



CHƯƠNG 4: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MÃ NGUỒN MỞ

Giảng viên: ThS. Lê Tuấn Anh

Điện thoại: 034.619.6934

Email: anhlt@dainam.edu.vn

CHƯƠNG 4: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MÃ NGUỒN MỞ

1

Tổng quan

2

PostgreSQL

3

MySQL

- Hệ quản trị cơ sở dữ liệu mã nguồn mở (Open-source DBMS) là phần mềm cho phép người dùng quản lý, lưu trữ và truy vấn dữ liệu mà không phải trả phí bản quyền.
- Một số DBMS mã nguồn mở phổ biến: PostgreSQL, MySQL, MariaDB, SQLite.



PostgreSQL

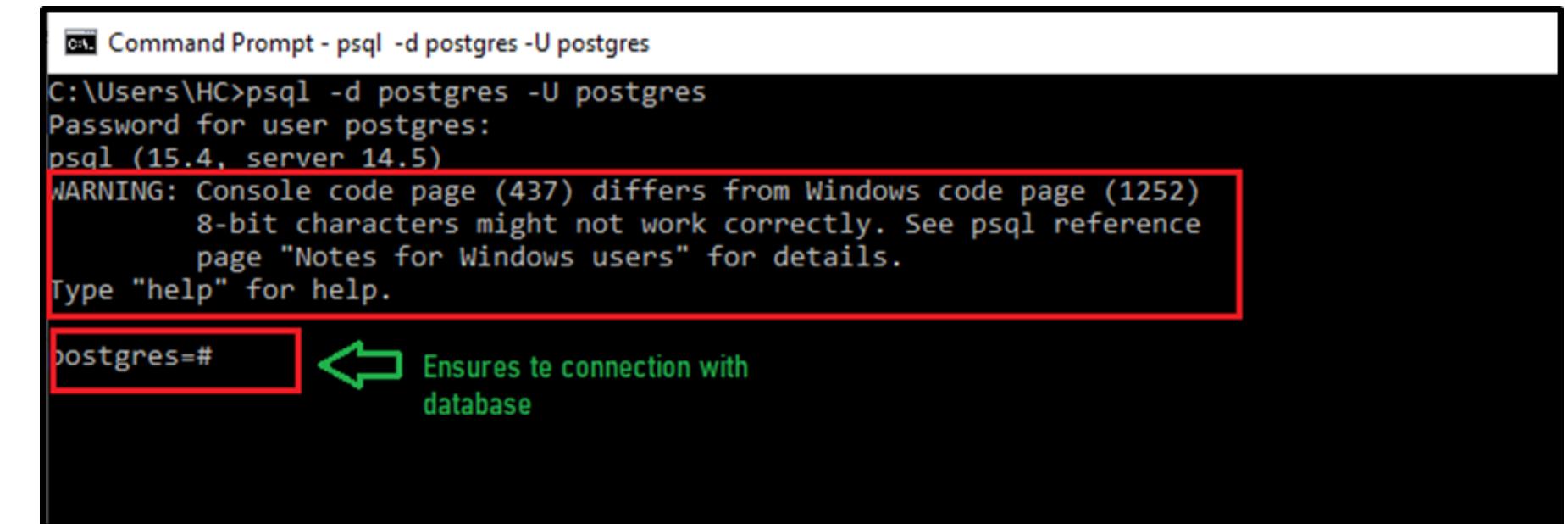
❖ Giới thiệu:

- PostgreSQL là hệ quản trị CSDL quan hệ mạnh mẽ, mã nguồn mở, miễn phí
- Hỗ trợ SQL tiêu chuẩn, JSON, stored procedure, transaction, mã hóa, ACID, và mở rộng dễ dàng
- Có thể dùng để xây dựng hệ thống quản lý sinh viên, ngân hàng, kế toán, thống kê,...



❖ Quản trị CSDL PostgreSQL:

- Công cụ phổ biến:
 - *pgAdmin*: giao diện đồ họa quản lý PostgreSQL
 - *psql*: công cụ dòng lệnh



A screenshot of a Windows Command Prompt window titled "Command Prompt - psql -d postgres -U postgres". The window shows the following text:

```
C:\Users\HC>psql -d postgres -U postgres
Password for user postgres:
psql (15.4, server 14.5)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

postgres=#
```

A green arrow points from the text "Ensures te connection with database" to the command prompt line "postgres=#".

❖ Các thao tác cơ bản:

- Tạo CSDL:

```
CREATE DATABASE qlsv;
```

- Tạo bảng:

```
CREATE TABLE students (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100),
    gpa NUMERIC(3,2)
);
```

❖ Các thao tác cơ bản:

- Thêm dữ liệu:

```
INSERT INTO students (name, gpa) VALUES ('Nguyễn Văn A', 3.5);
```

- Xem dữ liệu:

```
SELECT * FROM students;
```

❖ Truy vấn cơ sở dữ liệu:

| Mục đích | Câu truy vấn |
|---------------------|---|
| ----- | ----- |
| Lấy toàn bộ dữ liệu | <code>^SELECT * FROM students;</code> |
| Lọc theo điều kiện | <code>^SELECT * FROM students WHERE gpa > 3.0;</code> |
| Cập nhật dữ liệu | <code>^UPDATE students SET gpa = 3.8 WHERE id = 1;</code> |
| Xóa sinh viên | <code>^DELETE FROM students WHERE id = 2;</code> |
| Đếm sinh viên | <code>^SELECT COUNT(*) FROM students;</code> |

❖ Truy cập cơ sở dữ liệu PostgreSQL với PHP:

```
<?php

// 1. Kết nối PostgreSQL
$conn = pg_connect("host=localhost dbname=qlsv user=postgres password=123456");
if (!$conn) {
    echo "Kết nối thất bại.";
    exit;
}

// 2. Truy vấn dữ liệu
$result = pg_query($conn, "SELECT * FROM students");
echo "<h3>DANH SÁCH SINH VIÊN</h3>";
while ($row = pg_fetch_assoc($result)) {
    echo "ID: " . $row['id'] . " - Tên: " . $row['name'] . " - GPA: " . $row['gpa'] . "<br>";
}
// 3. Đóng kết nối
pg_close($conn);
?>
```

MySQLi

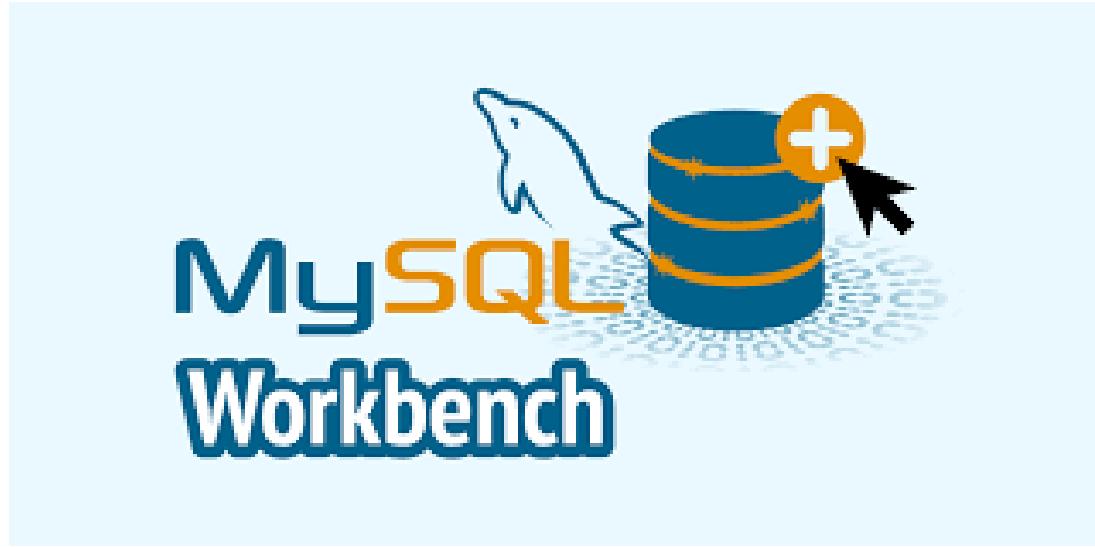
❖ Giới thiệu:

- MySQLi là thư viện mở rộng của PHP để làm việc với cơ sở dữ liệu MySQL.
- "i" trong MySQLi là viết tắt của Improved – tức là nâng cấp hơn so với mysql_* cũ (đã bị loại bỏ từ PHP 7).
- MySQLi hỗ trợ:
 - ✓ Kết nối an toàn hơn
 - ✓ Truy vấn chuẩn SQL
 - ✓ Hỗ trợ prepared statements (chống SQL Injection)
 - ✓ Kết nối theo hướng đối tượng hoặc hàm thủ tục



❖ Quản trị CSDL MySQLi:

- Công cụ phổ biến:
 - phpMyAdmin (giao diện web)
 - MySQL CLI (giao diện dòng lệnh)
 - MySql Workbench



❖ Các thao tác cơ bản:

- Tạo CSDL:

```
CREATE DATABASE qlsv;
```

- Tạo bảng:

```
CREATE TABLE students (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    gpa FLOAT
);
```

- Thêm dữ liệu:

```
INSERT INTO students (name, gpa) VALUES ('Nguyễn Văn A', 3.5);
```

❖ Truy vấn cơ sở dữ liệu:

| Mục đích | Câu lệnh SQL |
|----------------------|--|
| ----- | ----- |
| Lấy tất cả sinh viên | ^SELECT * FROM students; |
| Lọc theo điểm GPA | ^SELECT * FROM students WHERE gpa > 3.0; |
| Cập nhật thông tin | ^UPDATE students SET gpa = 3.9 WHERE id = 1; |
| Xoá sinh viên | ^DELETE FROM students WHERE id = 2; |

❖ Truy cập cơ sở dữ liệu MySQLi với PHP:

```
<?php  
// 1. Kết nối  
$conn = mysqli_connect("localhost", "root", "", "qlsv");  
// Kiểm tra kết nối  
if (!$conn) {  
    die("Kết nối thất bại: " . mysqli_connect_error());  
}  
// 2. Truy vấn  
$sql = "SELECT * FROM students";  
$result = mysqli_query($conn, $sql);  
// 3. Hiển thị  
while ($row = mysqli_fetch_assoc($result)) {  
    echo "ID: {$row['id']} - Tên: {$row['name']} - GPA: {$row['gpa']}<br>";  
}  
// 4. Đóng kết nối  
mysqli_close($conn);  
?>
```

❖ Dùng PDO + Prepared Statement (cách an toàn hơn so với query thuần):

- Thay vì viết:

```
$sql = "SELECT * FROM students WHERE id = $id";
```

- Có thể viết:

```
$stmt = $pdo->prepare("SELECT * FROM students WHERE id = :id");  
$stmt->execute(['id' => $id]);  
$data = $stmt->fetch();
```







Thank
you