



Phân loại các câu lệnh SQL

Các lệnh SQL được phân vào 1 trong 3 nhóm sau:

- ****DDL**** - ****Data Definition Language****: Ngôn ngữ định nghĩa dữ liệu, ví dụ như các lệnh CREATE, ALTER, DROP
- ****DML**** - ****Data Manipulation Language****: Ngôn ngữ thao tác dữ liệu, ví dụ như các lệnh SELECT, INSERT, UPDATE, DELETE
- ****DCL**** - ****Data Control Language****: Ngôn ngữ điều khiển dữ liệu, ví dụ như các lệnh GRANT, REVOKE

Khởi tạo CSDL

- Tạo database:

```
CREATE DATABASE tên_database;
```

VD: create database ManageSchool;

- Hiển thị tất cả các database hiện tại:

```
SHOW DATABASES;
```

Các kiểu dữ liệu trong CSDL

MySQL DATA TYPES

DATE TYPE	SPEC	DATA TYPE	SPEC
CHAR	String (0 - 255)	INT	Integer (-2147483648 to 214748-3647)
VARCHAR	String (0 - 255)	BIGINT	Integer (-9223372036854775808 to 9223372036854775807)
TINYTEXT	String (0 - 255)	FLOAT	Decimal (precise to 23 digits)
TEXT	String (0 - 65535)	DOUBLE	Decimal (24 to 53 digits)
BLOB	String (0 - 65535)	DECIMAL	"DOUBLE" stored as string
MEDIUMTEXT	String (0 - 16777215)	DATE	YYYY-MM-DD
MEDIUMBLOB	String (0 - 16777215)	DATETIME	YYYY-MM-DD HH:MM:SS
LONGTEXT	String (0 - 4294967295)	TIMESTAMP	YYYYMMDDHHMMSS
LOBLOB	String (0 - 4294967295)	TIME	HH:MM:SS
TINYINT	Integer (-128 to 127)	ENUM	One of preset options
SMALLINT	Integer (-32768 to 32767)	SET	Selection of preset options
MEDIUMINT	Integer (-8388608 to 8388607)	BOOLEAN	TINYINT(1)

Các kiểu dữ liệu thường dùng nhất

- Numeric: *int, float, double*
- String: *varchar, char, text*
- Time: *Date, Datetime, Timestamp*

char vs. varchar

Value	CHAR(4)	Storage Required	VARCHAR(4)	Storage Required
"	' '	4 bytes	"	1 byte
'ab'	'ab '	4 bytes	'ab'	3 bytes
'abcd'	'abcd'	4 bytes	'abcd'	5 bytes
'abcdefgh'	'abcd'	4 bytes	'abcd'	5 bytes

Note:

char: Cấp phát bộ nhớ tĩnh (hiểu một cách đơn giản, nếu khai báo dung lượng 4 bytes mà thực tế chỉ dùng 3 ô nhớ thì kích thước vẫn là 4 bytes)

varchar: Cấp phát bộ nhớ động (kích thước tính theo số ô nhớ thực tế được sử dụng)

So sánh datetime và timestamp

datetime

- Được sử dụng để lưu trữ giá trị cụ thể và tĩnh, không bị ảnh hưởng bởi bất kỳ thay đổi nào.
- KHÔNG chịu ảnh hưởng bởi các cài đặt liên quan đến time zone.
- Giới hạn lưu trữ:

1000-01-01 00:00:00 đến 9999-12-31 23:59:59

timestamp

- Được sử dụng để theo dõi các thay đổi đối với các bản ghi và cập nhật mỗi khi bản ghi được thay đổi.
- Chịu ảnh hưởng bởi các cài đặt liên quan đến time zone.
- Giới hạn lưu trữ:

1970-01-01 00:00:01 UTC đến 2038-01-19 03:14:07

Ví dụ về Timestamp Input

- Nếu đang ở Việt Nam:

+ Insert dữ liệu vào column type *timestamp* với giá trị 1970-01-01 00:00:01 thì dữ liệu lưu sẽ bị **lỗi**.

!!! Lý do là ở Việt Nam chúng ta có múi giờ UTC +7

+ Nếu lưu với dữ liệu có giá trị 1970-01-01 07:00:01 thì sẽ thành công.

Tạo bảng (CREATE TABLE)

Cấu trúc:

```
CREATE TABLE tên_bảng (  
  cột_1 <kiểu_dữ_liệu_của_cột_1>,  
  cột_2 <kiểu_dữ_liệu_của_cột_2>,  
  ...  
  cột_n <kiểu_dữ_liệu_của_cột_n>  
)
```

Ví dụ:

```
CREATE TABLE Student(  
  id int,  
  name varchar(20),  
  age int,  
  birth_date date  
)
```

Câu lệnh INSERT

Cấu trúc:

```
INSERT INTO table_name  
VALUES(value, value,...)
```

Ví dụ:

```
INSERT INTO student  
VALUES(1, 'Chu Tien Dat', 24, '1998-05-01')
```

Câu lệnh **SELECT**

Cấu trúc:

```
SELECT *FROM tên_bảng
```

hoặc

```
SELECT cột_1, cột_2, ..., cột_n
```

```
FROM tên_bảng
```

```
WHERE điều_kiện
```

Ví dụ:

```
SELECT name, age FROM Student
```

```
SELECT * FROM Student
```

```
WHERE age > 20
```

Toán tử điều kiện trong **WHERE**

- AND: Nối 2 điều kiện
- OR: Chỉ cần 1 trong các điều kiện thỏa mãn
- IN: Có thể sử dụng thay cho nhiều toán tử OR
- BETWEEN: Tương đương <điều kiện 1> AND <điều kiện 2>

Toán tử điều kiện trong **WHERE**

```
SELECT *FROM Student WHERE age = 24 AND name='Hoang'
```

```
SELECT *FROM Student WHERE age = 24 OR name='Hoang'
```

```
SELECT *FROM Student WHERE age IN (20, 24)
```

tương đương $\text{age} = 20$ or $\text{age} = 24$

```
SELECT *FROM Student WHERE age BETWEEN 20 AND 24
```

tương đương $\text{age} \geq 20$ AND $\text{age} \leq 24$

Toán tử LIKE

LIKE kết hợp với %

- Đại diện cho 0 hoặc nhiều ký tự
- Ví dụ:

```
SELECT *FROM Student WHERE  
name LIKE 'H%'
```

-> Tìm tất cả học sinh có tên bắt đầu bằng chữ H

LIKE kết hợp với _

- Đại diện cho duy nhất 1 ký tự
- Ví dụ:

```
SELECT *FROM Student WHERE  
name LIKE '_n'
```

-> Tìm tất cả học sinh với tên có 2 ký tự và kết thúc là chữ cái 'n'

VD: An, ...

```
SELECT *FROM Student WHERE  
name Like '_n%'
```

VD: Anh, ...

SELECT DISTINCT

- Loại bỏ dữ liệu trùng lặp trong các cột được select
- Ví dụ:

SELECT DISTINCT name FROM Student

-> Lấy ra giá trị không trùng lặp của cột *name*

SELECT DISTINCT name, age FROM Student

-> Lấy ra các cặp giá trị không trùng lặp của 2 cột *name, age*

ORDER BY (ASC / DESC)

- Dùng để sắp xếp kết quả trả về (tăng dần / giảm dần)
- Ví dụ:

```
SELECT * FROM Student
```

```
WHERE name LIKE 'Hoang%' ORDER BY name ASC
```

-> Tìm tất cả những người có họ Hoàng và sắp xếp tên tăng dần