

URL GitHub : <https://github.com/readonebe14/learn-git/tree/master>

## 1. Create Table

The screenshot displays a database management tool interface. On the left, a sidebar shows the database structure with a tree view. The main area is divided into two panes. The top pane shows a SQL query: `SELECT * FROM latihan_users LIMIT 100`. Below the query, a table schema is displayed with columns: `id` (integer), `email` (text), `name` (text), `phone` (text), and `postal_code` (text). The bottom pane shows a Python script for creating a table in a PostgreSQL database using the `psycopg2` library.

```
import psycopg2

# Connection String
conn_string = "host=localhost port=5432 dbname=postgres user=postgres password=welcome123 sslmode=prefer connect_timeout=10"

# Membuat koneksi
conn = psycopg2.connect(conn_string)

# Membuat cursor
cur = conn.cursor()

# Create Table
cur.execute("""
CREATE TABLE IF NOT EXISTS latihan_users(
    id serial primary key,
    email text,
    name text,
    phone text,
    postal_code text
)
""")

# Commit perubahan ke database
conn.commit()
```

At the bottom of the interface, there is a prompt: "Type 'python' code here and press Shift+Enter to run".

## 2. Table from Files (users\_w\_postal\_code.csv & region.csv)

The screenshot displays a Jupyter Notebook interface with two SQL queries and their results. The left pane shows the database schema, including tables like `latihan_users`, `region`, `users_using_copy`, and `users_w_postal_code`. The main area shows two queries and their results.

**Query 1: users\_w\_postal\_code**

```
SELECT * FROM users_w_postal_code LIMIT 100
```

Cost: 2ms

	email	name	phone	postalZip
1	feugiat.metus@yahoo.edu	Jackson Boyer	1-326-267-1884	56-066
2	morbi@hotmail.org	Devin Wolfe	(315) 718-9194	572037
3	tempor.augue@google.cou	Dennis Blackwell	1-433-880-4656	51433
4	vulputate.posuere.vulputate	Maggie Hawkins	(779) 331-8323	965863
5	nisi.magna@google.couk	Oleg Hall	1-208-578-4206	80756

**Query 2: region**

```
SELECT * FROM region LIMIT 100
```

Cost: 3ms

	postalZip	region	country
1	37078	Jakarta	Indonesia
2	687233	Bandung	Indonesia
3	2495	Bali	Indonesia
4	5903	Semarang	Indonesia
5	4355	Banten	Indonesia

**Python Code:**

```
import pandas as pd
from sqlalchemy import create_engine

df_users_w_postal_code = pd.read_csv("/002_Documents/001_Private/L...")
df_region = pd.read_csv("/002_Documents/001_Private/LEARNING/DATA E...")

engine = create_engine("postgresql://postgres:welcome123@localhost:...")

df_users_w_postal_code.to_sql("users_w_postal_code", engine, if_exists="replace", index=False)
df_region.to_sql("region", engine, if_exists="replace", index=False)

engine.dispose()
```

Output: [1] ✓ 1.5s

### 3. Inserting Data

The screenshot displays a database management interface with a table named `users_using_copy` and a Python script for inserting data into a PostgreSQL database.

**Table Data:**

	id	email	name	phone	postal_code
1	1	feugiat.metus@yahoo.edu	Jackson Boyer	1-326-267-1884	56-066
2	2	morbi@hotmail.org	Devin Wolfe	(315) 718-9194	572037
3	3	tempor.augue@google.cou	Dennis Blackwell	1-433-880-4656	51433
4	4	vulputate.posuere.vulputat	Maggie Hawkins	(779) 331-8323	965863
5	5	nisi.magna@google.couk	Oleg Hall	1-208-578-4206	80756
6	6	feugiat.metus@yahoo.edu	Jackson Boyer	1-326-267-1884	56-066
7	7	morbi@hotmail.org	Devin Wolfe	(315) 718-9194	572037
8	8	tempor.augue@google.cou	Dennis Blackwell	1-433-880-4656	51433
9	9	vulputate.posuere.vulputat	Maggie Hawkins	(779) 331-8323	965863
10	10	nisi.magna@google.couk	Oleg Hall	1-208-578-4206	80756

**Python Script:**

```
import psycopg2

# Connection String
conn_string = "host=localhost port=5432 dbname=postgres user=postgres"

# Membuat koneksi
conn = psycopg2.connect(conn_string)

# Membuat cursor
cur = conn.cursor()

# Create Table
cur.execute("""
CREATE TABLE IF NOT EXISTS users_using_copy(
    id serial primary key,
    email text,
    name text,
    phone text,
    postal_code text
)
""")

with open ("/002_Documents/001_Private/LEARNING/DATA ENGINEER/Digital_Marketing/001_Data/001_Data.csv") as f:
    next(f)
    cur.copy_from(f, "users_using_copy", sep=";", columns=("email", "name", "phone", "postal_code"))

# Commit perubahan ke database
conn.commit()

# Menutup cursor dan koneksi
cur.close()
conn.close()
```

The script successfully inserts data from a CSV file into the `users_using_copy` table. The execution time is 0.0s.