
01-Tutorial

Python and

PostgreSQL



JUNE 25

Authored by: Mahdi Abbasi

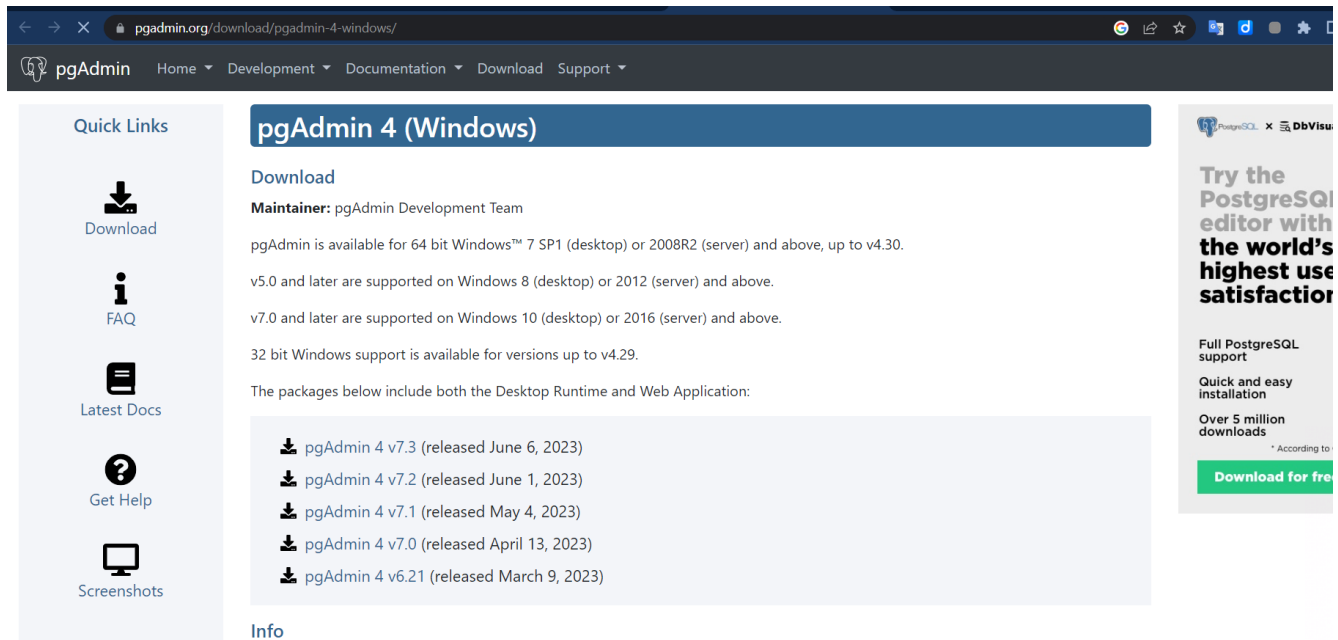
Email : abbasimahdi782@gmail.com



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.

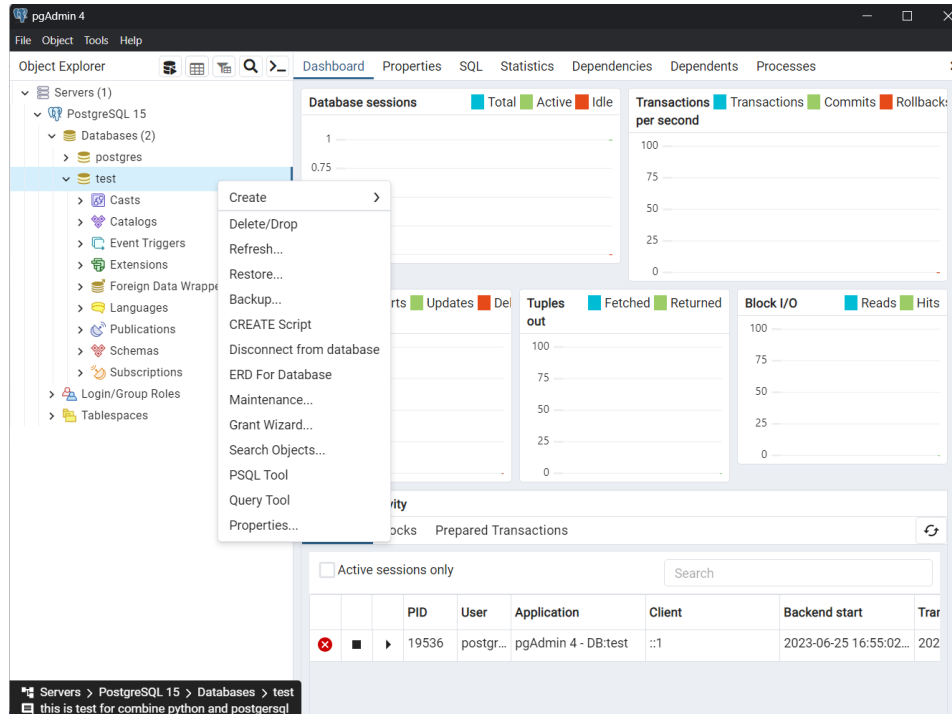
“On this Tutorial I worked PostgreSQL in windows 11”

Create new database

The screenshot shows the pgAdmin 4 interface. The 'Object Explorer' on the left shows a tree structure with 'Servers (1)' expanded, showing 'PostgreSQL 15'. Under 'PostgreSQL 15', 'Databases (2)' is selected. A context menu is open over 'Databases (2)' with 'Create' and 'Refresh' options. The 'Create' option is highlighted, and a sub-menu is open showing 'Database...'. The main panel displays the 'Dashboard' with various charts and tables. The 'Server sessions' chart shows 'Total', 'Active', and 'Idle' sessions. The 'Transactions per second' chart shows 'Transactions', 'Commits', and 'Rollbacks'. The 'Tuples in' chart shows 'Inserts', 'Updates', and 'Deletes'. The 'Tuples out' chart shows 'Fetched' and 'Returned'. The 'Block I/O' chart shows 'Reads' and 'Hits'. The 'Server activity' table shows a list of sessions with columns for PID, Database, User, Application, Client, and Backend start time.

		PID	Database	User	Application	Client	Backend start
✖	■	▶ 8544					2023-06-20 22:...
✖	■	▶ 8552					2023-06-20 22:...
✖	■	▶ 8796					2023-06-20 22:...

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



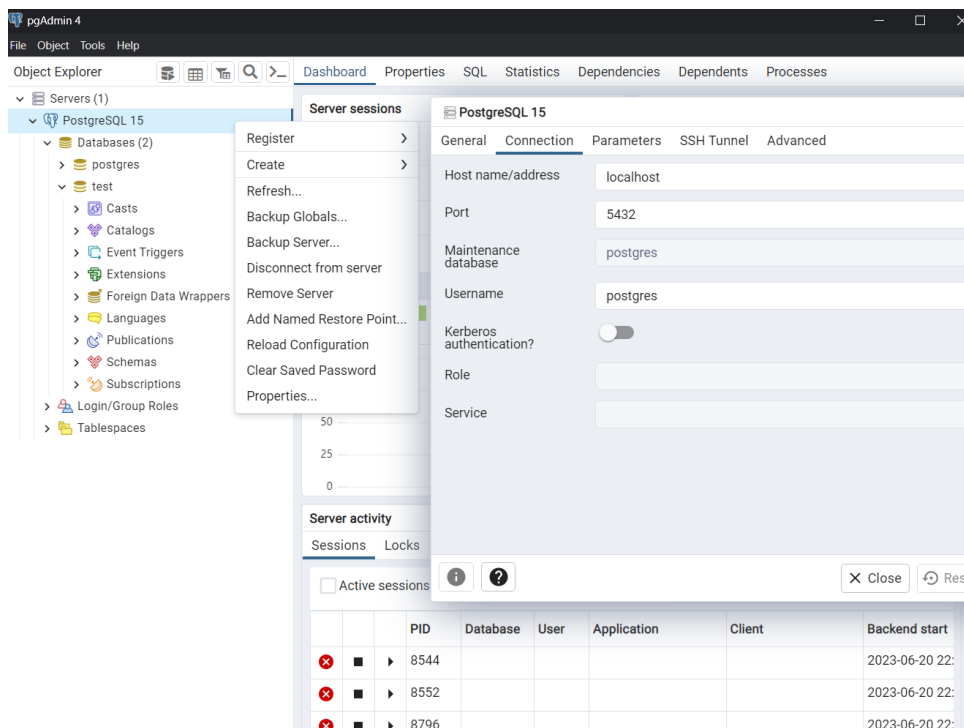
Install psycopg2

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.



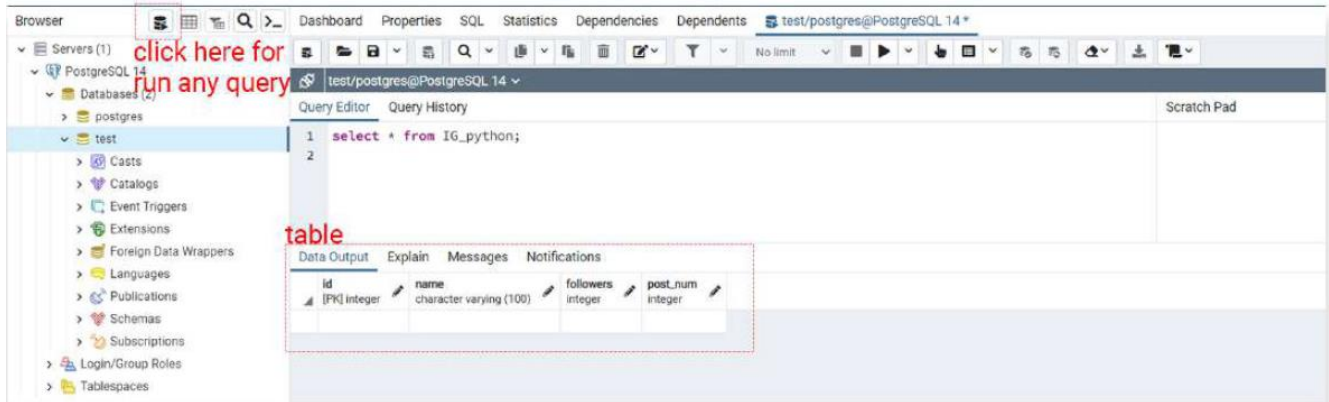
Connect with database.

```
connect.py X
connect.py > ...
1  # Practice for postgres and python
2  # Mahdi Abbasi
3
4  import psycopg2
5  import psycopg2.extras
6
7
8  host_name = 'localhost'
9  database = 'test'
10 username = 'postgres'
11 pw2 = 'miai09362092181'
12 port_id = 5432
13
14 conn = None
15
16 #----- CONNECT WITH POSTGRES DATABASE -----#
17
18 try:
19     conn = psycopg2.connect(
20         host = host_name,
21         database = database,
22         user = username,
23         password = pw2,
24         port = port_id,)
25     print('Connection Successfully 🔥 ')
26
27
28
29 except Exception as error :
30     print('Error Message: ', error)
31
32 finally:
33     if conn is not None:
34         conn.close()
35     print('Connection Closed Successfully')
```

Create table.

```
connect.py X
connect.py > ...
15
16 #----- CONNECT WITH POSTGRES DATABASE -----#
17
18 try:
19     conn = psycopg2.connect(
20         host = host_name,
21         database = database,
22         user = username,
23         password = pw2,
24         port = port_id,)
25     print('Connection Successfully 🔥 ')
26
27 #----- CREATE TABLE IN POSTGRES DATABASE -----#
28     cur = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)
29     cur.execute('DROP TABLE IF EXISTS table_python')
30     create_script = '''
31         CREATE TABLE IF NOT EXISTS table_python (
32             id int PRIMARY KEY,
33             name varchar(100) NOT NULL,
34             followers int,
35             post_num int
36         )
37     '''
38     cur.execute(create_script)
39     conn.commit()
40 except Exception as error :
41     print('Error Message: ', error)
42
43 finally:
44     if conn is not None:
45         conn.close()
46         cur.close()
```

For check table in database run select query in pgAdmin4



Replace IG_python to table_python or “name of table ”

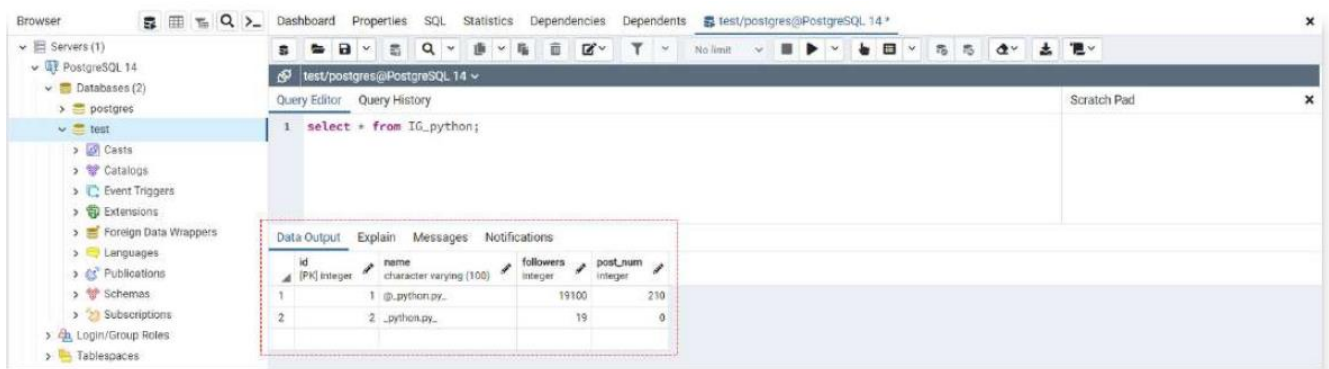
Insert data in table.

```
#----- 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.

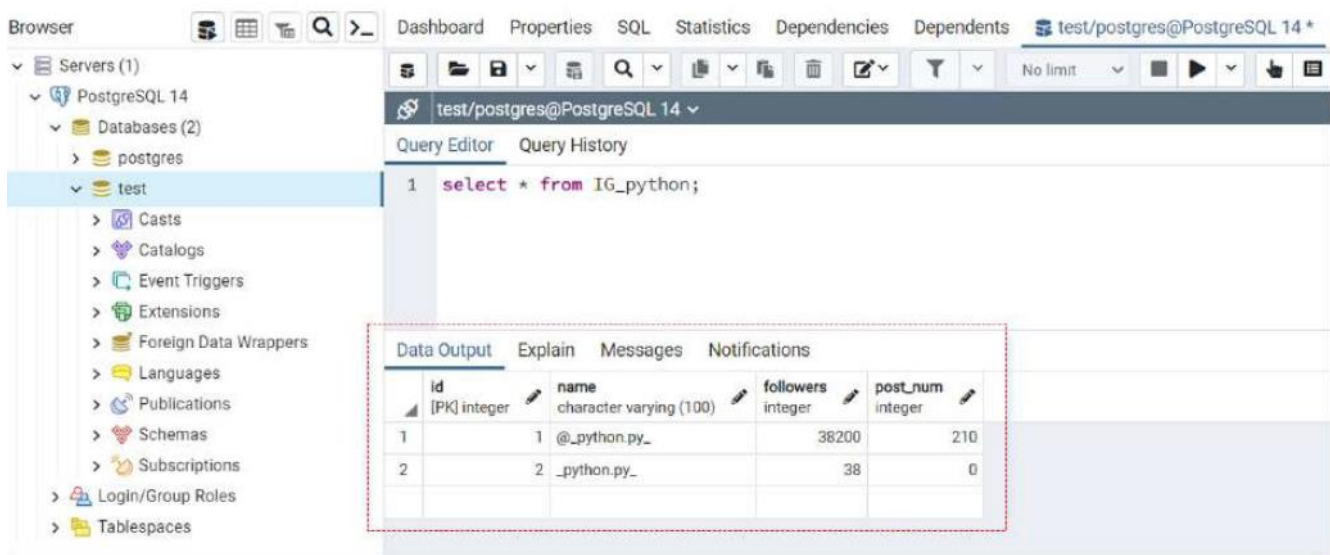
```
#----- SELECT DATA FROM table_python TABLE -----#
cur.execute('select * from table_python')
records = cur.fetchall()
print('select data is : \n',)
for record in records:
    print(record)
print()
conn.commit()
```



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()
```

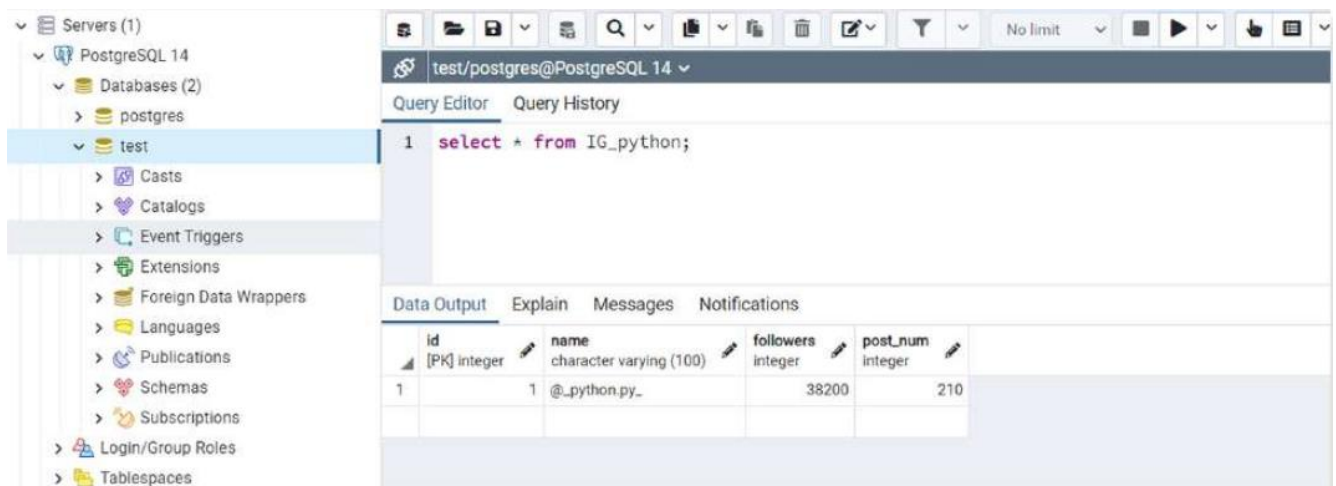


The screenshot shows a PostgreSQL client interface with a sidebar on the left listing servers and databases. The main window displays a query editor with the query `select * from IG_python;`. Below the query editor, a table titled "Data Output" shows the results of the query. The table has five columns: `id` (integer), `name` (character varying), `followers` (integer), and `post_num` (integer). The results show two rows of data.

id	name	followers	post_num
1	@_python.py_	38200	210
2	_python.py_	38	0

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/PostgreSQL>