Tạo database

CREATE DATABASE IF NOT EXISTS database\_name CHARACTER SET 'utf8' COLLATE 'utf8\_vietnamese\_ci'

Tạo bảng với các column

CREATE TABLE table\_name (

cột 1 INT(10) UNSIGNED NOT NULL PRIMARY KEY, // khóa chính

cột 2 VARCHAR(70) NOT NULL

) ENGINE = INNODB;

Ví dụ tạo bảng có khóa chính, khóa ngoại:

CREATE TABLE Groups (

groupid INT(11) NOT NULL AUTO\_INCREMENT PRIMARY KEY,

title INT(11) NOT NULL,

LEVEL TINYINT(1) DEFAULT 1 NOT NULL

);

CREATE TABLE Users(

userid INT(11) NOT NULL PRIMARY KEY AUTO\_INCREMENT,

username VARCHAR(50) NOT NULL ,

email VARCHAR (50) NOT NULL ,

groupid INT(11),

FOREIGN KEY (groupid) REFERENCES Groups(groupid)

);

INSER\_ chèn dữ liệu

INSERT INTO

table\_name(field1, field2, field2, ..., fieldn)

VALUES('field1', 'field2', 'field3', ...,'fieldn')

SELECT Mục đích: show ra dữ liệu

SELECT field1, field2, field3, ...

FROM table\_name

WHERE <dieu\_kien\_loc>

ORDER BY field\_name ASC|DESC

LIMIT start, limit

Note:

SELECT field1, field2, field3, ... là danh sách các fields cần lấy

FROM table\_name tên table cần lấy

WHERE <dieu\_kien\_loc> là các điều kiện để lấy các dòng dữ liệu

ORDER BY field\_name, ASC|DESC: là cách sắp xếp cho field\_name theo kiểu ASC (tăng dần) hoặc DESC (giảm dần)

LIMIT start, limit là lấy limit records kể từ record thứ start trong kết quả.

Ví dụ 1 về thao tác với database cơ bản bằng (PDO)

file ConnectDB.php

// kết nối bằng PDO

class ConnectDB

{

public static function connect()

{

try {

$dsn = 'mysql:host=localhost; dbname=student';

$userName = 'root';

$password = '';

$db = new PDO ($dsn, $userName, $password);

return $db;

} catch (PDOException $e) {

echo $e->getMessage();

}

}

}

File Student.php

<?php

require('ConnectDB.php');

Class Student

{

public $code;

public $name;

public $age;

public $address;

public $db;

function \_\_construct()

{

$this->db = ConnectDB::connect();

}

public function setCode($code)

{

$this->code = $code;

}

public function getCode()

{

return $this->code;

}

public function setName($name)

{

$this->name = $name;

}

public function getName()

{

return $this->name;

}

public function setAge($age)

{

$this->age = $age;

}

public function getAge()

{

return $this->age;

}

public function setAddress($address)

{

$this->address = $address;

}

public function getAddress()

{

return $this->address;

}

public function list()

{

try {

$query = "select \* from users";

$statement = $this->db->prepare($query);

$statement->execute();

$students = $statement->fetchAll();

$statement->closeCursor();

return $students ? $students : [];

} catch (PDOException $e) {

echo $e->getMessage();

}

}

// Chèn dữ liệu

public function insert($code, $name, $age, $address)

{

try {

$query = "insert into users

(code, name, age, address) values

(:code, :name, :age, :address)";

$statement = $this->db->prepare($query);

$statement->bindValue(':code', $code);

$statement->bindValue(':name', $name);

$statement->bindValue(':age', $age);

$statement->bindValue(':address', $address);

$statement->execute();

$statement->closeCursor();

} catch (PDOException $e) {

echo $e->getMessage();

}

}

// show dữ liệu chỗ chỉnh sửa để coi dữ liệu cũ

public function show($id)

{

try {

$query = "select \* from users where id = :id";

$statement = $this->db->prepare($query);

$statement->bindValue(':id', $id);

$statement->execute();

$students = $statement->fetch();

$statement->closeCursor();

return $students ? $students : [];

} catch (PDOException $e) {

echo $e->getMessage();

}

}

// Update dữ liệu đã chỉnh sửa

public function update($code, $name, $age, $address,$id)

{

try {

$query = "UPDATE users SET code = :code, name = :name, age = :age, address = :address WHERE id = :id";

$statement = $this->db->prepare($query);

$statement->bindValue(':id', $id);

$statement->bindValue(':code', $code);

$statement->bindValue(':name', $name);

$statement->bindValue(':age', $age);

$statement->bindValue(':address', $address);

$statement->execute();

$statement->closeCursor();

} catch (PDOException $e) {

echo $e->getMessage();

}

}

// Xóa dữ liệu

public function delete($id)

{

try {

$query = "delete from users where id = :id";

$statement = $this->db->prepare($query);

$statement->bindValue(':id', $id);

$statement->execute();

} catch (PDOException $e) {

echo $e->getMessage();

}

}

}