend\instance
>>> from app import db
>>> db.create_all()
>>> █
```

With this, your database is now connected with Django, and the required tables are created.

## Importing data into the database:

- To import the users, services, and customers data into your database, run the following command in the terminal of your “flask backend” directory.
  - python import\_data.py
- If the python command is not recognized, try running the above command using python3.
  - python3 import\_data.py

## Starting backend server:

- You are now set to start the server. Run the following command for the same in the terminal of your “flask backend” directory
  - flask run
- You will see the following output in the terminal. Your app is now serving at <http://127.0.0.1:5000/>
  - Please note this address, if it differs from <http://127.0.0.1:5000/>

```
PS C:\Users\shubham.kshatriya\Desktop\flask backend> flask run
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
█
```