

Khoa Công Nghệ Thông Tin

Trường Đại Học Đà Lạt

Giới thiệu về hệ quản trị cơ sở dữ liệu quan hệ

MySQL

Trần Thị Phương Linh
linhttp@dlu.edu.vn

Nội dung

MySQL là gì?

Những tính năng được cung cấp

Quản trị CSDL

Các lệnh SQL cơ bản

Giới thiệu về MySQL

MySQL là gì?

- Hệ quản trị cơ sở dữ liệu quan hệ (RDBMS)
- Chạy trên nhiều platforms (Unix, Linux, Windows)
- Phổ biến
- Đơn giản, tốc độ nhanh
- Phần mềm mã nguồn mở, miễn phí (**MySQL Community Server**)
- Hỗ trợ chuẩn SQL
- Tồn tại hơn 16 năm qua, có nguồn gốc từ mSQL (ISAM)

Giới thiệu về MySQL

399 systems in ranking, December 2022

Rank			DBMS	Database Model	Score		
Dec 2022	Nov 2022	Dec 2021			Dec 2022	Nov 2022	Dec 2021
1.	1.	1.	Oracle	Relational, Multi-model	1250.31	+8.62	-31.43
2.	2.	2.	MySQL	Relational, Multi-model	1199.40	-6.14	-6.64
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	924.35	+11.84	-29.67
4.	4.	4.	PostgreSQL	Relational, Multi-model	617.97	-5.18	+9.76
5.	5.	5.	MongoDB	Document, Multi-model	469.33	-8.57	-15.34
6.	6.	6.	Redis	Key-value, Multi-model	182.57	+0.52	+9.03
7.	8.	7.	IBM Db2	Relational, Multi-model	146.61	-2.95	-20.56
8.	7.	8.	Elasticsearch	Search engine, Multi-model	144.93	-5.40	-12.80
9.	9.	10.	Microsoft Access	Relational	133.83	-1.20	+7.84
10.	10.	9.	SQLite	Relational	132.44	-2.19	+3.76
11.	12.	17.	Snowflake	Relational	114.77	+4.61	+43.73
12.	11.	11.	Cassandra	Wide column	114.65	-3.47	-4.55
13.	13.	12.	MariaDB	Relational, Multi-model	100.93	-3.98	-3.43
14.	14.	13.	Splunk	Search engine	90.79	-3.44	-3.53
15.	15.	16.	Amazon DynamoDB	Multi-model	83.85	-1.55	+6.22
16.	16.	14.	Microsoft Azure SQL Database	Relational, Multi-model	81.98	-1.68	-1.27

<https://db-engines.com/en/ranking>

Những tính năng của MySQL

MySQL hỗ trợ:

- SQL cơ bản (tạo bảng, chèn/xóa/cập nhật mẫu tin, truy vấn, etc)
- Những câu truy vấn phức tạp
- Ràng buộc khóa, ràng buộc dữ liệu, Trigger
- View (bảng ảo)



MySQL®

MySQL

Ưu điểm của MySQL:

- Linh hoạt và dễ sử dụng
- Hiệu năng cao
- An toàn
- Hỗ trợ kỹ thuật



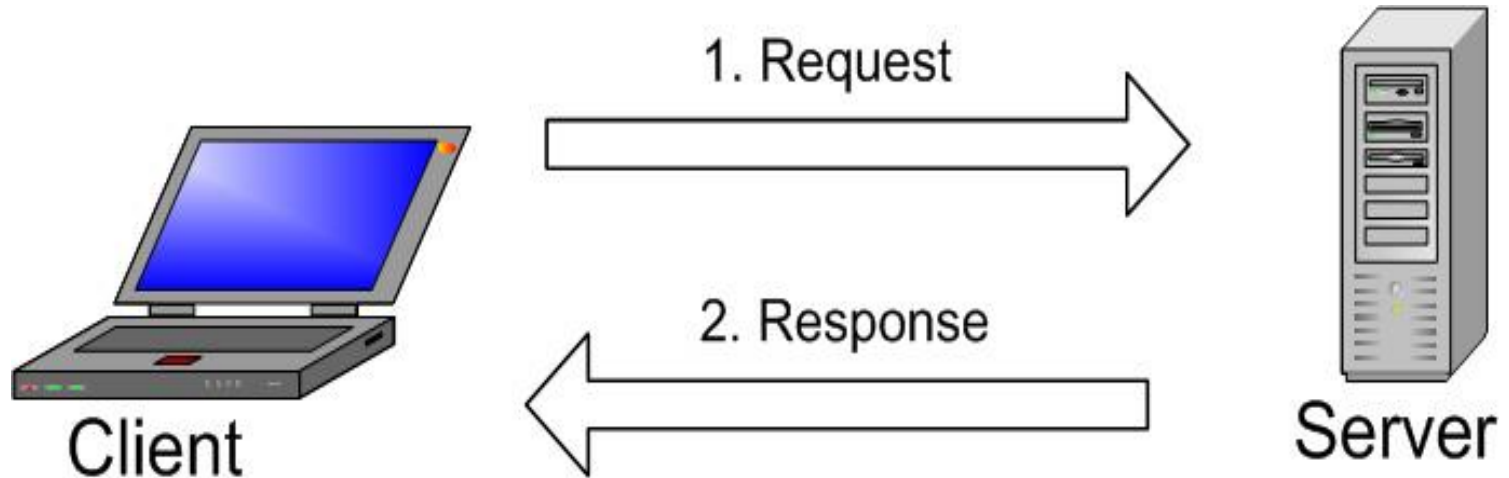
MySQL

Nhược điểm của MySQL:

- Dung lượng hạn chế
- Không hiệu quả với các cơ sở dữ liệu rất lớn
- Gỡ lỗi kém



Hoạt động



Dữ liệu trong MySQL

Data type	Description
CHAR(size)	Holds a fixed length string (can contain letters, numbers, and special characters). The fixed size is specified in parenthesis. Can store up to 255 characters
VARCHAR(size)	Holds a variable length string (can contain letters, numbers, and special characters). The maximum size is specified in parenthesis. Can store up to 255 characters. Note: If you put a greater value than 255 it will be converted to a TEXT type
TINYTEXT	Holds a string with a maximum length of 255 characters
TEXT	Holds a string with a maximum length of 65,535 characters
BLOB	For BLOBs (Binary Large Objects). Holds up to 65,535 bytes of data
MEDIUMTEXT	Holds a string with a maximum length of 16,777,215 characters
MEDIUMBLOB	For BLOBs (Binary Large Objects). Holds up to 16,777,215 bytes of data
LONGTEXT	Holds a string with a maximum length of 4,294,967,295 characters
LOBLOB	For BLOBs (Binary Large Objects). Holds up to 4,294,967,295 bytes of data
ENUM(x,y,z,etc.)	Let you enter a list of possible values. You can list up to 65535 values in an ENUM list. If a value is inserted that is not in the list, a blank value will be inserted. Note: The values are sorted in the order you enter them. You enter the possible values in this format: ENUM('X','Y','Z')
SET	Similar to ENUM except that SET may contain up to 64 list items and can store more than one choice

Dữ liệu trong MySQL

Data type	Description
TINYINT(size)	-128 to 127 normal. 0 to 255 UNSIGNED*. The maximum number of digits may be specified in parenthesis
SMALLINT(size)	-32768 to 32767 normal. 0 to 65535 UNSIGNED*. The maximum number of digits may be specified in parenthesis
MEDIUMINT(size)	-8388608 to 8388607 normal. 0 to 16777215 UNSIGNED*. The maximum number of digits may be specified in parenthesis
INT(size)	-2147483648 to 2147483647 normal. 0 to 4294967295 UNSIGNED*. The maximum number of digits may be specified in parenthesis
BIGINT(size)	-9223372036854775808 to 9223372036854775807 normal. 0 to 18446744073709551615 UNSIGNED*. The maximum number of digits may be specified in parenthesis
FLOAT(size,d)	A small number with a floating decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter
DOUBLE(size,d)	A large number with a floating decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter
DECIMAL(size,d)	A DOUBLE stored as a string , allowing for a fixed decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter

Dữ liệu trong MySQL

Data type	Description
DATE()	<p>A date. Format: YYYY-MM-DD</p> <p>Note: The supported range is from '1000-01-01' to '9999-12-31'</p>
DATETIME()	<p>*A date and time combination. Format: YYYY-MM-DD HH:MM:SS</p> <p>Note: The supported range is from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'</p>
TIMESTAMP()	<p>*A timestamp. TIMESTAMP values are stored as the number of seconds since the Unix epoch ('1970-01-01 00:00:00' UTC). Format: YYYY-MM-DD HH:MM:SS</p> <p>Note: The supported range is from '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC</p>
TIME()	<p>A time. Format: HH:MM:SS</p> <p>Note: The supported range is from '-838:59:59' to '838:59:59'</p>
YEAR()	<p>A year in two-digit or four-digit format.</p> <p>Note: Values allowed in four-digit format: 1901 to 2155. Values allowed in two-digit format: 70 to 69, representing years from 1970 to 2069</p>

Một số thuật ngữ trong MySQL

NULL: Giá trị cho phép rỗng.

NOT NULL: Giá trị không được rỗng.

AUTO_INCREMENT: Cho phép giá trị tăng dần (tự động).

UNSIGNED: Phải là số nguyên dương.

PRIMARY KEY: Cho phép nó là khóa chính trong bảng.

ENGINE: Cấu trúc lưu trữ (thường là MyISAM / InnoDB).

COMMENT: Chú thích cho trường hoặc bảng.

Lệnh MySQL thường dùng

- **CREATE** Tạo CSDL hoặc bảng
- **ALTER** Thay đổi bảng có sẵn
- **INSERT INTO** Chèn dữ liệu vào bảng
- **SELECT** Lấy dữ liệu từ bảng
- **DELETE** Xóa dữ liệu khỏi bảng
- **UPDATE** Cập nhật dữ liệu đã có trong bảng
- **DROP** Xóa bảng hay toàn bộ CSDL.

Lệnh MySQL thường dùng

- Các hàm tổng hợp dữ liệu:

AVG, COUNT, SUM, MIN, MAX

- Các hàm riêng của MySQL:

FIRST, LAST

UCASE, LCASE, MID, LENGTH

ROUND, FORMAT

NOW

Quản trị căn bản MySQL

Chạy/tắt MySQL server:

- Từ console gõ lệnh: `/etc/init.d/mysqld [start|stop]`

Đổi password của DBAdmin (**root**)

- Từ console gõ lệnh: `mysqladmin -u root password xxxx`

Làm việc từ mysql client

- Từ console gõ lệnh: `mysql -u root -p`
- Nhập password x x x x
- Dấu nhắc: `mysql>`
- exit: thoát, help: trợ giúp, etc.

Quản trị căn bản MySQL

Tạo một cơ sở dữ liệu: **create database db_name;**

- Ví dụ: **mysql> create database mydb;**

Chọn một cơ sở dữ liệu để làm việc: **use db_name;**

- Ví dụ: **mysql> use mydb;**

Xem thông tin về cơ sở dữ liệu và bảng:

- Ví dụ: **mysql> show databases;**
- Ví dụ: **mysql> show tables;**
- Ví dụ: **mysql> describe table_name;**

Xóa một cơ sở dữ liệu: **drop database db_name;**

- Ví dụ: **mysql> drop database mydb;**

Quản trị căn bản MySQL

Tạo và phân quyền cho user mới:

- `grant priv_type privileges on db_obj to username@"servername" identified by 'password';`
- Ví dụ: `mysql> grant all privileges on mydb.* to ivy_tran@"localhost" identified by 'mypasswd';`

Xóa phân quyền của user:

- `revoke priv_type privileges on db_obj from username@"servername";`
- Ví dụ: `mysql> revoke all privileges on mydb.* from ivy_tran@"localhost";`
- `mysql> drop user ivy_tran@"localhost";`

Kiểu phân quyền

Privilege	Column	Table	Database	Server
ALTER		X		
CREATE		X	X	
DELETE		X		
DROP		X	X	
GRANT		X	X	X
INSERT	X	X		
SELECT	X	X		
UPDATE	X	X		

Quản trị căn bản MySQL

Backup cơ sở dữ liệu

- `mysqldump --add-drop-table -u [username] -p[password] [db_name] > [backup_file]`
- Ví dụ từ console: `mysqldump --add-drop-table -u root - pxxxx mydb > mydb.bak`

Phục hồi cơ sở dữ liệu

- `mysql -u [username] -p[password] [db_name] < [backup_file]`
- Ví dụ từ console: `mysql -u root - pxxxx mydb < mydb.bak`

Quản trị căn bản MySQL

Backup bảng của cơ sở dữ liệu:

`select att into outfile desfile from sourcetable`

- Ví dụ: `mysql> select * into outfile '/tmp/test.sql' from test;`

Phục hồi bảng của cơ sở dữ liệu:

`load data infile sourcefile replace into table destable`

- Ví dụ: `mysql> load data infile '/tmp/test.sql' replace into table test;`

Tạo bảng

Cú pháp:

```
CREATE TABLE tbl_name (  
    column_definition  
    | [CONSTRAINT [symbol]] PRIMARY KEY [index_type]  
    (index_col_name,...)  
    | [CONSTRAINT [symbol]] UNIQUE [INDEX]  
    [index_name] [index_type] (index_col_name,...)  
    | [CONSTRAINT [symbol]] FOREIGN KEY  
    [index_name] (index_col_name,...) [reference_definition]  
    | CHECK (expr)  
);
```

Định nghĩa trường (cột dữ liệu) của bảng

Cú pháp:

column_definition:

col_name type [NOT NULL | NULL] [DEFAULT default_value]
[AUTO_INCREMENT]
[UNIQUE [KEY] | [PRIMARY] KEY]
[COMMENT 'string'] [reference_definition]

Các kiểu Constraint

NOT NULL	Thiết lập dữ liệu cột không được nhận giá trị NULL
UNIQUE	Thiết lập dữ liệu trong một cột không được phép trùng nhau
PRIMARY KEY	Thiết lập khóa chính
FOREIGN KEY	Thiết lập khóa ngoại
CHECK	Bảo đảm tất cả giá trị trong cột thỏa mãn điều kiện nào đó
DEFAULT	Gán giá trị mặc định cho dữ liệu

Kiểu dữ liệu

Các kiểu thông dụng

- Số: TINYINT, SMALLINT, MEDIUMINT, INT, INTEGER, BIGINT, REAL, DOUBLE, FLOAT, DECIMAL, NUMERIC
- Chuỗi ký tự: char, varchar, text
- Ngày giờ: date, time, timestamp
- Nhị phân: BLOB, TINYBLOB, BLOB, MEDIUMBLOB, LONGBLOB
- Tập hợp: SET

Tạo bảng

Cú pháp đơn giản

```
CREATE TABLE tbl_name (  
    col1 type1 [option1], col2 type2 [option2], ....  
);
```

Ví dụ 1

```
CREATE TABLE s (  
    sid varchar(3) PRIMARY KEY,  
    sname text NOT NULL,  
    status smallint,  
    city text);
```

```
CREATE TABLE p (  
    pid varchar(3) UNIQUE NOT NULL,  
    pname text NOT NULL,  
    color text,  
    weight smallint,  
    city text);
```

Ví dụ 2

```
CREATE TABLE sp (  
    sid varchar(3),  
    pid varchar(3),  
    qty integer NOT NULL CHECK (qty > 0),  
    CONSTRAINT sp_pkey PRIMARY KEY (sid, pid));
```

```
CREATE TABLE films (  
    code char(5) PRIMARY KEY,  
    title text NOT NULL,  
    date_prod date DEFAULT '2000/01/01');
```

Chỉnh sửa bảng

Cú pháp:

```
ALTER TABLE tbl_name alter_specification [,  
    alter_specification] ...
```

alter_specification:

```
    ADD [COLUMN] column_definition [FIRST | AFTER col_name ]  
| ADD [COLUMN] (column_definition,...)  
| ADD [CONSTRAINT [symbol]]  
    PRIMARY KEY [index_type] (index_col_name,...)  
| RENAME [TO] new_tbl_name  
| DROP [COLUMN] col_name
```

.....

Ví dụ 5

```
ALTER TABLE films ADD COLUMN nb int;
```

```
ALTER TABLE films DROP COLUMN nb;
```

```
ALTER TABLE films CHANGE title title varchar(30);
```

Chèn dữ liệu vào bảng

Cú pháp:

```
INSERT INTO table [ ( column [, ...] ) ]  
{ VALUES ( expression [, ...] ) | SELECT query }
```

Cú pháp đơn giản

```
INSERT INTO table(col1, col2, ...) VALUES (val1, val2, ...);
```

Ví dụ 6

Chèn dữ liệu vào bảng s:

```
INSERT INTO s VALUES ('S01', 'Smith', 20, 'London');
```

```
INSERT INTO s VALUES ('S02', 'Jones', 10, 'Paris');
```

```
INSERT INTO s VALUES ('S03', 'Blacke', 30, 'Paris');
```

Chèn dữ liệu vào bảng p:

```
INSERT INTO p VALUES ('P01', 'Nut', 'red', 12, 'London');
```

```
INSERT INTO p VALUES ('P02', 'Bolt', 'green', 17, 'Paris');
```

```
INSERT INTO p VALUES ('P03', 'Screw', 'blue', 17, 'Roma');
```

```
INSERT INTO p VALUES ('P04', 'Screw', 'red', 14, 'London');
```

Ví dụ 7

Chèn dữ liệu vào bảng sp:

```
INSERT INTO sp VALUES ('S01', 'P01', 300);
```

```
INSERT INTO sp VALUES ('S01', 'P02', 200);
```

```
INSERT INTO sp VALUES ('S01', 'P03', 400);
```

```
INSERT INTO sp VALUES ('S02', 'P01', 300);
```

```
INSERT INTO sp VALUES ('S02', 'P02', 400);
```

```
INSERT INTO sp VALUES ('S03', 'P02', 200);
```


Ví dụ 7

Chèn dữ liệu vào bảng films:

```
INSERT INTO films VALUES ('00001', 'Anh Hung Xa Dieu',  
'1990/12/07');
```

```
INSERT INTO films VALUES ('00002', 'Than Dieu Dai  
Hiep', '1991/07/22');
```

```
INSERT INTO films VALUES ('00013', 'Vo My  
Nuong', '1999/10/15');
```

Xoá dữ liệu từ bảng

Cú pháp:

DELETE FROM table [**WHERE** condition]

Ví dụ:

DELETE FROM films;

DELETE FROM films **WHERE** code = '00013';

Cập nhật dữ liệu của bảng

Cú pháp:

UPDATE table

SET col = expression [, ...]

[**WHERE** condition]

Ví dụ:

UPDATE films

SET title = 'Co Gai Do Long'

WHERE code = '00001';

Truy vấn dữ liệu

Cú pháp:

SELECT

[ALL | DISTINCT | DISTINCTROW]

select_expr, ...

[INTO OUTFILE 'file_name' export_options | INTO DUMPFILE
'file_name']

[**FROM** table_references

[**WHERE** where_definition]

[GROUP BY {col_name | expr | position} [ASC | DESC], ... [WITH
ROLLUP]]

[HAVING where_definition]

[ORDER BY {col_name | expr | position} [ASC | DESC] , ...]

[LIMIT {[offset,] row_count | row_count OFFSET offset}]

[PROCEDURE procedure_name(argument_list)]

[FOR UPDATE | LOCK IN SHARE MODE]]

Truy vấn dữ liệu

Cú pháp đơn giản:

```
SELECT select_expr, ...  
      FROM table_references  
      WHERE condition_expr
```

```
SELECT title  
      FROM films  
      WHERE code = '00013';
```

Ví dụ 8

```
mysql>
mysql> select 3*3-4, sqrt(16), round(2.6), abs(-2);
```

3*3-4	sqrt(16)	round(2.6)	abs(-2)
5	4	3	2

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

```
mysql> select current_date(), rand(), ln(2.7);
```

current_date()	rand()	ln(2.7)
2006-03-12	0.99811484664503	0.99325177301028

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

Ví dụ 9

```
mysql> select *  
-> from s;
```

sid	sname	status	city
S01	Smith	20	London
S02	Jones	10	Paris
S03	Blacke	30	Paris

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

```
mysql> select *  
-> from p;
```

pid	pname	color	weight	city
P01	Nut	red	12	London
P02	Bolt	green	17	Paris
P03	Screw	blue	17	Roma
P04	Screw	red	14	London

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

Ví dụ 10

```
mysql> select *
-> from sp;
+-----+-----+-----+
| sid | pid | qty |
+-----+-----+-----+
| S01 | P01 | 300 |
| S01 | P02 | 200 |
| S01 | P03 | 400 |
| S02 | P01 | 300 |
| S02 | P02 | 400 |
| S03 | P02 | 200 |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

```
mysql> select *
-> from sp
-> limit 3;
+-----+-----+-----+
| sid | pid | qty |
+-----+-----+-----+
| S01 | P01 | 300 |
| S01 | P02 | 200 |
| S01 | P03 | 400 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> select *
-> from sp
-> limit 3
-> offset 2;
+-----+-----+-----+
| sid | pid | qty |
+-----+-----+-----+
| S01 | P03 | 400 |
| S02 | P01 | 300 |
| S02 | P02 | 400 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```


Ví dụ 11

sid	sname	status	city
S01	Smith	20	London
S02	Jones	10	Paris
S03	Blacke	30	Paris

```
mysql> select sid
-> from s
-> where city='Paris';
+-----+
| sid |
+-----+
| S02 |
| S03 |
+-----+
2 rows in set (0.00 sec)

mysql> select *
-> from s
-> where city like 'P%';
+-----+-----+-----+-----+
| sid | sname | status | city |
+-----+-----+-----+-----+
| S02 | Jones | 10 | Paris |
| S03 | Blacke | 30 | Paris |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Ví dụ 12

sid	sname	status	city
S01	Smith	20	London
S02	Jones	10	Paris
S03	Blacke	30	Paris

sid	pid	qty
S01	P01	300
S01	P02	200
S01	P03	400
S02	P01	300
S02	P02	400
S03	P02	200

```
mysql> select distinct pid
-> from sp;
+-----+
| pid |
+-----+
| P01 |
| P02 |
| P03 |
+-----+
3 rows in set (0.00 sec)

mysql> select sid
-> from s
-> where city='Paris' and status > 20;
+-----+
| sid |
+-----+
| S03 |
+-----+
1 row in set (0.00 sec)
```

Ví dụ 13

```
mysql> select sid, status
-> from s
-> where city='Paris' order by status desc;
```

sid	status
S03	30
S02	10

2 rows in set (0.00 sec)

```
mysql> select sname
-> from s
-> where sid in
-> (select sid
-> from sp
-> where pid='P02');
```

sname
Smith
Jones
Blacke

3 rows in set (0.00 sec)

Ví dụ 14

pid	pname	color	weight	city
P01	Nut	red	12	London
P02	Bolt	green	17	Paris
P03	Screw	blue	17	Roma
P04	Screw	red	14	London

sid	pid	qty
S01	P01	300
S01	P02	200
S01	P03	400
S02	P01	300
S02	P02	400
S03	P02	200

```
mysql> select pid
-> from sp
-> where sid='S01'
-> union
-> select pid
-> from p
-> where weight>13;

+----+
| pid |
+----+
| P01 |
| P02 |
| P03 |
| P04 |
+----+

4 rows in set (0.00 sec)
```

Ví dụ 15

sid	pid	qty
S01	P01	300
S01	P02	200
S01	P03	400
S02	P01	300
S02	P02	400
S03	P02	200

```
mysql> select count(sid)
-> from sp
-> where pid='P02';
+-----+
| count(sid) |
+-----+
|          3 |
+-----+
1 row in set (0.00 sec)

mysql> select sum(qty)
-> from sp
-> where pid='P02';
+-----+
| sum(qty) |
+-----+
|        800 |
+-----+
1 row in set (0.00 sec)
```

Ví dụ 16

sid	pid	qty
S01	P01	300
S01	P02	200
S01	P03	400
S02	P01	300
S02	P02	400
S03	P02	200

```
mysql> select pid, count(sid)
-> from sp
-> group by pid;
+-----+-----+
| pid | count(sid) |
+-----+-----+
| P01 |          2 |
| P02 |          3 |
| P03 |          1 |
+-----+-----+
3 rows in set (0.00 sec)
```

Ví dụ 17

sid	pid	qty
S01	P01	300
S01	P02	200
S01	P03	400
S02	P01	300
S02	P02	400
S03	P02	200

```
mysql> select max(qty), avg(qty)
-> from sp;
+-----+-----+
| max(qty) | avg(qty) |
+-----+-----+
|      400 | 300.0000 |
+-----+-----+
1 row in set (0.00 sec)

mysql> select pid, sum(qty)
-> from sp
-> group by pid having pid<>'P02';
+-----+-----+
| pid | sum(qty) |
+-----+-----+
| P01 |      600 |
| P03 |      400 |
+-----+-----+
2 rows in set (0.00 sec)
```

JOIN trong MySQL

Các loại JOIN:

- Inner JOIN
- Left outer JOIN
- Right outer JOIN
- Full outer JOIN
- Cross JOIN

Ví dụ 18

```
mysql> select *  
-> from t1;  
+-----+-----+  
| id     | name  |  
+-----+-----+  
|      1 | Peter |  
|      2 | Bob   |  
|      3 | Joe   |  
+-----+-----+  
3 rows in set (0.00 sec)
```

```
mysql> select *  
-> from t2;  
+-----+-----+  
| code   | cname |  
+-----+-----+  
|      5 | Laval |  
|      2 | Paris |  
|      3 | Nantes|  
+-----+-----+  
3 rows in set (0.00 sec)
```

Ví dụ 20

```
mysql> select *  
      -> from t1 inner join t2 on t1.id=t2.code;
```

id	name	code	cname
2	Bob	2	Paris
3	Joe	3	Nantes

2 rows in set (0.00 sec)

```
mysql> select *  
      -> from t1 left join t2 on t1.id=t2.code;
```

id	name	code	cname
1	Peter	NULL	NULL
2	Bob	2	Paris
3	Joe	3	Nantes

3 rows in set (0.00 sec)

```
mysql> select *  
      -> from t1 right join t2 on t1.id=t2.code;
```

id	name	code	cname
NULL	NULL	5	Laval
2	Bob	2	Paris
3	Joe	3	Nantes

3 rows in set (0.01 sec)

Ví dụ 19

```
mysql> select *  
      -> from t1 cross join t2;
```

id	name	code	cname
1	Peter	5	Laval
2	Bob	5	Laval
3	Joe	5	Laval
1	Peter	2	Paris
2	Bob	2	Paris
3	Joe	2	Paris
1	Peter	3	Nantes
2	Bob	3	Nantes
3	Joe	3	Nantes

9 rows in set (0.00 sec)

Tài liệu tham khảo

www.mysql.com

Google

Tài liệu tham khảo

Sách:

- [Jon Stephens & Chad Russell, 2004] Jon Stephens & Chad Russell, “*Beginning MySQL Database Design and Optimization: From Novice to Professional*”, Apress, 2004.
- [Vikram Vaswani, 2003] Vikram Vaswani, “*MySQL (TM): The Complete Reference*”, McGraw Hill, 2003.
- [Paul DuBois, 2003] Paul DuBois, “*MySQL*”, SAMS, 2003.

Tài liệu tham khảo

Tài liệu hướng dẫn:

- [MySQL AB, 2005a] MySQL AB, “MySQL Administrator”, MySQL AB, 2005.
- [MySQL AB, 2005b] MySQL AB, “MySQL Reference Manual”, MySQL AB, 2005.



Cám ơn !



Câu hỏi ôn tập

1. Khái quát Ưu & Nhược điểm của MySQL?
2. So sánh 2 hệ quản trị CSDL: MySQL và SQL?
3. Các hàm riêng của MySQL? Nêu chức năng từng hàm và cho ví dụ?

MAKH	HOTEN	DCHI	SODT	NGSINH	DOANH SO	NGGMM
KH01	Nguyen Van A	731 Tran Hung Dao, Q5, TpHCM	08823451	22/10/1960	13,060,000	22/07/2006
KH02	Tran Ngoc Han	23/5 Nguyen Trai, Q5, TpHCM	0908256478	3/4/1974	280,000	30/07/2006
KH03	Tran Ngoc Linh	45 Nguyen Canh Chan, Q1, TpHCM	0938776266	12/6/1980	3,860,000	05/08/2006
KH04	Tran Minh Long	50/34 Le Dai Hanh, Q10, TpHCM	0917325476	9/3/1965	250,000	02/10/2006
KH05	Le Nhat Minh	34 Truong Dinh, Q3, TpHCM	08246108	10/3/1950	21,000	28/10/2006

NHANVIEN

MANV	HOTEN	DTHOAI	NGVL
NV01	Nguyen Nhu Nhut	0927345678	13/4/2006
NV02	Le Thi Phi Yen	0987567390	21/4/2006
NV03	Nguyen Van B	0997047382	27/4/2006
NV04	Ngo Thanh Tuan	0913758498	24/6/2006
NV05	Nguyen Thi Truc Thanh	0918590387	20/7/2006

SANPHAM

MASP	TENSP	DVT	NUOCSX	GIA
BC01	But chi	cay	Trung Quoc	3,000
BC02	But chi	cay	Thai Lan	5,000
BC03	But chi	cay	Viet Nam	3,500
BC04	But chi	hop	Viet Nam	30,000
BB01	But bi	cay	Viet Nam	5,000

HOADON

SOHD	NGHD	MAKH	MANV	TRIGIA
1001	23/07/2006	KH01	NV01	320,000
1002	12/08/2006	KH01	NV02	840,000
1003	23/08/2006	KH02	NV01	100,000
1004	01/09/2006	KH02	NV01	180,000
1005	20/10/2006	KH01	NV02	3,800,000

CTHD

SOHD	MASP	SL
1001	TV02	10
1001	ST01	5
1001	BC01	5
1001	BC02	10
1001	ST08	10
1002	BC04	20
1002	BB01	20
1002	BB02	20
1003	BB03	10
1004	TV01	20
1004	TV02	10
1004	TV03	10
1004	TV04	10
1005	TV05	50
1005	TV06	50

Yêu cầu

1. Viết lệnh tạo các bảng trên?
2. Viết lệnh nhập dữ liệu cho các bảng?
3. Tìm thông tin khách hàng có mua hàng trong tháng 08/2006?
4. In ra Tên khách hàng và những sản phẩm người đó đã mua?
5. Tìm các hóa đơn mua cả 2 sản phẩm “BC01” và “BC04”
6. Tìm các đơn hàng chỉ mua những sản phẩm do TQ sản xuất?