01-Tutorial Python and PostgreSQL

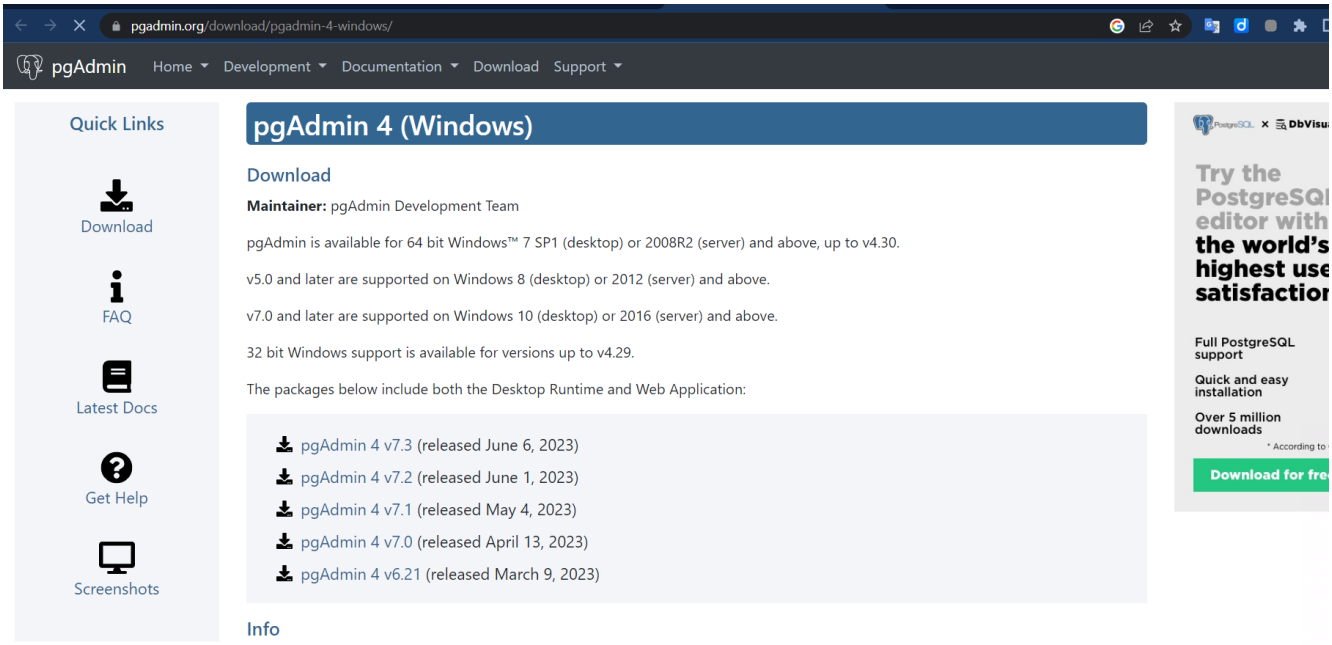
JUNE 25

Authored by: Mahdi Abbasi
Email : abbasimahdi782@gmail.com

PostgreSQL

Download pgAdmin4

https://www.postgresql.org/ftp/pgadmin/pgadmin4/v7.3/windows/



Full compatibility with different versions of the PostgreSQL database

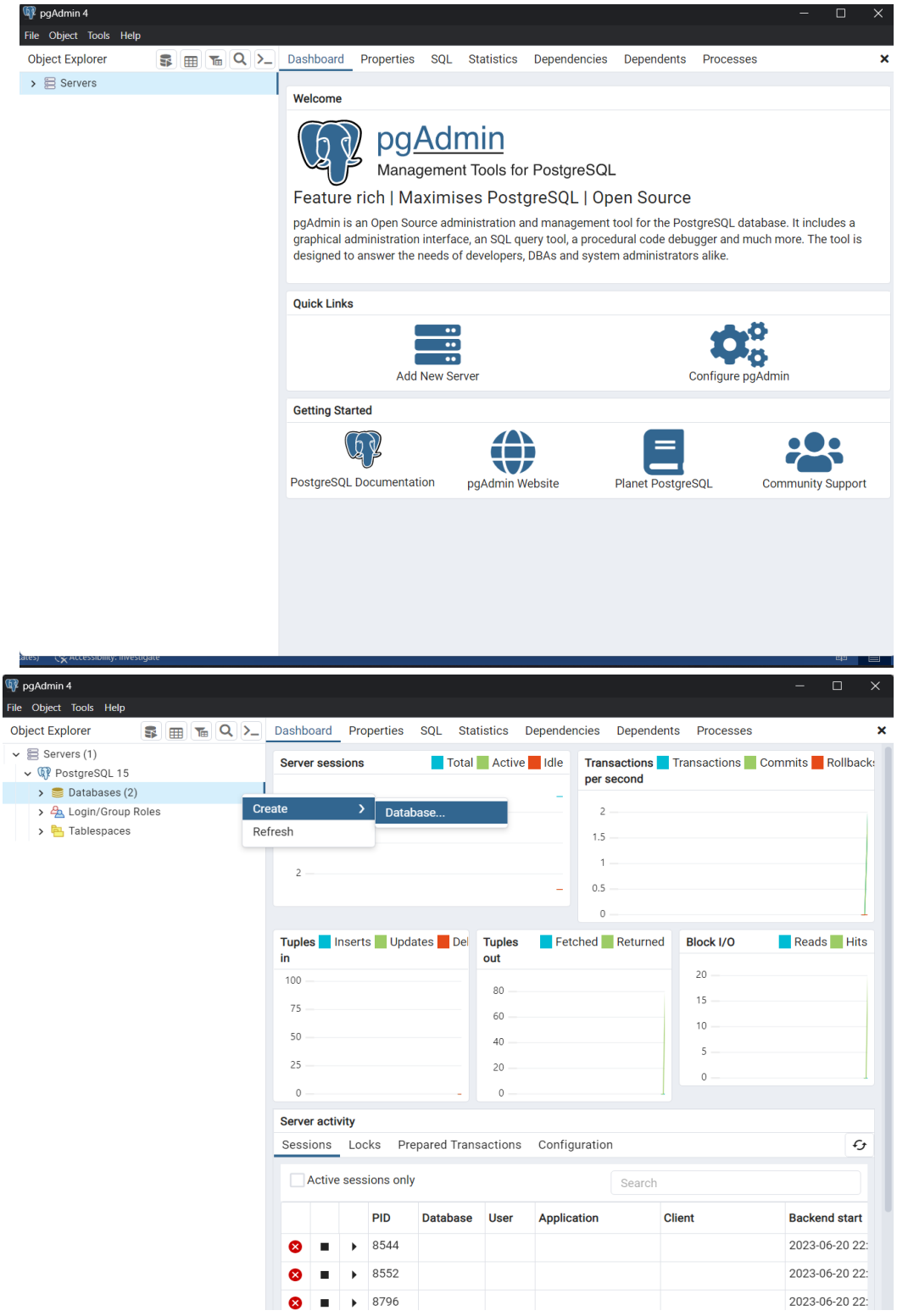
- The possibility of designing a database with extensive capabilities

- The ability to communicate with different databases and servers at the same time.

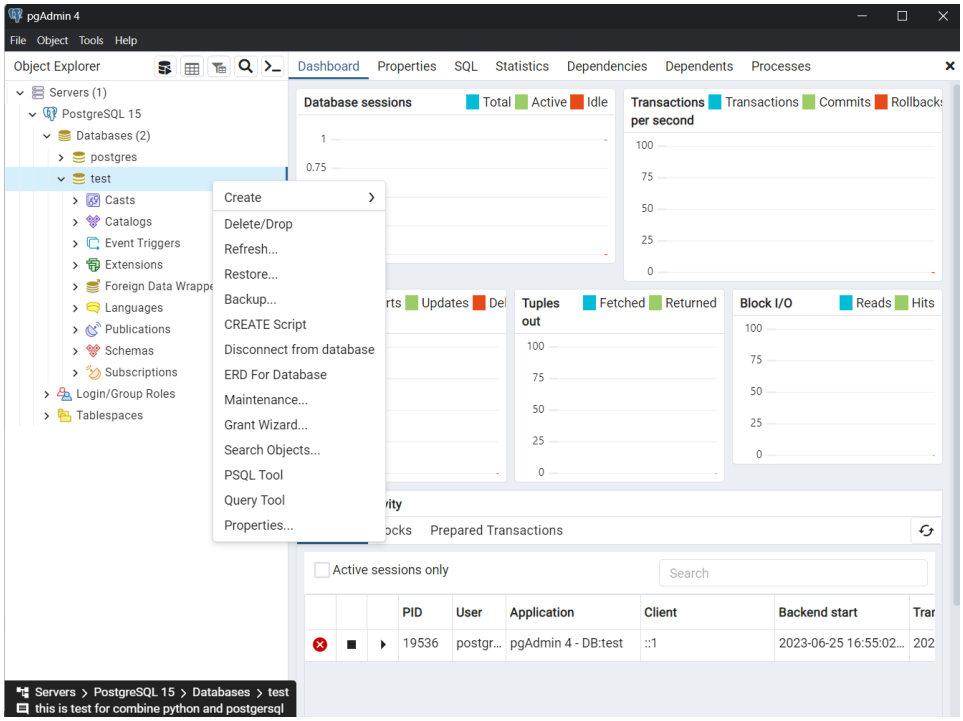
- Ability to design tables and write SQL queries.

"On this Tutorial I worked PostgreSQL in windows 11"

Create new database



Here two database, PostgreSQL and test. we are using test database.



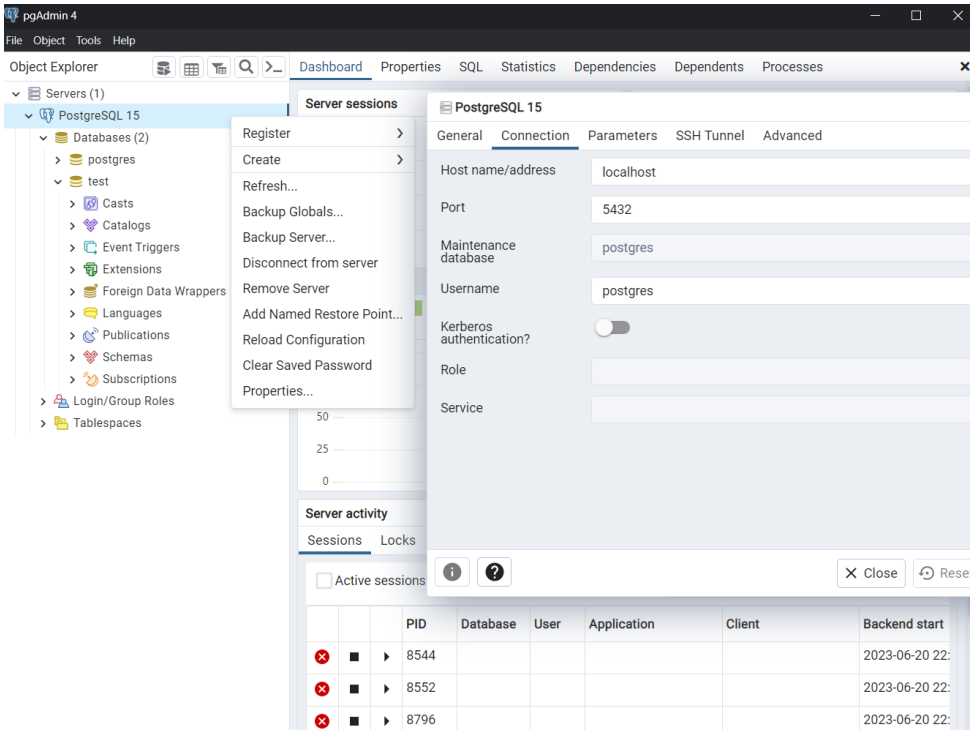
Install psycopg2

Run this command: pip install psycopg2.

After installing check hostname, port and username.

Right click + properties and go on connection.

Password is same as you set installation time.



Connect with database.

connect.py

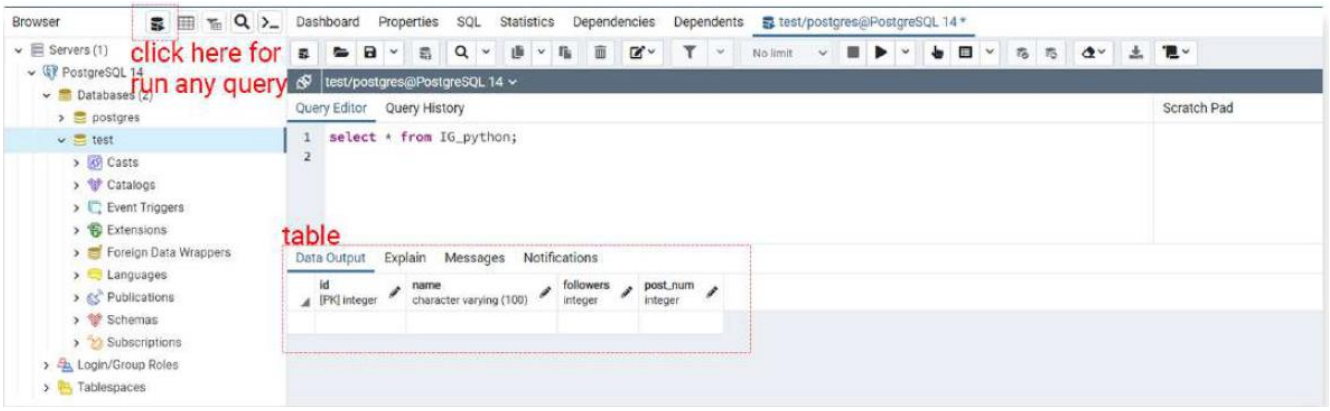
```python
# Practice for postgres and python
# Mahdi Abbasi

import psycopg2
import psycopg2.extras


host_name = 'localhost'
database = 'test'
username = 'postgres'
pw2 = 'miai09362092181'
port_id = 5432

conn = None

#------------- CONNECT WITH POSTGRES DATABASE -------------#

try:
    conn = psycopg2.connect(
        host = host_name,
        database = database,
        user = username,
        password = pw2,
        port = port_id,)
    print('Connection Successfully🔥 ')



except Exception as error :
    print('Error Message: ', error)

finally:
    if conn is not None:
        conn.close()
        print('Connection Closed Successfully')
```

Create table.

connect.py

```python

#------------- CONNECT WITH POSTGRES DATABASE -------------#

try:
    conn = psycopg2.connect(
        host = host_name,
        database = database,
        user = username,
        password = pw2,
        port = port_id,)
    print('Connection Successfully🔥 ')

#------------- CREATE TABLE IN POSTGRES DATABASE -------------#
    cur = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)
    cur.execute('DROP TABLE IF EXISTS table_python')
    create_script = '''
                CREATE TABLE IF NOT EXISTS table_python (
                    id int PRIMARY KEY,
                    name varchar(100) NOT NULL,
                    followers int,
                    post_num int
                )
    '''
    cur.execute(create_script)
    conn.commit()
except Exception as error :
    print('Error Message: ', error)

finally:
    if conn is not None:
        conn.close()
        cur.close()
```

For check table in database run select query in pgAdmin4



Replace IG_python to table_python or “name of table ”

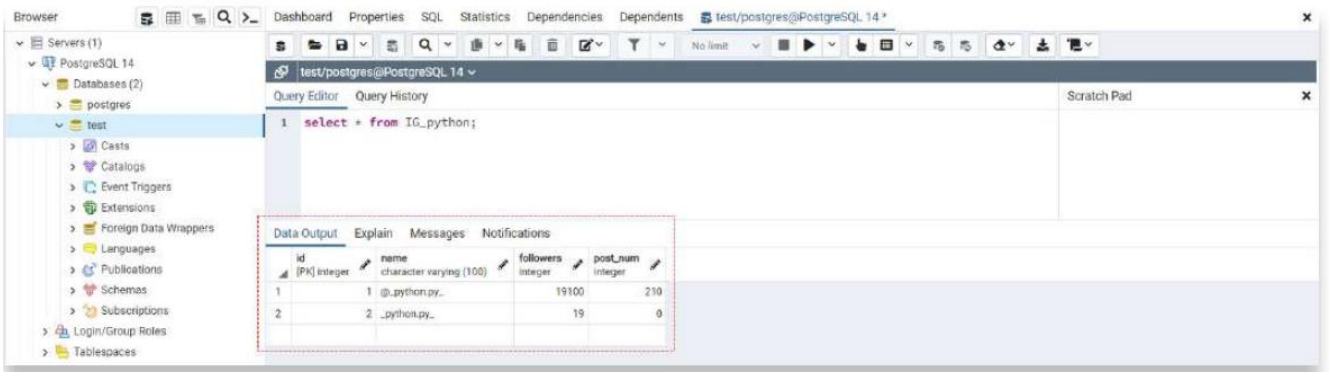
Insert data in table.

```
#------------- INSERT DATA IN TABLE table_python -------------#
    insert_scripts = 'INSERT INTO table_python (id, name, followers, post_num) values(%s, %s, %s, %s)'
    insert_values = [(1,'Mahdi_Abbasi', 1710, 3), (2,'Jadi_Mirmirani', 9900, 898)]
    for insert_value in insert_values:
        cur.execute(insert_scripts, insert_value)
```

Note: write code between in try and except.

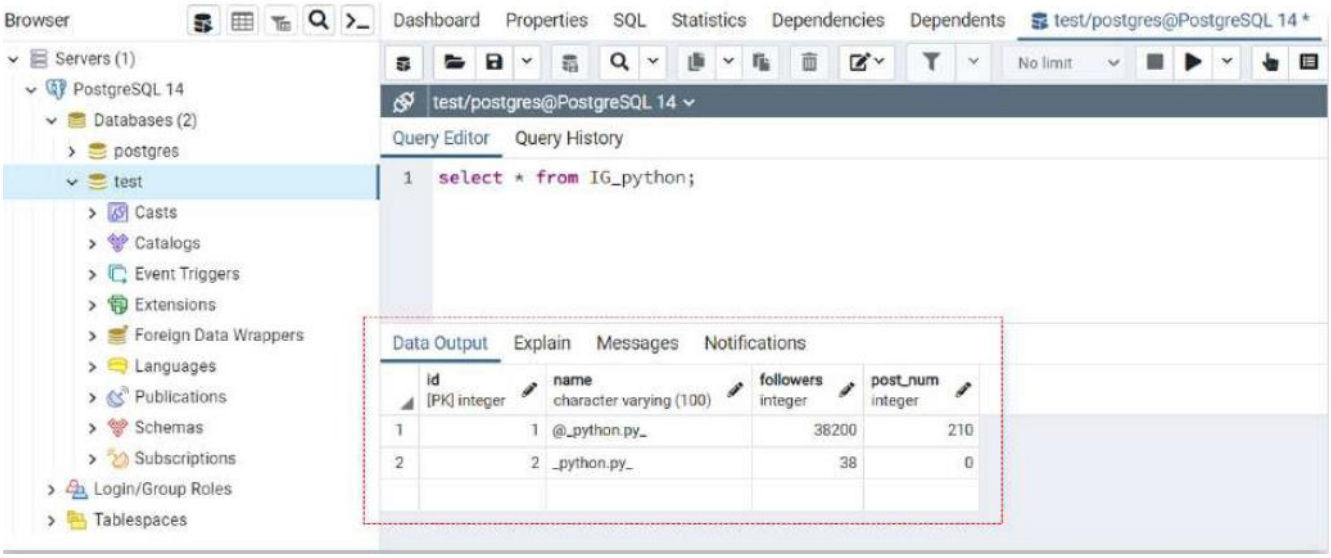
Select data.

```
#------------- SELECT DATA FROM table_python TABLE -------------#
    cur.execute('select * from table_python')
    records = cur.fetchall()
    print('select data is : \n',)
    for record in records:
        print(record)
    print()
    conn.commit()
```



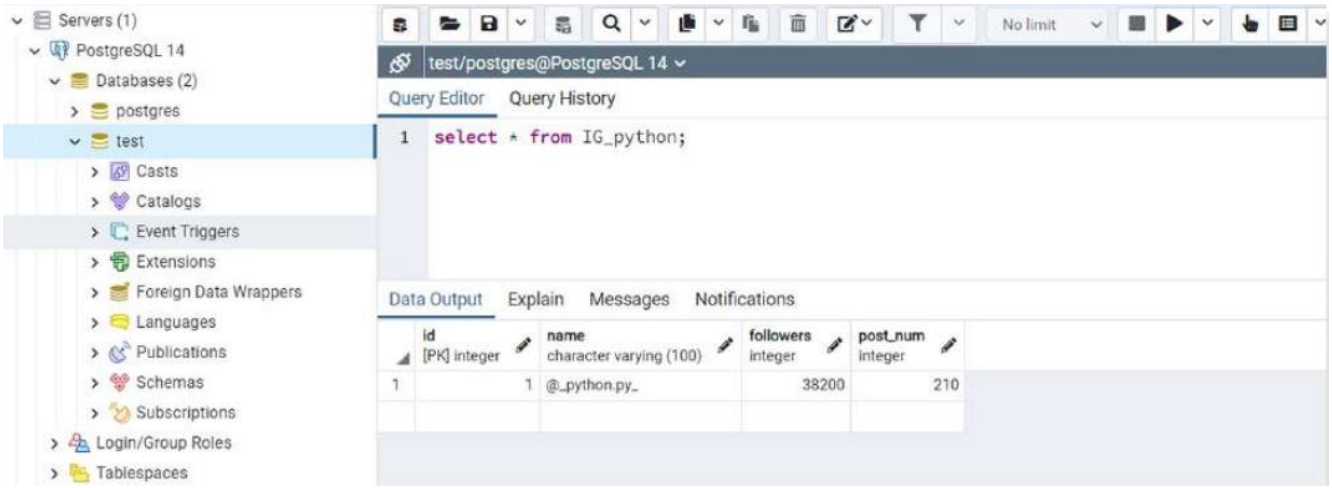
Update indexes table.

```
#------------- UPDATE DATE IN table_python TABLE -------------#
    update_scripts = 'UPDATE table_python SET followers = followers * 2.5'
    cur.execute(update_scripts)
    conn.commit
##### select data after update
    cur.execute('select * from table_python')
    records = cur.fetchall()
    print('select after update date in table : ',end= '\n',)
    for record in records:
        print(record)
    print()
    conn.commit()
```



Delete from table.

```
#------------- DELETE SOME DATE FROM table_python TABLE -------------#
    delete_script = 'DELETE FROM table_python where name = %s'
    delete_id = ('Jadi_Mirmirani',)
    cur.execute(delete_script,delete_id)
#### select data after delete
    cur.execute('select * from table_python')
    records = cur.fetchall()
    print('select after delete date in table : ',end= '\n',)
    for record in records:
        print(record)
    print()
```



Source code in this repo and my GitHub account

https://github.com/MahdiAbbasi7/PostgreSQ