

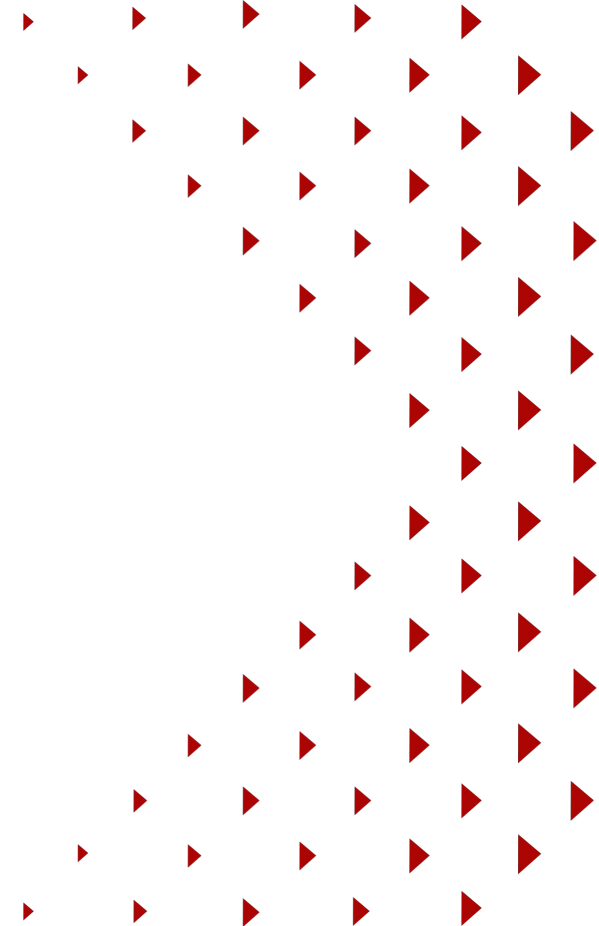


## **BÀI 2:**

# **Hệ quản trị CSDL MySQL**

Module: Fundamental Database

Phiên bản: 1.0

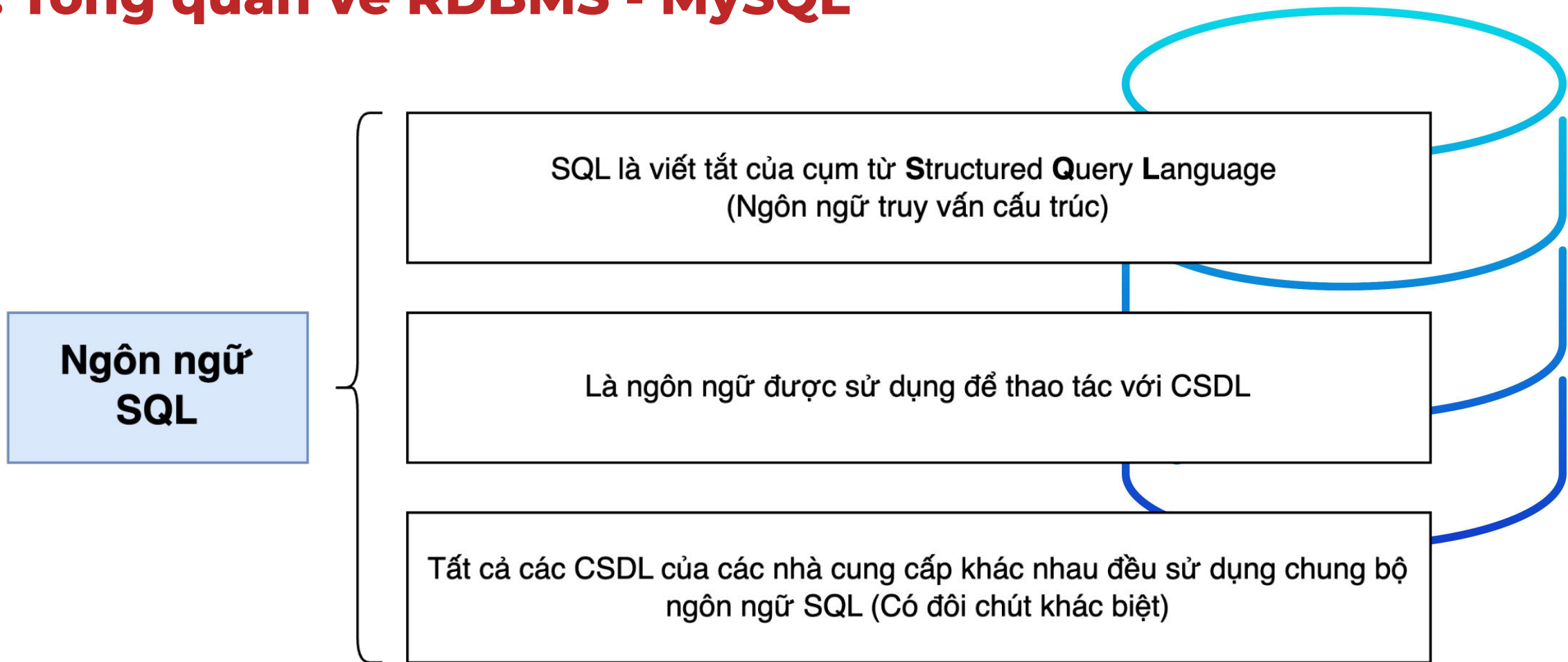


- 1. Tổng quan về RDBMS MySQL**
- 2. Làm việc với bảng - Các câu lệnh DDL**
- 3. Các ràng buộc (constraints)**

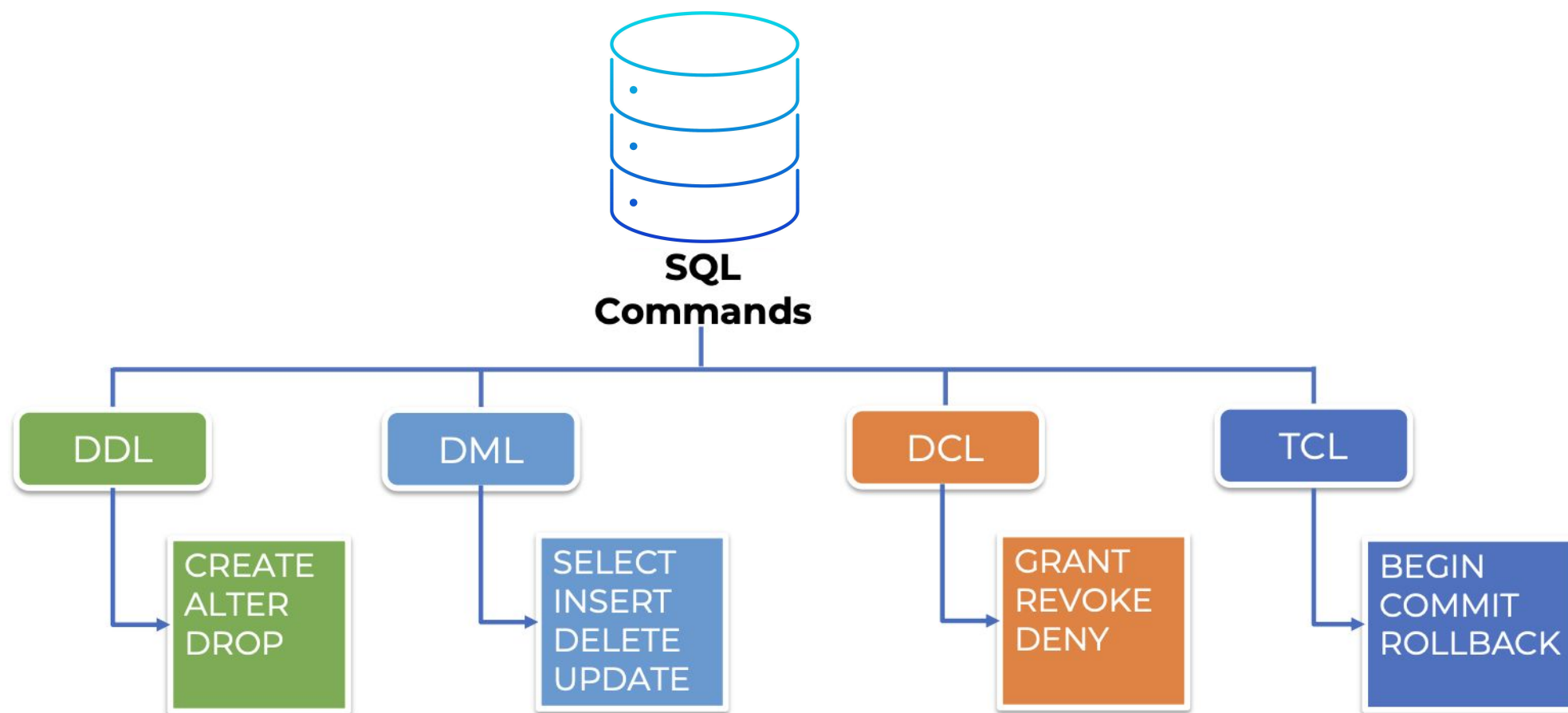
# 1. Tổng quan về RDBMS - MySQL



# 1. Tổng quan về RDBMS - MySQL



# 1. Tổng quan về RDBMS - MySQL



# 1. Tổng quan về RDBMS - MySQL

## Hệ quản trị CSDL MySQL

**Hệ quản trị CSDL MySQL** là một hệ quản trị CSDL dạng quan hệ (RDBMS) mã nguồn mở tương thích với nhiều nền tảng hệ điều hành (MacOs, Win ...)

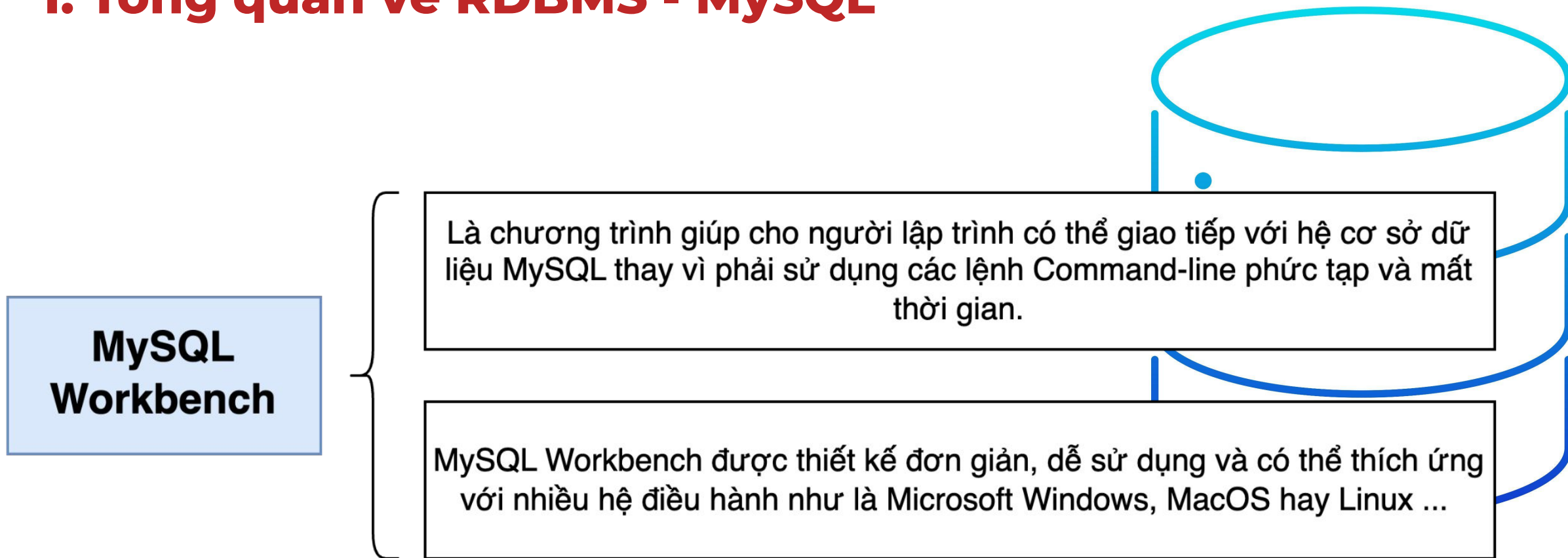
Hệ quản trị CSDL MySQL có rất nhiều phần mềm trong hệ sinh thái để phục vụ việc quản trị dữ liệu trong CSDL một cách đơn giản, hiệu quả

Có thể kể đến các phần mềm như **MySQL server** (Máy chủ quản trị dịch vụ MySQL), **MySQL client** (Ứng dụng kết nối và tương tác với MySQL server)

# 1. Tổng quan về RDBMS - MySQL



# 1. Tổng quan về RDBMS - MySQL





## 2. Làm việc với bảng - Các câu lệnh DDL

### DDL - Data Definition Language

*Ngôn ngữ định nghĩa dữ liệu giúp xác định cấu trúc hoặc lược đồ của cơ sở dữ liệu.*

*DDL bao gồm những câu lệnh **CREATE, DROP, ALTER, TRUNCATE**.*

## 2. Làm việc với bảng - Các câu lệnh DDL

### CREATE - Tạo mới bảng

```
CREATE TABLE table_name (  
    column1 data_type,  
    column2 data_type,  
    column3 data_type,  
    ...  
)
```

- **table\_name**: tên bảng
- **column**: tên cột
- **data\_type**: kiểu dữ liệu



```
CREATE TABLE Persons (  
    PersonID int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)  
);
```

## 2. Làm việc với bảng - Các câu lệnh DDL

### Kiểu dữ liệu trong MySQL - Kiểu dữ liệu số

<b>INT</b>	Một số nguyên với kích cỡ thông thường, có thể là signed (có dấu) hoặc unsigned (không dấu). Nếu có dấu, thì dãy giá trị có thể là từ -2147483648 tới 2147483647, Nếu không dấu thì dãy giá trị là từ 0 tới 4294967295. Bạn có thể xác định một độ rộng lên tới <b>11 chữ số</b> .
<b>TINYINT</b>	Một số nguyên với kích cỡ rất nhỏ, có thể là signed hoặc unsigned. Nếu có dấu, thì dãy giá trị có thể là từ -128 đến 127, Nếu không dấu thì dãy giá trị là từ 0 tới 255. Bạn có thể xác định một độ rộng lên tới <b>4 chữ số</b> .
<b>SMALLINT</b>	Một số nguyên với kích cỡ nhỏ, có thể là signed hoặc unsigned. Nếu có dấu, thì dãy giá trị có thể là từ -32768 tới 32767, nếu không dấu thì dãy giá trị là từ 0 tới 65535. Bạn có thể xác định một độ rộng lên tới <b>5 chữ số</b> .
<b>MEDIUMINT</b>	Một số nguyên với kích cỡ trung bình, có thể là signed hoặc unsigned. Nếu có dấu, thì dãy giá trị có thể là từ -8388608 tới 8388607. Nếu không dấu thì dãy giá trị là từ 0 tới 16777215. Bạn có thể xác định một độ rộng lên tới <b>9 chữ số</b> .
<b>BIGINT</b>	Một số nguyên với kích cỡ lớn, có thể là signed hoặc unsigned. Nếu có dấu, thì dãy giá trị có thể là từ -9223372036854775808 tới 9223372036854775807 Nếu không dấu thì dãy giá trị là từ 0 tới 18446744073709551615. Bạn có thể xác định một độ rộng lên tới <b>20 chữ số</b> .

## 2. Làm việc với bảng - Các câu lệnh DDL

### Kiểu dữ liệu trong MySQL - Kiểu dữ liệu số

<b>FLOAT(M,D)</b>	<p>Một số thực dấu chấm động không dấu. Bạn có thể định nghĩa <b>độ dài hiển thị (M)</b> và <b>số vị trí sau dấu phẩy (D)</b>. Điều này là không bắt buộc và sẽ có mặc định là <b>10,2</b> với <b>2</b> là <b>số vị trí sau dấu phẩy</b> và <b>10</b> là <b>số chữ số</b> (bao gồm các phần thập phân)</p> <p>Phần thập phân có thể lên tới <b>24 vị trí sau dấu phẩy</b> đối với một số FLOAT.</p>
<b>DOUBLE(M,D)</b>	<p>Một số thực dấu chấm động không dấu. Bạn có thể định nghĩa <b>độ dài hiển thị (M)</b> và <b>số vị trí sau dấu phẩy (D)</b>. Điều này là không bắt buộc và sẽ có mặc định là <b>16,4</b> với <b>4</b> là <b>số vị trí sau dấu phẩy</b> và <b>16</b> là <b>số chữ số</b> (bao gồm các phần thập phân).</p> <p>Phần thập phân có thể lên tới <b>53 vị trí sau dấu phẩy</b> đối với một số DOUBLE.</p> <p><b>REAL</b> là một từ khoá kiểu đồng nghĩa với <b>DOUBLE</b></p>
<b>DECIMAL(M,D)</b>	<p>Một kiểu khác của dấu chấm động không dấu. Mỗi chữ số thập phân chiếm 1 byte.</p> <p><b>Việc định nghĩa độ dài hiển thị (M) và số vị trí sau dấu phẩy (D) là bắt buộc.</b></p> <p><b>NUMERIC</b> là một từ đồng nghĩa cho <b>DECIMAL</b>.</p>

## 2. Làm việc với bảng - Các câu lệnh DDL

### Kiểu dữ liệu trong MySQL - Kiểu dữ liệu ngày/giờ

<b>DATE</b>	Một date trong định dạng <b>YYYY-MM-DD</b> , giữa <b>1000-01-01</b> và <b>9999-12-31</b> . Ví dụ, ngày 25 tháng 12 năm 2015 sẽ được lưu ở dạng 2015-12-25.
<b>DATETIME</b>	Một tổ hợp Date và Time trong định dạng <b>YYYY-MM-DD HH:MM:SS</b> , giữa <b>1000-01-01 00:00:00</b> và <b>9999-12-31 23:59:59</b> . Ví dụ, 3:30 chiều ngày 25 tháng 12, năm 2015 sẽ được lưu ở dạng 2015-12-25 15:30:00.
<b>TIMESTAMP</b>	Một Timestamp từ giữa nửa đêm ngày 1/1/1970 và 2037. Trông khá giống với định dạng DATETIME trước, khác biệt ở chỗ <b>không có dấu gạch nối giữa các số (YYYYMMDDHHMMSS)</b> Ví dụ, 3:30 chiều ngày 25 tháng 12, năm 2015 sẽ được lưu dưới dạng 20151225153000
<b>TIME</b>	Lưu time trong định dạng <b>HH:MM:SS</b> Ví dụ, 3:30 chiều sẽ có định dạng 15:30:00
<b>YEAR(M)</b>	Lưu 1 năm trong định dạng <b>2 chữ số hoặc 4 chữ số</b> . Nếu độ dài được xác định là 2 (ví dụ: YEAR(2)), YEAR có thể từ 1970 tới 2069 (70 tới 69). Nếu độ dài được xác định là 4, YEAR có thể từ 1901 tới 2155. Độ dài mặc định là 4

## 2. Làm việc với bảng - Các câu lệnh DDL

### Kiểu dữ liệu trong MySQL - Kiểu dữ liệu chuỗi ký tự

<b>CHAR(M)</b>	Một chuỗi có độ dài cố định có độ dài từ <b>1 tới 255 ký tự (ví dụ CHAR(5))</b> . Nếu giá trị thật của một trường kiểu Char không bằng với độ dài khai báo thì phần thiếu bên phải của nó sẽ được thêm bằng các ký tự trắng một cách tự động. Định nghĩa độ dài là không bắt buộc, giá trị mặc định là 1.
<b>VARCHAR(M)</b>	Dữ liệu kiểu chuỗi có độ dài thay đổi, có độ dài từ <b>1 đến 255 ký tự (ví dụ Varchar(24))</b> . Bạn phải định nghĩa độ dài khi tạo một trường VARCHAR.
<b>BLOB hoặc TEXT</b>	Trường kiểu này có <b>độ dài tối đa 65535 ký tự</b> . <b>BLOBs</b> là viết tắt của " <b>Binary Large Objects</b> ", và được sử dụng để <b>lưu trữ một lượng lớn dữ liệu nhị phân như các bức ảnh hoặc các loại tập tin khác</b> . Với <b>TEXT</b> , trường cũng lưu trữ được một lượng lớn dữ liệu. Điểm khác nhau giữa chúng là: khi sắp xếp và so sánh dữ liệu đã lưu trữ thì với <b>BLOBs là phân biệt kiểu chữ, còn với TEXT là không phân biệt kiểu chữ</b> . Bạn không phải xác định độ dài với BLOBs hoặc TEXT.
<b>ENUM</b>	Khi định nghĩa một trường kiểu này, tức là, ta đã chỉ ra một danh sách các đối tượng mà trường phải nhận (có thể là Null). Ví dụ, nếu ta muốn một trường nào đó chỉ nhận một trong các giá trị "A" hoặc "B" hoặc "C" thì ta phải định nghĩa kiểu ENUM cho nó như sau: <b>ENUM ('A', 'B', 'C')</b> . Và chỉ có các giá trị này (hoặc NULL) có thể xuất hiện trong trường đó.

## 2. Làm việc với bảng - Các câu lệnh DDL

### DROP - Xoá bảng

**DROP TABLE** *table\_name*

- ***table\_name***: tên bảng

→ **DROP TABLE** Shippers;

## 2. Làm việc với bảng - Các câu lệnh DDL

### ALTER - Chỉnh sửa bảng - Thêm mới cột

**ALTER TABLE** *table\_name*  
**ADD** *column data\_type*

- **table\_name**: tên bảng
- **column**: tên cột thêm mới
- **data\_type**: kiểu dữ liệu



```
ALTER TABLE Customers  
ADD Email  
varchar(255);
```



## 2. Làm việc với bảng - Các câu lệnh DDL

### ALTER - Chỉnh sửa bảng - Xoá cột

**ALTER TABLE** *table\_name*  
**DROP COLUMN** *column*

- **table\_name**: tên bảng
- **column**: tên cột cần xóa



**ALTER TABLE** Customers  
**DROP COLUMN** Email;

## 2. Làm việc với bảng - Các câu lệnh DDL

### ALTER - Chỉnh sửa bảng - Sửa kiểu dữ liệu cột

**ALTER TABLE** *table\_name*  
**MODIFY COLUMN** *column data\_type*

- **table\_name**: tên bảng
- **column**: tên cột cần sửa
- **data\_type**: kiểu dữ liệu



```
ALTER TABLE Customers  
MODIFY COLUMN Email  
varchar(15);
```

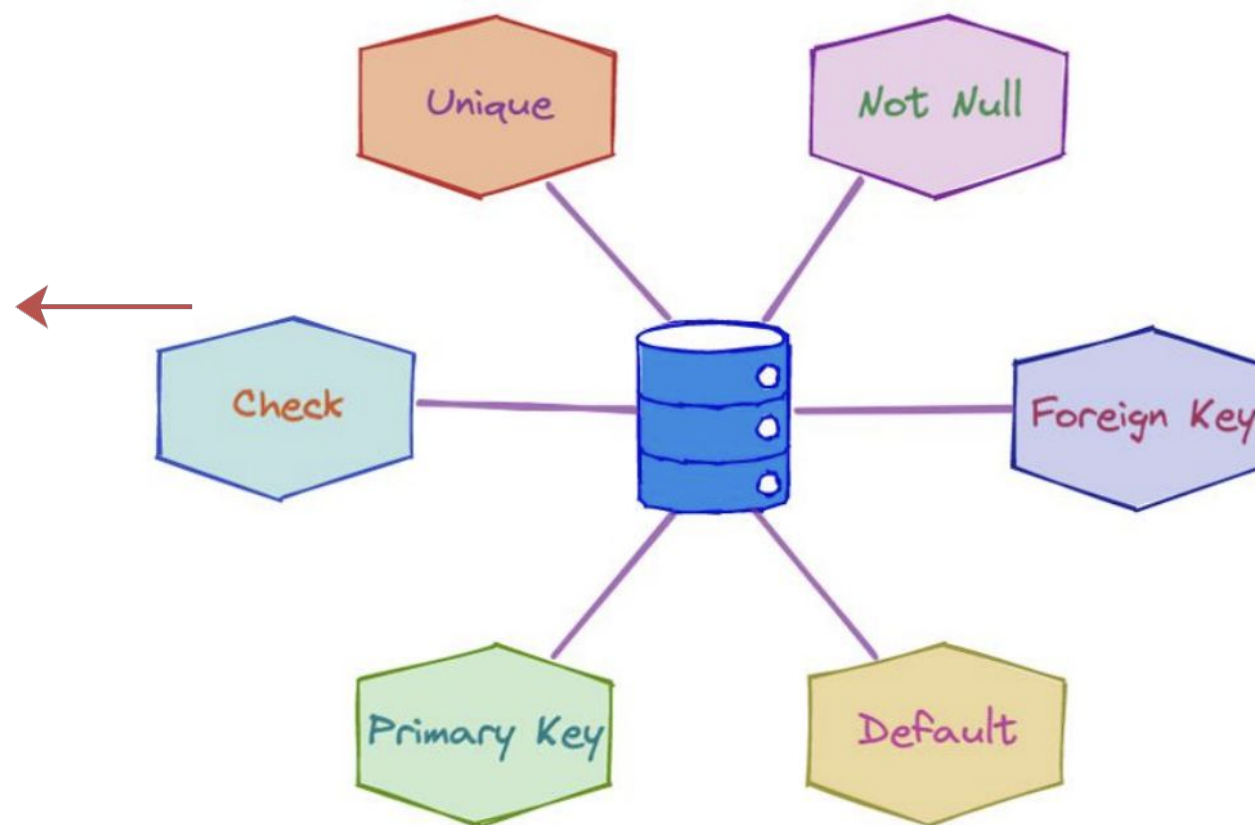
### 3. Constraints - Các ràng buộc trong SQL

## Constraints - Các ràng buộc trong SQL

- *Constraint là các quy tắc (rule) được quy định cho bảng*
- *Sử dụng constraint, chúng ta có thể hạn chế những dữ liệu không phù hợp 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*

### 3. Constraints - Các ràng buộc trong SQL

```
CREATE TABLE table_name (  
  column1 data_type constraint,  
  column2 data_type constraint,  
  column3 data_type constraint,  
  ...  
)
```



### 3. Constraints - Các ràng buộc trong SQL

#### Ràng buộc NOT NULL

```
CREATE TABLE student (  
  id INT NOT NULL  
  firstName VARCHAR(255) NOT NULL,  
  lastName VARCHAR(255) NOT NULL,  
  ...  
)
```

- Mặc định các trường trong bảng có thể chứa giá trị NULL (rỗng)
- Sử dụng **NOT NULL** để bắt các trường giá trị phải có giá trị khác rỗng

## 3. Constraints - Các ràng buộc trong SQL

### Ràng buộc UNIQUE

```
CREATE TABLE student (  
  id INT NOT NULL  
  fullName VARCHAR(255) NOT NULL,  
  email VARCHAR(255) UNIQUE,  
  ...  
)
```

- Mặc định các trường trong bảng có thể chứa giá trị giống nhau
- Sử dụng **UNIQUE** để bắt các trường giá trị phải phải có giá trị là độc nhất

### 3. Constraints - Các ràng buộc trong SQL

#### Ràng buộc PRIMARY KEY

```
CREATE TABLE student (  
  id INT PRIMARY KEY  
  fullName VARCHAR(255) NOT NULL,  
  email VARCHAR(255) UNIQUE,  
  ...  
)
```

- Mặc định các trường trong bảng có thể là các trường không khoá
- Sử dụng **PRIMARY KEY** giúp biến cột hoặc tập hợp cột thành khoá chính
- Giá trị nằm trong cột khoá chính phải là độc nhất và khác NULL

### 3. Constraints - Các ràng buộc trong SQL

#### Ràng buộc FOREIGN KEY

```
CREATE TABLE order (  
    id INT NOT NULL  
    fullName VARCHAR(255) NOT NULL,  
    PRIMARY KEY (id)  
    FOREIGN KEY (personId) REFERENCES Person(personId)  
)
```

- Khóa ngoại (**Foreign key**) dùng để liên kết các bảng lại với nhau
- • Khóa ngoại được tạo nên từ 1 hoặc tập hợp các cột tham chiếu đến khóa chính của những bảng khác
- Bảng chứa khóa ngoại gọi là bảng con, bảng chứa khóa chính gọi là bảng cha



### 3. Constraints - Các ràng buộc trong SQL

#### Ràng buộc CHECK

```
CREATE TABLE Student (  
  id INT NOT NULL  
  fullName VARCHAR(255) NOT NULL,  
  age INT NOT NULL,  
  CHECK (age <= 19)  
)
```

- Ràng buộc **CHECK** giới hạn giá trị có thể được nhập vào trong một cột hoặc logic nhiều cột với nhau.

### 3. Constraints - Các ràng buộc trong SQL

#### Ràng buộc DEFAULT

```
CREATE TABLE Person (  
  id INT NOT NULL  
  fullName VARCHAR(255) NOT NULL,  
  email VARCHAR(255) UNIQUE,  
  city VARCHAR(255) DEFAULT 'Hanoi'  
)
```

- Ràng buộc **DEFAULT** định nghĩa cho một cột trong bảng, nó gán giá trị mặc định cho cột đó nếu khi thêm bản ghi mới mà không nhập dữ liệu cho cột đó.

- ❑ **Nắm được cách làm việc cơ bản với RDBMS MySQL**
- ❑ **Nắm được cách làm việc với bảng**
- ❑ **Nắm được các ràng buộc có trong SQL (constraints)**



# KẾT THÚC

HỌC VIỆN ĐÀO TẠO LẬP TRÌNH CHẤT LƯỢNG NHẬT BẢN