

ASSIGNMENT-7

Module 16: Building Database Apps with PostgreSQL & Python:

Creating a database: create database database_name;

```
SQL Shell (psql)  x + v
Server [localhost]:
Database [postgres]:
Port [5433]:
Username [postgres]:
Password for user postgres:

psql (18.1)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \l
                                         List of databases
   Name   |  Owner   | Encoding | Locale Provider | Collate           | Ctype            | Locale | ICU Rules | Access privileges
---+-----+-----+-----+-----+-----+-----+-----+-----+-----+
postgres | postgres | UTF8    | libc       | English_United States.1252 | English_United States.1252 |          |          |          |
template0 | postgres | UTF8    | libc       | English_United States.1252 | English_United States.1252 |          |          |          |
template1 | postgres | UTF8    | libc       | English_United States.1252 | English_United States.1252 |          |          |          |
(3 rows)

postgres=# create database demodb;
CREATE DATABASE
postgres=# \l
                                         List of databases
   Name   |  Owner   | Encoding | Locale Provider | Collate           | Ctype            | Locale | ICU Rules | Access privileges
---+-----+-----+-----+-----+-----+-----+-----+-----+
demodb  | postgres | UTF8    | libc       | English_United States.1252 | English_United States.1252 |          |          |          |
postgres | postgres | UTF8    | libc       | English_United States.1252 | English_United States.1252 |          |          |          |
template0 | postgres | UTF8    | libc       | English_United States.1252 | English_United States.1252 |          |          |          |
template1 | postgres | UTF8    | libc       | English_United States.1252 | English_United States.1252 |          |          |          |
(4 rows)

postgres=# \c demodb
You are now connected to database "demodb" as user "postgres".
```

Deleting a database: drop database database_name;

```
SQL Shell (psql)  x + v
You are now connected to database "demodb" as user "postgres".
demodb=# create database testdb;
CREATE DATABASE
demodb=# \l
                                         List of databases
   Name   |  Owner   | Encoding | Locale Provider | Collate           | Ctype            | Locale | ICU Rules | Access privileges
---+-----+-----+-----+-----+-----+-----+-----+-----+
demodb  | postgres | UTF8    | libc       | English_United States.1252 | English_United States.1252 |          |          |          |
postgres | postgres | UTF8    | libc       | English_United States.1252 | English_United States.1252 |          |          |          |
template0 | postgres | UTF8    | libc       | English_United States.1252 | English_United States.1252 |          |          |          |
template1 | postgres | UTF8    | libc       | English_United States.1252 | English_United States.1252 |          |          |          |
testdb   | postgres | UTF8    | libc       | English_United States.1252 | English_United States.1252 |          |          |          |
(5 rows)

demodb=# drop database testdb;
DROP DATABASE
demodb=# \l
                                         List of databases
   Name   |  Owner   | Encoding | Locale Provider | Collate           | Ctype            | Locale | ICU Rules | Access privileges
---+-----+-----+-----+-----+-----+-----+-----+-----+
demodb  | postgres | UTF8    | libc       | English_United States.1252 | English_United States.1252 |          |          |          |
postgres | postgres | UTF8    | libc       | English_United States.1252 | English_United States.1252 |          |          |          |
template0 | postgres | UTF8    | libc       | English_United States.1252 | English_United States.1252 |          |          |          |
template1 | postgres | UTF8    | libc       | English_United States.1252 | English_United States.1252 |          |          |          |
(4 rows)

demodb=#

```

Creating table and adding data:

```
CREATE TABLE TABLE_NAME(COLUMN_NAMES);
```

```
INSERT INTO TABLE_NAME VALUES(COLUMN_DATA);
```

```
postgres=# create database student;
CREATE DATABASE
postgres=# \c student
You are now connected to database "student" as user "postgres"
student=# CREATE TABLE students(name text,number int,age int);
CREATE TABLE
student=# \d
List of relations
 Schema | Name   | Type  | Owner
-----+-----+-----+
 public | students | table | postgres
(1 row)

student=# INSERT INTO students(name,number,age) VALUES('Pragathi',528,24);
INSERT 0 1
student# INSERT INTO students(name,number,age) VALUES('Nithish',514,25);
INSERT 0 1
student# |
```



Retrieving data from database and deleting contents in the table:

```
SELECT * FROM TABLE_NAME;
```

```
TRUNCATE TABLE TABLE_NAME;
```

```
SQL Shell (psql)      x  +  v
student#
student# SELECT * FROM students;
 name | number | age
-----+
 Pragathi |    528 |  24
 Nithish  |    514 |  25
(2 rows)

student# SELECT name FROM students;
 name
-----
 Pragathi
 Nithish
(2 rows)

student# SELECT * FROM students WHERE number=528;
 name | number | age
-----+
 Pragathi |    528 |  24
(1 row)

student# SELECT * FROM students WHERE age=24;
 name | number | age
-----+
 Pragathi |    528 |  24
(1 row)

student# SELECT number FROM students where age=24;
 number
-----
 528
(1 row)

student# SELECT number FROM students where name='Nithish';
 number
-----+

```



SQL Shell (psql)

```
(1 row)

student=# SELECT number FROM students where name='Nithish';
number
-----
514
(1 row)

student=# TRUNCATE TABLE students;
TRUNCATE TABLE
student=# \d
          List of relations
 Schema |   Name   | Type  | Owner
-----+----------+-----+-----
 public | students | table | postgres
(1 row)

student=# SELECT * FROM students;
 name | number | age
-----+-----+
(0 rows)

student=# |
```

18°C Clear

Search

10:46 PM 12/28/2025

Setting up virtualenv:

```
Command Prompt
```

```
Downloading virtualenv-20.35.4-py3-none-any.whl (6.0 MB) 6.0/6.0 MB 8.7 MB/s 0:00:00
  Downloading distlib-0.4.0-py2.py3-none-any.whl (469 kB)
  Downloading filelock-3.20.1-py3-none-any.whl (16 kB)
  Installing collected packages: distlib, filelock, virtualenv
  Successfully installed distlib-0.4.0 filelock-3.20.1 virtualenv-20.35.4
[notice] A new release of pip is available: 25.2 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\DELL\Desktop\Visual Studio>virtualenv env
created virtual environment CPython3.14.0.final.0-64 in 19460ms
  creator CPython2Windows(dest=C:\Users\DELL\Desktop\Visual Studio\env, clear=False, no_vcs_ignore=False, global=False)
  seeders FromAppData(download=False, pip=bundle, via=copy, app_data_dir=C:\Users\DELL\AppData\Local\pypa\virtualenv)
    added seed packages: pip==25.3
  activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator

C:\Users\DELL\Desktop\Visual Studio>cd env

C:\Users\DELL\Desktop\Visual Studio\env>cd scripts

C:\Users\DELL\Desktop\Visual Studio\env\Scripts>activate

(env) C:\Users\DELL\Desktop\Visual Studio\env\Scripts>cd..

(env) C:\Users\DELL\Desktop\Visual Studio\env>cd..

(env) C:\Users\DELL\Desktop\Visual Studio>python test.py

(env) C:\Users\DELL\Desktop\Visual Studio>python test.py
Python

(env) C:\Users\DELL\Desktop\Visual Studio>deactivate
C:\Users\DELL\Desktop\Visual Studio>cd env

C:\Users\DELL\Desktop\Visual Studio\env>cd scripts

C:\Users\DELL\Desktop\Visual Studio\env\Scripts>activate

(env) C:\Users\DELL\Desktop\Visual Studio\env\Scripts>
```

18°C Clear

Search

11:01 PM 12/28/2025

File Edit Selection View Go Run Terminal Help

EXPLORER

VISUAL STUDIO

env

test.py

Walkthrough: Setup VS Code test.py

```
1 print("Python")
```

CHAT

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\DELL\Desktop\Visual Studio & "C:\Users\DELL\Desktop\Visual Studio\env\Scripts\Activate.ps1"
& : File C:\Users\DELL\Desktop\Visual Studio\env\Scripts\Activate.ps1 cannot be loaded because running scripts is
disabled on this system. For more information, see about_Execution_Policies at
<https://go.microsoft.com/fwlink/?LinkID=135170>.
At line:1 char:3
+ & "C:\Users\DELL\Desktop\Visual Studio\env\Scripts\Activate.ps1"
+ CategoryInfo : SecurityError: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess

PS C:\Users\DELL\Desktop\Visual Studio> python test.py
Python
PS C:\Users\DELL\Desktop\Visual Studio>

SUGGESTED ACTIONS

Build Workspace Show Config

test.py +

Describe what to build next

Agent Auto

Ln 1, Col 16 Spaces: 4 UTF-8 CRLF Python env (3.14.0)

18°C Clear

Search

11:02 PM 12/28/2025

Connecting to the database:

File Edit Selection View Go Run Terminal Help

EXPLORER

VISUAL STUDIO

env

test.py

Walkthrough: Setup VS Code test.py

```
1 import psycopg2  
2 connect = psycopg2.connect(dbname="postgres", user="postgres", password="11032001", host="localhost", port="5433")  
3 print("Connected successfully!")
```

CHAT

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\DELL\Desktop\Visual Studio > ...
1 import psycopg2
2 connect = psycopg2.connect(dbname="postgres", user="postgres", password="11032001", host="localhost", port="5433")
3 print("Connected successfully!")

SUGGESTED ACTIONS

Build Workspace Show Config

test.py +

Describe what to build next

Agent Auto

Ln 3, Col 33 Spaces: 4 UTF-8 CRLF Python env (3.14.0)

19°C Mostly clear

Search

11:17 PM 12/28/2025

```
Command Prompt
Microsoft Windows [Version 10.0.26200.7462]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL>cd desktop
C:\Users\DELL\Desktop>cd visual studio
C:\Users\DELL\Desktop\Visual Studio>cd env
C:\Users\DELL\Desktop\Visual Studio\env>cd scripts
C:\Users\DELL\Desktop\Visual Studio\env\Scripts>activate
(env) C:\Users\DELL\Desktop\Visual Studio\env\Scripts>cd..
(env) C:\Users\DELL\Desktop\Visual Studio>python test.py
Connected successfully!
(env) C:\Users\DELL\Desktop\Visual Studio>
```

The screenshot shows a Microsoft Windows Command Prompt window titled "Command Prompt". The window displays a series of commands being run in a terminal session. The user navigates to their desktop, then to the "visual studio" folder, followed by "env" and "scripts". They activate the environment using "activate". Then they change directory back up to the parent folder ("..") and run a Python script named "test.py". The output of the script, "Connected successfully!", is shown in the terminal. The system tray at the bottom of the screen indicates the date and time as 12/28/2025, 11:17 PM. The weather icon shows 19°C and "Mostly clear".

Creating table using Python:

Python code:

```
import psycopg2
connect =
psycopg2.connect(dbname="postgres",user="postgres",password="11032001",host="localhost",port="5433")

cursor = connect.cursor()

cursor.execute("""create table employees(Name text, ID int, Age int);""")

print("Table created successfully")

connect.commit()

connect.close()
```

```
Command Prompt
Microsoft Windows [Version 10.0.26200.7462]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL>cd desktop
C:\Users\DELL\Desktop>cd visual studio
C:\Users\DELL\Desktop\Visual Studio>cd env
C:\Users\DELL\Desktop\Visual Studio\env>cd scripts
C:\Users\DELL\Desktop\Visual Studio\env\Scripts>activate
(env) C:\Users\DELL\Desktop\Visual Studio\env\Scripts>cd..
(env) C:\Users\DELL\Desktop\Visual Studio\env>cd..
(env) C:\Users\DELL\Desktop\Visual Studio>python test.py
Connected successfully!
(env) C:\Users\DELL\Desktop\Visual Studio>python test.py
Table created successfully

(env) C:\Users\DELL\Desktop\Visual Studio>
```

Inserting the data using python:

Python code:

```
import psycopg2

def table():

    connect =
    psycopg2.connect(dbname="postgres",user="postgres",password="11032001",host="localhost",port="5
433")

    cursor = connect.cursor()

    cursor.execute("create table employees(Name text, ID int, Age int);"")

    print("Table created successfully")

    connect.commit()

    connect.close()

def data():
```

```

connect =
psycopg2.connect(dbname="postgres",user="postgres",password="11032001",host="localhost",port="5
433")

cursor = connect.cursor()

cursor.execute("insert into employees(Name,ID,Age) values('Ram',01,31);")

print("Data inserted successfully")

connect.commit()

connect.close()

data()

```

(env) C:\Users\DELL\Desktop\Visual Studio>python test.py
Data inserted successfully
(env) C:\Users\DELL\Desktop\Visual Studio>

postgres=# select * from employees;
name | id | age
-----+---+---
Ram | 1 | 31
(1 row)

postgres=# |

Extracting data from database:

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder named "VISUAL STUDIO" containing "env" and "test.py".
- Code Editor:** Displays Python code for creating a database table and extracting data from it.
- Terminal:** Shows the command to install the psycopg2-binary package via pip.
- Status Bar:** Shows the file path as "PS C:\Users\DELL\Desktop\Visual Studio> python test.py", line 27, column 1, and other status information like "Ln 27, Col 1 Spaces: 4 UTF-8 CRLF {} Python env (3.14.0)".

```
2 def table():
3     cursor.execute(" create table employees(Name text, ID int, Age int); ")
4     print("Table created successfully")
5     connect.commit()
6     connect.close()
7
8 def data():
9     connect = psycopg2.connect(dbname="postgres",user="postgres",password="11032001",host="localhost",port="5432")
10
11    cursor = connect.cursor()
12    cursor.execute("insert into employees(Name,ID,Age) values('Ram',01,31);")
13    print("Data inserted successfully")
14    connect.commit()
15    connect.close()
16
17 def extract():
18    connect = psycopg2.connect(dbname="postgres",user="postgres",password="11032001",host="localhost",port="5432")
19    cursor = connect.cursor()
20    cursor.execute("select * from employees;")
21    show=cursor.fetchone()
22    print(show[0])
23    connect.commit()
24    connect.close()
25
26 extract()
```

Build with Agent panel is visible on the right, showing AI responses and build actions.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder named "VISUAL STUDIO" containing "env" and "test.py".
- Code Editor:** Displays the same Python code as the first screenshot.
- Terminal:** Shows the command to install the psycopg2-binary package via pip, followed by a success message.
- Status Bar:** Shows the file path as "PS C:\Users\DELL\Desktop\Visual Studio> python test.py", line 25, column 20, and other status information like "Ln 25, Col 20 Spaces: 4 UTF-8 CRLF {} Python env (3.14.0)".

```
Successfully installed psycopg2-binary-2.9.11
[notice] A new release of pip is available: 25.2 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\DELL\Desktop\Visual Studio> python test.py
('Ram', 1, 31)
PS C:\Users\DELL\Desktop\Visual Studio> python test.py
1
PS C:\Users\DELL\Desktop\Visual Studio> python test.py
1
PS C:\Users\DELL\Desktop\Visual Studio> python test.py
31
PS C:\Users\DELL\Desktop\Visual Studio>
```

Build with Agent panel is visible on the right, showing AI responses and build actions.

Adding input from user:

The screenshot shows the Visual Studio Code interface. The left sidebar has 'EXPLORER' and 'V р VISUAL STUDIO' sections. The main editor window displays a Python script named 'test.py'. The code creates a 'employees' table and inserts data into it. A tooltip 'Build with Agent' is visible on the right, along with 'SUGGESTED ACTIONS' for 'Build Workspace' and 'Show Config'. The status bar at the bottom shows file path 'C:\Users\DELL\Desktop\Visual Studio>python test.py', line count 'Ln 21, Col 7', and other system details.

```
1 import psycopg2
2 def table():
3     connect = psycopg2.connect(dbname="postgres",user="postgres",password="11032001",host="localhost",port="5432")
4     cursor = connect.cursor()
5     cursor.execute(''create table employees(Name text, ID int, Age int);''')
6     print("Table created successfully")
7     connect.commit()
8     connect.close()
9
10 def data():
11     connect = psycopg2.connect(dbname="postgres",user="postgres",password="11032001",host="localhost",port="5432")
12     name = input("Enter name: ")
13     id = input("Enter ID: ")
14     age = input("Enter age: ")
15     cursor = connect.cursor()
16     query='''insert into employees(Name,ID,Age) values(%s,%s,%s);'''
17     cursor.execute(query,(name,id,age))
18     print("Data inserted successfully")
19     connect.commit()
20     connect.close()
21
22 data()
```

The screenshot shows a terminal window with the command '(env) C:\Users\DELL\Desktop\Visual Studio>python test.py' entered. The output shows the creation of the 'employees' table and the insertion of two rows of data ('Pragathi', '2', '24'). The terminal prompt then changes to '(env) C:\Users\DELL\Desktop\Visual Studio>'.

```
(env) C:\Users\DELL\Desktop\Visual Studio>python test.py
Enter name: Pragathi
Enter ID: 2
Enter age: 24
Data inserted successfully

(env) C:\Users\DELL\Desktop\Visual Studio>
```

The screenshot shows a terminal window with the command 'postgres=# select * from employees;' entered. The output displays the contents of the 'employees' table, showing two rows: 'Ram' with ID 1 and age 31, and 'Pragathi' with ID 2 and age 24. The terminal prompt then changes to 'postgres=# |'.

```
postgres=# select * from employees;
   name   | id | age
-----+-----+-----
  Ram    |  1 | 31
 Pragathi |  2 | 24
(2 rows)

postgres=# |
```