

# DATABASE DESIGN

# Outline



1. Định nghĩa
2. Tổng quan CSDL MySQL
3. Truy vấn trong CSDL MySQL – SELECT
4. Truy vấn trong CSDL MySQL – Truy vấn con & cập nhật dữ liệu
5. Kết hợp PHP & MySQL – Thư viện PDO 1
6. Kết hợp PHP & MySQL – Thư viện PDO 2

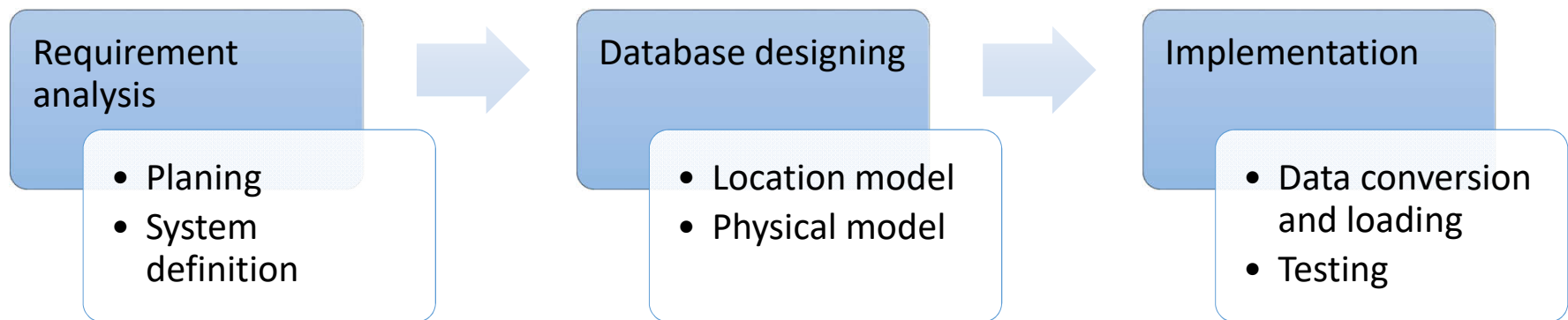
# 1. ĐỊNH NGHĨA

- **Database Design** is a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems. Properly designed database are easy to maintain, improves data consistency and are cost effective in terms of disk storage space. The database designer decides how the data elements correlate and what data must be stored.
- The main objectives of database designing are to produce logical and physical designs models of the proposed database system.
- The logical model concentrates on the data requirements and the data to be stored independent of physical considerations. It does not concern itself with how the data will be stored or where it will be stored physically.
- The physical data design model involves translating the logical design of the database onto physical media using hardware resources and software systems such as database management systems (DBMS).

# Why Database Design is Important?

- It helps produce database systems.
- That meet the requirements of the users.
- Have high performance.
- Database designing is crucial to **high performance** database system.
- Note, the genius of a database is in its design. Data operations using SQL is relatively simple.

# Database development life cycle



- The database development life cycle has a number of stages that are followed when developing database systems.
- The steps in the development life cycle do not necessary have to be followed religiously in a sequential manner.
- On small database systems, the database system development life cycle is usually very simple and does not involve a lot of steps.
- In order to fully appreciate the above diagram, let's look at the individual components listed in each step.

# Requirements analysis

- **Planning** - This stages concerns with planning of entire Database Development Life Cycle It takes into consideration the Information Systems strategy of the organization.
- **System definition** - This stage defines the scope and boundaries of the proposed database system.

# Database designing



- **Logical model** - This stage is concerned with developing a database model based on requirements. The entire design is on paper without any physical implementations or specific DBMS considerations.
- **Physical model** - This stage implements the logical model of the database taking into account the DBMS and physical implementation factors.

# Implementation

- **Data conversion and loading** - this stage is concerned with importing and converting data from the old system into the new database.
- **Testing** - this stage is concerned with the identification of errors in the newly implemented system .It checks the database against requirement specifications.



## 2. TỔNG QUAN VỀ CSDL

- Khái niệm
- Chức năng
- Các đối tượng chính của CSDL quan hệ
- Hệ quản trị CSDL
- SQL (Structure Query Language)

# KHÁI NIỆM CSDL



- **Khái niệm:** CSDL là tập hợp dữ liệu được lưu trữ một cách có tổ chức nhằm giúp việc xem, tìm kiếm và lấy thông tin được nhanh chóng và chính xác, giúp công sức và thời gian quản lý thông tin cần thiết.
- **Chức năng:**
  - ✓ Lưu trữ
  - ✓ Truy cập
  - ✓ Tổ chức
  - ✓ Xử lý
- **Các đối tượng chính của CSDL quan hệ:**
  - ✓ Bảng (table)
  - ✓ Quan hệ (relationship)

# CSDL - BẢNG

- **Bảng – dòng (row):**
  - ✓ Là tập hợp các thông tin của tất cả cột dữ liệu trong bảng.
- **Bảng – cột (column):**
  - ✓ Là khối dữ liệu trong bảng, có cùng loại dữ liệu, có các thông tin chính:
  - ✓ Tên cột: dùng để phân biệt với các cột khác trong bảng. Tên cột trong bảng phải duy nhất và không các ký tự đặc biệt.
  - ✓ Kiểu dữ liệu của cột: xác định loại giá trị nào được phép lưu trữ trong cột.
- **Ví dụ: Bảng Khách hàng KHACH\_HANG**

MKH	TEN_KH	PHAI	DIA_CHI	DT	EMAIL
<u>KH001</u>	Trần Văn An	0	123 Nguyễn Du	0235544756	vanan@gmail.com
<u>KH002</u>	Nguyễn Văn B	0	234 Nguyễn Du	0343455475	edt@gmail.com
<u>KH003</u>	Lê Thi B	1	567 Nguyễn Du	0888554756	leb@gmail.com

- ✓ Mỗi dòng trong bảng lưu trữ thông tin về khách hàng trong thực tế
- ✓ Trong bảng trên có các cột: MKH (mã khách hàng), TEN\_KH (tên khách hàng)...

# CSDL - QUAN HỆ TRONG CSDL



- Quan hệ:
  - ✓ Là thành phần được dùng để tạo mối liên kết giữa các bảng dữ liệu với nhau nhằm đảm bảo tính nhất quán, đúng đắn của dữ liệu trong CSDL.
- Các loại quan hệ:
  - ✓ Quan hệ 1 – 1
  - ✓ Quan hệ 1 – nhiều
  - ✓ Quan hệ nhiều – nhiều

# CSDL - QUAN HỆ TRONG CSDL

## ▪ Quan hệ 1 – 1:

- ✓ Mô tả mối quan hệ giữa hai bảng mà trong đó một dòng dữ liệu bên bảng liên hệ với duy nhất với một dòng dữ liệu bên bảng kia và ngược lại.
- ✓ Ví dụ:



*Một nhân viên chỉ có một sơ yếu lý lịch.*

*Một sơ yếu lý lịch chỉ thuộc về một nhân viên.*

# CSDL - QUAN HỆ TRONG CSDL

## ▪ Quan hệ 1 – nhiều:

- ✓ Mô tả mối quan hệ giữa hai bảng mà trong đó một dòng dữ liệu bên bảng liên hệ với nhiều dòng dữ liệu bên bảng kia và một dòng dữ liệu bên bảng kia liên hệ với duy nhất với một dòng dữ liệu bên bảng này.
- ✓ Quan hệ này thường gặp nhất trong CSDL.
- ✓ Ví dụ:



*Một phòng có nhiều nhân viên.*

*Một nhân viên chỉ thuộc về một phòng.*

# CSDL - QUAN HỆ TRONG CSDL

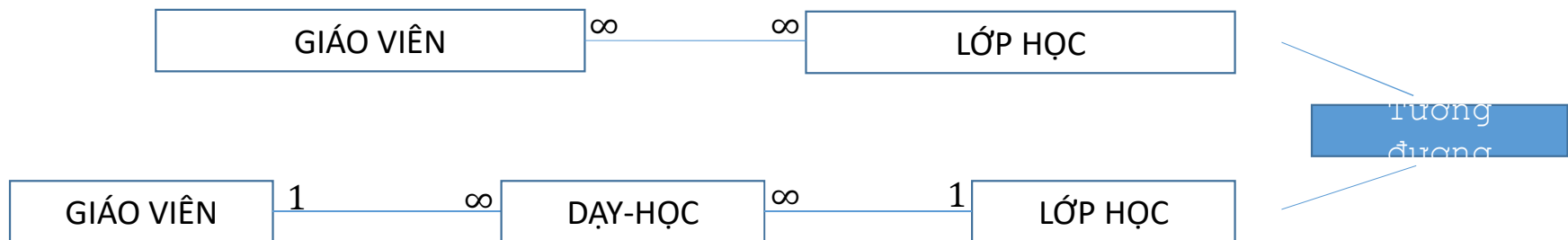
## ▪ Quan hệ nhiều – nhiều:

- ✓ Mô tả mối quan hệ giữa hai bảng mà trong đó một dòng dữ liệu bên bảng liên hệ với nhiều dòng dữ liệu bên bảng kia và ngược lại.
- ✓ Trong CSDL không lưu trữ quan hệ nhiều - nhiều nên cần chuyển thành quan một nhiều.

✓ Ví dụ:

*Một giáo viên dạy nhiều lớp.*

*Một lớp học có nhiều giáo viên.*



## ▪ HỆ QUẢN TRỊ CSDL:

- ✓ Quản lý các dữ liệu được lưu trữ bên trong các CSDL, giúp cho CSDL đến với người dùng khi cần truy cập thông tin. Có khả năng lưu trữ dữ liệu và cho phép dữ liệu có thể trao đổi với các CSDL khác.
- ✓ Có khả năng: bảo vệ dữ liệu, duy trì dữ liệu, quản lý các giao dịch...

## ▪ SQL:

- ✓ Là loại ngôn ngữ cho phép thực hiện các thao tác rút trích, tính toán, cập nhật trên các dữ liệu được lưu trữ trong CSDL.



# CSDL - MySQL

- Giới thiệu
- Đặc điểm
- Các tập tin vật lý lưu trữ CSDL
- Quy tắc đặt tên
- Tạo CSDL
- Xóa CSDL

## ■ Giới thiệu:

- ✓ Là tập hợp các đối tượng: bảng, bảng ảo... cho phép người dùng lưu trữ và xuất các thông tin đã được tổ chức và lưu trữ bên trong đó.

## ■ Đặc điểm:

- ✓ Sử dụng cho các ứng dụng web có quy mô vừa và nhỏ.
- ✓ Để thực hiện các thao tác trên CSDL, có thể sử dụng giao diện đồ họa hay dòng lệnh (command line)

## ■ Các tập tin vật lý lưu trữ CSDL:

- ✓ Mỗi bảng sẽ được lưu trữ dưới ba tập tin vật lý.
  - .frm: lưu định dạng (cấu trúc) của bảng
  - .MYD: lưu nội dung của bảng
  - .MYI: lưu chỉ mục của bảng
- ✓ Các tập tin này sẽ được tự động lưu trữ trong thư mục:  
wamp\mysql\data\tên\_CSDL

## ■ Giới thiệu:

- ✓ Là tập hợp các đối tượng: bảng, bảng ảo... cho phép người dùng lưu trữ và xuất các thông tin đã được tổ chức và lưu trữ bên trong đó.

## ■ Đặc điểm:

- ✓ Sử dụng cho các ứng dụng web có quy mô vừa và nhỏ.
- ✓ Để thực hiện các thao tác trên CSDL, có thể sử dụng giao diện đồ họa hay dòng lệnh (command line)

## ■ Các tập tin vật lý lưu trữ CSDL:

- ✓ Mỗi bảng sẽ được lưu trữ dưới ba tập tin vật lý.
  - .frm: lưu định dạng (cấu trúc) của bảng
  - .MYD: lưu nội dung của bảng
  - .MYI: lưu chỉ mục của bảng
- ✓ Các tập tin này sẽ được tự động lưu trữ trong thư mục:  
wamp\mysql\data\tên\_CSDL

## ❑ Chiều dài của tên CSDL, bảng, chỉ mục, cột, định danh

Loại	Chiều dài tối đa (bytes)	Chiều dài tối đa (ký tự không dấu)
CSDL (database)	64	64
Bảng (table)	64	64
Chỉ mục (Index)	64	64
Cột (Column)	64	64
Định danh (Alias)	255	255

## ❑ Quy tắc đặt tên CSDL:

- Tên không kết thúc bằng khoảng trắng
- Tên CSDL không có ký tự /, \, ., :, \*, ", <, >
- Chiều dài tên tối đa là 64 ký tự không dấu

# CSDL

- Các thuộc tính của CSDL:
  - Tên CSDL: phải duy nhất trong hệ CSDL MySQL
  - Vị trí lưu trữ: Khi tạo mới CSDL hệ thống auto tạo thư mục có tên của CSDL và được lưu tại thư mục wamp\mysql\data
- Cách tạo CSDL: Có thể tạo CSDL bằng giao diện hoặc dòng lệnh  
`CREATE DATABASE xxx`
- Xóa CSDL: Dùng lệnh DROP DATABASE  
`DROP DATABASE xxx`

## ▪ Quy tắc đặt tên

- **Tên bảng:** Do người dùng tạo ra, duy nhất trong CSDL. Tên bảng không có ký tự /, \, ., :, \*, ", <, >, |
- **Tên cột:** Do người dùng tạo ra, duy nhất trong bảng

## ▪ Các kiểu dữ liệu:

- **Số nguyên:** *tiny, Smallint, mediumint, int, bigint*
- **Boolean:** có hai giá trị *true/false*
- **Số thập phân:** *decimal, numeric*
- **Số thực:** *float, double*
- **Thời gian:** *date, datetime, time, year, timestamp*
- **Chuỗi:** *char, varchar, text,...*

# CSDL – BẢNG

## ▪ Thuộc tính cột trong bảng:

- Độ dài dữ liệu
- Kiểu hiển thị
- Thuộc tính
- Cho phép để trống dữ liệu
- Giá trị mặc định
- Thuộc tính mở rộng
- Ghi chú
- Tăng giá trị tự động

# CSDL – BẢNG

## Thao tác với bảng

- **TẠO BẢNG:** Cú pháp

```
CREATE TABLE ten_bang  
(  
    ten_cot_1 kieu_du_lieu[(kich_co)] [NOT NULL] PRIMARY KEY,....  
    ten_cot_2 kieu_du_lieu[(kich_co)] [NOT NULL] DEFAULT gia_tri,....  
)
```

- **THAY ĐỔI CẤU TRÚC BẢNG:** Cú pháp

Thêm cột:

```
ALTER TABLE ten_bang  
    ADD  
        ten_cot kieu_du_lieu[(kich_co)] [...]
```



# CSDL – BẢNG

## ▪ THAY ĐỔI CẤU TRÚC BẢNG: Cú pháp

Hủy cột:

`ALTER TABLE` ten\_bang

`DROP COLUMN` ten\_cot

## ▪ XÓA BẢNG: Cú pháp

`DROP TABLE` ten\_bang

## ▪ Tạo mối quan hệ các bảng

- *Tạo mối quan hệ giữa các bảng với nhau (giữa bảng cha (là khóa chính - primary key) và bảng con (là khóa ngoại - foreign key))*
- *Tạo khóa ngoại:*

`FOREIGN KEY` (ten\_cot\_khoa\_ngoai) `REFERENCES` ten\_bang\_cha  
(ten\_cot\_khoa\_chinh) `ON UPDATE CASCADE`;

# IMPORT - EXPORT DỮ LIỆU

- Import dữ liệu:
  - Đưa dữ liệu từ tập tin vào CSDL MySQL
- Export dữ liệu:
  - Xuất dữ liệu từ database trong mySQL ra tập tin

# **3. TRUY VẤN TRONG CSDL MYSQL - SELECT**

### 3. TRUY VẤN TRONG CSDL MYSQL - SELECT

- Truy vấn - Chọn ra dữ liệu của các cột có trong một bảng.
- Cú pháp:  
    SELECT danh\_sach\_cot FROM ten\_bang  
    Hoặc  
    SELECT \* FROM ten\_bang
- Yêu cầu:
  - Tên các cột phải chính xác
  - Dùng \* khi muốn lấy dữ liệu tất cả các cột

### 3. TRUY VẤN TRONG CSDL MYSQL - SELECT

#### ▪ Sắp xếp :

- Mệnh đề **ORDER BY** giúp sắp xếp dữ liệu theo thứ tự tăng/giảm dần.
- Cú pháp:  
**SELECT** danh\_sach\_cot **FROM** ten\_bang  
**ORDER BY** ten\_cot\_sap\_xep [**DESC**, ...]
- Lưu ý: Mặc định các cột sẽ tăng dần, nếu muốn thứ tự ngược lại thì phía sau cột cần sắp xếp thêm khóa **DESC**

#### ▪ Truy vấn có sử dụng **AS** :

- Sử dụng **AS** để đặt lên tên (alias) cho các cột.
- Cú pháp:  
**SELECT** ten\_cot **AS** dinh\_danh\_cot,... **FROM** ten\_bang  
[**ORDER BY** ten\_cot\_sap\_xep [**DESC**, ...]]

### 3. TRUY VẤN TRONG CSDL MYSQL - SELECT

#### ■ Giới hạn mẫu tin LIMIT :

- Mệnh đề **LIMIT** m, n giúp lấy ra m mẫu tin trong bảng tính từ vị trí n, theo tiêu chuẩn sắp xếp nào đó.
- Cú pháp:

```
SELECT danh_sach_cot FROM ten_bang  
ORDER BY ten_cot_sap_xep [DESC, ...]  
LIMIT n,m
```

#### ■ DISTINCT :

- Trong 1 bảng, 1 cột có thể chứa nhiều giá trị trùng, nhưng user chỉ muốn lấy trị duy nhất, không trùng lặp => Sử dụng từ khóa **DISTINCT**
- Cú pháp:

```
SELECT DISTINCT ten_cot_1, ten_cot_2,... FROM ten_bang  
[ORDER BY ten_cot_sap_xep [DESC, ...]]
```

### 3. TRUY VẤN TRONG CSDL MYSQL - SELECT

#### ▪ Lọc theo mẫu tin :

- Mệnh đề **WHERE** giúp lọc các dòng dữ liệu bên trong bảng, dữ liệu này phải điều kiện đưa ra trong mệnh đề WHERE.
- Cú pháp:  
**SELECT** danh\_sach\_cot **FROM** ten\_bang  
**WHERE** dieu\_kien\_loc  
[**ORDER BY** ten\_cot\_sap\_xep [**DESC**, ...]]
- Các phép toán trong điều kiện lọc:
  - So sánh: >, >=, <, <=, =, !=, <>
  - Logic: and, or, not, in, not in, between, like, not like

# TRUY VẤN TRÊN NHIỀU BẢNG

## ▪ LOẠI 1 :

- Có thể sử dụng mệnh đề WHERE để liên kết dữ liệu của nhiều bảng với nhau
- Cú pháp:  
`SELECT danh_sach_cot FROM ten_bang_1, ten_bang_2,...`  
`WHERE ten_bang_1.ten_cot = ten_bang_2.ten_cot`  
`[ORDER BY ten_cot_sap_xep [DESC, ...]]`



# TRUY VẤN TRÊN NHIỀU BẢNG

## ▪ LOẠI 2 :

- Khi muốn liên kết các bảng có quan hệ với nhau để lấy ra dữ liệu chung. User có thể hợp **SELECT ... FROM** với mệnh đề **JOIN**.
- Khi sử dụng JOIN để nối các bảng cần phải lưu ý những bảng này phải có các cột liên với nhau và thứ tự quan hệ được chỉ định giữa các bảng sẽ làm ảnh hưởng tới kết truy vấn.

## ▪ LOẠI 2 – INNER JOIN :

- Khi dùng INNER JOIN thì chỉ định việc so sánh giá trị trong các cột của các bảng là tương đương, nghĩa là dữ liệu có ở CẢ HAI BẢNG.
- Kết quả sau khi thực hiện câu truy vấn là các mẫu tin thỏa điều kiện quan hệ ở cả hai bảng.
- Cú pháp:

```
SELECT danh_sach_cot FROM ten_bang  
INNER JOIN ten_bang_lien_ket ON dieu_kien_lien_ket  
[WHERE dieu_kien_loc]  
[ORDER BY danh_sach_cot_sap_xep [DESC, ...]]
```

# TRUY VẤN TRÊN NHIỀU BẢNG

## ▪ LOẠI 2 – LEFT JOIN, RIGHT JOIN :

- Khi dùng LEFT JOIN / RIGHT JOIN thì chỉ định việc so sánh giá trị trong các cột cột của các bảng là được ưu tiên cho mối quan hệ bên nhánh TRÁI/PHẢI. Việc thay đổi thứ tự ưu tiên này sẽ làm thay đổi kết quả truy vấn.
- Cú pháp:

```
SELECT danh_sach_cot FROM ten_bang  
LEFT|RIGHT JOIN ten_bang_lien_ket ON điều_kien_lien_ket  
[WHERE điều_kien_loc]  
[ORDER BY danh_sach_cot_sap_xep [DESC, ...]]
```

# TRUY VẤN TRÊN NHIỀU BẢNG

## ▪ MỆNH ĐỀ LIÊN KẾT DỮ LIỆU NHIỀU BẢNG :

- Cú pháp:

```
SELECT danh_sach_cot FROM ten_bang_1  
INNER|LEFT|RIGHT JOIN ten_bang_2 ON dieu_kien_lien_ket_bang_1_2  
INNER|LEFT|RIGHT JOIN ten_bang_3 ON dieu_kien_lien_ket_...  
[WHERE dieu_kien_loc]  
[ORDER BY danh_sach_cot_sap_xep [DESC, ...]]
```

# TRUY VẤN CÓ NHÓM VÀ CÁC HÀM THỐNG KÊ



## ■ GROUP BY:

- Mệnh đề **GROUP BY** giúp nhóm dữ liệu của các dòng dữ liệu bên trong bảng sử dụng thêm các hàm thống kê đi kèm để tính toán dữ liệu có tính chất kê.
- Cú pháp:  
**SELECT** danh\_sach\_cot, ham\_thong\_ke [AS ten]  
**FROM** ten\_bang  
[**WHERE** dieu\_kien\_loc]  
**GROUP BY** danh\_sach\_cac\_cot\_nhom\_du\_lieu  
[**ORDER BY** danh\_sach\_cot\_sap\_xep [**DESC**, ...]]

# TRUY VẤN CÓ NHÓM VÀ CÁC HÀM THỐNG KÊ



## ■ GROUP BY:

Các hàm thống kê:

- AVG: trả về giá trị trung bình
- COUNT: trả về số lượng
- MIN: trả về giá trị nhỏ nhất
- MAX: trả về giá trị lớn nhất
- SUM: trả về tổng giá trị

# LỌC THEO NHÓM

## ■ HAVING:

- Mệnh đề **HAVING** giúp lọc lại dữ liệu sau khi đã gom nhóm dữ liệu bằng **BY**.
- Cú pháp:  
**SELECT** danh\_sach\_cot, ham\_thong\_ke [AS ten]  
**FROM** ten\_bang  
[**WHERE** dieu\_kien\_loc]  
**GROUP BY** danh\_sach\_cac\_cot\_nhom\_du\_lieu  
**HAVING** dieu\_kien\_loc\_sau\_khi\_nhom  
[**ORDER BY** danh\_sach\_cot\_sap\_xep [**DESC**, ...]]

## **4. TRUY VẤN TRONG CSDL MYSQL**

### **- TRUY VẤN CON & CẬP NHẬT DỮ LIỆU**

# TRUY VẤN CON



## ▪ Khái niệm:

- Truy vấn con là một câu lệnh truy vấn SELECT được lồng vào các câu lệnh truy vấn khác nhằm thực hiện các truy vấn tính toán phức tạp.
- Lưu ý: cần tuân thủ quy tắc
  - Phải được đặt trong dấu ngoặc đơn
  - Chỉ có thể tham chiếu đến một cột hoặc một biểu thức
- Kết quả trả về: có thể là một giá trị hoặc danh sách các giá trị



# TRUY VẤN CON



- **Truy vấn con trả về một giá trị:**
  - Là truy vấn con mà kết quả trả về của nó là một giá trị duy nhất.
- **Truy vấn con trả về danh sách các giá trị:**
  - Là truy vấn con mà kết quả trả về là tập hợp các giá trị.
  - Toán tử IN hoặc NOT IN thường được sử dụng trong trường hợp này vì nó so sánh một phần tử thuộc/không thuộc tập hợp các giá trị không.

## ▪ Làm việc với toán tử truy vấn con:

- Các toán tử so sánh:  $>$ ,  $>=$ ,  $<$ ,  $<=$ ,  $=$ ,  $<>$  (thường trả về một giá trị)
- Các toán tử truy vấn: ANY, SOME, ALL, IN, NOT IN, EXISTS, NOT EXISTS (thường trả về tập các giá trị)
- Quy tắc:
  - $IN \Leftrightarrow = ANY$
  - $NOT IN \Leftrightarrow <> ALL$
- Sử dụng từ khóa EXISTS hoặc NOT EXISTS để kiểm tra sự tồn tại/ không tồn tại của dữ liệu.
- Sau EXISTS hoặc NOT EXISTS là câu lệnh truy vấn con mà kết quả trả về là một tập hợp trống hoặc có phần tử.

# INSERT

## ▪ Định nghĩa:

- Câu lệnh INSERT INTO được dùng để thêm một hoặc nhiều dòng dữ liệu vào trong một bảng.
- Có thể thêm vào bảng giá trị trực tiếp hoặc lấy từ bảng khác.

## ▪ Insert giá trị trực tiếp: Cú pháp

```
INSERT INTO ten_bang[(danh_sach_cac_cot)]  
VALUE ("gia_tri_cot1", "gia_tri_cot2",...)
```

## ▪ Insert bằng cách lấy dữ liệu từ bảng khác: Cú pháp

```
INSERT INTO ten_bang[(danh_sach_cac_cot_can_them_du_lieu)]  
SELECT danh_sach_cac_cot_lay_du_lieu FROM ten_bang_nguon  
WHERE dieu_kien_loc_du_lieu
```

Lưu ý: danh\_sach\_cac\_cot\_can\_them\_du\_lieu phải tương ứng  
danh\_sach\_cac\_cot\_lay\_du\_lieu

# UPDATE

- **Định nghĩa:**

- Câu lệnh UPDATE được dùng để cập nhật giá trị trực tiếp hay một biểu thức giá trị trả về cho mẫu tin trong bảng.

- **Update trực tiếp:** Cú pháp

```
UPDATE ten_bang[(danh_sach_cac_cot)]  
SET ten_cot = gia_tri (hoặc biểu thức) [...]  
WHERE điều_kiện_cập_nhật
```

- **Update bằng cách lấy dữ liệu từ bảng khác:** Cú pháp

```
UPDATE ten_bang  
SET ten_cot = (SELECT ... FROM ... WHERE)  
WHERE điều_kiện_lọc_dữ_liệu
```

# DELETE

- **Định nghĩa:**
  - Câu lệnh DELETE được dùng để xóa dữ liệu trong bảng.
- **Lệnh xóa dữ liệu đơn giản:** Cú pháp  
`DELETE FROM ten_bang`  
`WHERE dieu_kien_xoa`
- **Lệnh xóa dữ liệu có điều kiện được lấy từ bảng khác:** Cú pháp  
`DELETE FROM ten_bang`  
`WHERE ten_cot toan_tu (SELECT ... FROM ... WHERE)`

# 5. KẾT HỢP PHP & MYSQL

## *THƯ VIỆN PDO 1*

## ▪ Định nghĩa:

- PDO – PHP Data Object – là một lớp truy cập dữ liệu cung cấp phương thức chuẩn để truy cập vào nhiều loại CSDL khác nhau.
- PDO cung cấp một lớp truy cập dữ liệu trừu tượng. Dù chúng ta sử dụng nào thì chúng ta đều có thể sử dụng các phương thức như nhau để truy vấn lấy dữ liệu.

## ▪ Lý do chọn PDO:

- Hướng đối tượng: phù hợp mô hình MVC
- Nhanh và đơn giản
- Linh động: đổi sang CSDL khác
- An toàn: không lo sợ SQL injection

# THƯ VIỆN PDO 1

## ▪ Tạo kết nối:

- Cú pháp: `$dbh = new PDO(<driver>,`
- Tham số:
  - `<driver>`: tên driver dùng để làm việc CSDL
  - `<username>`: tên đăng nhập CSDL
  - `<password>`: mật khẩu truy cập CSDL
  - `<driver_options>`: tham số tùy chọn, để bổ sung thêm các trạng thái kết nối của PDO

## ▪ Hủy kết nối:

- Cú pháp: `$dbh = NULL;`



# THƯ VIỆN PDO 1



- Xử lý lỗi kết nối:

```
try {  
    $dbh = new PDO('mysql:host=localhost;dbname=test','root','');  
    $dbh = NULL;  
} catch (PDOException $e) {  
    print "Error: ". $e->getMessage();  
    die();  
}
```

# THỰC THI SQL



- Câu lệnh Insert, Update, Delete:
  - Cú pháp: `$dbh->exec(<query>);`
- Câu lệnh Select:
  - Cú pháp: `$dbh-> query(<query>);`

## 6. KẾT HỢP PHP & MYSQL

### *THƯ VIỆN PDO 2*

# TRANSACTION

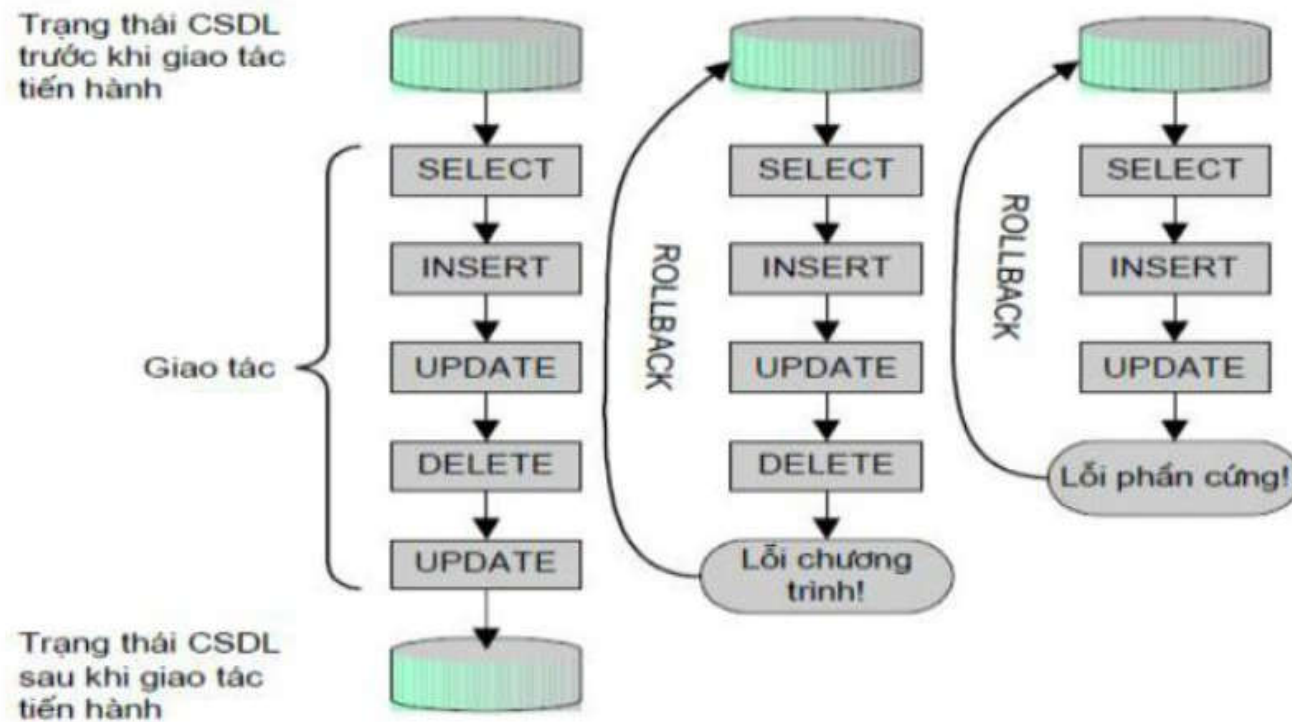


## ▪ Định nghĩa:

- Một giao tác (transaction) là một chuỗi một hoặc nhiều câu lệnh SQL được kết hợp lại với nhau thành một khối công việc, nhằm đảm bảo toàn vẹn dữ liệu và khả năng phục hồi dữ liệu.
- Trong một giao tác, các câu lệnh có thể độc lập với nhau nhưng tất cả các câu lệnh trong một giao tác đòi hỏi phải thực thi trọn vẹn hoặc không một câu nào được thực thi.

# TRANSACTION

## ■ Ví dụ:



# TRANSACTION

- Giao tác SQL được định nghĩa các câu lệnh:
  - **BEGIN TRANSACTION**: bắt đầu một giao tác
  - **COMMIT**: Đánh dấu kết thúc một giao tác. Câu lệnh báo hiệu sự kết thúc công của một giao tác. Sau câu lệnh này, một giao tác mới sẽ được bắt đầu.
  - **ROLLBACK**: Hủy bỏ một giao tác và đưa CSDL về trạng thái như trước khi giao tác bắt đầu. Một giao tác mới sẽ được bắt đầu sau khi câu lệnh này được thực thi

# PDO Statement

- Prepare statement:
  - Sử dụng Prepare statement sẽ giúp chúng ta tránh SQL injection.
  - Cú pháp: `$sth = $dbh->prepare(<query>);`
- Thực thi:
  - Cú pháp: `$dbh->execute();`

# PDO Statement

## ▪ Duyệt dữ liệu – duyệt lần lượt từng mẫu tin:

- Cú pháp: `PDOStatement->fetch([int $fetch_style])`
- Có các `$fetch_style` sau:
  - `PDO::FETCH_NUM`: kết quả trả về là một mảng (có chỉ số) có chứa giá trị của một dòng dữ liệu với mỗi phần tử là nội dung của một cột -> truy cập bằng cách gọi từng phần tử của mảng
  - `PDO::FETCH_ASSOC`: kết quả trả về là một mảng (có chỉ số chuỗi) có chứa giá trị của một dòng dữ liệu với mỗi phần tử là nội dung của một cột -> truy cập bằng cách gọi từng phần tử của mảng
  - `PDO::FETCH_BOTH`: kết quả trả về là một mảng (có chỉ số và chỉ số chuỗi) có chứa giá trị của một dòng dữ liệu với mỗi phần tử là nội dung của một cột -> truy cập bằng cách gọi từng phần tử của mảng
  - `PDO::FETCH_OBJ`: kết quả trả về là một mẫu tin trong bộ các mẫu tin như là một đối tượng -> truy cập bằng cách gọi từng thuộc tính của đối tượng



# PDO Statement

## ▪ Duyệt dữ liệu – duyệt danh sách các mẫu tin:

- Cú pháp: `PDOStatement->fetchAll([int $fetch_style, [$column_index]])`
- Có các `$fetch_style` sau:
  - `PDO::FETCH_BOTH`: giá trị mặc định. Kết quả trả về là mảng hai chiều, một phần tử là một chiều (có chỉ số và chỉ số chuỗi) có chứa giá trị của một dòng dữ liệu với mỗi phần tử nội dung của một cột
  - `PDO::FETCH_COLUMN`: Kết quả trả về là mảng một chiều chứa tất cả các giá trị của một được chỉ định
- `$column_index`: kết quả trả về là cột được chỉ định khi `$fetch_style` có giá trị là `PDO::FETCH_COLUMN`

# PDO Statement

## ▪ Truyền tham số trong câu truy vấn:

- Cú pháp: `PDOStatement->bindParam(mixed $parameter, mixed &$variable [, $data_type [, int $length [, mixed $driver_options]])`
- Tham số:
  - `$parameter` : tên tham số
  - `$variable` : giá trị tham số
  - `$data_type` : tùy chọn, kiểu dữ liệu tham số
  - `$variable` : tùy chọn, chiều dài kiểu dữ liệu

## ▪ Đếm số lượng mẫu tin:

- Cú pháp: `PDOStatement->rowCount()`
- Hàm này có kết quả trả về là số lượng mẫu tin bị tác động với câu lệnh insert, update, delete

# Exercise 1

## ➤ Manufacturer

A manufacturing company produces products. The following product information is stored: **product name, product ID and quantity on hand**. These products are made up of many components. Each component can be supplied by **one or more suppliers**. The following component information is kept: **component ID, name, description, suppliers who supply them, and products in which they are used**.

**Create an ERD** to show how you would track this information.

**Show entity names, primary keys, attributes for each entity, relationships between the entities and cardinality.**

## ➤ Assumptions

- A supplier can exist without providing components.
- A component does not have to be associated with a supplier.
- A component does not have to be associated with a product. Not all components are used in products.
- A product cannot exist without components.

## Exercise 2

Create an ERD for a car dealership. The dealership sells both new and used cars, and it operates a service facility. Base your design on the following business rules:

- A salesperson may sell many cars, but each car is sold by only one salesperson.
- A customer may buy many cars, but each car is bought by only one customer.
- A salesperson writes a single invoice for each car he or she sells.
- A customer gets an invoice for each car he or she buys.
- A customer may come in just to have his or her car serviced; that is, a customer need not buy a car to be classified as a customer.
- When a customer takes one or more cars in for repair or service, one service ticket is written for each car.
- The car dealership maintains a service history for each of the cars serviced. The service records are referenced by the car's serial number.
- A car brought in for service can be worked on by many mechanics, and each mechanic may work on many cars.
- A car that is serviced may or may not need parts (e.g., adjusting a carburetor or cleaning a fuel injector nozzle does not require providing new parts).



# Thank you

# Why Cloud Nine Solutions

## Competitive Low TCO

We always commit: high quality output and result, on-time delivery, proactive communication and reports, add-in consultancy in technology, methodology and business domain. We “speak” client’s languages: English, Japanese, and Vietnamese (native).

## End To End Value Chain

Key player in the entire IT service chain: consulting to implementation & support – system operation and maintenance service (from Level 2) – one-stop SW services: engineering, re-engineering, migrating, porting, customizing for Independent Software Vendors (ISVs).

## People, Process & Technology

Certified resources pool, across various domains/technologies, coupled with investments in infrastructure, process excellence and technology partnerships - to ensure best-in-class solutions & services.

## Repeat Business

Enduring relationships with fortune 100 & enterprise customers across Japan, UK and globally - as a result of consistent SLA based service delivery and process improvements.

## Niche Solutions & Rich Experience

Niche solutions / real experience in the business verticals of logistics/location tracking, real estate management, property matching/recommendation, automotive, etc. Additionally, strong product development capabilities on multiple platforms.

# Contact Us



## ONE TEAM - ONE VOICE - ONE VISION



 [sales@cloud9-solutions.com](mailto:sales@cloud9-solutions.com) |  <https://www.cloud9-solutions.com> |  +81-806-543-786

### CLOUD NINE SOLUTIONS COMPANY LIMITED

Slot 1 & 4 Floor 5 The Scetpa Building, 19A Cong Hoa  
Str., Tan Binh Dist., Ho Chi Minh City, Vietnam  
+84-286-296-7086  
<https://www.cloud9-solutions.com>