

Lập trình Website tĩnh - SQL ADVANCE

Ths. Vũ Duy Khương



Nội dung

1

Kiểu dữ liệu trong SQL

2

Định nghĩa dữ liệu

3

SQL Join

4

SQL union

5

Một số lệnh truy vấn nâng cao

6

Bài tập thực hành



Kiểu dữ liệu trong SQL

- ❖ Kiểu dữ liệu trong SQL định nghĩa một kiểu giá trị mà một cột có thể chứa
- ❖ Trong một bảng, mỗi cột cần có một tên và một kiểu dữ liệu
- ❖ Kiểu dữ liệu thì khá đa dạng với các Database khác nhau



Các kiểu dữ liệu trong MySQL

- ❖ Kiểu numeric
- ❖ Kiểu datetime, date, timestamp
- ❖ Kiểu string

Kiểu numeric

Kiểu số nguyên

Kiểu dữ liệu	Số byte	Giá trị nhỏ nhất	Giá trị lớn nhất
TINYINT	1	-128	127
SMALLINT	2	-32768	32767
MEDIUMINT	3	-8388608	8388607
INT	4	-2147483648	2147483647
BIGINT	8	9223372036854775808	9223372036854775807

Kiểu numeric

Kiểu dấu chấm động

Kiểu dữ liệu	Số byte	Giá trị nhỏ nhất	Giá trị lớn nhất
FLOAT	4	-3.402823466E+38	-1.175494351E-38
DOUBLE	8	-1.7976931348623 157E+ 308	-2.2250738585072014 E- 308



Kiểu datetime,date, timestamps

Kiểu dữ liệu	hiển thị	Phạm vi	Mô tả
DATETIME	YYYY-MM-DD HH:MM:SS	'1000-01-01 00:00:00' to '9999-12-31 23:59:59'.	Sử dụng khi bạn cần giá trị lưu trữ cả hai thông tin ngày tháng và thời gian.
DATE	YYYY-MM-DD	'1000-01-01' to '9999-12-31'.	Sử dụng khi bạn muốn lưu trữ chỉ thông tin ngày tháng.
TIMESTAMP	YYYY-MM-DD HH:MM:SS	'1970-01-01 00:00:01' UTC to '2038-01-19 03:14:07' UTC	lưu trữ cả hai thông tin ngày tháng và thời gian



So sánh datetime và timestamp

❖ Giống nhau:

Đều lưu trữ cả 2 thông tin là ngày tháng và thời gian

❖ Khác nhau:



So sánh datetime và timestamp

Khác nhau

	datetime	timestamp
Phạm vi	'1000-01-01 00:00:00' tới '9999-12-31 23:59:59'	'1970-01-01 00:00:01' UTC tới '2038-01-19 03:14:07'
Chuyển đổi múi giờ	Không thay đổi	được chuyển đổi từ múi giờ hiện tại sang UTC trong khi lưu trữ, và chuyển ngược trở lại từ UTC sang múi giờ hiện tại trong lúc lấy ra



Kiểu varchar

Kiểu dữ liệu	Phạm vi	Ý nghĩa
Char	Giá trị từ 0 tới 255	Lưu chuỗi. Các cột lúc khai báo ban đầu phải có kích thước cụ thể
Varchar	Giá trị từ 0 tới 255	Lưu chuỗi. Kích thước có thể thay đổi

Kiểu text

Kiểu dữ liệu	Loại	Phạm vi
TEXT	TINYBLOB	Chiều dài tối đa là 255 ký tự
	MEDIUMBLOB	Chiều dài tối đa là 16777215 ký tự
	LOBLOB	Chiều dài tối đa là 4294967295 ký tự.



Định nghĩa dữ liệu

❖ SQL Database

- Lệnh create database
- Lệnh drop database
- Lệnh select database

❖ SQL Table

- Lệnh Create table
- Lệnh insert into table
- Lệnh Drop table
- Lệnh Delete table
- Lệnh Rename table
- Lệnh Truncate table
- Lệnh Alter table



SQL database

Create database

Cú pháp: **create database** ten_database

Ví dụ: **create database** lib_management

Result		
<code><T> show databases</code> <i>Enter a SQL expression to filter results (use Ctrl+Space)</i>		
Grid		Database
1		design-csdl
2		hibernate
3		information_schema
4		jdbc
5		lib_management



SQL database

Select database

Cú pháp: **use** ten_cSDL

Kiểm tra các database khả dụng: **show databases**

Result	
<code>show databases</code> Enter a SQL expression to filter results (use Ctrl+Space)	
Grid	Database
	1 design-csdl
	2 hibernate
	3 information_schema
	4 jdbc
	5 lib_management

SQL database

Drop database

Cú pháp: **drop database** ten_database

Ví dụ: **drop database** lib_management;

Statistics	
drop database lib_management	
Name	Value
Updated Rows	0
Query	drop database lib_management
Finish time	Sat Jan 23 14:51:18 ICT 2021



SQL table – Create table

Cú pháp:

```
CREATE TABLE <Table_Name> (  
    Column_Name_Id    DataType [NOT NULL]    [AUTO_INCREMENT]  
    ,  
    Column_name2      DataType [NOT NULL] ,  
    .....  
    Column_nameN      DataType [NOT NULL] ,  
    Primary Key (Column_Name_Id)  
);
```




SQL table – Create table

- ❖ Chú ý một bảng không nhất thiết phải có khóa chính.
- ❖ Phải có dấu chấm phẩy cuối dòng lệnh.
- ❖ Thuộc tính **NOT NULL** được sử dụng nếu bạn không muốn trường này null
- ❖ Thuộc tính **AUTO_INCREMENT** nói với MySQL tự gán giá trị tăng dần cho trường ID
- ❖ Từ khóa **PRIMARY KEY** được sử dụng để định nghĩa cột này là một khóa chính. Bạn có thể sử dụng nhiều cột ngăn cách nhau bởi dấu phẩy để định nghĩa một khóa chính



SQL table – Create table

Ví dụ: Tạo bảng sinh viên

```
create table students (  
    id int NOT NULL AUTO_INCREMENT,  
    full_name VARCHAR(255) not null,  
    id_student varchar(255) not null,  
    course_th int not null,  
    major varchar(255) not null,  
    address varchar(255) not null,  
    PRIMARY KEY (id)  
);
```



SQL table – insert into table

❖ Cú pháp 1:

```
INSERT INTO TABLE_TEN (cot1, cot2,...cotN)  
VALUES (giatri1, giatri2, giatri3,...giatriN);
```

❖ Cú pháp 2:

```
INSERT INTO ten_bang_1[(cot1, cot2, ... cotN)] SELECT cot1, cot2,  
...cotN FROM ten_bang_2[WHERE dieu_kien];
```



SQL table – insert into table

Ví dụ:

```
insert into students(full_name,id_student,course_th,major,address)
values('Anh','20122006',57,'IT','Hà nội');
insert into students(full_name,id_student,course_th,major,address)
values('Hồng','20132009',57,'Hóa dược','Hà nội');
insert into students(full_name,id_student,course_th,major,address)
values('Nam','20152009',56,'Cơ khí','Hà nội');
```



123 id 🔼🔼	ABC full_name 🔼🔼	ABC id_student 🔼🔼	123 course_th 🔼🔼	ABC major 🔼🔼	ABC address 🔼🔼
1	Anh	20122006	57	IT	Hà nội
2	Hồng	20132009	57	Hóa dược	Hà nội
3	Nam	20152009	56	Cơ khí	Hà nội



SQL table – drop table

Drop table:

Cú pháp:

drop table ten_table

Ví dụ:

drop table students;



SQL table – delete from table

Delete from table:

❖ Cú pháp:

delete from ten_table **where** [condition]

❖ Ví dụ:

delete from students **where** course_th = 56



SQL table – rename, truncate

Rename table

- **Cú pháp:** `alter table ten_cu rename to ten_moi`
- **Ví dụ:**

`alter table students1 rename to students2;`

Truncate table

- **Cú pháp:** `truncate table ten_table`
- **Ví dụ:** `truncate table students1`



SQL table – alter

Thêm cột vào bảng

➤ Cú pháp:

ALTER TABLE <Table_Name>

Add <Column_Name> Data_Type [NOT NULL DEFAULT value];

➤ Ví dụ:

ALTER TABLE students **ADD** phone varchar(255);



SQL table – alter

Thay đổi tên cột

➤ Cú pháp

ALTER TABLE table_name **CHANGE** old_column new_column
datatype(length);

➤ Ví dụ

ALTER TABLE students **change** phone s_phone varchar(255);



SQL table – alter

Xóa cột trong bảng

➤ Cú pháp:

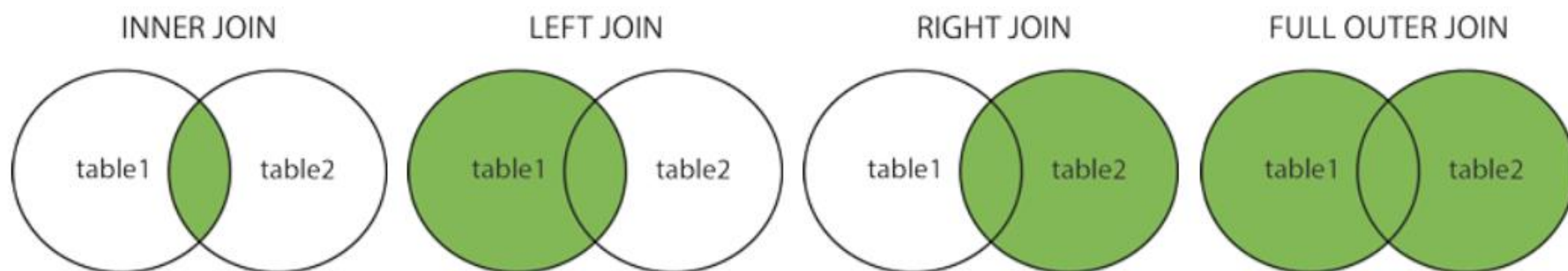
ALTER TABLE <Table_Name> **DROP** <Column_Name>;

➤ Ví dụ

ALTER TABLE students **drop** s_phone;

SQL join

- ❖ Mệnh đề join trong SQL được dùng để kết hợp các bản ghi từ 2 hoặc nhiều bảng trong 1 database
- ❖ Các loại join trong SQL
inner join, left join, right join, full join



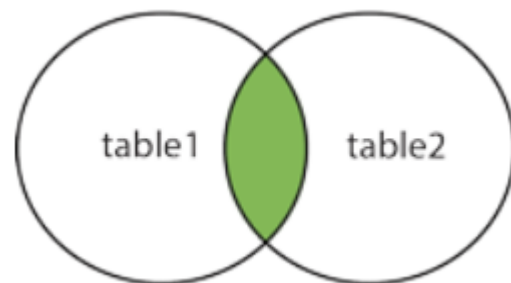


SQL join

❖ Inner join: Trả lại các bản ghi có giá trị thỏa mãn điều kiện join trong 2 bảng

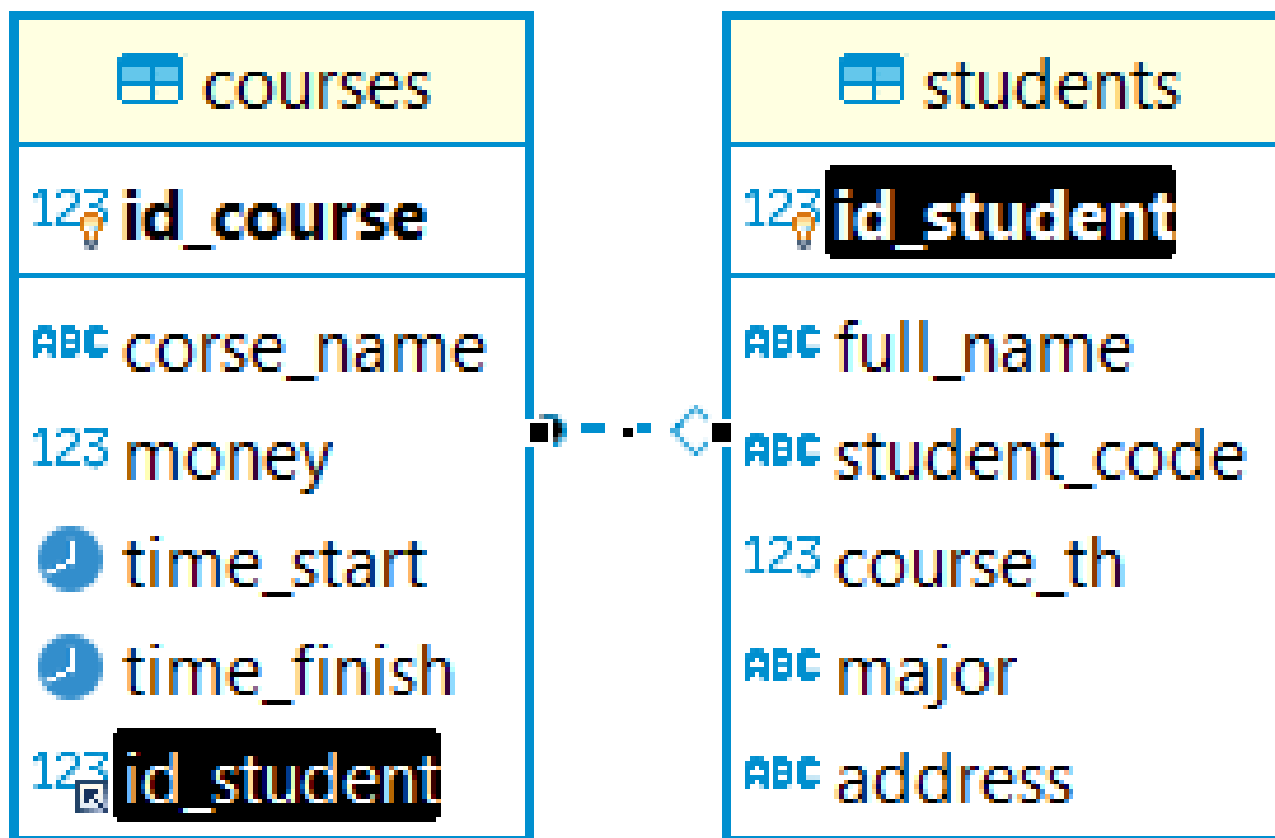
❖ Cú pháp:

```
SELECT column_name(s)  
FROM table1  
INNER JOIN table2  
ON table1.column_name = table2.column_name;
```



SQL join

❖ Ví dụ: có 2 bảng students và courses liên kết với nhau





SQL join

Lấy tất cả sinh viên có đăng ký môn học và thông tin các môn đã đăng ký

```
select
    s.id_student,
    s.full_name,
    c.corse_name
from
    students s
join courses c on
    s.id_student = c.id_student;
```

`<>T select s.id_student,s.full_name,c.corse_name from stu` | `Enter c`

	123 id_student	ABC full_name	ABC corse_name
Grid	1	Anh	Lập trình java
	2	Anh	Lập trình C
Text	3	Hồng	Hóa công
<>T	4	Nam	Thiết kế CNC



SQL join

❖ **left join:** Trả lại tất cả các bản ghi của bảng bên trái(table1) và các bản ghi bảng bên phải nếu thỏa mãn điều kiện join, nếu không thỏa mãn điều kiện join thì bên phải là các giá trị null

❖ **Cú pháp:**

```
SELECT column_name(s)  
FROM table1  
LEFT JOIN table2  
ON table1.column_name = table2.column_name;
```



SQL join

Ví dụ: Lấy tất cả danh sách sinh viên và môn học mà sinh viên đã đăng ký nếu có

```
select
    s.id_student,
    s.full_name,
    c.corse_name
from
    students s
left join courses c on
    s.id_student = c.id_student;
```

123 id_student 🔍	ABC full_name 🔍	ABC corse_name 🔍
1	Anh	Lập trình java
1	Anh	Lập trình C
2	Hồng	Hóa công
4	Nam	Thiết kế CNC
5	Sơn	[NULL]
6	Hà	[NULL]



SQL join

❖ **Full join:** Trả lại tất cả các bản ghi có ở bảng bên trái, bên phải.

❖ **Cú pháp:**

SELECT *column_name(s)*

FROM *table1*

FULL JOIN *table2*

ON *table1.column_name = table2.column_name;*

Chú ý: full join không hỗ trợ trong MySql



SQL join

❖ **Right join:** Trả lại tất cả các bản ghi của bảng bên phải (table2) và các bản ghi bảng bên trái (table1). Nếu không thỏa mãn điều kiện thì bên trái là các giá trị null

❖ **Cú pháp:**

```
SELECT column_name(s)  
FROM table1  
RIGHT JOIN table2  
ON table1.column_name = table2.column_name;
```

SQL join

Ví dụ: Lấy tất cả các khóa học và các sinh viên học môn đó nếu có

select

```
s.id_student,  
s.full_name,  
c.corse_name
```

from

```
students s
```

RIGHT join courses c **on**

```
s.id_student = c.id_student;
```

123 id_student 🔍	ABC full_name 🔍	ABC corse_name 🔍
1	Anh	Lập trình java
1	Anh	Lập trình C
4	Nam	Thiết kế CNC
2	Hồng	Hóa công



SQL union

- ❖ Mệnh đề **UNION** trong SQL được sử dụng để kết hợp kết quả của hai hoặc nhiều lệnh **SELECT** mà không trả về bất cứ bản sao nào của bản ghi
- ❖ Để sử dụng **UNION**, mỗi lệnh **SELECT** phải có cùng số cột đã chọn, cùng kiểu dữ liệu, và chúng có cùng thứ tự.



SQL union

Cú pháp:

```
select
    cot1 [,cot2 ]
from
    bang1 [bang2]
    [where dieu_kien]
union
select
    cot1 [cot2 ]
from
    bang1 [bang2 ]
    [where dieu_kien]
```

SQL union

Ví dụ:

```
select s.id_student, s.full_name,c.corse_name
from
    students s
RIGHT join courses c on
    s.id_student = c.id_student
union
select s.id_student, s.full_name,c.corse_name
from
    students s
RIGHT join courses c on
    s.id_student = c.id_student;
```

SQL union

Kết quả

Result ✕				
<code><>T select s.id_student, s.full_name, c.corse_name from stu</code> <code>↕ ↗ ↘ ↖</code> Enter				
Grid		<code>123</code> id_student <code>↕ ↗ ↘ ↖</code>	<code>ABC</code> full_name <code>↕ ↗ ↘ ↖</code>	<code>ABC</code> corse_name <code>↕ ↗ ↘ ↖</code>
	1	1	Anh	Lập trình java
Text	2	1	Anh	Lập trình C
	3	4	Nam	Thiết kế CNC
	4	2	Hồng	Hóa công



SQL union

Union all: Mệnh đề **UNION ALL** trong SQL được sử dụng để kết hợp kết quả của hai hoặc nhiều lệnh **SELECT** mà không trả về cả bản sao của các bản ghi

Cú pháp: Tương tự union



SQL union

Ví dụ: `select s.id_student, s.full_name, c.corse_name
from
students s
RIGHT join courses c on
s.id_student = c.id_student
union all
select s.id_student, s.full_name, c.corse_name
from
students s
RIGHT join courses c on
s.id_student = c.id_student;`

SQL union

Kết quả:

Result				
<pre>select s.id_student, s.full_name,c.corse_name from stu</pre>				
Grid		123 id_student	ABC full_name	ABC corse_name
	1	1	Anh	Lập trình java
	2	1	Anh	Lập trình C
Text	3	4	Nam	Thiết kế CNC
	4	2	Hồng	Hóa công
	5	1	Anh	Lập trình java
	6	1	Anh	Lập trình C
Chart	7	4	Nam	Thiết kế CNC
	8	2	Hồng	Hóa công



Một số lệnh truy vấn nâng cao

Mệnh đề order by:

- Được sử dụng để sắp xếp dữ liệu theo thứ tự tăng dần hoặc theo thứ tự giảm dần, trên một hoặc nhiều cột
- Một số Database sắp xếp kết quả truy vấn theo thứ tự tăng dần theo mặc định

Một số lệnh truy vấn nâng cao

Cú pháp:

```
SELECT danh_sach_cot  
FROM ten_bang  
[WHERE dieu_kien]  
[ORDER BY cot1, cot2, .. cotN] [ASC | DESC];
```

Ví dụ: Lấy tất cả các môn học và sắp xếp theo học phí giảm dần

```
select * from courses order by money desc;
```



Một số lệnh truy vấn nâng cao

Mệnh đề group by:

- Được sử dụng kết hợp với lệnh SELECT để sắp xếp dữ liệu đồng nhất vào trong các nhóm.
- Mệnh đề GROUP BY theo sau mệnh đề WHERE trong một lệnh SELECT và ở trước mệnh đề ORDER BY

Một số lệnh truy vấn nâng cao

Cú pháp:

```
SELECT cot1, cot2  
FROM ten_bang  
WHERE [ dieu_kien ]  
GROUP BY cot1, cot2  
ORDER BY cot1, cot2;
```



Bài tập thực hành SQL

Bài tập 1:

1. Tạo một cơ sở dữ liệu có tên là **Thuchanh**
2. Tạo một bảng có tên là **DOCGIA**, có các thuộc tính như sau:

```
CREATE TABLE DOCGIA(  
    MaDG INT NOT NULL auto_increment,  
    TenDG Text(30) NOT NULL,  
    DiaChi Text(50) NOT NULL,  
    TuoI INT , primary key(MaDG));
```

3. Thêm một thuộc tính mới có tên là Ghichu cho bảng DOCGIA.

```
ALTER TABLE DOCGIA  
ADD COLUMN GhiChu Text(50));
```



Bài tập thực hành SQL

Bài tập 1:

4. Thay đổi kiểu dữ liệu của thuộc tính **Ghichu** thành kiểu dữ liệu **Text(50)**.

ALTER TABLE DOCGIA

MODIFY COLUMN GhiChu LongText;

5. Xóa thuộc tính **Ghichu** trong bảng **DOCGIA**.

ALTER TABLE DOCGIA

DROP COLUMN GhiChu;

6. Thực hiện các lệnh sau đây, mỗi lệnh này sẽ cho kết quả như thế nào?
Nêu sự khác nhau giữa chúng.

Câu 1:

INSERT INTO DOCGIA

VALUES('01','Nguyễn Công Thành','Lớp 41NC',22);



Bài tập thực hành SQL

Bài tập 1:

Câu 2:

```
INSERT INTO DOCGIA(MaDG, TenDG, DiaChi)  
VALUES('DHTL06','Nguyễn Phương Lan','Lớp 41NC');
```

7. Xóa một bản ghi có **MaDG**= '01' trong bảng **DOCGIA**.

```
DELETE FROM DOCGIA WHERE MaDG='01';
```

8. Xóa những độc giả có địa chỉ là: 41NC trong bảng **DOCGIA**.

```
DELETE FROM DOCGIA WHERE Diachi like '41NC';
```

9. Sửa địa chỉ của độc giả có **MaDG** là 01 thành địa chỉ mới là CVK3I.

```
UPDATE DOCGIA  
SET Diachi='CVK3I'  
WHERE MaDG='01';
```

Q&A





THANK YOU