

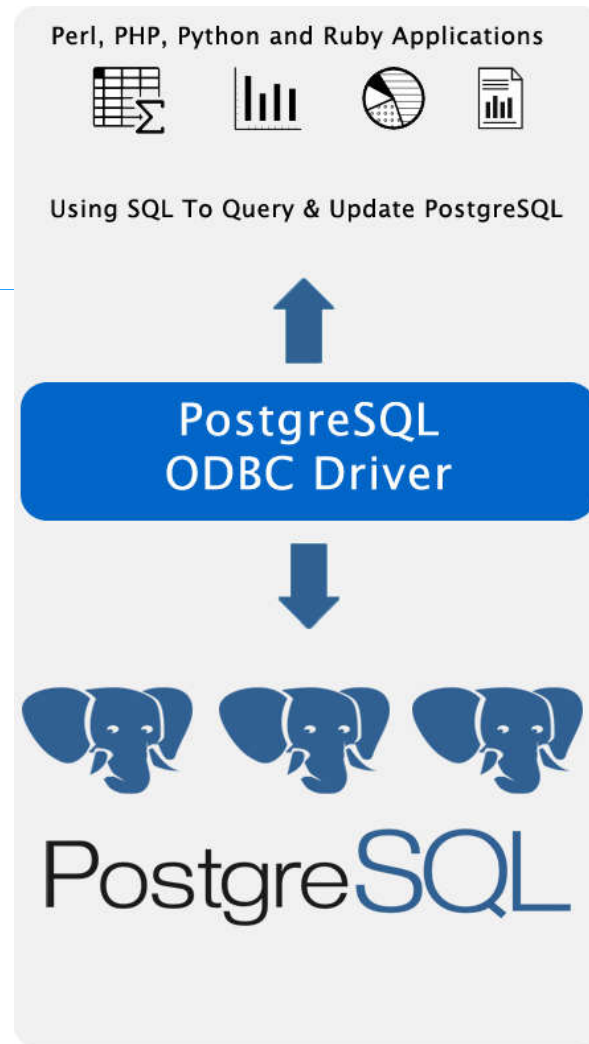
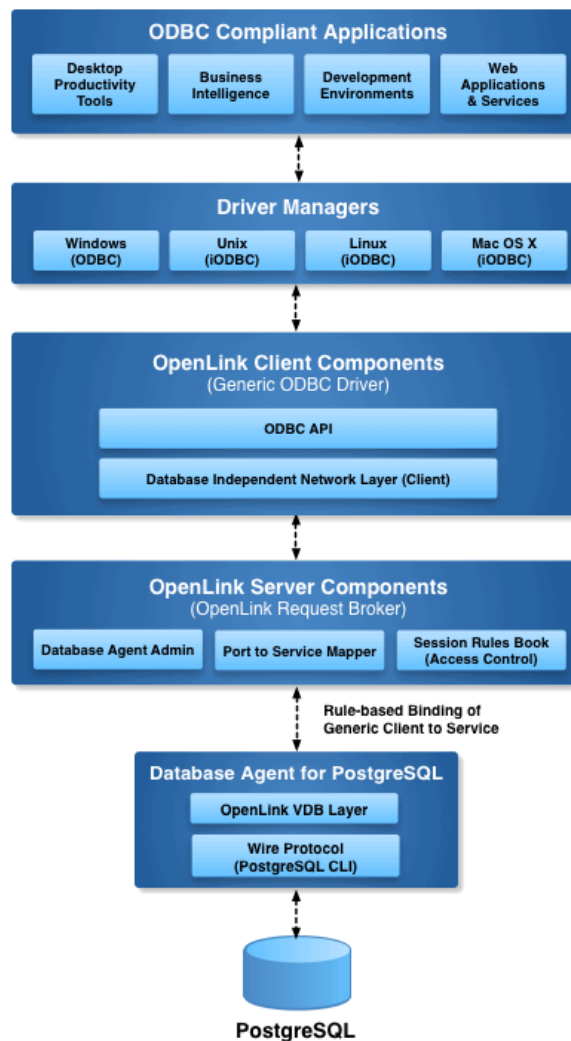
Kết nối postgres trong python

Nội dung

1. Psycpg2 connector
2. Kết nối postgres từ python với psycpg2
3. Excute query
4. Create, Insert, Update, Select, Delete
5. Bài tập thực hành

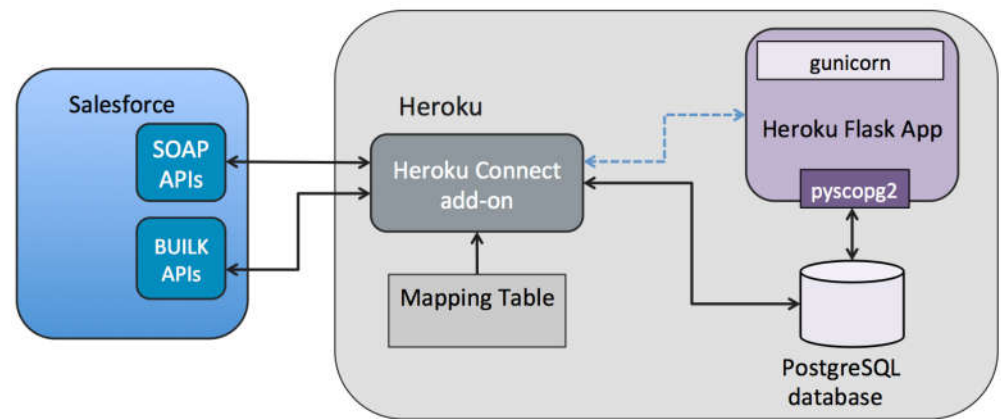
DB driver

❖ ODBC (*Open Database Connectivity*) driver: là một công nghệ Windows cho phép ứng dụng client nối với CSDL từ xa.



Postgres connector

- ❖ Để kết nối từ Python vào một cơ sở dữ liệu cần một database connector, nó là một thư viện dùng để giao tiếp với cơ sở dữ liệu.
- ❖ Đối với cơ sở dữ liệu postgres sử dụng: **psycopg2**.



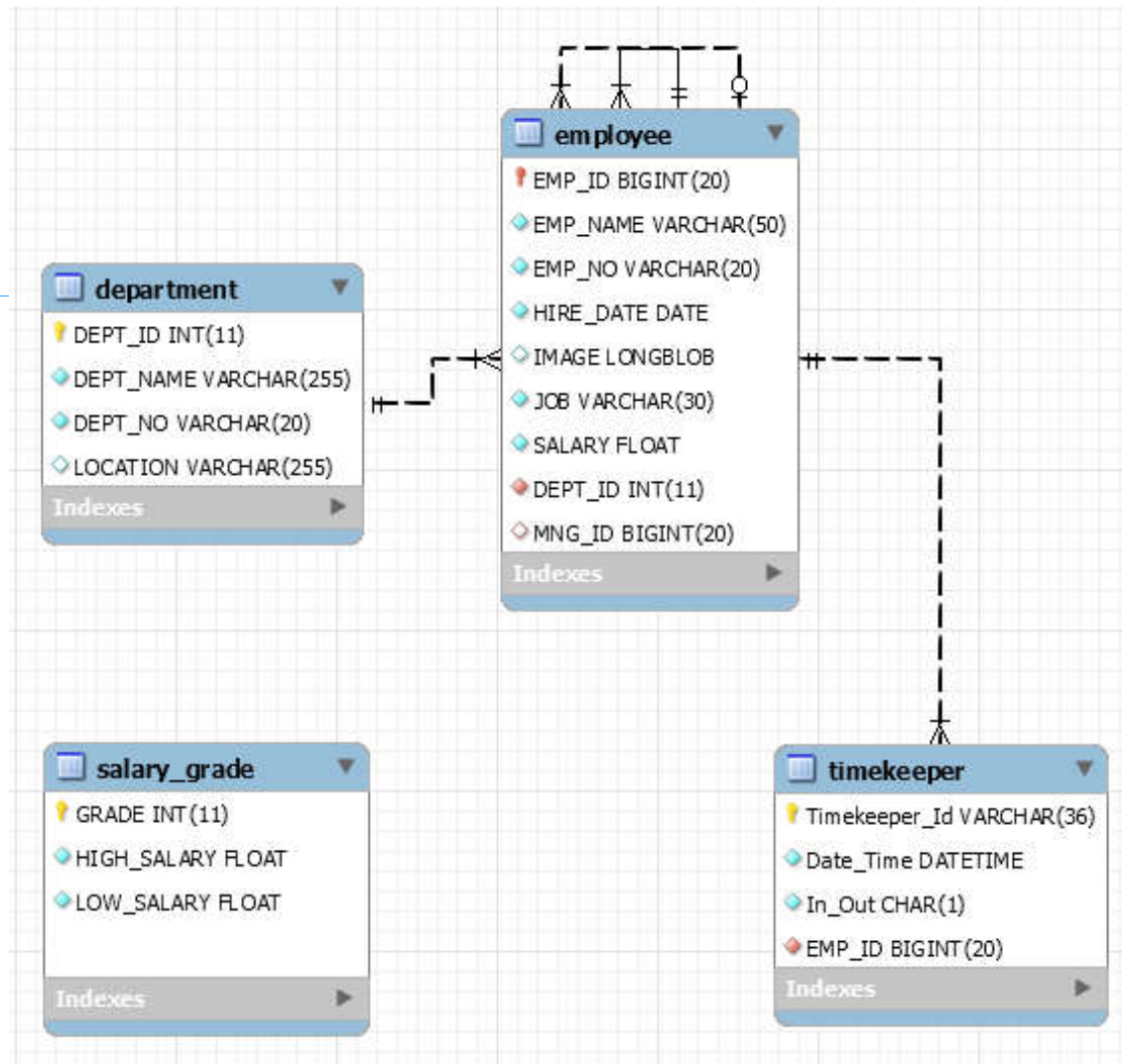
Flow python làm việc với DB

- ❖ Bước 1: Tạo Database trên postgres.
- ❖ Bước 2: Initial (khởi tạo) connection kết nối đến DB postgres (sử dụng **psycopg2** connector trên python).
- ❖ Bước 3: Open connection.
- ❖ Bước 4: Excuted các câu query thao tác với DB:
 - 4.1. Table: Create
 - 4.2. Record: Insert, Update, Select, Delete
- ❖ Bước 5: Close connection.

Tạo database

❖ Thực hiện tạo Database thực hành có tên 'itplusdb' để thực hành với các câu lệnh postgres trong Python. Bao gồm các bảng:

- DEPARTMENT
- EMPLOYEE
- SALARY_GRADE
- TIMEKEEPER



Initial & open connection

❖ Để kết nối được đến postgres , thực hiện khởi tạo đối tượng connection:

```
conn = psycopg2.connect(database='pythondb', user='openpg',  
                        password='openpgpwd', host='127.0.0.1', port='5432')
```


Module xử lý connectivity

- ❖ Nên tạo ra một module tiện ích để tạo một kết nối tới cơ sở dữ liệu. Ở đây module có tên "connectionutils", module này định nghĩa hàm `getConnection()` trả về một `connection`.

Excuted query

- ❖ Có 2 trường hợp excute query trong Python:
 - TH1: Câu query không có tham số truyền vào
 - TH2: Câu query có tham số truyền vào

Create table

```
from database.connectionutils import *
# connect to the PostgreSQL server
conn = get_connection()
# create cursor to execute sql statement
cur = conn.cursor()

def create_tables():
    """ Create new tables in the PostgreSQL database"""
    commands = (
        """
        CREATE TABLE department
        (
            dept_id integer NOT NULL,
            dept_name character varying(255) NOT NULL,
            dept_no character varying(20) NOT NULL,
            location character varying(255),
            CONSTRAINT department_pkey PRIMARY KEY (dept_id),
            CONSTRAINT department_dept_no_key UNIQUE (dept_no)
        )
        """)
    try:
        # create table one by one
        for command in commands:
            cur.execute(command)
        # close communication with the PostgreSQL database server
        cur.close()
        # commit the changes
        conn.commit()
    except (Exception, psycopg2.DatabaseError) as error:
        print(error)
    finally:
        if conn is not None:
            conn.close()
```

Insert data – No params

```
def insert_department():  
    """ insert a new vendor into the vendors table """  
    sql = "insert into Department (DEPT_ID, DEPT_NAME, DEPT_NO, LOCATION)" + \  
        "values (10, 'ACCOUNTING', 'D10', 'NEW YORK');"   
    try:  
        # execute the INSERT statement  
        cur.execute(sql)  
        conn.commit()  
        cur.close()  
        print('Insert successfully...')  
    except (Exception, psycopg2.DatabaseError) as error:  
        print('Error:', error)  
    finally:  
        if conn is not None:  
            conn.close()
```

Truy vấn truyền tham số

- ❖ Python sử dụng duy nhất mã lệnh %s như là một placeholder (tức là nơi tham số sẽ được truyền vào) cho các tham số.
- ❖ Placeholder trong python không phụ thuộc kiểu dữ liệu của tham số như một số ngôn ngữ lập trình khác.
- ❖ Ví dụ khi cần thực hiện 2 câu lệnh sau đối với DB bằng Python:

```
sql1 = "Insert into Department (Dept_Id, Dept_No, Dept_Name, Location)  
       values (%s, %s, %s, %s) "
```

```
sql2 = "Select * from Department Where Dept_Name = %s "
```

Insert data – Using input params

```
def insert_department():  
    """ insert a new vendor into the vendors table """  
    sql = "insert into Department (DEPT_ID, DEPT_NAME, DEPT_NO, LOCATION) " + \  
        "values (%s,%s,%s,%s);"  
  
    params = [20, 'RESEARCH', 'D20', 'DALLAS']  
    try:  
        # execute the INSERT statement  
        cur.execute(sql, (params))  
        conn.commit()  
        cur.close()  
        print('Insert successfully...')  
    except (Exception, psycopg2.DatabaseError) as error:  
        print('Error:', error)  
    finally:  
        if conn is not None:  
            conn.close()
```

Insert multi data

```
def insert_department():  
    """ insert a new vendor into the vendors table """  
    sql = "insert into Department (DEPT_ID, DEPT_NAME, DEPT_NO, LOCATION)" + \  
        "values (%s,%s,%s,%s);"  
  
    params = [[30, 'SALES', 'D30', 'CHICAGO'],  
              [40, 'OPERATIONS', 'D40', 'BOSTON'],  
              [50, 'ITPLUS', 'D50', 'HA NOI']]  
  
    try:  
        # execute the INSERT statement  
        for param in params:  
            cur.execute(sql, param)  
        conn.commit()  
        cur.close()  
        print('Insert successfully...')  
    except (Exception, psycopg2.DatabaseError) as error:  
        print('Error:', error)  
    finally:  
        if conn is not None:  
            conn.close()
```

Update

```
def update_department():  
    """ insert record to table in the PostgreSQL database """  
    # sql = "SELECT * FROM Department"  
    sql = "UPDATE department SET location = %s where dept_name = %s"  
    try:  
        # execute a statement  
        cur.execute(sql, ('HO CHI MINH', 'SALES', ))  
        conn.commit()  
        cur.close()  
        print('Update successfully...')  
    except (Exception, psycopg2.DatabaseError) as error:  
        print('Error: ', error)  
    finally:  
        if conn is not None:  
            conn.close()
```


Select (Truy vấn) – No params

```
def query():  
    """ insert record to table in the PostgreSQL database """  
    sql = "SELECT * FROM Department"  
    try:  
        # execute a statement  
        cur.execute(sql)  
        print("The number of rows: ", cur.rowcount)  
        # get first line of result  
        row = cur.fetchone()  
        while row is not None:  
            print(row)  
            row = cur.fetchone()  
        # close communication with the PostgreSQL database server  
        cur.close()  
  
    except (Exception, psycopg2.DatabaseError) as error:  
        print('Error: ', error)  
    finally:  
        if conn is not None:  
            conn.close()
```

Select (Truy vấn) – Using input params

```
def query():
    """ insert record to table in the PostgreSQL database"""
    # sql = "SELECT * FROM Department"
    sql = "SELECT * FROM department where dept_name = %s"
    try:
        # execute a statement
        cur.execute(sql, ('ITPLUS',))
        print("The number of rows: ", cur.rowcount)
        # get first line of result
        row = cur.fetchone()
        while row is not None:
            print(row)
            row = cur.fetchone()
        # close communication with the PostgreSQL database server
        cur.close()
    except (Exception, psycopg2.DatabaseError) as error:
        print('Error: ', error)
    finally:
        if conn is not None:
            conn.close()
```

Bài tập

❖Viết chương trình python để:

- A) Lấy ra danh sách các nhân viên có chức vụ là MANAGER
- B) Insert thông tin phòng làm việc thực tế của bạn vào bảng department
- C) Insert thông tin thực tế của bản thân bạn vào bảng employee
- D) Cập nhật thông tin của nhân viên có tên là CLARK trong bảng deployment thành thông tin cá nhân của bạn
- E) Xóa thông tin của nhân viên có tên là MILLER trong bảng employee

THANKS FOR ATTENDING!