

THỰC HÀNH QUẢN TRỊ DỮ LIỆU (CT467)

Nội dung thực hành

Nội dung thực hành của học phần này bao gồm **6 buổi**:

1. Giới thiệu MySQL
2. MySQL nâng cao (transaction, trigger,...)
3. Giới thiệu về đồ án và cơ sở lý thuyết để thực hiện đồ án (JDBC)
4. Tìm hiểu và phân tích yêu cầu của đồ án, thiết kế CSDL, tạo CSDL
5. Thiết kế giao diện, kết nối và xử lý dữ liệu
6. Báo cáo kết quả đồ án

BUỔI 1 - GIỚI THIỆU MYSQL

Nội dung:

- Giới thiệu Hệ quản trị cơ sở dữ liệu MySQL
- Căn bản về MySQL
- Tạo các ràng buộc toàn vẹn trong MySQL

1.1 Giới thiệu MySQL

MySQL là một trong những Hệ quản trị cơ sở dữ liệu (CSDL) quan hệ, mã nguồn mở thông dụng nhất hiện nay. Đây được đánh giá là một hệ quản trị có hiệu năng tốt, đáng tin cậy, tính dân nở (scalability) cao, dễ sử dụng và đa nền (Unix, Linux, BSD, Windows,...). MySQL được sử dụng cho các ứng dụng có kiến trúc client/server và trong các hệ thống nhúng. Ngoài ra, đây cũng chính là hệ quản trị trong LAMP (Linux , Apache, MySQL, PHP/Perl/Python), một trong những nền tảng phát triển web thông dụng nhất hiện nay.

Hiện nay, MySQL bao gồm 5 phiên bản (chi tiết về các phiên bản này có thể được tham khảo tại www.mysql.com/products):

- MySQL Enterprise Edition
- MySQL Standard Edition
- MySQL Classic Edition
- MySQL Cluster CGE
- MySQL Embedded

1.2 Cài đặt MySQL

1.2.1 Trên hệ điều hành Windows

MySQL gồm 2 phần chính và quan trọng nhất là **MySQL Server** và **MySQL Workbench** (công cụ trực quan cho việc học và làm việc trên MySQL)

❖ **MySQL Server:**

- Link download: <https://downloads.mysql.com/archives/installer/>
- Chọn Product Version cho phù hợp với cấu hình máy tính hiện tại
- Chọn download ở dòng 2 (mysql-installer-community-[version](#).msi)
- Tài liệu: <https://dev.mysql.com/doc/refman/8.0/en/mysql-installer.html>

❖ **MySQL Workbench:**

- Link download: <https://downloads.mysql.com/archives/workbench/>
- Chọn Product Version phù hợp với cấu hình MySQL Server đã chọn
- Tài liệu: <https://dev.mysql.com/doc/workbench/en/wb-installing-windows.html>

1.2.1 Trên hệ điều hành Linux (Ubuntu và Debian)

❖ **MySQL Server:** Lựa chọn gói cài đặt cho phù hợp

- Tài liệu version 8.0: <https://dev.mysql.com/doc/refman/8.0/en/linux-installation.html>
- Cài đặt với gói Debian: <https://dev.mysql.com/doc/refman/8.0/en/linux-installation-debian.html>

❖ **MySQL Workbench:**

- Tài liệu: <https://dev.mysql.com/doc/workbench/en/wb-linux.html>
- Cài đặt: <https://dev.mysql.com/doc/workbench/en/wb-installing-linux.html>

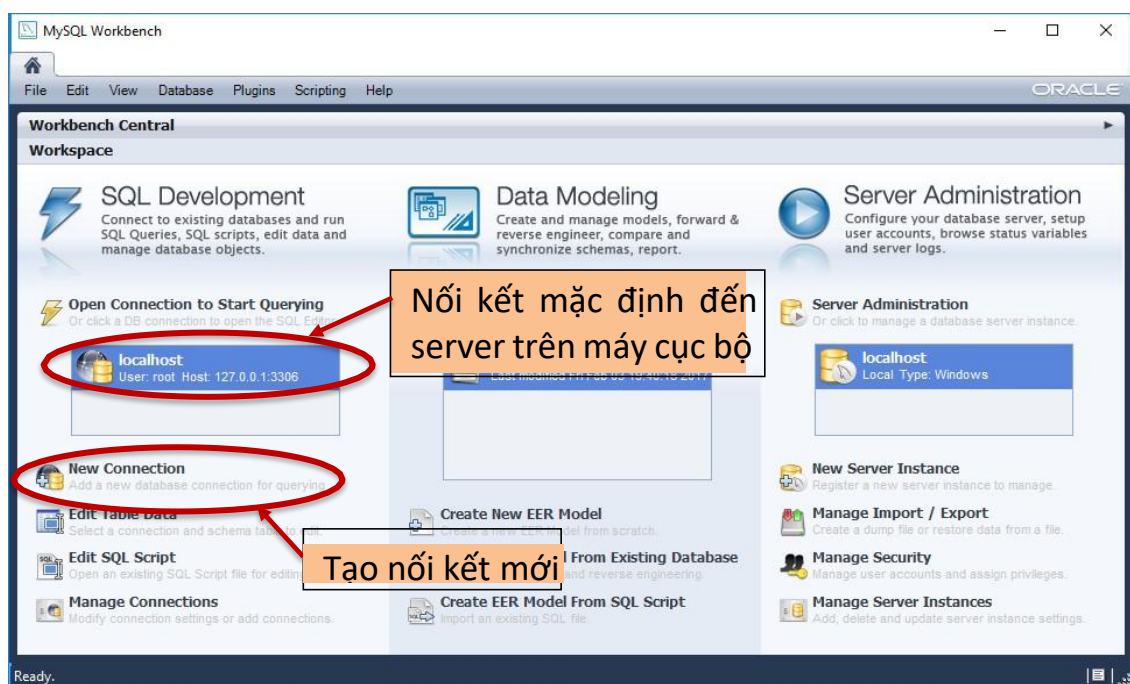
1.3 Nối kết và làm việc với MySQL

Để nối kết và thao tác trên MySQL, có thể dùng công cụ dòng lệnh (shell), các công cụ giao diện đồ họa, hoặc các API (để nối kết và truy xuất từ các chương trình).

- Tài liệu: <https://dev.mysql.com/doc/refman/8.0/en/connection-options.html>

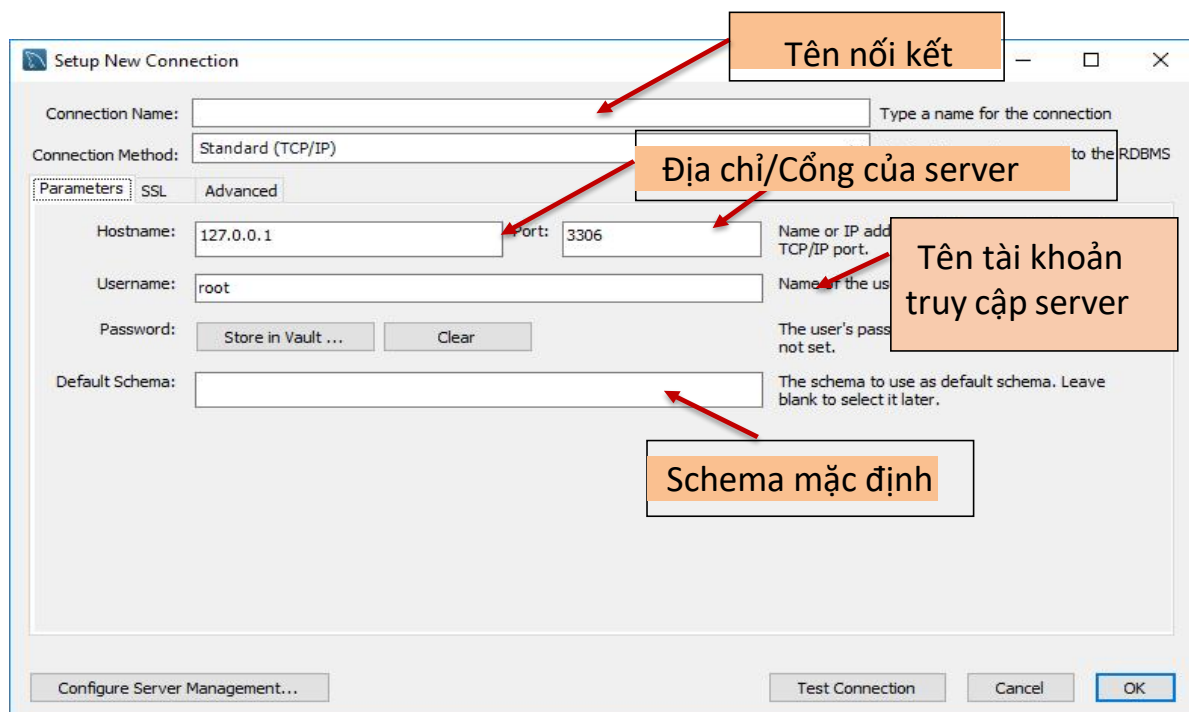
1.3 Giao diện đồ họa

Giao diện chính của chương trình MySQL Workbench khi mới cài đặt xong:

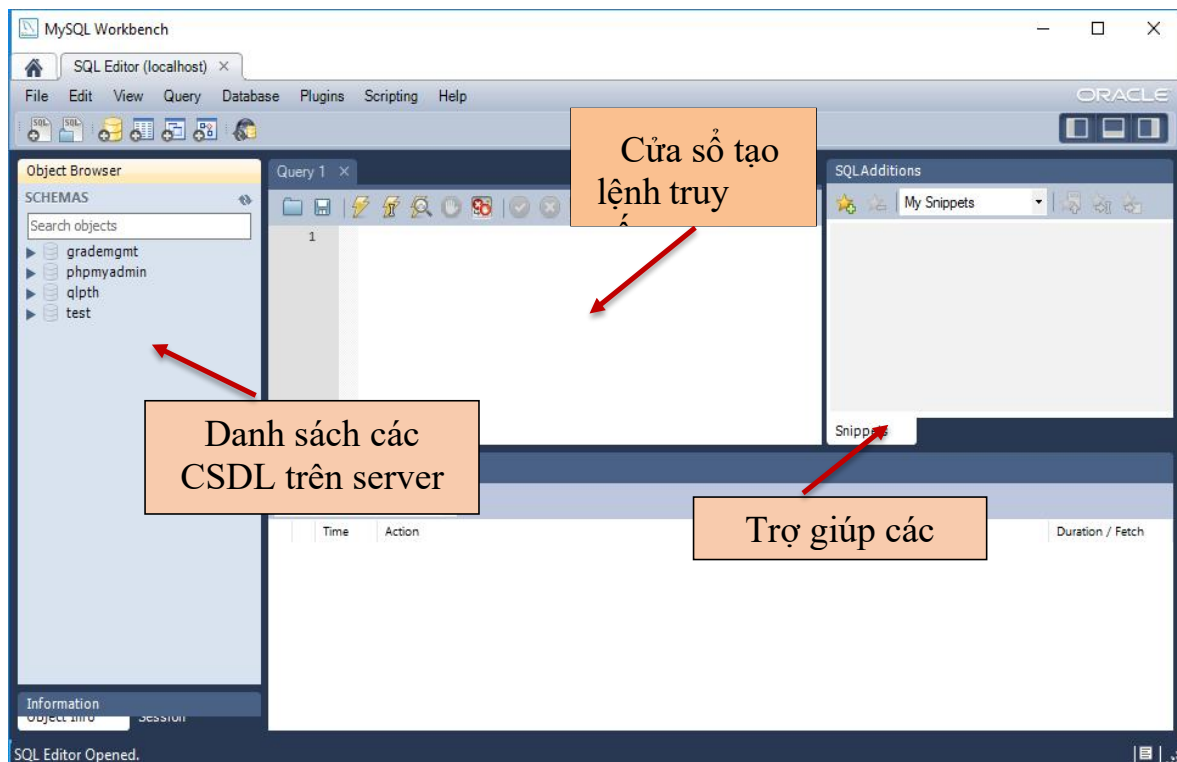


Tài liệu MySQL Workbench: <https://dev.mysql.com/doc/workbench/en/>

Giao diện này của chương trình cho phép tạo nối kết đến server trên máy cục bộ hoặc nối kết đến các MySQL server trên máy khác. Để tạo nối kết đến 1 MySQL Server ta cần khai các thông tin như trong hình sau:



Giao diện của tiện ích này sau khi nối kết vào server:



Công cụ này bao gồm các chức năng cho phép tạo CSDL, các đối tượng trong CSDL như table, procedure, view,...bằng các lệnh DDL.

1.4 Căn bản về MySQL

1.4.1. Cú pháp dòng lệnh

Một số quy tắc chính đối với cú pháp dòng lệnh của MySQL:

- Các lệnh MySQL từ dòng lệnh luôn kết thúc bằng dấu chấm phẩy ;
- Một lệnh có thể bao gồm nhiều dòng
- Một lệnh đang nhập có thể được hủy bằng cách thêm “\c” ở cuối dòng
- Các lệnh không phân biệt chữ hoa, chữ thường

<pre>mysql> SELECT version();</pre> <table border="1"> <tr><td>version()</td></tr> <tr><td>5.6.14</td></tr> </table> <pre>1 row in set (0.00 sec)</pre>	version()	5.6.14	<pre>mysql> SELECT -> curdate();</pre> <table border="1"> <tr><td>curdate()</td></tr> <tr><td>2014-02-20</td></tr> </table> <pre>1 row in set (0.00 sec)</pre>	curdate()	2014-02-20
version()					
5.6.14					
curdate()					
2014-02-20					

1.4.2. Các lệnh trên cơ sở dữ liệu (DDL, Data Definition Language)

Một server có thể quản trị nhiều CSDL. Một số lệnh cơ bản liên quan đến việc quản lý các CSDL:

- **SHOW DATABASES:** liệt kê các CSDL trên server
- **USE <db_name>:** chọn CSDL sẽ được sử dụng (làm việc)

- **CREATE DATABASE <db_name>**: tạo CSDL
- **DROP DATABASE <db_name>**: xóa CSDL
- **ALTER DATABASE <db_name>**: thay đổi thông số của CSDL

1.4.3 Kiểm soát dữ liệu

- **GRANT** <quyền> **ON** <tên bảng> **TO** <user> [WITH GRANT OPTION]:
để trao quyền cho một account nào đó.
 - *Các quyền có thể trao*: All, Select, Update, Delete, Insert, Alter,...
 - *User có thể*: Public, tên một user cụ thể,...
 - *[WITH GRANT OPTION]*: chỉ định trao lại quyền người này cho người khác
- **REVOKE** <quyền> **ON** <tên bảng> **FROM** <user>: thu hồi quyền của một account nào đó.

1.4.4 Các kiểu dữ liệu của MySQL

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

Các thuộc tính và giá trị mặc định của các kiểu dữ liệu được cho trong bảng sau:

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
Nhị phân	BINARY, CHARACTER SET	NULL	Rỗng
Ngày tháng		NULL	Rỗng
Timestamp		curdate()	curdate()

1.4.5. Các lệnh DDL trên table

Một số lệnh DDL cơ bản cho phép tạo và thay đổi cấu trúc table:

1.4.5.1. Tạo table – CREATE TABLE

❖ Cú pháp cơ bản:

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

```
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)
```

❖ Một số thông số (options) áp dụng cho các trường:

- **NOT NULL**: trường không được rỗng.
- **DEFAULT <value>**: chỉ định giá trị mặc nhiên
- **AUTO_INCREMENT**: giá trị tự động tăng dần.
- **UNSIGNED**: số không dấu (số dương)
- **PRIMARY KEY**: khóa chính.

1.4.5.2. Thay đổi cấu trúc table – ALTER TABLE

Một số lệnh cơ bản để thay đổi cấu trúc table:

- **ALTER TABLE <tb_name> RENAME <new_tb_name>**: đổi tên table
- **ALTER TABLE <tb_name> MODIFY <col_name> <column definition>**: đổi kiểu dữ liệu của các trường
- **ALTER TABLE <tb_name> CHANGE <col_name> <new_name> <col def.>**: đổi tên trường
- **ALTER TABLE <tb_name> ADD <col_name> <column definition>**: thêm trường
- **ALTER TABLE <tb_name> DROP <col_name>**: xóa trường


```
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)

```
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)

1.4.5.3. Một số lưu ý

- Cần phải chọn CSDL (lệnh: **USE <db_name>**) trước khi thực hiện các lệnh này.
- Lệnh để lấy danh sách các table trong CSDL: **SHOW TABLES**
- Để truy vấn cấu trúc table, dùng lệnh: **DESCRIBE <tb_name>**
- Để xóa table ra khỏi CSDL, dùng lệnh: **DROP TABLE <tb_name>**

1.4.6. Các lệnh thao tác trên dữ liệu DML

Các lệnh DML cho phép thao tác trên dữ liệu như thêm dữ liệu vào table, xóa dữ liệu ra khỏi table, cập nhật giá trị các mẫu tin trong table,...

- **INSERT INTO...:** thêm mẫu tin vào table

```
INSERT [INTO] tbl_name
```

```
[(col_name,...)]
```

```
{VALUES | VALUE} ({expr | DEFAULT},...),(...),...
```

- **DELETE FROM...:** xóa mẫu tin khỏi table

```
DELETE FROM tbl_name
```

```
[WHERE where_condition]
```

```
[LIMIT row_count]
```

- **UPDATE...:** cập nhật dữ liệu của table

```
UPDATE [LOW_PRIORITY] [IGNORE] table_references
  SET col_name1={expr1|DEFAULT} [,col_name2={expr2|DEFAULT}]...
  [WHERE where_condition]
```

- **LOAD DATA...:** thêm dữ liệu từ một tập tin bên ngoài

```
LOAD DATA INFILE 'file_name'
  INTO TABLE tbl_name
  [CHARACTER SET charset_name]
  [{FIELDS | COLUMNS}
   [TERMINATED BY 'string']
   [ENCLOSED BY 'char']
   [ESCAPED BY 'char']
  ]
  [LINES
   [STARTING BY 'string']
   [TERMINATED BY 'string']
  ]
  [IGNORE number {LINES | ROWS}]
```

- ✓ **FIELDS TERMINATED BY:** chỉ định chuỗi ngăn cách các trường trong tập tin dữ liệu
- ✓ **FIELDS ENCLOSED BY:** chỉ định ký tự bao quanh giá trị của 1 trường
- ✓ **LINE STARTING BY:** chỉ định ký tự (dấu hiệu) bắt đầu của 1 mẫu tin
- ✓ **LINE TERMINATED BY:** chỉ định ký tự ngăn cách các mẫu tin
- ✓ **IGNORE number LINES:** bỏ qua *number* dòng đầu tiên trong file

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)
```

Tương tự các lệnh DDL, ta cần phải chọn CSDL trước khi thực hiện các lệnh DML.

1.4.7. Truy vấn dữ liệu MySQL

Tương tự các DBMS khác, MySQL cũng hỗ trợ lệnh SELECT để truy vấn dữ liệu. Cú pháp căn bản của lệnh SELECT trong MySQL như sau:

SELECT

```
[ALL | DISTINCT | DISTINCTROW ]
select_expr [, select_expr ...]
[FROM table_references
[WHERE where_condition]
[GROUP BY {col_name | expr | position}
[ASC | DESC], ... [WITH ROLLUP]]
[HAVING where_condition]
[ORDER BY {col_name | expr | position}
[ASC | DESC], ...]
[LIMIT {[offset,] row_count | row_count OFFSET offset}]
[INTO OUTFILE 'file_name'
[CHARACTER SET charset_name]
export_options
| INTO DUMPFILE 'file_name'
| INTO var_name [, var_name]]
```

Các thông số cơ bản của lệnh này cũng tương tự như ANSI SQL. Ý nghĩa một số thông số đặc trưng của MySQL như sau:

- **DISTINCTROW**: từ khóa khác của **DISTINCT**
- **WITH ROLLUP**: yêu cầu sinh ra kết quả thống kê được nhóm theo nhiều cấp.
- **INTO OUTFILE**: yêu cầu lưu kết quả truy vấn vào file hoặc biến

Một số hàm thông dụng trong MySQL:

- **Hàm xử lý chuỗi**: CONCAT(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), CONCAT_WS(sep, str1, str2, ...)
- **Hàm xử lý dữ liệu ngày tháng**: CURDATE(), DATE(expr), DATE_ADD(date, INTERVAL expr unit), DAY/MONTH/YEAR(date), DATENAME(date), DAYOFWEEK(date)
- **Hàm xử lý dữ liệu giờ**: CURTIME(), HOUR/MINUTE/SECOND(time), TIMEDIFF(expr1, expr2)
- **Hàm toán học**: ABS(expr), FLOOR(expr), MOD(expr), ROUND(expr)

Ví dụ: chọn ra các sinh viên có tháng sinh là tháng hiện tại

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

id	name	birthday
0123458	Doremon	2112-02-20

1.4.8.Nối kết (join) các bảng

Join dùng để truy vấn dữ liệu từ nhiều table, dựa vào mối quan hệ luận lý giữa chúng (giá trị của các trường). Có 3 lệnh để nối kết các table:

- **NATURAL JOIN <tables>**: tự động nối kết các table trong mệnh đề **FROM** dựa trên các trường cùng tên
- **JOIN <tables> ON <fields>**: nối kết các table ở mệnh đề **FROM** và **JOIN** dựa trên các trường chỉ định trong mệnh đề **ON**
- **FROM <tables> WHERE <join_condition>**: kết nối các table trong mệnh đề **FROM** sử dụng điều kiện trong mệnh đề **WHERE**

Các nối kết được phân làm 4 loại: **INNER JOIN**, **LEFT/RIGHT/FULL OUTER JOIN**.

Để minh họa cho các câu lệnh trong phần này, các table sau sẽ được sử dụng:

◆ Table Student:

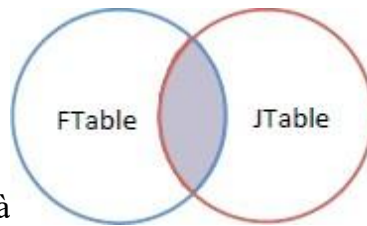
id	name	birthday	province	enroll
0123400	Nobisuke	NULL	4	2000-09-01 00:00:00
0123401	Tamako	1971-12-23	4	2000-09-01 00:00:00
0123456	Tom	1996-01-15	1	2014-02-20 00:00:00
0123457	Jerry	2000-01-15	1	2014-02-21 15:19:36
0123458	Doremon	2112-02-20	6	2014-02-21 23:46:50
0123459	Nobita	1992-01-02	2	2014-02-21 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

◆ Table Province:

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

1.4.8.1.Inner join

Đây là loại nối kết mặc định trong MySQL. Inner join trả về các mẫu tin có giá trị của trường được chỉ định nối kết bằng nhau như được mô tả trong hình sau:



Trong đó, **FTable** là table trong mệnh đề **FROM** và **JTable** là table trong mệnh đề **JOIN**. Kết quả trả về là phần giao nhau của hai table, dựa trên điều kiện được chỉ định trong mệnh đề **JOIN** hoặc **WHERE**.

Ví dụ: Truy vấn mã sinh viên, họ tên sinh viên cùng với tên quê quán của SV.

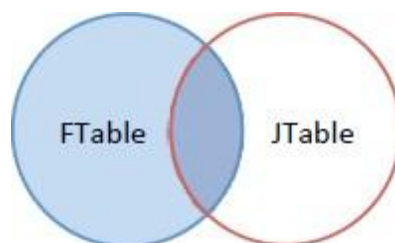
```
SELECT id, student.name,
       province.name
FROM student
JOIN province
ON student.province =
   province.province
```

```
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

1.4.8.2. Left outer join

Left outer join trả về các mẫu tin có giá trị trường được chỉ định nối kết bằng nhau cộng với các mẫu tin của table trong mệnh đề FROM. Hình sau minh họa cách thức làm việc của loại nối kết này:



Phần có tô nền là phần được trả về bởi câu truy vấn. Đối với các mẫu tin thuộc table trong mệnh đề **FROM** nhưng không giao với table trong mệnh đề **JOIN**, thuộc tính các trường thuộc table trong mệnh đề **JOIN** sẽ có giá trị **NULL**.

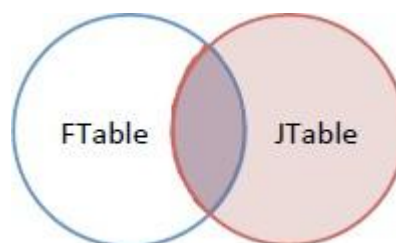
Ví dụ: tương tự như ví dụ trong phần 1.4.8.1, trả về cả các sinh viên có mã tỉnh không hợp lệ (không có trong danh sách các tỉnh trong table Province). Câu truy vấn và kết quả tương ứng như sau:

```
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

1.4.8.3 Right outer join

Ngược với left outer join, right outer join trả về các mẫu tin giao nhau giữa các table và các mẫu tin của table trong mệnh đề **JOIN**.



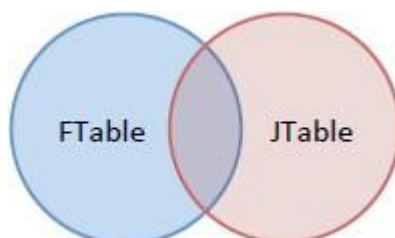
Đối với các mẫu tin của table trong mệnh đề **JOIN** mà không giao với table trong mệnh đề **FROM**, các trường trong mệnh đề **SELECT** sẽ mang giá trị **NULL**. Hoạt động của loại nối kết này được minh họa bằng câu truy vấn và kết quả tương ứng như sau:

```
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

1.4.8.4 Full outer join (union)

Kết quả của full outer join là sự kết hợp của left outer join và right 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
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

1.4.9. Các câu truy vấn lồng nhau

Câu truy vấn lồng nhau có nghĩa là một câu truy vấn được sử dụng trong mệnh đề điều kiện của một câu truy vấn khác. Sau đây là 2 ví dụ về câu truy vấn lồng nhau:

```

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

1.5 Ràng buộc (Constraint) trong MySQL

Ràng buộc là các qui tắc (rule) do người dùng thiết đặt để cho phép hay ngăn cấm các giá trị có thể được lưu trữ trong các trường của một mẫu tin. Mục đích của việc thiết đặt các ràng buộc là để đảm bảo tính toàn vẹn của dữ liệu.

Ràng buộc trong MySQL có thể được phân làm 2 mức:

- Ràng buộc mức cột: dùng để thiết đặt cho các trường.
- Ràng buộc mức table: áp dụng cho toàn bộ table.

Các kiểu ràng buộc được hỗ trợ trong MySQL:

- **NOT NULL:** không cho phép chứa giá trị NULL.
- **UNIQUE:** không cho phép một trường của các mẫu tin có giá trị trùng nhau.
- **PRIMARY KEY:** ràng buộc khóa chính, không cho phép trùng nhau.

- **FOREIGN KEY:** ràng buộc khóa ngoại, bắt buộc giá trị khóa ngoại phải có giá trị khóa chính tương ứng.
- **CHECK:** dùng để chỉ định các giá trị cho phép được lưu trữ trong một trường.
- **DEFAULT:** dùng để chỉ định giá trị mặc nhiên cho một trường khi giá trị của trường không được cung cấp.

Các ràng buộc được chỉ định trong câu lệnh **CREATE TABLE** hoặc **ALTER TABLE**.

Cú pháp tạo table với các thông số liên quan đến ràng buộc bao gồm:

```
CREATE TABLE [IF NOT EXISTS] tbl_name
    (create_definition,...)

create_definition:
    col_name column_definition
    | [CONSTRAINT [symbol]] PRIMARY KEY (index_col_name,...)
    | [CONSTRAINT [symbol]] UNIQUE
        [index_name] [index_type] (index_col_name,...)
    | [CONSTRAINT [symbol]] FOREIGN KEY
        [index_name] (index_col_name,...) reference_definition
    | CHECK (expr)

column_definition:
    data_type [NOT NULL | NULL] [DEFAULT default_value]
    [AUTO_INCREMENT] [UNIQUE | [PRIMARY] KEY]
    [reference_definition]

reference_definition:
    REFERENCES tbl_name (index_col_name,...)
```

Ví dụ: tạo table **Publisher** (nhà xuất bản) với các trường và ràng buộc như sau:

```
CREATE TABLE IF NOT EXISTS Publisher
( pub_id varchar(8) PRIMARY KEY,    --khóa chính
  pub_name varchar(50) NOT NULL UNIQUE, --ko NULL, ko trùng nhau
  pub_city varchar(25) NOT NULL,      --ko được rỗng
  country varchar(25) DEFAULT 'Vietnam', --giá trị mặc nhiên
);
```

- ✓ **pub_id:** mã nhà xuất bản, khóa chính của table
- ✓ **pub_name:** tên nhà xuất bản, không được rỗng và trùng nhau
- ✓ **pub_city:** thành phố đặt NXB, không được rỗng
- ✓ **country:** quốc gia đặt NXB, giá trị mặc nhiên là “Vietnam”

Tương tự, table Book (sách) bao gồm các trường và ràng buộc sau:

```
CREATE TABLE IF NOT EXISTS Book
( book_id varchar(15) PRIMARY KEY,           --thiết đặt khóa chính
  book_name varchar(50) NOT NULL,           --tên sách ko được NULL
  isbn_no varchar(15) NOT NULL UNIQUE,      --ISBN ko được trùng nhau
  pub_id varchar(8),
  date_of_pub date CHECK (dt_of_pub LIKE '--/--/-----'),
  no_page decimal(5,0) CHECK(no_page > 0),
  FOREIGN KEY(pub_id) REFERENCES Publisher(pub_id)
);
```

- ✓ **book_id**: mã sách, là khóa chính
- ✓ **book_name**: tên sách, không được rỗng
- ✓ **isbn_no**: số xuất bản, không được rỗng và không được trùng nhau
- ✓ **pub_id**: mã NXB, là khóa ngoại tham chiếu đến pub_id trong table Publisher
- ✓ **date_of_pub**: ngày xuất bản, giá trị phải đúng định dạng của ngày tháng năm
- ✓ **no_page**: số trang, giá trị phải > 0

1.6 Thực hành

Bài 1: Làm quen với MySQL bằng giao diện đồ họa với **MySQL Workbench**

❖ **Yêu cầu:**

- 1) Tạo nối kết đến MySQL trên máy tính cục bộ bằng tài khoản root
- 2) Sau đó, thực hiện các việc như sau:
 - a) Tạo 2 CSDL đặt tên là **dbms** và **dbms1** trên server
 - b) Liệt kê các CSDL trên server
 - c) Xóa CSDL tên **dbms1**

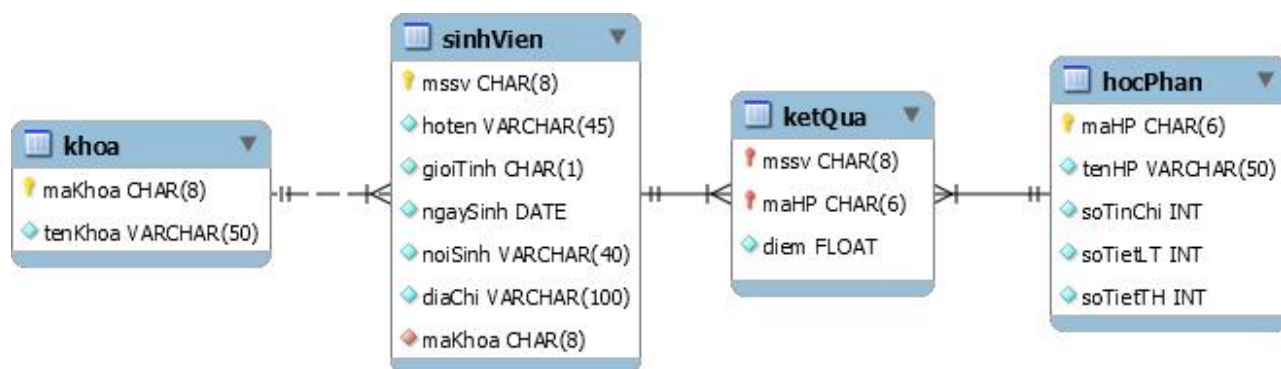
Bài 2: Tạo table, chèn dữ liệu vào table

❖ **Mục tiêu:**

- Ôn tập các lệnh DML và DDL.

❖ **Yêu cầu:**

- 1) Tạo một CSDL tên là **QLDIEM** bao gồm các tables như sau:



2) Dùng các câu lệnh SQL để thêm dữ liệu cho các table trên như sau:

Học phần (hocPhan):

maHP	tenHP	soTinChi	soTietLT	soTietTH
CT101	Lập trình căn bản	4	30	60
CT176	Lập trình Hướng đối tượng	3	30	30
CT237	Nguyên lý Hệ điều hành	3	30	30
TN001	Vi tích phân A1	3	45	0
TN101	Xác suất thống kê	3	45	0
SP102	Đại số tuyến tính	4	60	0
TN172	Toán rời rạc	4	60	0
XH023	Anh văn căn bản 1	4	60	0
TN021	Hóa học đại cương	3	60	0

Khoa:

maKhoa	tenKhoa
NNG	Khoa Ngoại ngữ
CNTT&TT	Công nghệ thông tin và Truyền thông
TN	Khoa Khoa học tự nhiên
TS	Khoa Thủy sản
SP	Khoa Sư phạm
KT	Khoa Kinh tế

Sinh viên (sinhVien):

mssv	hoten	GT	ngaySinh	noiSinh	diaChi	maKhoa
B1234567	Nguyễn Thanh Hải	M	2001-12-02	Bạc Liêu	Phòng 01, KTX Khu B, ĐHCT	CNTT&TT
B1234568	Trần Thanh Mai	M	2001-01-20	Cần Thơ	232, Nguyễn Văn Khéo, Q. Ninh Kiều, TPCT	CNTT&TT
B1234569	Trần Thu Thủy	F	2001-07-01	Cần Thơ	02, Đại lộ Hòa Bình, Q. Ninh Kiều, TPCT	CNTT&TT
B1334569	Nguyễn Thị Trúc Mã	F	2002-05-25	Sóc Trăng	343, Đường 30/4, Q/ Ninh Kiều, TPCT	SP

B1345678	Trần Hồng Trúc	F	2002-03-02	Cần Thơ	123, Trần Hưng Đạo, Q. Ninh Kiều, TPCT	CNTT&TT
B1345679	Bùi Hoàng Yến	F	2001-11-05	Vĩnh Long	Phòng 201, KTX Khu A, TPCT	CNTT&TT
B1345680	Trần Minh Tâm	M	2001-02-04	Cà Mau	07, Đại lộ Hòa Bình, Q. Ninh Kiều, TPCT	KT
B1456789	Nguyễn Hồng Thắm	F	2003-05-09	An Giang	133, Nguyễn Văn Cừ, Q. Ninh Kiều, TPCT	NNG
B1459230	Lê Văn Khang	M	2002-06-02	Đồng Tháp	312, Nguyễn Văn Linh, Q. Ninh Kiều, TPCT	NN

Kết quả (ketQua):

mssv	maHP	diem
B1234567	CT101	4
B1234568	CT176	8
B1234569	CT237	9
B1334569	SP102	2
B1345678	CT101	6
B1345679	CT176	5
B1456789	TN172	10
B1456790	TN172	7
B1456789	XH023	6
B1459230	XH023	8

mssv	maHP	diem
B1234567	CT176	1
B1234568	CT101	9
B1234569	CT101	8
B1334569	CT101	4
B1345678	TN001	6
B1345679	CT101	2
B1456789	CT101	7
B1456790	CT101	6
B1345680	TN101	7
B1345680	XH023	7

Bài 3: Thực hiện một số yêu cầu sau dựa trên CSDL ở Bài 3:

- 1) Hiển thị danh sách SV bao gồm MSSV, họ tên, ngày tháng năm sinh, sắp xếp theo tăng dần theo MSSV.
- 2) Hiển thị danh sách các SV (MSSV, họ tên) thuộc khoa “Công nghệ thông tin và Truyền thông”
- 3) Hiển thị bảng điểm bao gồm các thông tin sau: MSSV, họ tên, tên học phần, điểm.
- 4) Hiển thị MSSV và họ tên các SV nữ sinh năm 2001.
- 5) Hiển thị danh sách bao gồm MSSV, họ tên, mã học phần đã học và điểm của từng học phần
- 6) Hiển thị danh sách các môn học chưa có SV học.
- 7) Hiển thị các sinh viên có điểm trung bình lớn nhất.
- 8) Hiển thị danh sách các Khoa cùng với số sinh viên tương ứng của Khoa đó.
- 9) Hiển thị khoa có nhiều SV nhất.
- 10) Cho biết mã học phần có sinh viên tên ‘Mai’

11) Cập nhật lại điểm của các sinh viên không thuộc Khoa CNTT&TT học môn CT101 lên mỗi SV 1 điểm.

Bài 4: Tạo các ràng buộc toàn vẹn trên các table trong CSDL QLDIEM như sau:

12) MSSV: Ký tự đầu tiên phải là một chữ cái từ 'A' đến 'Z', các ký tự còn lại phải là các số từ '0' đến '9' (CHECK (mssv REGEXP '^[A-Z][0-9]{8}\$'))

13) hoten: chỉ cho phép các ký tự từ 'a' đến 'z' (thường và hoa) và khoảng trắng (CHECK (hoten REGEXP '^[a-z]+\$'))

14) diem: có giá trị từ 0 đến 10 (CHECK (diem BETWEEN 0 AND 10)).

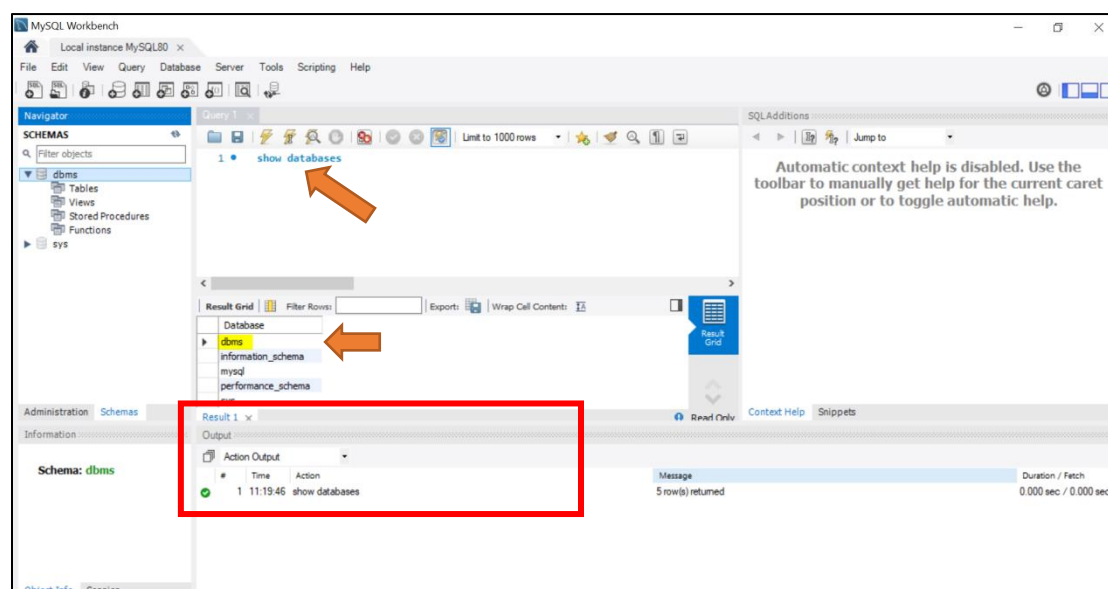
15) Phai: chỉ cho phép 1 trong 2 giá trị 'F' hoặc 'M' (CHECK phai IN ('M', 'F'))

-----Hết Buổi 1-----

HƯỚNG DẪN CÁCH TRẢ LỜI VÀ NỘP FILE BÀI LÀM

- Các bạn làm thực hành từ phần **1.6 Thực hành - nhưng chỉ nộp bài làm Bài 3 và Bài 4**
- Mỗi câu chụp màn hình phải có câu lệnh truy vấn và kết quả hiện thị bên dưới (15 câu tương đương có 15 hình)

Ví dụ: Thực hiện lệnh `show databases`. Như **Hình 1** nhìn thấy được câu truy vấn và kết quả sau khi chạy lệnh `show databases`



Hình 1

- Nộp file có phần mở rộng **.pdf** và lưu với tên: <MSSV>_<Họ và tên>_TH1.pdf

Ví dụ: Sinh viên có MSSV là B1234567 tên Nguyễn Văn An, thì lưu:

B1234567_NguyenVanAn_TH1.pdf