

THỰC HÀNH LAB01

Nội dung thực hành:

- Cài đặt PostgreSQL
 - Tạo, xóa DATABASE
 - Tạo, chỉnh sửa, xoá TABLE với khóa chính, khoá ngoại, các ràng buộc
 - Nhập, chỉnh sửa, xoá dữ liệu trong TABLE
 - Truy xuất dữ liệu cơ bản
-

1. Cài đặt trực tiếp vào máy

<https://www.enterprisedb.com/docs/supported-open-source/postgresql/installing/windows/>

2. Sử dụng Docker

i. Cài đặt Docker

<https://docs.docker.com/>

ii. Cài đặt PostgreSQL

<https://www.commandprompt.com/education/how-to-create-a-postgresql-data-base-in-docker/>

```
docker pull postgres
```

```
docker run --name postgresQL -p 5432:5432 -e
```

```
POSTGRES_PASSWORD=secret123 -d postgres
```

```
docker exec -it postgresQL bash
```

- Kết nối PostgreSQL Database Server

```
psql -h localhost -U postgres
```

Trong Command Prompt của Window:

```
\cd "C:\Program Files\PostgreSQL\16\bin"
```

```
psql -h localhost -U postgres
```

- Tạo một database tên mydb

```
CREATE DATABASE mydb;
```

- Xem danh sách các database hiện có

`\l`

- Kết nối với database mydb

`\c mydb;`

Xóa DATABASE:

- Xóa kết nối tới database cần xóa

```
SELECT pg_terminate_backend (pg_stat_activity.pid)
FROM pg_stat_activity
WHERE pg_stat_activity.datname='mydb';
```

- Xóa DATABASE

```
DROP DATABASE mydb;
```

3. Các câu lệnh cơ bản

3.1 Tạo bảng: **CREATE TABLE table-name (column-name datatype [constraints]);**

Tạo bảng `friend` gồm các thuộc tính: id, firstname, lastname, city, state, age.

Trong đó id (khoá chính) là số nguyên tự động tăng, age ≥ 0 .

```
CREATE TABLE friend(
    id          SERIAL PRIMARY KEY,
    firstname   VARCHAR(15), – character varying (15)
    lastname    VARCHAR(15),
    city        CHAR(10),
    state       CHAR(2), – character (2)
    age        INTEGER CHECK (age  $\geq$  0)
);
```

Xem cấu trúc bảng friend trong command prompt

`\d friend`

Các kiểu dữ liệu cơ bản

Category	Type	Description
character string	CHAR(length) VARCHAR(length)	blank-padded string, fixed storage length variable storage length
number	INTEGER FLOAT NUMERIC(precision, decimal)	integer, +/-2 billion range floating point number, 15-digit precision number with user-defined precision and decimal location
date/time	DATE TIME TIMESTAMP	date time date and time

DATE: 'YYYY-MM-DD'

TIME: 'HH:MM:SS'

TIMESTAMP: 'YYYY-MM-DD HH:MM:SS'

TIMESTAMPTZ: TZ = time zone

3.2 Nhập dữ liệu vào bảng: **INSERT INTO table-name (column-names) VALUES (column-values);**

```
INSERT INTO friend (firstname, lastname, city, state, age) VALUES
    ('Mike', 'Nichols', 'Tampa', 'FL', 19),
    ('Jim', 'Barnes', 'Ocean City', 'NJ', 25),
    ('Dick', 'Gleason', 'Ocean City', 'NJ', 24),
    ('Dean', 'Yeager', 'Plymouth', 'MA', 24);
INSERT INTO friend (age, state, city, lastname, firstname) VALUES (23, 'CO',
'Denver', 'Anderson', 'Cindy');
INSERT INTO friend (lastname, age, state, city) VALUES ('Jackson', 22, 'PA',
'Allentown');
INSERT INTO friend (firstname, city, state) VALUES ('Mark', 'Indiana', 'IN');
```

Copy dữ liệu từ file csv vào bảng

COPY table-name FROM 'file-name.csv' DELIMITER '|' CSV HEADER;

3.3 Giá trị **DEFAULT**

```
CREATE TABLE account (
    account_id SERIAL PRIMARY KEY,
    name VARCHAR(32),
    balance NUMERIC(16,2) DEFAULT 0,
    active CHAR(1) DEFAULT 'Y',
```

```
        created      TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
        friend_id    INTEGER REFERENCES friend(id) ON DELETE CASCADE  
    );
```

```
INSERT INTO account (name, friend_id) VALUES ('Federated Builders', 1);
```

3.4 Hiển thị dữ liệu: **SELECT column-names FROM table-name WHERE condition;**

```
SELECT * FROM friend;           # hiển thị tất cả dữ liệu trong bảng friend
```

```
SELECT city FROM friend;        # chỉ hiển thị thông tin của cột city
```

Hiển thị firstname, city của những người có độ tuổi lớn hơn hoặc bằng 22

```
SELECT firstname, city FROM friend WHERE age>22;
```

Hiển thị thông tin của những người có firstname là Sam

```
SELECT * FROM friend WHERE firstname='Sam';
```

3.5 Đặt lại tên cột khi hiển thị

```
SELECT firstname AS buddy FROM friend ORDER BY buddy;
```

3.6 Sắp xếp dữ liệu: **ORDER BY**

Hiển thị dữ liệu trong bảng friend theo độ tuổi giảm dần

```
SELECT * FROM friend ORDER BY age DESC;
```

3.7 **WHERE AND, OR**

```
SELECT * FROM friend WHERE firstname = 'Sam' AND lastname = 'Jackson';
```

```
SELECT * FROM friend WHERE state = 'PA' OR city = 'Denver';
```

3.8 **WHERE BETWEEN ... AND**

Hiển thị thông tin của những người có độ tuổi từ 22 đến 25, sắp xếp theo thứ tự chữ cái của firstname

```
SELECT * FROM friend WHERE age >= 22 AND age <= 25 ORDER BY  
firstname;
```

hoặc

```
SELECT * FROM friend WHERE age BETWEEN 22 AND 25 ORDER BY  
firstname;
```

3.9 WHERE LIKE

Hiển thị thông tin của những người có firstname bắt đầu bằng chữ cái D

```
SELECT * FROM friend WHERE firstname LIKE 'D%';
```

Comparison	Operation
begins with D	LIKE 'D%'
contains a D	LIKE '%D%'
has D in second position	LIKE '_D%'
begins with D and contains e	LIKE 'D%e%'
begins with D, contains e, then f	LIKE 'D%e%f%'
begins with non-D	NOT LIKE 'D%'

3.10 SELECT DISTINCT

```
SELECT DISTINCT state FROM friend ORDER BY state;
```

3.11 Cập nhật dữ liệu: UPDATE table-name SET new value WHERE condition;

Cập nhật độ tuổi là 20 của người có firstname là Cindy

```
UPDATE friend SET age=20 WHERE firstname='Cindy';
```

3.12 Xóa dữ liệu: DELETE FROM table-name WHERE condition;

Xóa thông tin của những người có độ tuổi bằng 19

```
DELETE FROM friend WHERE age=19;
```

3.13 Xóa tất cả dữ liệu của 1 bảng

```
TRUNCATE TABLE account;
```

3.14 Xóa bảng: DROP TABLE table-name

```
DROP TABLE account;
```

3.15 Store schema

```
:/# pg_dump -h localhost -U postgres -s mydb > schema.sql
```

3.16 Lưu dữ liệu ra file text

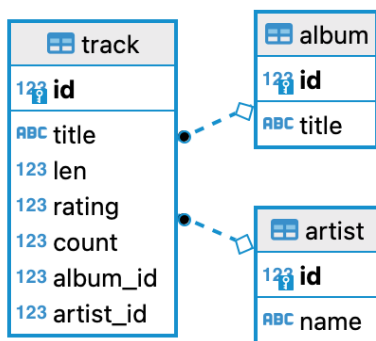
```
mydb=# \o output.txt
```

```
mydb=# SELECT * FROM friend;
mydb=# \o
mydb=# \q – quit ra khỏi database mydb
```

Bài tập

1. Tạo bảng myemployees gồm các thuộc tính: employee_id, firstname, lastname, title, age và salary với các ràng buộc sau: employee_id (khoá chính) duy nhất và NOT NULL, title NOT NULL và giá trị mặc định là ' ' (BLANK), age không âm, salary tối đa là 100000.
2. Nhập các dữ liệu sau vào bảng myemployees
 - 1, Jonie, Weber, Secretary, 28, 19500
 - 2, Potsy, Weber, Programmer, 32, 45300
 - 3, Dirk, Smith, Programmer II, 45, 75020
 - 4, Mike, Nicols, Programmer, 25, 35000
 - 5, Jim, Smith, Secretary, 24, 17000
 - 6, Dean, Yeager, Programmer II, 39, 73000
 - 7, Mark, Middleton, NULL , 21, 10000
3. Hiển thị tất cả dữ liệu trong bảng myemployees
4. Hiển thị những nhân viên có mức lương dưới 30000
5. Hiển thị firstname, lastname của những nhân viên có độ tuổi trên 30
6. Hiển thị firstname, lastname, salary của nhân viên có title là 'Programmer'
7. Hiển thị thông tin của những nhân viên có lastname chứa 'ebe'
8. Hiển thị thông tin của nhân viên có firstname là 'Potsy'
9. Hiển thị thông tin của những nhân viên có lastname kết thúc bằng 'ith'
10. Jonie Weber vừa kết hôn với Bob Williams. Đổi họ tên của Jonie Weber lại thành Jonie Williams.
11. Hôm nay là sinh nhật của Dirk Smith, cập nhật tuổi của anh ta (cộng thêm 1 vào tuổi)
12. Công ty đổi title 'Secretary' thành 'Administrative Assistant'. Cập nhật lại dữ liệu.
13. Các nhân viên có mức lương nhỏ hơn 30000 được tăng thêm 3500
14. Các nhân viên có mức lương trên 33500 được tăng thêm 4500

15. Các nhân viên có title “Programmer II” được thăng chức thành “Programmer III” và nhân viên có title “Programmer” được thăng chức thành “Programmer II”.
16. Jonie Williams vừa xin nghỉ việc, xóa thông tin của cô ấy ra khỏi bảng
17. Hiện tại đang thực hiện việc cắt giảm ngân sách, xóa những nhân viên có mức lương trên 70000
18. Tạo 1 database tên music với owner là tên của bạn, encoding ‘UTF8’
19. Tạo các bảng với khóa chính và khóa ngoại theo mô hình thực thể bên dưới
20. Nhập dữ liệu cho database music từ file các file: ‘artist.csv’, ‘album.csv’, ‘track.csv’



Bài tập nâng cao (không bắt buộc):

Nhập dữ liệu cho database music ở trên từ file ‘track_raw.csv’