

Creado por:

Isabel Maniega

PostgreSQL: Psycopg2 librería python

Será necesario realizar la instalación de la librería:

<https://pypi.org/project/psycopg2-binary/> (<https://pypi.org/project/psycopg2-binary/>).

<https://www.psycopg.org/docs/> (<https://www.psycopg.org/docs/>).

- pip install psycopg2-binary

Importamos las principales dependencias:

In []:

```
import psycopg2
from psycopg2 import sql
from psycopg2.extensions import ISOLATION_LEVEL_AUTOCOMMIT
```

Creación Base de datos:

Una vez instalada creamos un script de Python, ejemplo connection_sql.py, en ese script realizaremos la creación de la base de datos mediante:

In []:

```
def createDatabase():
    conn = psycopg2.connect(user="IsaMan", password="cursoPython",
                            host="localhost", port=5432)

    conn.set_isolation_level(ISOLATION_LEVEL_AUTOCOMMIT)

    cur = conn.cursor()

    try:
        cur.execute(sql.SQL("CREATE DATABASE {};").format(sql.Identifier("test")))
    except psycopg2.Error as e:
        print("Error al crear la base de datos: %s" % str(e))

    cur.close()
    conn.close()
```

```
def createDatabase():
    conn = psycopg2.connect(user="IsaMan", password="cursoPython",
                             host="localhost", port=5432)

    conn.set_isolation_level(ISOLATION_LEVEL_AUTOCOMMIT)

    cur = conn.cursor()

    try:
        cur.execute(sql.SQL("CREATE DATABASE {};").format(sql.Identifier("test")))
    except psycopg2.Error as e:
        print("Error al crear la base de datos: %s" % str(e))

    cur.close()
    conn.close()
```

donde:

- conn: es la conexión a la base de datos, especificar user, password, host y port
- conn.set_isolation_level() para realizar una conexión donde se mantengan los cambios, hacerlos definitivos
- cur: definimos el cursor para realizar las ejecuciones.
- cur.execute(): instrucción para ejecutar los comandos.

Creación de la Tabla:

Una vez creada la base de datos modificamos la conexión para conectarnos a una base de datos concreta:

- database="test"

In []:

```
conn = psycopg2.connect(database="test", user="IsaMan", password="cursoPython",
                        host="localhost", port=5432)

cur = conn.cursor()
```

Una vez que tenemos la base de datos realizada crearemos la tabla donde insertaremos los registros, esto lo realizamos:

In []:

```
def createTabla(cur):
    try:
        cur.execute("CREATE TABLE notas(name varchar(80), age int, grades real, dat
    except psycopg2.Error as e:
        print("Error crear la tabla: %s" % str(e))

# createTabla(cur)
```

```
def createTabla(cur):
    try:
        cur.execute("CREATE TABLE notas(name varchar(80), age int, grades real, date date);")
    except psycopg2.Error as e:
        print("Error crear la tabla: %s" % str(e))

# createTabla(cur)
```

Insertar un registro:

In []:

```
def insertar(cur):
    try:
        cur.execute("INSERT INTO notas VALUES(%s, %s, %s, %s);", ('Jose Perez', 215
    except psycopg2.Error as e:
        print("Error crear registro: %s" % str(e))

# insertar(cur)
```

```
def insertar(cur):
    try:
        cur.execute("INSERT INTO notas VALUES(%s, %s, %s, %s);", ('Jose Perez', 215, 8.6, '07/07/2022'))
    except psycopg2.Error as e:
        print("Error crear registro: %s" % str(e))

# insertar(cur)
```

Actualizar un registro:

In [1]:

```
def actualizar(cur):
    try:
        cur.execute("UPDATE notas SET grades=7.9 WHERE name='Isabel Maniega';")
    except psycopg2.Error as e:
        print("Error actualizar registro: %s" % str(e))

# actualizar(cur)
```

```
def actualizar(cur):
    try:
        cur.execute("UPDATE notas SET grades=7.9 WHERE name='Isabel Maniega';")
    except psycopg2.Error as e:
        print("Error actualizar registro: %s" % str(e))

# actualizar(cur)
```

Eliminar un registro:

In []:

```
def delete(cur):
    try:
        cur.execute("DELETE FROM notas WHERE name='Isabel Maniega'")
    except psycopg2.Error as e:
        print("Error actualizar registro: %s" % str(e))

# delete(cur)
```

```
def delete(cur):
    try:
        cur.execute("DELETE FROM notas WHERE name='Isabel Maniega'")
    except psycopg2.Error as e:
        print("Error actualizar registro: %s" % str(e))

# delete(cur)
```

Leer los registros:

In []:

```
def mostrar(cur):
    try:
        cur.execute("SELECT * FROM notas;")
        rows = cur.fetchall()
        for row in rows:
            print(row)
    except psycopg2.Error as e:
        print("Error mostrar registros: %s" % str(e))

mostrar(cur)
```

```
def mostrar(cur):
    try:
        cur.execute("SELECT * FROM notas;")
        rows = cur.fetchall()
        for row in rows:
            print(row)
    except psycopg2.Error as e:
        print("Error mostrar registros: %s" % str(e))

mostrar(cur)
```

In []:

```
conn.commit()

cur.close()
conn.close()
```

Cerrar el cursor y la conexión al final.

Creado por:

Isabel Maniega