

---

# Bài 16

# Thiết kế và tạo CSDL

Module: **ADVANCED PROGRAMMING WITH PHP 2.0**

# Mục tiêu

---

- Trình bày được mục đích của việc thiết kế CSDL
- Mô tả được các bước để thiết kế cơ sở dữ liệu
- Thiết kế được cơ sở dữ liệu đơn giản
- Trình bày được các ràng buộc trong CSDL
- Trình bày được ý nghĩa của khoá chính
- Tạo được khoá chính trong một bảng
- Tạo được khoá ngoại trong một bảng
- Sử dụng được các kiểu dữ liệu cơ bản của MySQL

# Mục tiêu

---



- Trình bày được cú pháp câu lệnh INSERT
- Trình bày được cú pháp câu lệnh SELECT
- Trình bày được cú pháp câu lệnh UPDATE
- Trình bày được cú pháp câu lệnh DELETE
- Trình bày được cú pháp câu lệnh WHERE cơ bản

---

# Thảo luận

Thiết kế CSDL

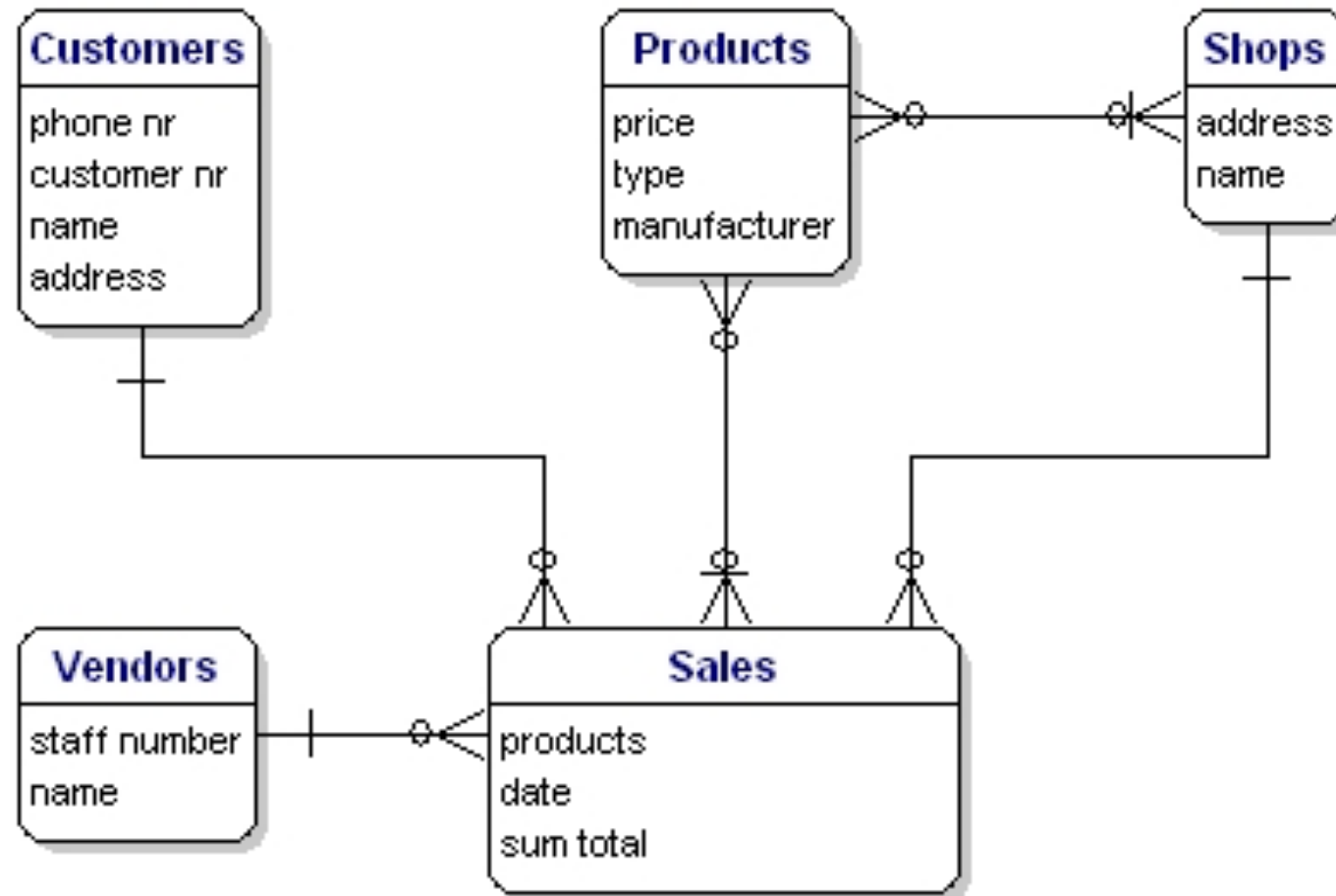
# Phân tích và Thiết kế CSDL

---



- Phân tích và Thiết kế CSDL là các thao tác được thực hiện để tìm ra một mô hình CSDL trong một tình huống nhất định
- Kết quả của bước này là một bản thiết kế của CSDL
- Dựa vào bản thiết kế của CSDL, chúng ta có thể viết các câu lệnh để tạo ra CSDL đó
- Bản thiết kế của CSDL thường được biểu diễn dưới dạng một Lưu đồ Thực thể Quan hệ (ERD - Entity-Relationship Diagram)

# ERD: Ví dụ



# Các bước phân tích và thiết kế CSDL Quan hệ

---



1. Xác định mục đích của CSDL
2. Tìm hiểu và tổ chức các thông tin cần lưu trữ
3. Phân chia thông tin vào trong các bảng
4. Xác định các trường dữ liệu của từng bảng
5. Xác định khóa chính của các bảng
6. Xác định mối quan hệ giữa các bảng
7. Làm mịn thiết kế
8. Áp dụng các quy tắc chuẩn hóa



# Ví dụ về hệ thống quản lý bán hàng

---

- Danh sách các cửa hàng
- Danh sách các sản phẩm
- Danh sách các nhà cung cấp
- Danh sách các nhà sản xuất
- Danh sách khách hàng
- Danh sách hóa đơn





# Tập hợp Dữ liệu, Tổ chức các bảng

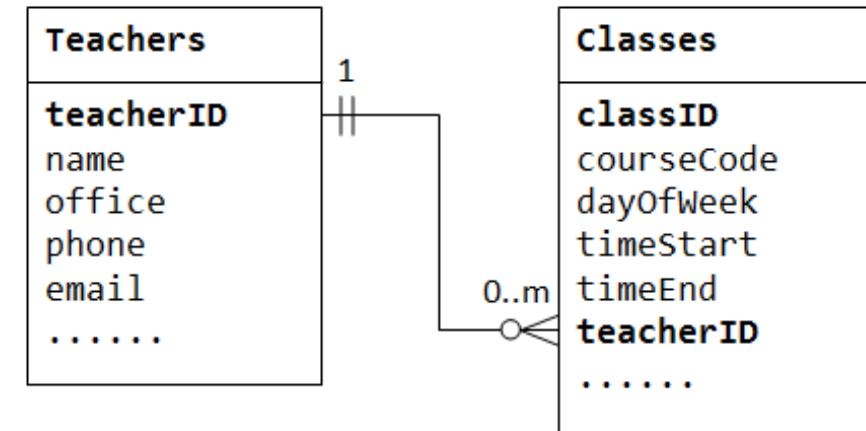
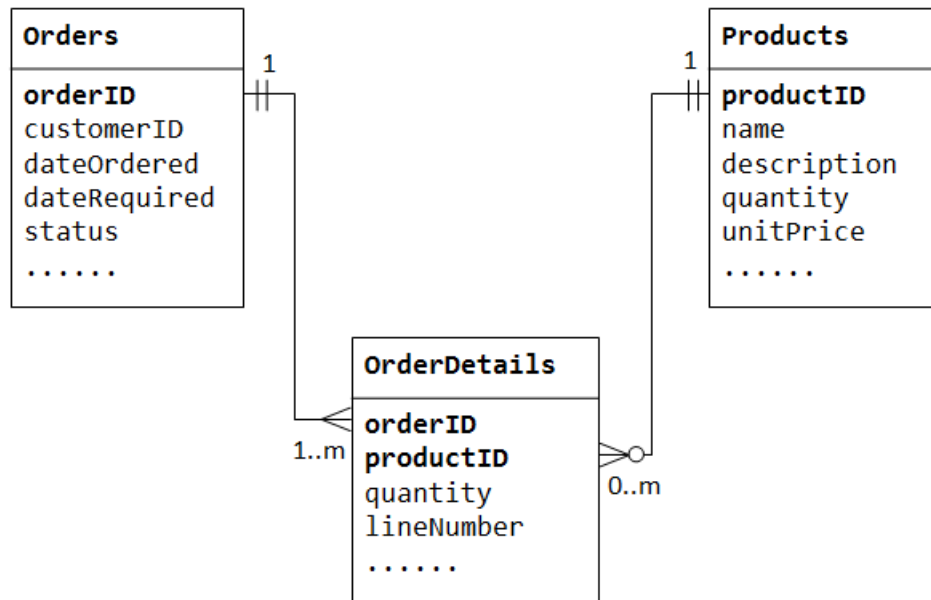
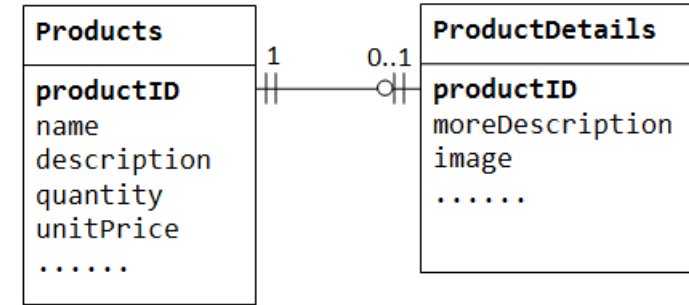
---

- Tìm hiểu và tổ chức các thông tin cần lưu trữ
- Phân chia thông tin vào trong các bảng
- Xác định các trường dữ liệu của từng bảng
- Xác định khóa chính của các bảng

# Tạo mối quan hệ giữa các bảng

Các loại quan hệ gồm:

- one-to-many(một ~ nhiều)
- many-to-many(nhiều ~ nhiều)
- one-to-one(một ~ một)



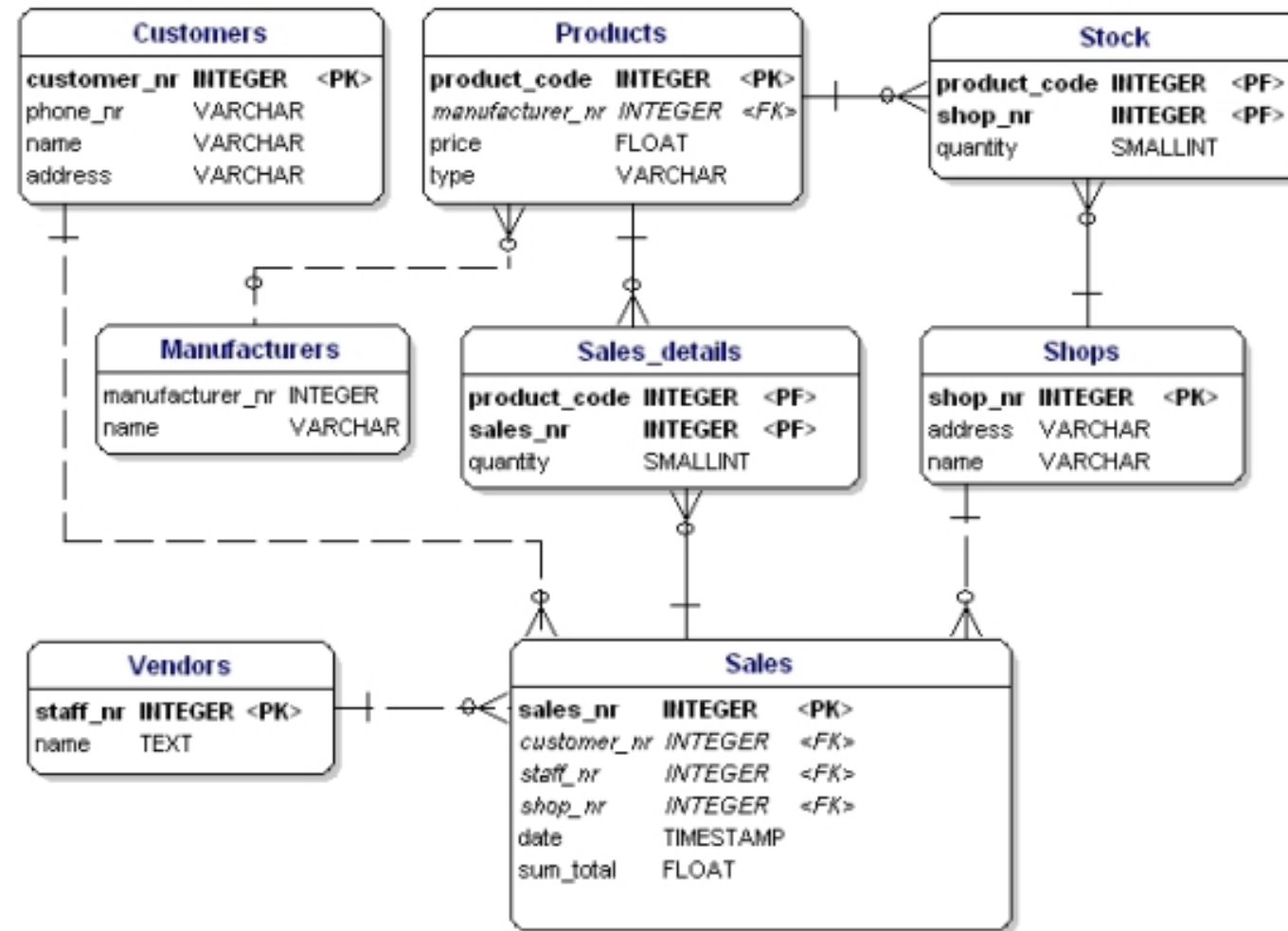
# Tính chỉnh và chuẩn hóa thiết kế

---



- Chuẩn hóa
  - Chuẩn thứ nhất(1NF)
  - Chuẩn thứ hai (2NF)
  - Chuẩn thứ ba (3NF)
- Quy tắc về tính toàn vẹn của Entity: Khóa chính không được phép NULL.

# ERD Website bán hàng





---

# Thảo luận

Tạo CSDL

# Tạo CSDL

---



- Câu lệnh tạo CSDL:  
`CREATE DATABASE`
- Câu lệnh tạo bảng:  
`CREATE TABLE`
- Cấu trúc của một bảng cần tuân theo các ràng buộc

# Constraint (ràng buộc)

---

- Constraint là các quy tắc (rule) được quy định cho bảng
- Sử dụng constraint, chúng ta có thể hạn chế những dữ liệu có thể đưa vào trong bảng
- Constraint giúp cho dữ liệu chính xác, tin cậy, toàn vẹn
- Dữ liệu đưa vào trong bảng cần tuân thủ các constraint
- Có thể quy định constraint cho bảng hoặc cột



# Một số ràng buộc thông dụng

---

- NOT NULL: Không cho phép giá trị NULL
- UNIQUE: Mỗi giá trị là duy nhất
- PRIMARY KEY: Khóa chính (Không NULL và là UNIQUE)
- FOREIGN KEY: Khóa ngoại (tham chiếu sang bảng khác)
- CHECK: Kiểm tra dựa vào một điều kiện
- DEFAULT: Quy định giá trị mặc định cho trường (nếu không có giá trị nào được nhập vào)
- INDEX: Giúp tăng tốc độ truy vấn dữ liệu



# NOT NULL

---



- Mặc định thì các trường trong bảng có thể chứa giá trị NULL
- Sử dụng từ khóa NOT NULL để bắt buộc các trường phải có giá trị khác NULL
- Ví dụ:

```
CREATE TABLE persons (  
    id int NOT NULL,  
    last_name varchar(255) NOT NULL,  
    first_name varchar(255) NOT NULL,  
    age int  
);
```

# UNIQUE

---



- Mặc định thì các trường trong bảng có thể chứa giá trị giống nhau
- Sử dụng từ khóa UNIQUE để bắt buộc các giá trị trong cột phải khác nhau
- Chẳng hạn: Số điện thoại, email...
- Ví dụ:

```
CREATE TABLE persons (  
    id int NOT NULL UNIQUE,  
    name varchar(255) NOT NULL,  
    email varchar(100) UNIQUE,  
    phone varchar(13) UNIQUE  
);
```

# CHECK

---



- Từ khóa CHECK được sử dụng để quy định điều kiện ràng buộc cho giá trị của một cột
- Ví dụ:

```
CREATE TABLE persons (  
    id int NOT NULL,  
    last_name varchar(255) NOT NULL,  
    first_name varchar(255),  
    age int,  
    CHECK (age>=18)  
);
```

# DEFAULT

---



- Từ khóa DEFAULT được sử dụng để quy định giá trị mặc định cho một trường
- Giá trị mặc định sẽ được sử dụng nếu không NULL được nhập vào

- Ví dụ:

```
CREATE TABLE persons (  
    id int NOT NULL,  
    last_name varchar(255) NOT NULL,  
    first_name varchar(255),  
    age int DEFAULT 0,  
    city varchar(255) DEFAULT 'Sandnes'  
);
```



# Primary Key (Khóa chính)

---

- Khóa chính là một cột (hoặc nhiều cột) được sử dụng để xác định một bản ghi duy nhất trong bảng
- Khóa chính là UNIQUE và NOT NULL
- Mỗi bảng chỉ có thể có 1 khóa chính
- Từ khóa PRIMARY KEY được sử dụng để định nghĩa khóa chính

# Primary Key: Ví dụ

---



```
CREATE TABLE users(  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    username VARCHAR(40),  
    password VARCHAR(255),  
    email VARCHAR(255)  
);
```

- Hoặc

```
CREATE TABLE roles(  
    id INT AUTO_INCREMENT,  
    name VARCHAR(50),  
    PRIMARY KEY(id)  
);
```



# Tạo khóa chính phức hợp

---

- Khóa chính phức hợp (composite) là khóa được tạo nên từ 2 hoặc nhiều cột:
- Ví dụ:

```
CREATE TABLE user_roles(  
    user_id INT NOT NULL,  
    role_id INT NOT NULL,  
    PRIMARY KEY(user_id,role_id),  
    ...  
);
```



# Tạo khóa chính sau khi tạo bảng

---

- Nếu một bảng đã tồn tại và chưa có khóa chính, chúng ta có thể bổ sung khóa chính thông qua câu lệnh ALTER TABLE.
- Ví dụ:

```
ALTER TABLE users  
  ADD PRIMARY KEY(id);
```





# Khóa ngoại (Foreign Key)

---

- Khóa ngoại là cơ chế để tạo liên kết giữa 2 bảng trong cùng CSDL
- Khóa ngoại được đặt trên một cột của bảng này và tham chiếu đến khóa chính của một bảng khác
- Kiểu dữ liệu của khóa chính và khóa ngoại phải giống nhau

# Khóa ngoại: Ví dụ

---



- CSDL bán hàng: Bảng **customers** và bảng **orders** có liên kết với nhau

```
CREATE TABLE customers(  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(40),  
  address VARCHAR(255),  
  email VARCHAR(255)  
);
```

- Cột **customer\_id** trong bảng **orders** là khóa ngoại, tham chiếu đến cột **id** của bảng **customers**

```
CREATE TABLE orders(  
  id INT AUTO_INCREMENT,  
  staff VARCHAR(50),  
  PRIMARY KEY(id),  
  customer_id INT, FOREIGN KEY (customer_id) REFERENCES customers(id)  
);
```

# Các kiểu dữ liệu thông dụng

---



- CHAR
- VARCHAR
- TEXT
- LONGTEXT
- INT
- BIGINT
- FLOAT
- DOUBLE
- DATE
- DATETIME
- TIMESTAMP
- TIME

# Tóm tắt bài học

---

- Constraint là các quy tắc (rule) được quy định cho bảng
- Dữ liệu đưa vào trong bảng cần tuân thủ các constraint
- PRIMARY KEY: Khóa chính (Không NULL và là UNIQUE)
- FOREIGN KEY: Khóa ngoại (tham chiếu sang bảng khác)
- Phân tích và Thiết kế CSDL là các thao tác được thực hiện để tìm ra một mô hình CSDL trong một tình huống nhất định
- Kết quả của bước này là một bản thiết kế của CSDL

---

# Hướng dẫn

- Hướng dẫn làm bài thực hành và bài tập
- Chuẩn bị bài tiếp: ***Thao tác với CSDL***