Session 4 - Connecting and Basic Operation

September 5, 2018

1 Python Training: Day 2 Session 4

1.1 2.4.1 Creating sample database

This session covers database creation, connection, and simple manipulation.

1. Connecting to server
2. Create a database
3. Name it 'jualan' (or anything else) and save
4. Create table
5. Name the table 'produk'
6. Fill up the colum as the following, then click 'save'

1.2 2.4.2 Connecting to sample database

Note that

```
import psycopg2
```

requires package psycopg2 to be installed.

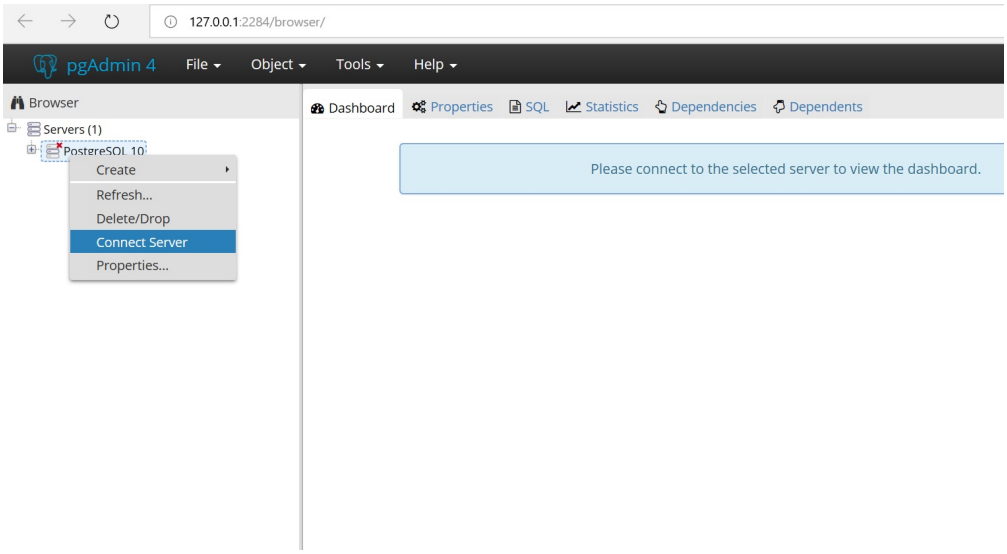
Install with the following command if you encounter ModuleNotFoundError: No module named 'psycopg2'

```
pip install psycopg2
```

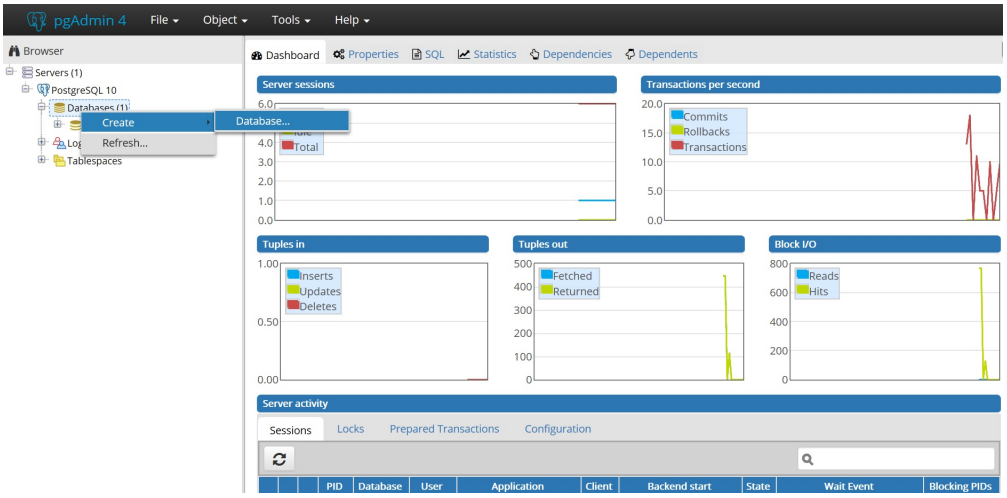
If using Anaconda distribution, install using Anaconda prompt. Or, your default command prompt if it supports pip.

Note:

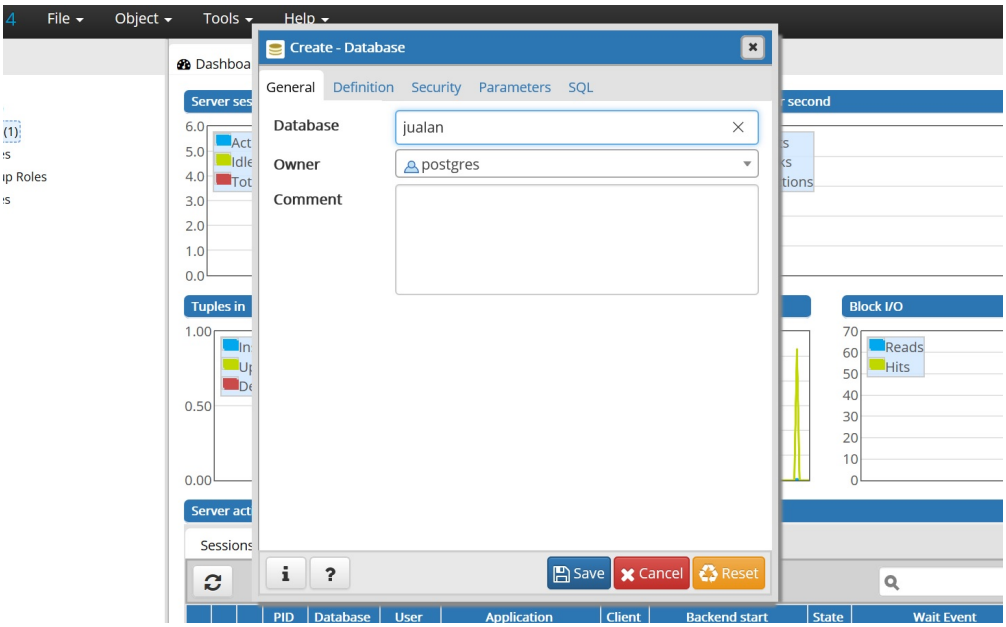
Connection information could be checked here



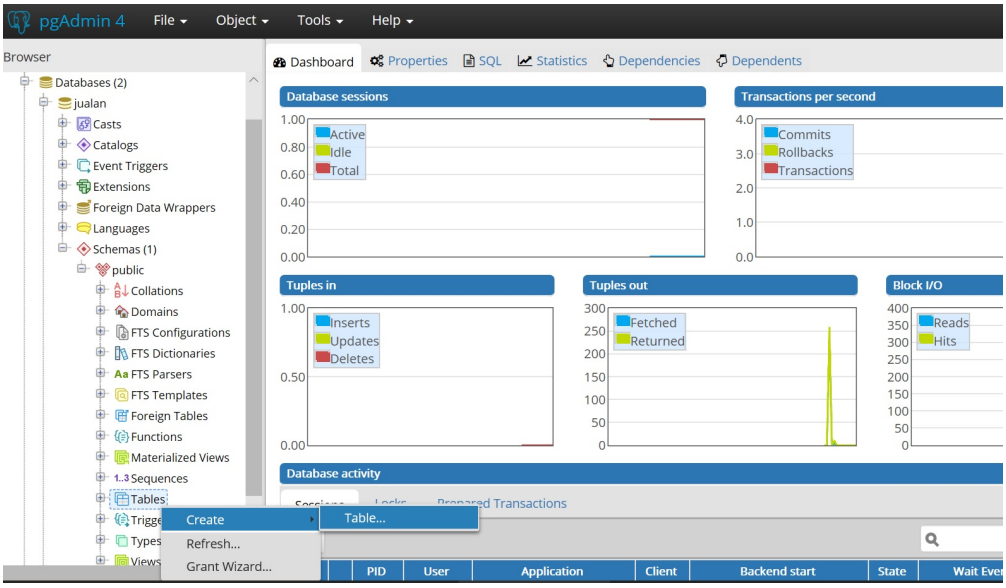
alt text



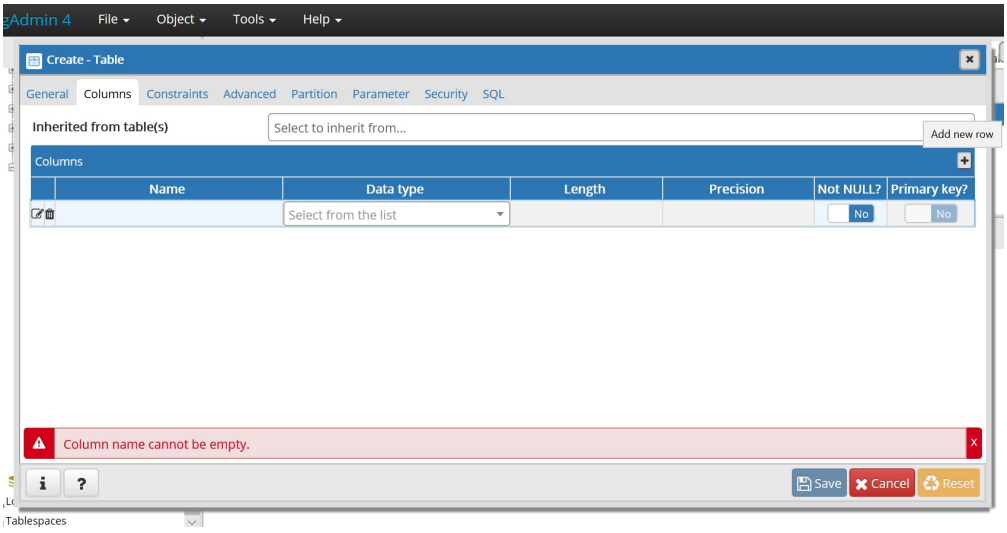
alt text



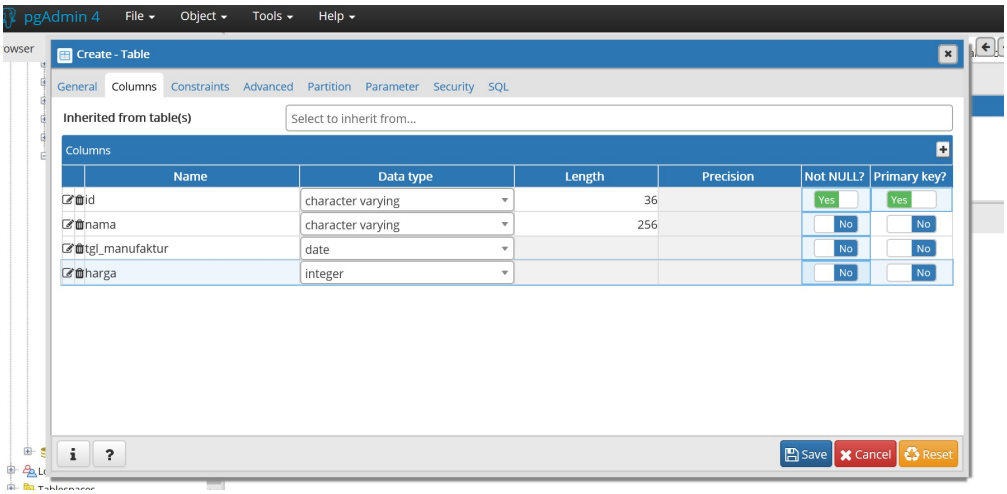
alt text



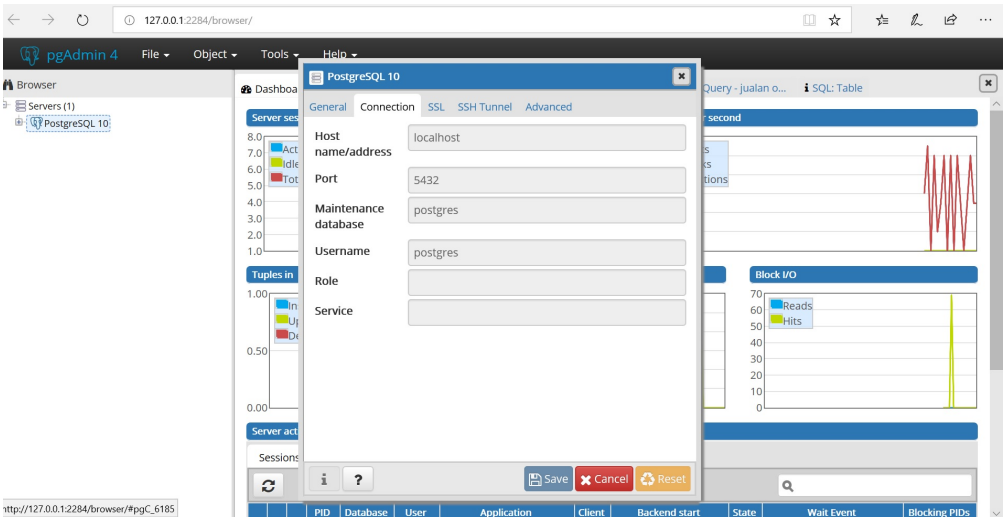
alt text



alt text



alt text



alt text

1.3 2.4.3 Connecting to database

```
In [ ]: import psycopg2
```

```
In [ ]: connection = psycopg2.connect(database = "jualan", user = "postgres",
                                      password = "postgres123",
                                      host = "localhost", port = "5432")
        cursor = connection.cursor()
        cursor.execute("SELECT version()")
        ver = cursor.fetchone()
        connection.commit()
        connection.close()
```

```
In [ ]: print(ver)
```

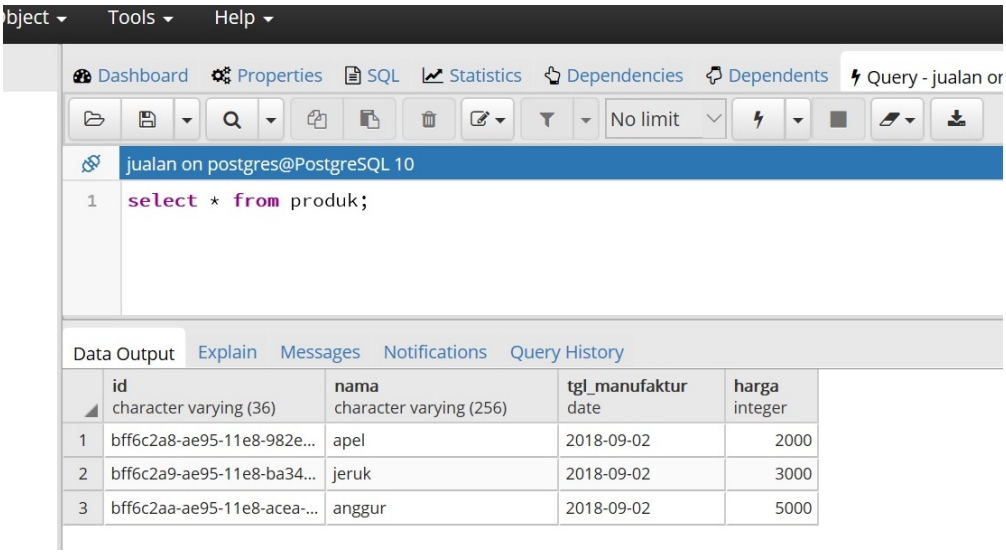
1.4 2.4.4. Database operations

1.4.1 Basic sql command execution

```
In [ ]: def execute_sql(command):
            connection = psycopg2.connect(database = "jualan",
                                          user = "postgres", password = "postgres123",
                                          host = "localhost", port = "5432")
            cursor = connection.cursor()
            cursor.execute(command)
            connection.commit()
            connection.close()
```

1.4.2 Insert

```
In [ ]: import uuid
        import datetime
```



alt text

```
        id_barang = uuid.uuid1()
        now = datetime.date.today()
```

```
In [ ]: print(id_barang)
        print(now)
```

```
In [ ]: execute_sql("INSERT INTO produk(id, nama, tgl_manufaktur, harga) VALUES ('"
                    + str(id_barang) + "', 'kantong ajaib', '" + str(now) + "', 1000000);")
```

```
In [ ]: # Create more !

        id_barang_1 = uuid.uuid1()
        id_barang_2 = uuid.uuid1()
        id_barang_3 = uuid.uuid1()

        execute_sql("INSERT INTO produk(id, nama, tgl_manufaktur, harga) VALUES ('"
                    + str(id_barang_1) + "', 'apel', '" + str(now) + "', 2000);")
        execute_sql("INSERT INTO produk(id, nama, tgl_manufaktur, harga) VALUES ('"
                    + str(id_barang_2) + "', 'jeruk', '" + str(now) + "', 3000);")
        execute_sql("INSERT INTO produk(id, nama, tgl_manufaktur, harga) VALUES ('"
                    + str(id_barang_3) + "', 'anggur', '" + str(now) + "', 5000);")
```

1.4.3 Delete

```
In [ ]: execute_sql("DELETE FROM produk WHERE id='" + str(id_barang) +"';")
```

The following is the result.

1.4.4 Assignment 2.4.1

Implement a function to execute update and query (sql select) command from table 'produk'