

240-216 Computer Engineering Laboratory

PostgreSQL

วัตถุประสงค์

1. รู้จักการใช้งานฐานข้อมูล PostgreSQL
2. รู้จักการใช้งาน PGADMIN กับฐานข้อมูล PostgreSQL
3. ทดลองเขียนโค้ดด้วยภาษา Python เพื่อเชื่อมต่อฐานข้อมูล PostgreSQL

กำหนดส่งงานและวิธีส่งงาน

ติดตามวิธีการส่งงานผ่านทาง LMS หรือ Facebook

สัดส่วนคะแนน

- Checkpoint ในระหว่างการทดลอง 80%
- สอบ 20%

งานก่อนการทดลอง

ให้นักศึกษาทบทวนการใช้งานด้วย Docker

Checkpoint งานหลังการทดลอง และสอบ

ส่งงานในระบบ LMS

1 . Introduction to PostgreSQL

ศึกษาความรู้พื้นฐานของระบบฐานข้อมูล PostgreSQL

<https://www.postgresql.org/docs/16/index.html>

<https://www.geeksforgeeks.org/postgresql-tutorial/?ref=lbp>

Checkpoint #1 (10 คะแนน)

อธิบายความสามารถของ PostgreSQL ข้อดี และข้อเสีย

2 . PostgreSQL with Docker

2.1 สร้างไฟล์ docker-compose.yml

การกำหนดค่าตัวแปรในรูปแบบต่างๆ

```
version: '3.4'

services:
  postgresql:
    image: docker.io/library/postgres:15
    restart: unless-stopped
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -d ${POSTGRES_DB} -U ${POSTGRES_USER}"]
      start_period: 20s
      interval: 60s
      retries: 5
      timeout: 5s
    ports:
      - 5432:5432
    volumes:
      - postgresql:/var/lib/postgresql/data
    environment:
      - POSTGRES_PASSWORD=CoEpasswd
      - POSTGRES_USER=coe
      - POSTGRES_DB=coedb
    logging:
      options:
        max-size: "10m"
        max-file: "3"

  pgadmin:
    container_name: pgadmin
    image: dpage/pgadmin4
    restart: unless-stopped
    volumes:
      - pgadmin:/var/lib/pgadmin
    environment:
      - PGADMIN_DEFAULT_EMAIL=coe@local.db
      - PGADMIN_DEFAULT_PASSWORD=CoEpasswd
    ports:
      - 7443:443
      - 7080:80
    logging:
      options:
        max-size: "10m"
        max-file: "3"
```

```
volumes:
  postgresql:
    driver: local
  pgadmin:
    driver: local
```

2.2 เริ่มการทำงาน PostgreSQL และ PGADMIN

เริ่มการทำงาน PostgreSQL และ PGADMIN โดยใช้คำสั่ง docker compose หรือ docker-compose

```
$ docker compose up -d
$ docker compose ps
```

| Name | Command | State | Ports |
|-------------------|--------------------------------------|--------------|--|
| code_postgresql_1 | docker- entrypoint.sh postgres | Up (healthy) | 0.0.0.0:5432->5432/tcp, :::5432->5432/tcp |
| pgadmin | /entrypoint.sh | Up | 0.0.0.0:7443->443/tcp, :::7443->443/tcp, 0.0.0.0:7080->80/tcp, :::7080->80/tcp |

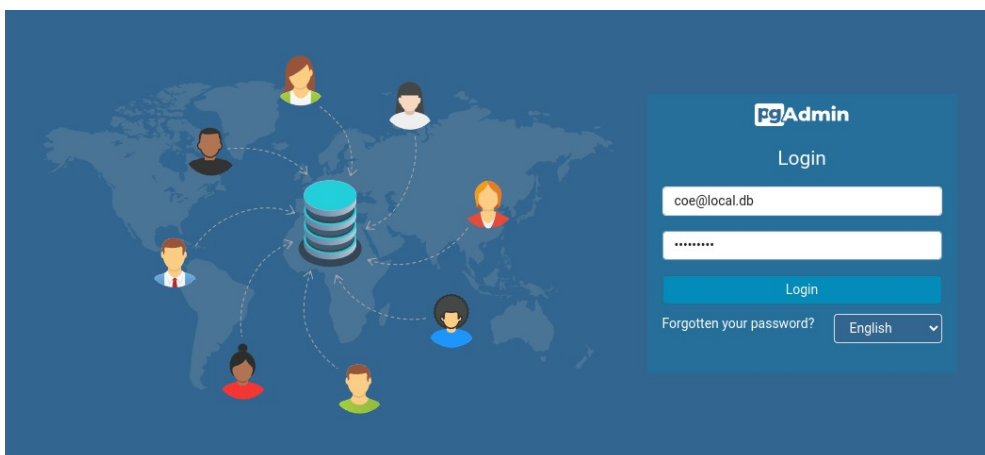
Checkpoint #2 (10 คะแนน)

จับภาพหน้าจอแสดงให้เห็นเซิร์ฟเวอร์ของ PostgreSQL และ PGADMIN ส่งเข้าระบบ

3 . PGADMIN

3.1 เปิดเว็บเบราว์เซอร์ url: <http://localhost:7080>

จะปรากฏดังภาพ

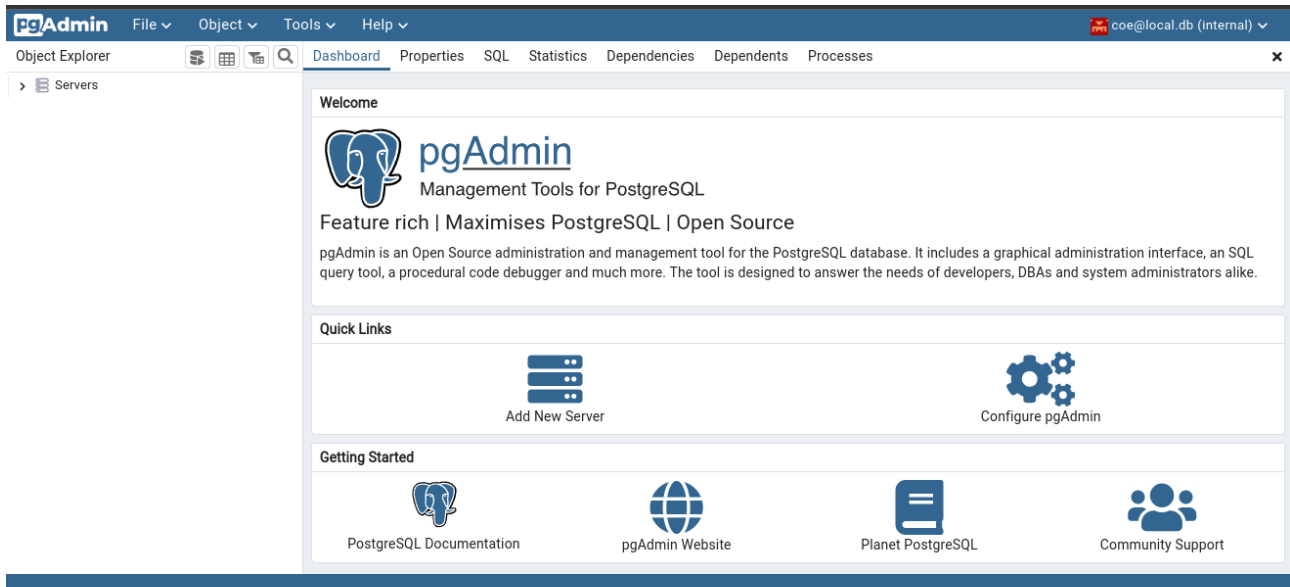


3.2 Login

เข้าสู่ระบบโดยใช้ username และ password

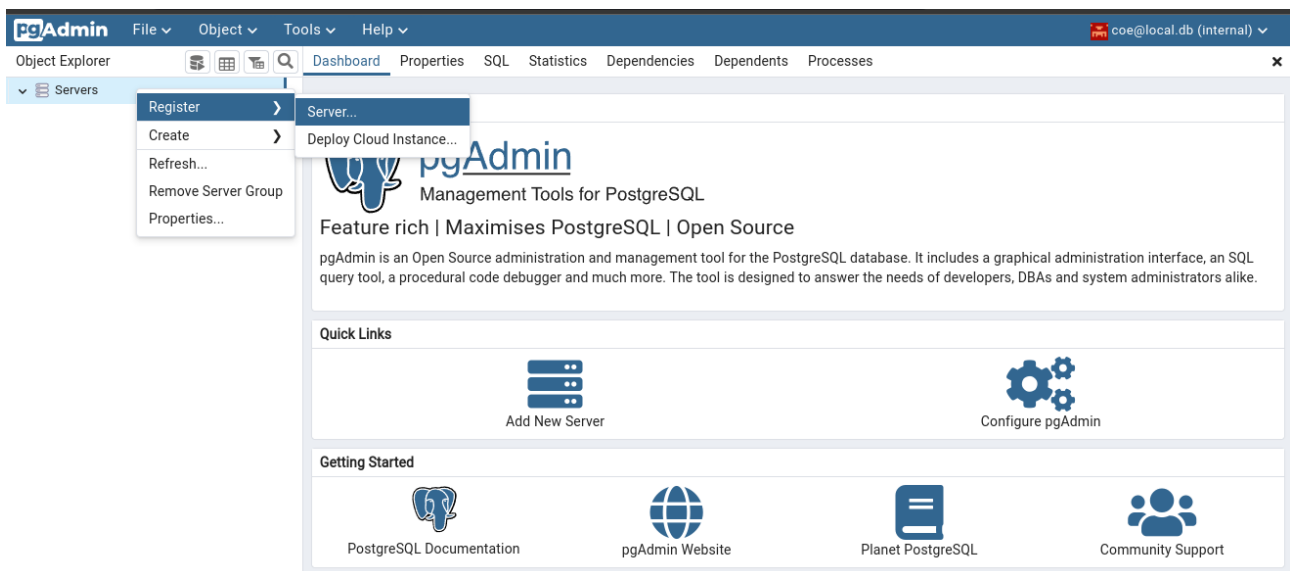
username: coe@local.db

password: CoEpasswd



3.3 Database Registration

เริ่มลงทะเบียน Database



กรอกรข้อมูลดังภาพ

Register - Server

General Connection Parameters SSH Tunnel Advanced

Name: postgresql

Server group: Servers

Background: ☐

Foreground: ☐

Connect now?: ☒

Shared?: ☐

Comments:

Either Host name or Service must be specified.

Close Reset Save

กดแท็บ Connection กรอกรข้อมูลดังภาพ ใช้ username: coe และ password: CoEpasswd เสร็จแล้วกด Save

Register - Server

General Connection Parameters SSH Tunnel Advanced

Host name/address: postgresql

Port: 5432

Maintenance database: postgres

Username: coe

Kerberos authentication?: ☐

Password: CoEpasswd

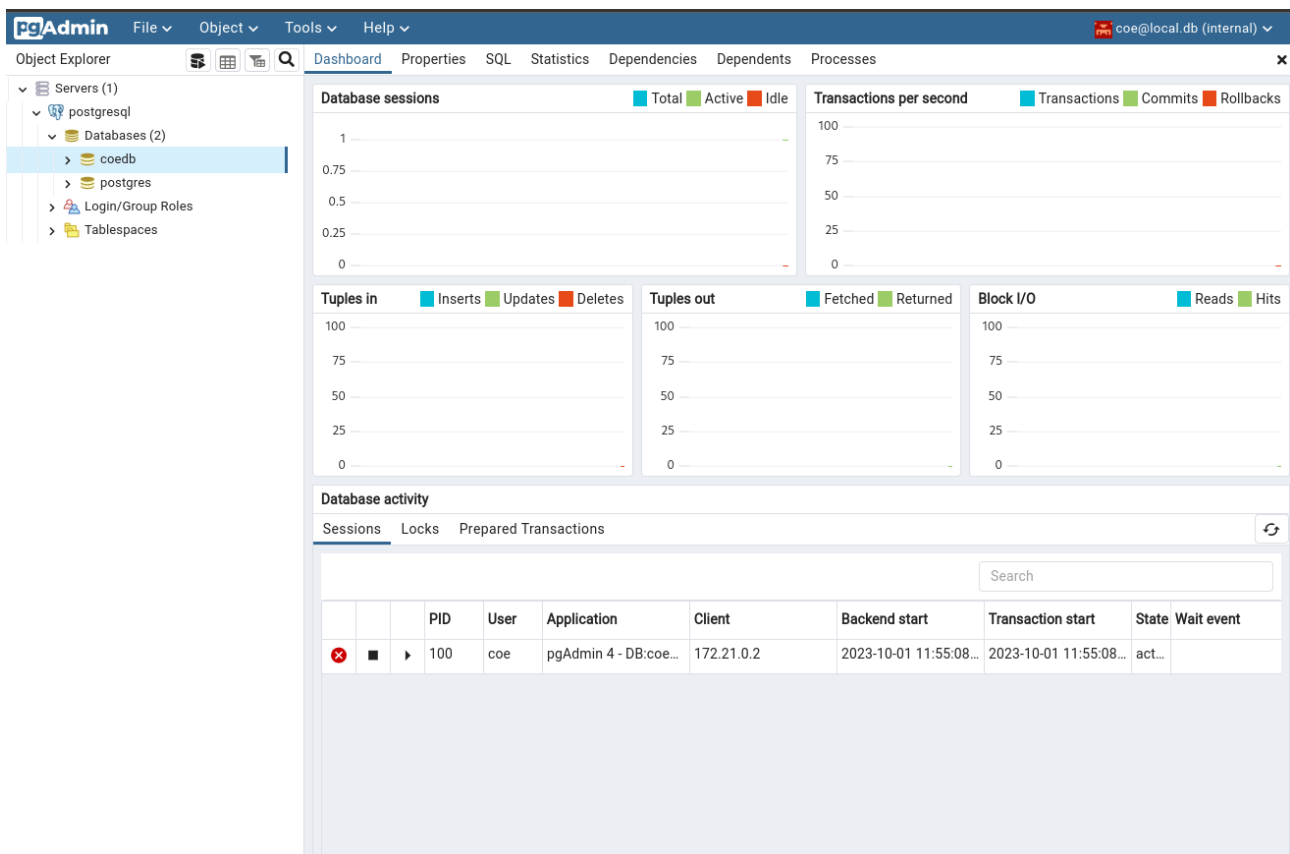
Save password?: ☒

Role:

Service:

Close Reset Save

เรียบร้อยแล้วจะได้ข้อมูลดังภาพ



Checkpoint #3 (10 คะแนน)

จับภาพหน้าจอส่งเข้าระบบ

3.4 สร้าง Table และเพิ่มข้อมูล

ทดลองสร้าง Table ชื่อ Activities

The screenshot shows the 'Create - Table' dialog box in pgAdmin 4. The 'General' tab is selected, and the following fields are filled:

- Name:** activities
- Owner:** coe
- Schema:** public
- Tablespace:** Select an item...
- Partitioned table?:** ☐
- Comment:** (empty text area)

At the bottom of the dialog, there are buttons for 'Close', 'Reset', and 'Save'.

กดแท็บ columns แล้วสร้าง columns ดังภาพ แล้วกด Save

Create - Table

General

Columns

Advanced

Constraints

Partitions

Parameters

Security

SQL

Inherited from table(s)

Select to inherit from...

Columns

| | | Name | Data type | Length/Precision | Scale | Not NULL? | Primary key? | Default |
|---|---|------|--------------|-----------------------|-------|-------------------------------------|-------------------------------------|---------|
| ⋮ | ✎ | ✖ | id | serial | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| ⋮ | ✎ | ✖ | name | character varying | 255 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| ⋮ | ✎ | ✖ | description | text | | <input type="checkbox"/> | <input type="checkbox"/> | |
| ⋮ | ✎ | ✖ | created_date | timestamp with tim... | | <input checked="" type="checkbox"/> | <input type="checkbox"/> | now() |
| ⋮ | ✎ | ✖ | updated_date | timestamp with tim... | | <input checked="" type="checkbox"/> | <input type="checkbox"/> | now() |

?

?

Close

Reset

Save

ทดลอง เพิ่มข้อมูลใน Table

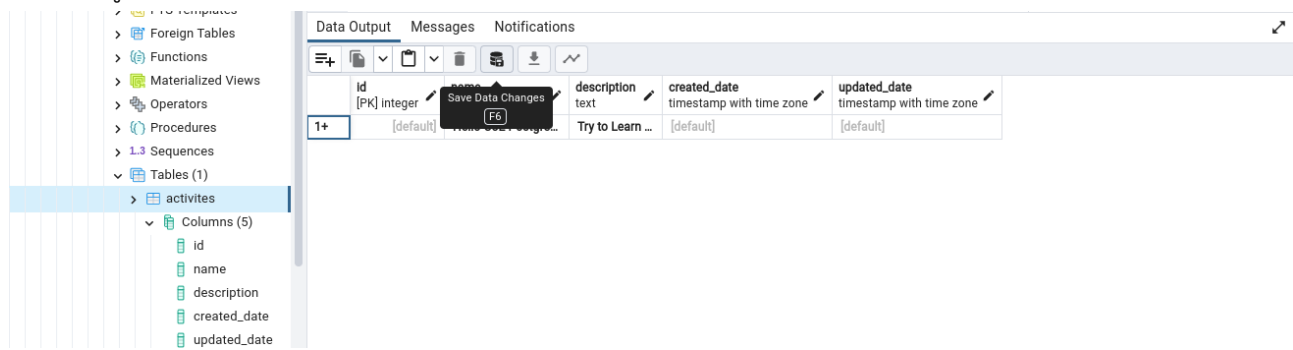
The screenshot displays the pgAdmin 4 web interface. On the left, the **Object Explorer** shows a tree structure: **Servers (1)** > **postgresql** > **Databases (2)** > **coedb** > **Schemas (1)** > **public** > **activities**. The **activities** table is selected, showing its columns: **id**, **name**, **description**, **created_date**, and **updated_date**.

The main area is the **Query Tool**, showing a SQL query:


```
SELECT * FROM public.activities
ORDER BY id ASC
```

 The **Query History** tab is active, showing the executed query. Below the query, the **Table Data View** for the **activities** table is displayed, showing columns: **id**, **name**, **description**, **created_date**, and **updated_date**. The **View/Edit Data** dropdown menu is open, showing options: **All Rows**, **First 100 Rows**, **Last 100 Rows**, and **Filtered Rows...**.

กรอกข้อมูลแล้ว Save



จะได้ตัวอย่างข้อมูลดังภาพ

| Data Output Messages Notifications | | | | | |
|------------------------------------|--------------|----------------------|------------------|-----------------------------|-----------------------------|
| | id | name | description | created_date | updated_date |
| | [PK] integer | character varying | text | timestamp with time zone | timestamp with time zone |
| 1 | 1 | Hello CoE Postgre... | Try to Learn ... | 2023-10-01 12:12:10.2016... | 2023-10-01 12:12:10.2016... |

Checkpoint #4 (10 คะแนน)

ทดลองเพิ่มข้อมูลเพิ่มเติมอย่างน้อย ตามรหัสนักศึกษา % 5 + 3 เช่น รหัส 6610110000 ให้เพิ่มข้อมูล 3 เร็คคอร์ด จับภาพหน้าจอแล้วส่งเข้าระบบ

3.5 SQL

หาข้อมูลเพิ่มเติมจาก Internet แล้วเขียน SQL ผ่านทาง PGADMIN เพื่อจัดการข้อมูลในตาราง activities ดังต่อไปนี้

- INSERT
- UPDATE
- DELETE

Checkpoint #5 (10 คะแนน)

อธิบายการเขียนโปรแกรม SQL อย่างละเอียดแล้วส่งเข้าระบบ

4 . SQLAlchemy

4.1 SQLAlchemy

ศึกษาข้อมูลจาก <https://www.sqlalchemy.org/>

Checkpoint #6 (10 คะแนน)

SQLAlchemy คืออะไร

4.2 ศึกษา Code จากโปรแกรม

Download ตัวอย่างโค้ดจาก [psunote.tar.gz](https://psunote.com/psunote.tar.gz)

ทดลองรันโค้ด

```
$ python -m pip install -r requirements.txt
$ python psunote/noteapp.py
* Serving Flask app 'noteapp'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production
WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 118-293-318
```

หลังจากรันโค้ดเรียบร้อยแล้วเปิดเว็บเบราว์เซอร์ <http://localhost:5000> จะได้ดังภาพ

Hello PSU Note

Create

4.3 แก้ไขโค้ดให้สามารถใช้งานได้ดังต่อไปนี้

1. แก้ไขโค้ด
2. แก้ไข Tag
3. ลบโค้ด
4. ลบ Tag

Checkpoint #7 (20 คะแนน)

เขียนโค้ดเพิ่มเติมพร้อมทั้งเก็บโค้ดเข้า Git ให้เรียบร้อย ส่งลิงค์ Git เข้าระบบ