PostgresSQL (Neon) with python

(To connect Python with PostgreSQL, we will use the psycopg2 package,)

PostgresSQL

PostgreSQL is a lightweight, free, and **open-source relational database**. Because of its proven architecture, reliability, data integrity, and smooth integration with other popular programming languages, such as Python and R, PostgreSQL is extremely well accepted by the industry, with companies of all sizes and regions using it.

Understanding psycopg2

In order to connect to a database that is already created in your system or on the Internet, you will have to instruct Python how to detect it. In other words, you will have to tell Python the database of your interest is a PostgreSQL database.
+there are several options in our case we will be choosing **psycopg2** :the most popular PostgreSQL database adapter for Python
documentation de psycopg2:
https://www.psycopg.org/docs/install.html#install-from-source

However, if you want to use psycopg2 straightforwardly, you could also install psycopg2-binary, a stand-alone version of the package, not requiring a compiler or external libraries. This is the preferred installation for new users.

```
pip install psycopg2-binary
```

Before creating the table, it’s important to explain how the connection instance you’ve just created works. In essence, the connection encapsulates a database session, and it allows you to execute SQL commands and queries, such as SELECT, INSERT, CREATE, UPDATE, OR DELETE, using the **cursor()** method, and to make changes persistent using the **commit()** method.
Once you have created the cursor instance, you can send commands to the database using the **execute()** method and retrieve data from a table using **fetchone()**, **fetchmany()**, or **fetchall().**

```
# Open a cursor to perform database operations
cur = conn.cursor()
# Execute a command: create datacamp_courses table
cur.execute("""CREATE TABLE datacamp_courses(
          course_id SERIAL PRIMARY KEY,
          course_name VARCHAR (50) UNIQUE NOT NULL,
          course_instructor VARCHAR (100) NOT NULL,
          topic VARCHAR (20) NOT NULL);
          """)
# Make the changes to the database persistent
conn.commit()
# Close cursor and communication with the database
cur.close()
conn.close()
```