



# CT428. LẬP TRÌNH WEB

## CHƯƠNG 5. PHP & MYSQL (MYSQL)

Giảng viên: Trần Công Ân ([tcan@cit.ctu.edu.vn](mailto:tcan@cit.ctu.edu.vn))

Bộ môn Mạng máy tính & Truyền thông  
Khoa Công Nghệ Thông Tin & Truyền Thông  
Đại học Cần Thơ

2013 – 2014

# MySQL

# NỘI DUNG

GIỚI THIỆU MySQL

SỬ DỤNG GIAO DIỆN DÒNG LỆNH

CÁC LỆNH DDL TRÊN CƠ SỞ DỮ LIỆU (DATABASE)

CÁC LỆNH DDL TRÊN TABLE

CÁC LỆNH THAO TÁC TRÊN DỮ LIỆU (DML)

MỘT SỐ CÔNG CỤ QUẢN LÝ MySQL MIỄN PHÍ

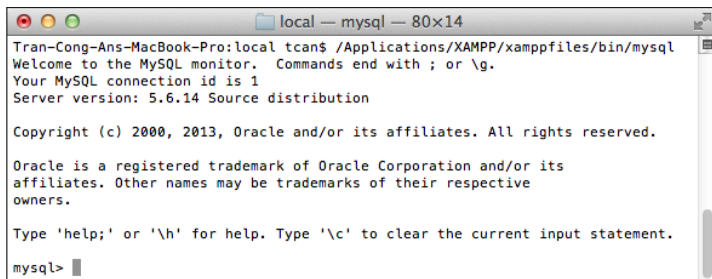
CÁC NỘI DUNG KHÁC LIÊN QUAN ĐẾN MySQL

# MySQL

- ▶ Là một Hệ quản trị CSDL mã nguồn mở và phổ biến nhất hiện nay.
- ▶ Có khả năng quản lý các CSDL lớn và hiệu năng cao.
- ▶ Tại sao sử dụng MySQL?
  - ▶ Đa nền.
  - ▶ Miễn phí.
  - ▶ Gọn nhẹ, có thể cài đặt trên các máy tính “bình thường”.
  - ▶ Có thể dễ dàng sử dụng giao diện Shell để quản trị.
  - ▶ Hỗ trợ nhiều cầu nối để sử dụng trong các ngôn ngữ khác.

# NỐI KẾT VỚI MySQL

- ▶ Nối kết đến MySQL: shell, công cụ với giao diện GUI, API.
- ▶ Shell: từ dòng lệnh gõ vào mysql (Linux: /usr/local/mysql/bin; Windows: c:\Program Files\MySQL\MySQL Server 5.0\bin).



```
local — mysql — 80x14
Tran-Cong-Ans-MacBook-Pro:local tcan$ /Applications/XAMPP/xamppfiles/bin/mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.6.14 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

- ▶ Thoát khỏi MySQL: exit hoặc QUIT

## CÚ PHÁP CÂU LỆNH TỪ DÒNG LỆNH

- ▶ Các lệnh MySQL từ cửa sổ dòng lệnh luôn **kết thúc bằng dấu ;**
- ▶ Một lệnh có thể bao gồm **nhiều dòng**.
- ▶ Một lệnh đang được nhập từ phím sẽ không được thực thi nếu nó kết thúc bằng **\c**
- ▶ Các lệnh **không phân biệt chữ hoa, chữ thường**.

```
mysql> SELECT version();  
+-----+  
| version() |  
+-----+  
| 5.6.14    |  
+-----+  
1 row in set (0.00 sec)
```

```
mysql> SELECT  
    -> curdate();  
+-----+  
| curdate() |  
+-----+  
| 2014-02-20 |  
+-----+  
1 row in set (0.00 sec)
```

# CÁC LỆNH TRÊN CƠ SỞ DỮ LIỆU

- ▶ Một MySQL server có thể quản lý **nhiều CSDL**.
- ▶ Các lệnh cơ bản trên CSDL bao gồm:
  - ▶ **SHOW DATABASES**: liệt kê các CSDL có trên server.
  - ▶ **USE** <db\_name>: chọn CSDL sẽ được sử dụng.
  - ▶ **CREATE DATABASE** <db\_name>: tạo CSDL.
  - ▶ **DROP DATABASE** <db\_name>: xóa CSDL.
  - ▶ **ALTER DATABASE** <db\_name> ...: thay đổi thông số của CSDL.
- ▶ **Chú ý**: muốn thực hiện các lệnh tác động đến CSDL, người dùng cần **được gán quyền** create database.

# NGƯỜI DÙNG (USERS) VÀ QUYỀN (PRIVILEGES)

- ▶ Trong MySQL, người dùng **root** có quyền cao nhất.
- ▶ Nối kết đến MySQL với người dùng cụ thể: `mysql -u <username>`
- ▶ Truy vấn tên người dùng đang nối kết: `SELECT user();`
- ▶ Gán quyền cho người dùng:

```
GRANT <privileges> ON <database.object>  
TO '<username>'@'<hostname>' IDENTIFIED BY '<password>'
```

- ▶ Nếu người dùng chỉ định trong lệnh GRANT chưa có trong server, người dùng sẽ tự động được tạo.
- ▶ Người dùng cũng có thể được tạo tường minh bằng lệnh `CREATE USER`



# NGƯỜI DÙNG (USERS) VÀ QUYỀN (PRIVILEGES)

```
Tran-Cong-Ans-MacBook-Pro:~ tcan$ mysql
```

```
...
```

```
mysql> SELECT user();
```

```
+-----+
```

```
| user() |
```

```
+-----+
```

```
| tcan@localhost |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

Hàm user() trả về người dùng hiện hành

```
mysql> CREATE DATABASE ltweb;
```

```
ERROR 1044 (42000): Access denied for user 'tcan'@'localhost' to  
database 'ltweb'
```

# NGƯỜI DÙNG (USERS) VÀ QUYỀN (PRIVILEGES)

```
Tran-Cong-Ans-MacBook-Pro:~ tcan$ mysql -u root
mysql> CREATE DATABASE ltweb;
Query OK, 1 row affected (0.11 sec)
mysql> GRANT ALL ON ltweb.* TO 'tcan'@'localhost';
Query OK, 0 rows affected (1.82 sec)
```

```
Tran-Cong-Ans-MacBook-Pro:~ tcan$ mysql
mysql> SHOW DATABASES ;
+-----+
| Database          |
+-----+
| information_schema |
| ltweb              |
| test              |
+-----+
3 rows in set (0.06 sec)
```

# CÁC LỆNH DDL CƠ BẢN TRÊN TABLE

- ▶ Các lệnh DDL (Data Manipulate Language) cho phép tạo và thay đổi cấu trúc table.
  - ▶ **CREATE TABLE...**: tạo table.
  - ▶ **ALTER TABLE...**: thay đổi cấu trúc table.
  - ▶ **DROP TABLE** <tb\_name>: xóa table.
  - ▶ **DESCRIBE** <tb\_name>: truy vấn cấu trúc table.
- ▶ Chú ý: cần phải **chọn CSDL** trước khi thực hiện các lệnh này.
- ▶ Truy vấn danh sách các table trong CSDL: **SHOW TABLES**;

# KIỂU DỮ LIỆU TRONG MySQL

- ▶ **Chuỗi ký tự:** CHAR(255), VARCHAR(65K), TINYTEXT(255), TEXT(65K), MEDIUMTEXT(16M), LONGTEXT(4G).
- ▶ **Nhị phân:** BINARY(255), VARBINARY(65K), TINYBLOB(255), BLOB(65K), MEDIUMBLOB(16M), LONGBLOB(4G).
- ▶ **Kiểu số:** TINYINT(1), SMALLINT(2), MEDIUMINT(3), INT/INTEGER(4), BIGINT(8), FLOAT(4), DOUBLE/REAL(8).
- ▶ **Ngày tháng:** DATETIME(yyyy-mm-dd hh-mm-ss), DATE(yyyy-mm-dd), TIMESTAMP(yyyy-mm-dd, 1970–2037), TIME, YEAR(1901–2155).

# GIÁ TRỊ MẶC ĐỊNH & THUỘC TÍNH CỦA KIỂU DL

Kiểu dữ liệu	Thuộc tính	Giá trị mặc định	
		NULL	NOT NULL
Số nguyên	AUTO_INCREMENT, UNSIGNED, SERIAL DEFAULT VALUE	NULL	0
Số thực	UNSIGNED, ZERO FILL	NULL	0
Chuỗi	BINARY, CHARACTER SET	NULL	""
Nhị phân		NULL	""
Ngày tháng		NULL	All 0
TIMESTAMP		curdate()	curdate()

# TẠO TABLE – CREATE TABLE

```
CREATE TABLE <tab_name> (  
    <col_name1> <datatype> [options],  
    <col_name2> <datatype> [options],  
    ...  
    [, PRIMARY KEY(col_name[,col_name, ...])]  
)
```

- Các thông số (options) của các trường:
  - NOT NULL: trường không được rỗng.
  - DEFAULT <value>: đặt giá trị mặc nhiên cho trường.
  - AUTO\_INCREMENT: giá trị tự tăng, tính từ m/tin được thêm sau cùng.
  - UNSIGNED: số không dấu (số dương).

# TẠO TABLE – CREATE TABLE

```
mysql> USE ltweb;  
Database changed
```

```
mysql> CREATE TABLE student (  
    -> id CHAR(10),  
    -> name VARCHAR(100),  
    -> birthday DATE,  
    -> province INT UNSIGNED,  
    -> PRIMARY KEY(id));  
Query OK, 0 rows affected (0.52 sec)
```

```
mysql> SHOW TABLES;  
+-----+  
| Tables_in_ltweb |  
+-----+  
| student          |  
+-----+  
1 row in set (0.04 sec)
```

- Tạo table có cấu trúc giống một table khác:

```
CREATE TABLE <tab_name> LIKE <old_tab_name>
```

# XEM CẤU TRÚC TABLE

```
mysql> DESCRIBE student;
```

Field	Type	Null	Key	Default	Extra
id	char(10)	NO	PRI		
name	varchar(100)	YES		NULL	
birthday	date	YES		NULL	
provine	int(10) unsigned	YES		NULL	

```
4 rows in set (0.85 sec)
```



# SỬA CẤU TRÚC TABLE

- ▶ Đổi tên table:

```
ALTER TABLE <tb_name> RENAME <new_tb_name>
```

- ▶ Đổi kiểu dữ liệu của trường:

```
ALTER TABLE <tb_name> MODIFY <col_name> <column definition>
```

- ▶ Đổi tên trường:

```
ALTER TABLE <tb_name> CHANGE <col_name> <new_name> <column def.>
```

- ▶ Thêm trường:

```
ALTER TABLE <tb_name> ADD <col_name> <column definition>
```

- ▶ Xóa trường:

```
ALTER TABLE <tb_name> DROP <col_name>
```

# SỬA CẤU TRÚC TABLE

```
mysql> ALTER TABLE student ADD enroll TIMESTAMP DEFAULT now();
Query OK, 0 rows affected (0.38 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> describe student;
```

Field	Type	Null	Key	Default	Extra
id	char(10)	NO	PRI		
name	varchar(100)	YES		NULL	
birthday	date	YES		NULL	
province	int(10) unsigned	YES		NULL	
enroll	timestamp	NO		CURRENT_TIMESTAMP	

```
5 rows in set (0.00 sec)
```

# CÁC LỆNH THAO TÁC DỮ LIỆU (DML)

- ▶ Bao gồm các lệnh thao tác trên dữ liệu của các tables.
  - ▶ **INSERT INTO...**: thêm mẫu tin vào table.
  - ▶ **DELETE FROM...**: xóa mẫu tin khỏi table.
  - ▶ **UPDATE...**: cập nhật các mẫu tin trong table.
  - ▶ **SELECT...**: truy vấn dữ liệu của table.
- ▶ Chú ý: cần phải **chọn CSDL** trước khi thực hiện các lệnh này.

# THÊM DỮ LIỆU VÀO TABLE – INSERT INTO

- ▶ Thêm **một mẫu tin** vào table:

```
INSERT INTO <tb_name> [<column list>] VALUES (<list of values>)
```

- ▶ Ví dụ:

```
mysql> INSERT INTO student  
-> VALUES('0123456', 'Tom', '1996-01-15', 1, '2014-2-20');  
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO student (id, name, birthday, province)  
-> VALUES('0123457', 'Jerry', '2000-01-15', 1);  
Query OK, 1 row affected (0.00 sec)
```

# THÊM DỮ LIỆU VÀO TABLE – INSERT INTO

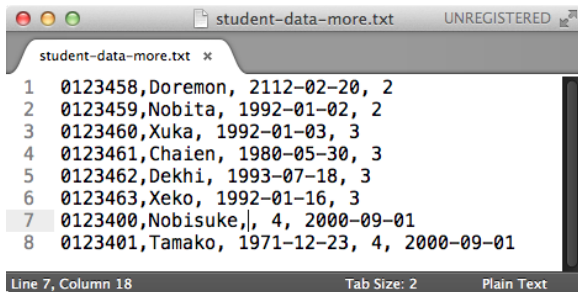
- ▶ Thêm dữ liệu từ một tập tin bên ngoài (text, csv) vào table:

```
LOAD DATA LOCAL INFILE <path> INTO TABLE <tab_name> [options]
```

- ▶ Các options cơ bản:

- ▶ **FIELDS TERMINATED BY '<str>'**: chuỗi ký tự ngăn cách các trường.
- ▶ **FIELDS ENCLOSED BY '<char>'**: ký tự bao quanh giá trị của trường.
- ▶ **LINE TERMINATED BY '<char>'**: ký tự ngăn cách các mẫu tin.
- ▶ **IGNORE <n> LINES**: bỏ qua n hàng đầu tiên.

# THÊM DỮ LIỆU VÀO TABLE – INSERT INTO



```
student-data-more.txt x
1 0123458,Doremon, 2112-02-20, 2
2 0123459,Nobita, 1992-01-02, 2
3 0123460,Xuka, 1992-01-03, 3
4 0123461,Chaien, 1980-05-30, 3
5 0123462,Dekhi, 1993-07-18, 3
6 0123463,Xeko, 1992-01-16, 3
7 0123400,Nobisuke,|, 4, 2000-09-01
8 0123401,Tamako, 1971-12-23, 4, 2000-09-01
```

Line 7, Column 18      Tab Size: 2      Plain Text

```
mysql> LOAD DATA LOCAL INFILE '/Users/tcan/student-data-more.txt'
-> INTO TABLE student FIELDS TERMINATED BY ',';
```

Query OK, 8 rows affected, 6 warnings (0.09 sec)

Records: 8 Deleted: 0 Skipped: 0 Warnings: 6

# CÚ PHÁP CÂU LỆNH TRUY VẤN – SELECT

- ▶ Cú pháp của lệnh truy vấn SELECT:

```
SELECT <list of fields/expressions> | *  
FROM <list of tables>  
[WHERE <conditions>]  
[GROUP BY <list of fields>  
[HAVING <condition>]]
```

- ▶ Diễn giải các mệnh đề cơ bản:

**lựa chọn** (SELECT) một số biểu thức/trường của các mẫu tin

**từ** (FROM) một/các table <list of tables>

**thỏa** (WHERE) điều kiện <conditions>

# DỮ LIỆU MINH HỌA

- Table **student** đã tạo.
- Table **classics** có cấu trúc và dữ liệu như sau:

Field	Type	Null	Key	Default	Extra
author	varchar(128)	YES	MUL	NULL	
title	varchar(128)	YES		NULL	
type	varchar(16)	YES		NULL	
year	char(4)	YES		NULL	
price	float	YES		0	

author	title	type	year	price
Mark Twain	The Adventures of Tom Sawyer	Fiction	1876	100
Jane Austen	Pride and Prejudice	Fiction	1811	99.9
Charles Darwin	The Origin of Species	Non-Fiction	1856	60.5
Charles Dickens	The Old Curiosity Shop	Fiction	1841	54.9
William Shakespeare	Romeo and Juliet	Play	1594	79.9



# LỰA CHỌN TẤT CẢ CÁC MẪU TIN

- Lựa chọn tất cả các mẫu tin của table **student**:

```
SELECT * FROM student;
```

id	name	birthday	province	enroll
0123400	Nobisuke	NULL	4	2000-09-01 00:00:00
0123401	Tamako	1971-12-23	4	2000-09-01 00:00:00
0123456	Tom	1996-01-15	1	2014-02-20 00:00:00
0123457	Jerry	2000-01-15	1	2014-02-21 15:19:36
0123458	Doremon	2112-02-20	2	2014-02-21 23:46:50
0123459	Nobita	1992-01-02	2	2014-02-21 23:46:50
0123460	Xuka	1992-01-03	3	2014-02-21 23:46:50
0123461	Chaien	1980-05-30	3	2014-02-21 23:46:50
0123462	Dekhi	1993-07-18	3	2014-02-21 23:46:50
0123463	Xeko	1992-01-16	3	2014-02-21 23:46:50

# LỰA CHỌN MỘT SỐ TRƯỜNG/BIỂU THỨC

```
SELECT id, name, birthday FROM student
```

```
SELECT id, name, year(curdate())-year(birthday) AS age FROM student
```

id	name	birthday
0123400	Nobisuke	NULL
0123401	Tamako	1971-12-23
0123456	Tom	1996-01-15
0123457	Jerry	2000-01-15
0123458	Doremon	2112-02-20
0123459	Nobita	1992-01-02
0123460	Xuka	1992-01-03
0123461	Chaien	1980-05-30
0123462	Dekhi	1993-07-18
0123463	Xeko	1992-01-16

id	name	age
0123400	Nobisuke	NULL
0123401	Tamako	43
0123456	Tom	18
0123457	Jerry	14
0123458	Doremon	-98
0123459	Nobita	22
0123460	Xuka	22
0123461	Chaien	34
0123462	Dekhi	21
0123463	Xeko	22

# MỘT SỐ HÀM THÔNG DỤNG CỦA MYSQL

- ▶ Hàm xử lý chuỗi: `CONCAT(str1, str2, ...)`,  
`CONCAT_WS(sep, str1, str2, ...)`, `LEFT/RIGHT(str, len)`,  
`MID(str, pos, len)`, `LENGTH(str)`, `LPAD/RPAD(str, len, padstr)`,  
`LOCATE(substr, str, pos)`, `LOWER/UPPER(str)`, `TRIM/LTRIM/RTRIM(str)`
- ▶ Hàm xử lý kiểu ngày tháng: `CURDATE()`, `DATE(expr)`,  
`DATE_ADD(date, INTERVAL expr unit)`, `DAY/MONTH/YEAR(date)`,  
`DATENAME(date)`, `DAYOFWEEK(date)`
- ▶ Hàm xử lý giờ: `CURTIME()`, `HOURL/MINUTE/SECOND(time)`,  
`TIMEDIFF(expr1, expr2)`
- ▶ Hàm toán học: `ABS(expr)`, `FLOOR(expr)`, `MOD(expr)`, `ROUND(expr)`

# TRUY VẤN CÓ ĐIỀU KIỆN – WHERE

```
SELECT id, name, birthday
FROM student
WHERE province=4
```

```
SELECT id, name, birthday
FROM student
WHERE MONTH(birthday)=MONTH(
CURDATE())
```

```
SELECT id, name, birthday
FROM student
WHERE province=3
AND YEAR(birthday)>=1990
```

id	name	birthday
0123400	Nobisuke	NULL
0123401	Tamako	1971-12-23

id	name	birthday
0123458	Doremon	2112-02-20

id	name	birthday
0123460	Xuka	1992-01-03
0123462	Dekhi	1993-07-18
0123463	Xeko	1992-01-16

# TRUY VẤN CÓ ĐK – MIỀN GIÁ TRỊ, DANH SÁCH

```
SELECT id, name, birthday
  FROM student
 WHERE birthday BETWEEN
      '1990/1/1' AND '2000/1/1'
```

```
SELECT id, name, birthday
  FROM student
 WHERE LOWER(name) IN
      ('tom', 'jerry')
```

id	name	birthday
0123456	Tom	1996-01-15
0123459	Nobita	1992-01-02
0123460	Xuka	1992-01-03
0123462	Dekhi	1993-07-18
0123463	Xeko	1992-01-16

id	name	birthday
0123456	Tom	1996-01-15
0123457	Jerry	2000-01-15

# TRUY VẤN CÓ ĐK – NULL, PATTERN, FULLTEXT

```
SELECT id, name, birthday
FROM student
WHERE birthday IS NULL
```

id	name	birthday
0123400	Nobisuke	NULL

```
SELECT id, name, province
FROM student
WHERE name LIKE '%nobi%'
```

id	name	province
0123400	Nobisuke	4
0123459	Nobita	2

```
ALTER TABLE classics
ADD FULLTEXT(author, title)
```

```
SELECT author, title
FROM classics
WHERE MATCH(author, title)
AGAINST('charles romeo')
```

author	title
William Shakespeare	Romeo and Juliet
Charles Darwin	The Origin of Species
Charles Dickens	The Old Curiosity Shop

# SẮP XẾP, TRÁNH TRÙNG LẬP DỮ LIỆU

- ▶ Sắp xếp dữ liệu: dùng mệnh đề **ORDER BY** <fields> | <expr>
- ▶ Loại dữ liệu bị trùng lặp: dùng từ khóa **DISTINCT** trong mệnh đề **SELECT**.

```
SELECT author, title
FROM classics
ORDER BY author
```

author	title
Charles Darwin	The Origin of Species
Charles Dickens	The Old Curiosity Shop
Jane Austen	Pride and Prejudice
Mark Twain	The Adventures of Tom Sawyer
William Shakespeare	Romeo and Juliet

```
SELECT DISTINCT type
FROM classics
```

type
Fiction
Non-Fiction
Play

# TỔNG HỢP DỮ LIỆU

- ▶ Dùng các **hàm tổng hợp** để tạo ra các dữ liệu có tính tổng hợp từ các trường của table (e.g. sách có giá cao nhất, ...).
- ▶ Cú pháp: `aggregate_function([ALL|DISTINCT] <expr>)`
- ▶ Các hàm tổng hợp tính toán trên một tập các giá trị và **trả về một giá trị đơn**.
- ▶ Các hàm thông dụng: **AVG**, **COUNT**, **MAX**, **MIN**, **SUM**



# TỔNG HỢP DỮ LIỆU – Ví Dụ

```
SELECT COUNT(*) NoOfStudent
  FROM student
```

NoOfStudent
10

```
SELECT COUNT(*)
  FROM student
 WHERE birthday IS NULL
```

COUNT(*)
1

```
SELECT MAX(price), MIN(price), AVG(price)
  FROM classics
```

MAX(price)	MIN(price)	AVG(price)
100.00	54.90	79.040001

```
SELECT COUNT(type), COUNT(DISTINCT type)
  FROM classics;
```

COUNT(type)	COUNT(DISTINCT type)
5	3

# NHÓM DỮ LIỆU – GROUP BY & HAVING

- ▶ Có thể nhóm dữ liệu dựa vào giá trị 1 trường hay một biểu thức.
- ▶ Cú pháp: `GROUP BY <fields> | <expr> [HAVING <conditions>]`
  - ▶ Mệnh đề **HAVING** dùng để lọc dữ liệu **sau khi nhóm**.
- ▶ Thường kết hợp với các hàm tổng hợp dữ liệu để tổng hợp dữ liệu cho từng nhóm.
- ▶ **Chú ý:** Nếu câu truy vấn có mệnh đề **WHERE**, việc nhóm gộp dữ liệu chỉ được thực hiện trên tập dữ liệu thỏa điều kiện mệnh đề **WHERE**.

# NHÓM DỮ LIỆU – GROUP BY & HAVING

```
SELECT type, COUNT(title), max(price)
  FROM classics
 GROUP BY type
```

type	COUNT(title)	max(price)
Fiction	3	100.00
Non-Fiction	1	60.50
Play	1	79.90

```
SELECT type, COUNT(title)
  FROM classics
 GROUP BY type
 HAVING count(title)>2
```

type	COUNT(title)
Fiction	3

```
SELECT type, COUNT(*)
  FROM classics
 WHERE year>1800
 GROUP BY type
```

type	COUNT(*)
Fiction	3
Non-Fiction	1

# NỐI KẾT DỮ LIỆU – JOIN | WHERE

- ▶ Join dùng để truy vấn dữ liệu từ nhiều table, dựa vào mối quan hệ luận lý giữa chúng (giá trị của các trường).
- ▶ Các cách kết nối:
  - ▶ **NATURAL JOIN** <table>: tự động kết nối table trong mệnh đề **FROM** và **NATURAL JOIN** dựa trên các trường cùng tên.
  - ▶ **JOIN** <table> **ON** <fields>: kết nối table ở mệnh đề **FROM** và **JOIN** dựa trên các trường chỉ định trong mệnh đề **ON**.
  - ▶ **FROM** <tables> **WHERE** <join\_conditions>: kết nối các table trong mệnh đề **FROM** sử dụng điều kiện trong mệnh đề **WHERE**.
- ▶ Các loại nối kết: **INNER JOIN**, **(LEFT|RIGHT)OUTER JOIN**

# DỮ LIỆU MINH HỌA

- Tạo thêm table **province** có cấu trúc và dữ liệu như sau:

Field	Type	Null	Key	Default	Extra
province	int(10) unsigned	NO	PRI	0	
name	varchar(100)	YES		NULL	

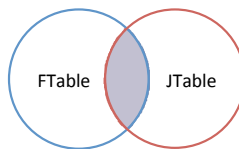
  

province	name
1	Tokyo
2	Kobe
3	Kagawa
4	Hirosima
5	Osaka

# KẾT NỐI DỮ LIỆU – INNER JOIN

```
SELECT id, student.name,
       province.name
FROM student
JOIN province
ON student.province =
   province.province
```

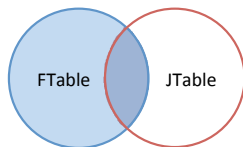
```
SELECT id, s.name, p.name
FROM student AS s, province AS p
WHERE s.province = p.province
```



id	name	name
0123456	Tom	Tokyo
0123457	Jerry	Tokyo
0123459	Nobita	Kobe
0123460	Xuka	Kagawa
0123461	Chaien	Kagawa
0123462	Dekhi	Kagawa
0123463	Xeko	Kagawa
0123400	Nobisuke	Hirosima
0123401	Tamako	Hirosima

# KẾT NỐI DỮ LIỆU – LEFT OUTER JOIN

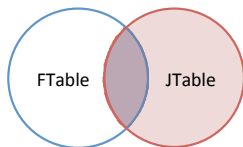
```
SELECT id, s.name AS SName, p.name
      AS PName
FROM student AS s
LEFT OUTER JOIN province as p
      ON s.province = p.province
```



id	SName	PName
0123456	Tom	Tokyo
0123457	Jerry	Tokyo
0123459	Nobita	Kobe
0123460	Xuka	Kagawa
0123461	Chaien	Kagawa
0123462	Dekhi	Kagawa
0123463	Xeko	Kagawa
0123400	Nobisuke	Hirosima
0123401	Tamako	Hirosima
0123458	Doremon	NULL

# KẾT NỐI DỮ LIỆU – RIGHT OUTER JOIN

```
SELECT id, s.name AS SName, p.name
      AS PName
FROM student AS s
RIGHT OUTER JOIN province as p
      ON s.province = p.province
```



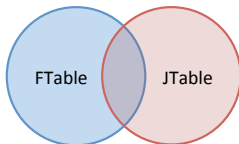
id	SName	PName
0123400	Nobisuke	Hirosima
0123401	Tamako	Hirosima
0123456	Tom	Tokyo
0123457	Jerry	Tokyo
0123459	Nobita	Kobe
0123460	Xuka	Kagawa
0123461	Chaien	Kagawa
0123462	Dekhi	Kagawa
0123463	Xeko	Kagawa
NULL	NULL	Osaka



# KẾT NỐI DỮ LIỆU – FULL OUTER JOIN (UNION)

```

SELECT id, s.name AS SName, p.name
      AS PName
FROM student AS s
LEFT OUTER JOIN province as p
      ON s.province = p.province
UNION
SELECT id, s.name AS SName, p.name
      AS PName
FROM student AS s
RIGHT OUTER JOIN province as p
      ON s.province = p.province
  
```



id	SName	PName
0123456	Tom	Tokyo
0123457	Jerry	Tokyo
0123459	Nobita	Kobe
0123460	Xuka	Kagawa
0123461	Chaien	Kagawa
0123462	Dekhi	Kagawa
0123463	Xeko	Kagawa
0123400	Nobisuke	Hirosima
0123401	Tamako	Hirosima
0123458	Doremon	NULL
NULL	NULL	Osaka

# CÁC CÂU TRUY VẤN LỒNG NHAU

- Kết quả một câu truy vấn có thể được sử dụng trong mệnh đề điều kiện của một câu truy vấn khác.

```
SELECT id, name, province
FROM student
WHERE province NOT IN
      (SELECT province FROM province)
```

id	name	province
0123458	Doremon	0

```
SELECT id, name, birthday
FROM student
WHERE birthday =
      (SELECT MAX(birthday) FROM student)
```

id	name	birthday
0123458	Doremon	2112-02-20

# GIỚI HẠN SỐ MẪU TIN TRẢ VỀ – LIMIT

- ▶ Cú pháp: `LIMIT [offset,] <number of rows>`  
(Chọn tối đa *number of rows* mẫu tin, tính từ *offset*)
- ▶ Thường được dùng trong phân trang kết quả truy vấn.

```
SELECT id, name, birthday
FROM student
LIMIT 2
```

id	name	birthday
0123400	Nobisuke	NULL
0123401	Tamako	1971-12-23

```
SELECT id, name, birthday
FROM student
LIMIT 2, 2
```

id	name	birthday
0123456	Tom	1996-01-15
0123457	Jerry	2000-01-15

# CẬP NHẬT DỮ LIỆU CỦA TABLE – UPDATE

## ► Cú pháp:

```
UPDATE <tab_name>
  SET <<field_name> = <value> [, <field_name> = <value> ...]>
  [WHERE <conditions>]
```

## ► Chú ý: Nếu không có điều kiện, tất cả các mẫu tin sẽ được c/nhập.

```
UPDATE classics
  SET price = price*0.9
```

```
UPDATE classics
  SET price = price*0.9
  WHERE type = 'Fiction'
```

title	price	price
The Adventures of Tom Sawyer	90.00	81.00
Pride and Prejudice	89.91	80.92
The Origin of Species	54.45	54.45
The Old Curiosity Shop	49.41	44.47
Romeo and Juliet	71.91	71.91

# XÓA DỮ LIỆU CỦA TABLE – DELETE

## ► Cú pháp:

```
DELETE FROM <tab_name> [WHERE <conditions>]
```

## ► Ví dụ:

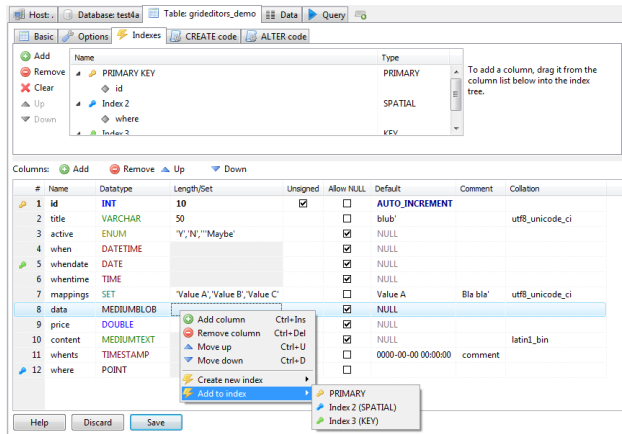
```
DELETE FROM classics WHERE year < 1800
```

author	title	year
Mark Twain	The Adventures of Tom Sawyer	1876
Jane Austen	Pride and Prejudice	1811
Charles Darwin	The Origin of Species	1856
Charles Dickens	The Old Curiosity Shop	1841

## ► **Chú ý:** Nếu không có điều kiện, tất cả các mẫu tin sẽ bị xóa.

# MỘT SỐ CÔNG CỤ QUẢN LÝ MYSQL

MySQL Workbench,  
HeidiSQL, Toad,  
Eclipse (+plugin),  
myPHPQdmin, ...

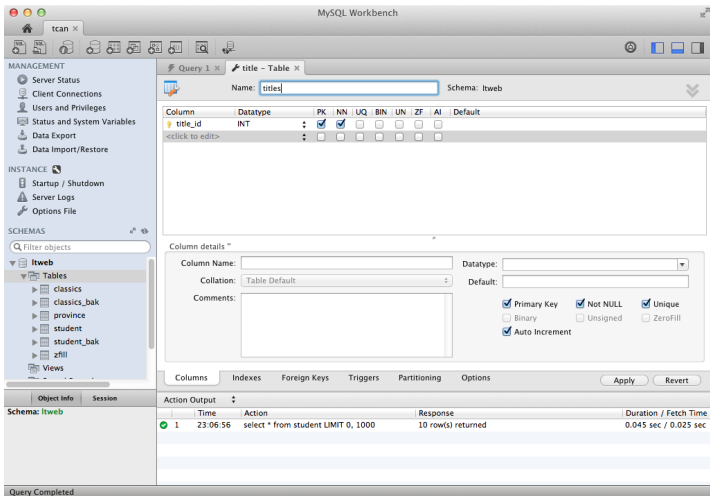


# MỘT SỐ CÔNG CỤ QUẢN LÝ MYSQL

The screenshot displays the MySQL Workbench application window. On the left, the 'MANAGEMENT' sidebar includes options like Server Status, Client Connections, and Users and Privileges. Below it, the 'INSTANCE' section shows Startup / Shutdown, Server Logs, and Options File. The 'SCHEMAS' section is active, showing a tree view of databases, with 'Itweb' selected and its tables listed. The main area shows 'Query 1' with the SQL statement `select * from student`. Below the query editor, the 'Result Set Filter' is set to 'Q', and a table of results is displayed. The table has columns: id, name, birthday, province, and enroll. The results show 10 rows of student data. At the bottom, the 'Action Output' tab shows the execution details: 'select \* from student LIMIT 0, 1000' returned 10 row(s) in 0.045 sec / 0.025 sec. The status bar at the very bottom indicates 'Query Completed'.

id	name	birthday	province	enroll
0123400	Nobisuke	1971-12-23	4	2000-09-01...
0123401	Tamako	1971-12-23	4	2000-09-01...
0123456	Tom	1996-01-15	1	2014-02-20...
0123457	Jerry	2000-01-15	1	2014-02-21...
0123458	Doremon	2112-02-20	0	2014-02-21...
0123459	Nobita	1992-01-02	2	2014-02-21...
0123460	Xuka	1992-01-03	3	2014-02-21...
0123461	Chalien	1980-05-30	3	2014-02-21...
0123462	Dekhi	1993-07-18	3	2014-02-21...
0123463	Xeko	1992-01-16	3	2014-02-21...

# MỘT SỐ CÔNG CỤ QUẢN LÝ MYSQL





# MỘT SỐ NỘI DUNG KHÁC LIÊN QUAN ĐẾN MYSQL

- ▶ Sử dụng ràng buộc (constraints): foreign key, cascade update/delete, ...
- ▶ Chỉ định bảng mã (character set), qui tắc sắp xếp (collation) cho dữ liệu của các table.
- ▶ Tạo các thủ tục/hàm (stored procedure/function) và view.
- ▶ Sử dụng bẫy sự kiện (trigger).
- ▶ Quản lý quyền (prililege), bảo mật (security).
- ▶ Sao lưu (backup) và phục hồi (recover) dữ liệu.

