

 免費電子書

學習

MySQL

Free unaffiliated eBook created from
Stack Overflow contributors.

#mysql

.....	1
1: MySQL	2
.....	2
.....	2
Examples.....	2
.....	2
.....	6
PROCESSLIST	6
.....	6
2: ENUM	7
Examples.....	7
ENUM.....	7
TINYINT.....	7
VARCHAR.....	7
.....	8
NULLNOT NULL.....	8
3: JOINS3id	10
Examples.....	10
3.....	10
4: JSON	11
.....	11
.....	11
Examples.....	11
JSON.....	11
JSON.....	11
JSON.....	11
JSON.....	11
CASTJSON.....	12
Json.....	12
5: MyISAM	13
.....	13

Examples.....	13
ENGINE = MyISAM.....	13
6: MySQL LOCK TABLE.....	14
.....	14
.....	14
Examples.....	14
Mysql.....	14
.....	15
7: MySQL.....	17
.....	17
.....	17
Examples.....	17
.....	17
.....	18
.....	18
.....	18
.....	18
8: Mysql.....	19
Examples.....	19
.....	19
InnoDB.....	19
.....	19
9: MySQL.....	20
Examples.....	20
root.....	20
.....	20
Atomic RENAME.....	20
10: MySQL.....	21
.....	21
.....	21
Examples.....	21

.....	21
ALL.....	21
UNION ALLWHERE.....	22
11: SSL.....	23
Examples.....	23
Debian.....	23
CASSL.....	23
MySQL.....	23
SSL.....	24
SSL.....	24
.....	25
CentOS7 / RHEL7.....	25
dbserver.....	25
.....	27
.....	27
.....	28
appclient.....	28
12: TRIGGERS.....	30
.....	30
.....	30
.....	30
.....	30
Examples.....	30
.....	30
.....	31
.....	31
.....	31
Insert.....	31
Update.....	31
.....	31

13: UPDATE	33
.....	33
Examples	33
.....	33
.....	33
.....	33
Join Pattern	33
ORDER BYLIMIT	34
.....	34
.....	35
14:	36
.....	36
.....	36
Examples	36
.....	36
.....	36
.....	36
15:	38
Examples	38
.....	38
COMMITROLLBACKAUTOCOMMIT	38
JDBC	41
16: Docker-ComposeMysql	44
Examples	44
docker-compose	44
17: mysqldump	45
.....	45
.....	45
.....	45
Examples	46
.....	46
.....	46

.....	46
mysqldump.....	46
gzipmysqldump.....	47
Amazon S3.....	47
MySQLMySQL.....	47
.....	47
18: UTF-8.....	48
Examples.....	48
.....	48
PHP.....	48
19: Un-Pivot.....	49
Examples.....	49
.....	49
20:	51
Examples.....	51
.....	51
Select.....	51
21:	54
.....	54
.....	54
Examples.....	58
.....	58
22:	60
.....	60
.....	60
Examples.....	60
FULLTEXT.....	60
BOOLEAN.....	60
FULLTEXT.....	60
23:	62
.....	62
Examples.....	62

.....	62
.....	62
HASH.....	63
24:	64
.....	64
.....	64
Examples.....	64
Where.....	64
.....	64
.....	64
.....	64
.....	66
.....	66
DELETETRUNCATE.....	66
.....	66
25:	68
.....	68
.....	68
Examples.....	68
.....	68
MyDatabase.....	70
.....	70
.....	70
26:	71
.....	71
Examples.....	71
MySQL.....	71
.....	71
.....	71
.....	71
27:	72
.....	72

Examples.....	72
.....	72
“”.....	72
-.....	73
.....	73
3.....	75
.....	76
28:	77
.....	77
Examples.....	77
LOAD DATA INFILE.....	77
CSVMySQL.....	78
.....	78
.....	78
LOAD DATA INFILE'fname'	78
LOAD DATA INFILE'fname'IGNORE	78
.....	78
.....	79
29:	80
Examples.....	80
.....	80
30:	81
.....	81
Examples.....	81
.....	81
31:	82
.....	82
Examples.....	83
.....	83
STR_TO_DATE -	84
LOWER/ LCASE.....	84

.....	84
SUBSTRING.....	84
UPPER/ UCASE.....	84
.....	85
CHAR_LENGTH.....	85
HEXSTR.....	85
32:	86
Examples.....	86
.....	86
.....	86
.....	86
.....	86
33:	88
.....	88
.....	88
Examples.....	88
.....	88
.....	89
INOUTINOUT.....	89
.....	90
ResultSet.....	92
.....	92
34:	93
.....	93
Examples.....	93
REGEXP / RLIKE.....	93
^.....	93
\$**.....	93
REGEXP.....	93
.....	94
[].....	94
.....	94

.....	94
35: JSON	95
.....	95
.....	95
.....	95
.....	95
Examples.....	95
JSON.....	95
JSON.....	95
36: MyISAMInnoDB	97
Examples.....	97
.....	97
.....	97
37: root	98
Examples.....	98
rootrootsockethttp.....	98
38:	99
.....	99
.....	99
Examples.....	99
.....	99
.....	99
.....	99
.....	100
.....	100
.....	100
.....	100
.....	100
.....	101
.....	101
+ GROUP BY.....	101
39: MySQL 5.7+root	102
.....	102
.....	102

Examples.....	102
.....	102
root.....	102
UNIX"/ var / run / mysqld"root.....	102
40:	105
.....	105
.....	105
Examples.....	105
.....	105
.....	105
41:	106
.....	106
.....	106
Examples.....	106
.....	106
INSERTON DUPLICATE KEY UPDATE.....	106
.....	107
.....	107
INSERT SELECT.....	108
AUTO_INCREMENT + LAST_INSERT_IDINSERT.....	108
AUTO_INCREMENT ID.....	109
42:	111
Examples.....	111
/.....	111
VARCHAR255 -	111
INTAUTO_INCREMENT.....	112
.....	112
.....	112
.....	112
.....	113
.....	113
.....	113
.....	113

CHARN.....	114
DATEDATETIMETIMESTAMPYEARTIME.....	114
43:	116
Examples.....	116
.....	116
.....	116
.....	116
.....	116
SYSDATENOWCURDATE.....	117
.....	117
.....	117
44:	118
Examples.....	118
.....	118
.....	118
.....	119
.....	120
45:	122
.....	122
Examples.....	122
.....	122
Javascript.....	122
.....	122
/.....	123
JavascriptTIMESTAMP.....	123
46:	124
Examples.....	124
LinuxMySQL root.....	124
WindowsMySQL root.....	125
.....	125
47:	126
.....	126
.....	126

Examples.....	127
;;file_per_table.....	127
.....	127
ALTERINDEX.....	127
.....	127
.....	127
.....	128
MySQL.....	128
MySQL.....	128
MySQL.....	129
MySQL.....	129
48:	130
.....	130
Examples.....	130
SHOW VARIABLES.....	130
.....	130
49:	132
Examples.....	132
.....	132
.....	132
21102.....	132
.....	133
.....	133
50:	135
.....	135
Examples.....	135
DEALLOCATE PREPARE.....	135
.....	135
.....	135
51: mysqlimport	136
.....	136

.....	136
Examples.....	136
.....	136
.....	136
.....	136
.....	137
.....	137
csv.....	137
52:	139
Examples.....	139
NULL.....	139
NULL.....	139
53:	140
.....	140
Examples.....	140
.....	140
BIGINT.....	140
.....	140
.....	140
.....	140
SINCOS.....	141
.....	141
.....	141
.....	141
.....	141
.....	141
.....	141
.....	141
ROUNDFLOORCEIL.....	142
.....	142
.....	142

.....	142
.....	142
POW.....	142
SQRT.....	142
.....	143
.....	143
.....	143
ABSSIGN.....	143
54:	145
.....	145
.....	145
.....	145
Examples.....	145
.....	145
.....	145
.....	145
.....	145
AUTO_INCREMENT.....	146
55:	147
Examples.....	147
.....	147
56:	148
.....	148
.....	148
Examples.....	148
SELECTUNION.....	148
.....	148
OFFSET.....	148
.....	149
UNION ALLUNION.....	149
MySQL.....	149
57:	151

Examples.....	151
.....	151
.....	151
58: PS1.....	152
Examples.....	152
MySQL PS1.....	152
MySQLPS1.....	152
59:	153
.....	153
Examples.....	153
.....	153
`DATE` `DATETIME`.....	153
`TIMESTAMP`.....	153
.....	154
time_zone.....	154
60:	155
Examples.....	155
NULL.....	155
61:	158
.....	158
.....	158
Examples.....	158
.....	158
.....	158
.....	159
.....	159
.....	159
.....	160
.....	160
SELECT.....	160
.....	161

TimeStamp.....	161
62:	163
.....	163
Examples.....	163
-	163
.....	165
63:	167
.....	167
.....	167
.....	167
Examples.....	167
.....	167
.....	168
VIEW.....	168
.....	168
64:	170
Examples.....	170
.....	170
.....	170
.....	170
.....	170
65: Mysql	172
.....	172
Examples.....	172
.....	172
.....	172
66:	173
.....	173
Examples.....	173
.....	173
67:	175

.....	175
.....	175
.....	175
Examples.....	175
GROUP BYSUM.....	175
MIN.....	175
GROUP BYCOUNT.....	175
GROUP BYHAVING.....	176
Group Concat.....	176
AGGREGATEGROUP BY.....	176
68: GRANTs.....	180
Examples.....	180
.....	180
@.....	180
69:	181
.....	181
.....	181
.....	181
Examples.....	181
.....	181
*.....	181
WHERE.....	182
WHERESELECT	182
SELECT with LIKE.....	182
SELECT with AliasAS.....	184
LIMITSELECT.....	184
DISTINCT.....	184
LIKE_.....	185
CASEIF.....	185
BETWEEN.....	186
.....	187

70:	188
.....	188
Examples.....	188
InnoDB.....	188
.....	188
group_concat.....	188
InnoDB.....	188
MySQL.....	189
71: 1055ONLY_FULL_GROUP_BY.....	190
.....	190
.....	190
Examples.....	190
GROUP BY.....	190
GROUP BY.....	191
GROUP BYSELECT *.....	191
ANY_VALUE.....	192
72:	193
Examples.....	193
1064.....	193
1175.....	193
1215.....	193
1045.....	194
1236"".....	194
20022003.....	195
1067,1292,1366,1411 -	195
126,127,134,144,145.....	195
139.....	196
1366.....	196
126,1054,1146,1062,24.....	196
73:	198
.....	198
.....	198

Examples.....	198
.....	198
LIMIT.....	198
LIMIT.....	198
OFFSET.....	199
.....	200

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [mysql](#)

It is an unofficial and free MySQL ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official MySQL.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: MySQL



MySQL RDBMS Oracle Corporation.

MySQL Linux OS X Windows. [API](#) [CC++](#) [Java](#) [Lua](#) [.Net](#) [Perl](#) [PHP](#) [Python](#) [Ruby](#).

[MariaDB](#) MySQL. MySQL.

1.0	1995-05-23
3.19	1996-12-01
3.20	1997-01-01
3.21	1998-10-01
3.22	1999-10-01
3.23	2001122
4	2003-03-01
4.1	2004-10-01
5	2005-10-01
5.1	2008-11-27
5.5	2010-11-01
5.6	201321
5.7	2015101

Examples

MySQL

```
CREATE DATABASE mydb;
```

OK10.05

mydb

```
USE mydb;
```

MySQL

```
CREATE TABLE mytable
(
  id            int unsigned NOT NULL auto_increment,
  username      varchar(100) NOT NULL,
  email        varchar(100) NOT NULL,
  PRIMARY KEY  (id)
);
```

```
CREATE TABLE mytablemytable
```

```
id int unsigned NOT NULL auto_incrementIDid MySQLid1
```

OK00.10

MySQL

```
INSERT INTO mytable ( username, email )
VALUES ( "myuser", "myuser@example.com" );
```

10.06

varchar **aka** strings

```
INSERT INTO mytable ( username, email )
VALUES ( 'username', 'username@example.com' );
```

MySQL

```
UPDATE mytable SET username="myuser" WHERE id=8
```

10.06

```
int ' ' °
```

MySQL

```
DELETE FROM mytable WHERE id=8
```

10.06

MySQL

```
SELECT * FROM mytable WHERE username = "myuser";
```

```
+----+-----+-----+
| id | username | email                |
+----+-----+-----+
|  1 | myuser   | myuser@example.com  |
+----+-----+-----+
```

10.00

```
SHOW databases;
```

```
+-----+
| Databases |
+-----+
| information_schema |
| mydb         |
+-----+
```

20.00

“information_schema””。

```
SHOW tables;
```

```
+-----+
| Tables_in_mydb |
+-----+
| mytable        |
+-----+
```

10.00

```
DESCRIBE databaseName.tableName;
```

```
DESCRIBE tableName;
```

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type                | Null | Key  | Default          | Extra |
+-----+-----+-----+-----+-----+-----+
| fieldname  | fieldvaluetype     | NO/YES | keytype | defaultfieldvalue |       |
+-----+-----+-----+-----+-----+-----+
```


Extraauto_increment°

Key° PRIUNI.....

n0.00

n°

/° °

```
CREATE USER 'user'@'localhost' IDENTIFIED BY 'some_password';
```

°

```
CREATE USER 'user'@'%' IDENTIFIED BY 'some_password';
```

°

00.00

```
GRANT SELECT, INSERT, UPDATE ON databaseName.* TO 'userName'@'localhost';
```

```
GRANT ALL ON *.* TO 'userName'@'localhost' WITH GRANT OPTION;
```

. databaseName.*° databaseName.tableName°

WITH GRANT OPTION °

```
ALL
```

°

```
SELECT  
INSERT  
UPDATE  
DELETE  
CREATE  
DROP
```

SQL° tablefirst name °

`,`°

```
CREATE TABLE `table`  
(  
  `first name` VARCHAR(30)  
);
```

```
SELECT `first name` FROM `table` WHERE `first name` LIKE 'a%';
```

PROCESSLIST

◦

```
SELECT * FROM information_schema.PROCESSLIST ORDER BY INFO DESC, TIME DESC;
```

```
SELECT ID, USER, HOST, DB, COMMAND,  
TIME as time_seconds,  
ROUND(TIME / 60, 2) as time_minutes,  
ROUND(TIME / 60 / 60, 2) as time_hours,  
STATE, INFO  
FROM information_schema.PROCESSLIST ORDER BY INFO DESC, TIME DESC;
```

Stored Procedures◦

```
SELECT * FROM information_schema.ROUTINES WHERE ROUTINE_DEFINITION LIKE '%word%';
```

MySQL <https://riptutorial.com/zh-TW/mysql/topic/302/mysql>

2: ENUM

Examples

ENUM

ENUM. ◦

```
reply ENUM('yes', 'no')
gender ENUM('male', 'female', 'other', 'decline-to-state')
```

```
INSERT ... VALUES ('yes', 'female')
SELECT ... --> yes female
```

TINYINT

```
type ENUM('fish','mammal','bird')
```

```
type TINYINT UNSIGNED
```

```
CREATE TABLE AnimalTypes (
  type TINYINT UNSIGNED NOT NULL AUTO_INCREMENT,
  name VARCHAR(20) NOT NULL COMMENT "('fish','mammal','bird')",
  PRIMARY KEY(type),
  INDEX(name)
) ENGINE=InnoDB
```

◦

ENUM

- INSERT_{type}
- SELECT JOIN ENUM
- ◦ ENUM ALTER TABLE ◦
- 2551 ◦
- TINYINT; ENUM SQL ◦ TINYINT / ◦ FOREIGN KEYS ◦

VARCHAR

```
type ENUM('fish','mammal','bird')
```

```
type VARCHAR(20) COMMENT "fish, bird, etc"
```

◦

ENUM

- INSERT
- INSERT
- SELECT
-

```
ALTER TABLE tbl MODIFY COLUMN type ENUM('fish','mammal','bird','insect');
```

- **MODIFY COLUMN**NOT NULL◦
- 256ALTER ◦ ◦ MySQL◦

NULLNOT NULL

NULL'bad-value'◦ +0◦

```
CREATE TABLE enum (
  e      ENUM('yes', 'no')  NOT NULL,
  enull  ENUM('x', 'y', 'z')  NULL
);
INSERT INTO enum (e, enull)
VALUES
  ('yes', 'x'),
  ('no', 'y'),
  (NULL, NULL),
  ('bad-value', 'bad-value');
Query OK, 4 rows affected, 3 warnings (0.00 sec)
Records: 4 Duplicates: 0 Warnings: 3
```

```
mysql>SHOW WARNINGS;
+-----+-----+-----+
| Level | Code | Message |
+-----+-----+-----+
| Warning | 1048 | Column 'e' cannot be null |
| Warning | 1265 | Data truncated for column 'e' at row 4 |
| Warning | 1265 | Data truncated for column 'enull' at row 4 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

◦ “+0”◦

```
mysql>SELECT e, e+0 FROM enum;
+-----+-----+
| e | e+0 |
+-----+-----+
| yes | 1 |
| no | 2 |
| | 0 | -- NULL
| | 0 | -- 'bad-value'
+-----+-----+
4 rows in set (0.00 sec)

mysql>SELECT enull, enull+0 FROM enum;
+-----+-----+
| enull | enull+0 |
+-----+-----+
```

```
+-----+-----+
| x      |      1 |
| y      |      2 |
| NULL   |    NULL |
|        |      0 | -- 'bad-value'
+-----+-----+
4 rows in set (0.00 sec)
```

ENUM <https://riptutorial.com/zh-TW/mysql/topic/4425/enum>

3: JOINS3id.

Examples

3

```
CREATE TABLE Table1 (  
  id INT UNSIGNED NOT NULL,  
  created_on DATE NOT NULL,  
  PRIMARY KEY (id)  
)  
CREATE TABLE Table2 (  
  id INT UNSIGNED NOT NULL,  
  personName VARCHAR(255) NOT NULL,  
  PRIMARY KEY (id)  
)  
CREATE TABLE Table3 (  
  id INT UNSIGNED NOT NULL,  
  accountName VARCHAR(255) NOT NULL,  
  PRIMARY KEY (id)  
)
```

selectid

```
SELECT  
  t1.id AS table1Id,  
  t2.id AS table2Id,  
  t3.id AS table3Id  
FROM Table1 t1  
LEFT JOIN Table2 t2 ON t2.id = t1.id  
LEFT JOIN Table3 t3 ON t3.id = t1.id
```

JOINS3id. <https://riptutorial.com/zh-TW/mysql/topic/9921/joins-3id->

4: JSON

MySQL 5.7.8MySQLJSONJavaScript Object Notation

<https://dev.mysql.com/doc/refman/5.7/en/json.html>

MySQL 5.7.8MySQLJSON. JSONJSON. .

Examples

JSON

```
CREATE TABLE table_name (  
    id INT NOT NULL AUTO_INCREMENT,  
    json_col JSON,  
    PRIMARY KEY(id)  
);
```

JSON

```
INSERT INTO  
    table_name (json_col)  
VALUES  
    ('{"City": "Galle", "Description": "Best damn city in the world"}');
```

JSON. .

JSON.

json.

```
INSERT INTO myjson(dict)  
VALUES ('{"opening": "Sicilian", "variations": ["pelikan", "dragon", "najdorf"]}');
```

. .

JSON

JSON. variations.

```
UPDATE  
    myjson  
SET  
    dict=JSON_ARRAY_APPEND(dict, '$.variations', 'scheveningen')  
WHERE  
    id = 2;
```

1. json\$.variations. \$json. mysqljson<https://dev.mysql.com/doc/refman/5.7/en/json-path->

5: MyISAM

InnoDBMyISAM。 MyISAM。

MyISAMInnoDB23。

InnoDBMyISAM。 XtraDB5.6InnoDBMyISAM“”。

InnoDBInnoDBMyISAM。

Examples

ENGINE = MyISAM

```
CREATE TABLE foo (  
    ...  
) ENGINE=MyISAM;
```

MyISAM <https://riptutorial.com/zh-TW/mysql/topic/4710/myisam>

6: MySQL LOCK TABLE

- LOCK TABLES table_name [READ]; //
- ;//

◦ ◦ ◦

READ LOCK - ◦

WRITE LOCK - ◦

WRITE LOCK ◦ READ LOCK READ LOCK WRITE LOCK ◦

InnoDBMySQL ◦

InnoDBMySQL ◦

Examples

Mysql

ENGINE=MyISAMENGINE=InnoDB InnoDB ◦

MySQL ◦ ◦ ◦

◦ ◦

```
LOCK TABLES table_name READ|WRITE;
```

```
;
```

READ LOCK

```
LOCK TABLES table_name READ;
```

WRITE LOCK

```
LOCK TABLES table_name WRITE;
```

```
SHOW OPEN TABLES;
```

```
/
```

```
UNLOCK TABLES;
```

```
LOCK TABLES products WRITE;
INSERT INTO products(id,product_name) SELECT id,old_product_name FROM old_products;
UNLOCK TABLES;
```

```
LOCK TABLES products READ;
INSERT INTO products(id,product_name) SELECT id,old_product_name FROM old_products;
UNLOCK TABLES;
```

InnoDBMySQL。

。

```
SELECT ... FOR UPDATE。
```

1

```
START TRANSACTION;
SELECT ledgerAmount FROM accDetails WHERE id = 1 FOR UPDATE;
```

```
1SELECT ... FOR UPDATE。
```

2

```
UPDATE accDetails SET ledgerAmount = ledgerAmount + 500 WHERE id=1;
```

21innodb_lock_wait_timeout50。

```
Error Code: 1205. Lock wait timeout exceeded; try restarting transaction
```

```
SHOW ENGINE INNODB STATUS
```

```
---TRANSACTION 1973004, ACTIVE 7 sec updating
mysql tables in use 1, locked 1
LOCK WAIT 2 lock struct(s), heap size 360, 1 row lock(s)
MySQL thread id 4, OS thread handle 0x7f996beac700, query id 30 localhost root update
UPDATE accDetails SET ledgerAmount = ledgerAmount + 500 WHERE id=1
----- TRX HAS BEEN WAITING 7 SEC FOR THIS LOCK TO BE GRANTED:
```

2

```
UPDATE accDetails SET ledgerAmount = ledgerAmount + 250 WHERE id=2;
1 row(s) affected
```

2。

1

```
UPDATE accDetails SET ledgerAmount = ledgerAmount + 750 WHERE id=1;
COMMIT;
1 row(s) affected
```

1。

2

```
UPDATE accDetails SET ledgerAmount = ledgerAmount + 500 WHERE id=1;  
1 row(s) affected
```

12。

MySQL LOCK TABLE <https://riptutorial.com/zh-TW/mysql/topic/5233/mysql-lock-table>

7: MySQL

- `mysql [OPTIONS] [database_name]`

<code>-D --database=name</code>	
<code>--delimiter=str</code>	◦ <code>';</code>
<code>-e --execute='command'</code>	
<code>-h --host=name</code>	
<code>-p --password=name</code>	<code>-p</code>
<code>-p</code>	
<code>-P --port=#</code>	
<code>-s --silent</code>	◦ <code>\t</code>
<code>-ss</code>	<code>-s</code>
<code>-S --socket=path</code>	UnixWindows
<code>--skip-column-names</code>	
<code>-u --user=name</code>	
<code>-U --safe-updates --i-am-a-dummy</code>	<code>sql_safe_updates=ON</code> ◦ <code>DELETEUPDATE</code>
<code>-V --version</code>	

Examples

MySQL

```
mysql --user=username --password=pwd --host=hostname test_db
```

```
mysql -u username -p password -h hostname test_db
```

passwordMySQL ◦ password “”

```
mysql -u=username -p -h=hostname test_db
```

```
--socket
```

```
mysql --user=username --password=pwd --host=localhost --socket=/path/to/mysql.sock test_db
```

socket◦ ◦

◦ **shell**◦

```
$ mysql -uroot -proot test -e'select * from people'
```

```
+----+-----+-----+
| id | name  | gender |
+----+-----+-----+
|  1 | Kathy | f      |
|  2 | John  | m      |
+----+-----+-----+
```

--silent

```
$ mysql -uroot -proot test -s -e'select * from people'
```

```
id      name    gender
1       Kathy   f
2       John    m
```

```
$ mysql -uroot -proot test -ss -e'select * from people'
```

```
1       Kathy   f
2       John    m
```

```
$ mysql -uroot -proot test < my_script.sql
```

```
$ mysql -uroot -proot test -e'source my_script.sql'
```

```
$ mysql -uroot -proot test < my_script.sql > out.txt
```

```
$ mysql -uroot -proot test -s -e'select * from people' > out.txt
```

MySQL <https://riptutorial.com/zh-TW/mysql/topic/5619/mysql>

8: Mysql

Examples

MySQL -

1. wherewhere。 - *user_id > 2000. user_id at lot。 ◦
2. limit。 Ex - *。 20 Ex - LIMIT 20*。
3. ◦ Ex - *。 ◦ - ID。
4. where NULL。 SELECT * FROM tbl_name WHERE key_col IS NULL; key_col。

InnoDB

1. InnoDB PRIMARY KEY。 ◦ AUTO_INCREMENT。
2. VARCHAR CHAR NULL。 NULL CHAR NN。 I / O。
COMPACT InnoDB utf8sjis CHAR NN。
3. COMPRESSED。 I / O. COMPRESSED COMPACT。 21 COMPRESSED。
4. OPTIMIZE TABLE。 I / O。 ◦ OPTIMIZE TABLE。 ◦ InnoDB +。

OPTIMIZE TABLE。 ◦ ◦ ◦ ◦ ◦ ◦

◦ ◦

- =WHERE◦ INDEX(a,b,...) WHERE a=12 AND b='xyz' ...
- IN;◦
- ""x BETWEEN 3 AND 9 name LIKE 'J%' ◦
- GROUP BY
- ORDER BY◦ ASC DESC 8.0◦
- ◦
- ◦
- WHERE GROUP BY◦
- ORDER BY WHERE ◦
- ""DATE(x) = ...x ◦
- " text_col(99) ;◦

◦

Mysql <https://riptutorial.com/zh-TW/mysql/topic/5752/mysql>

9: MySQL

Examples

root

```
mysqladmin -u root -p'old-password' password 'new-password'
```

```
mysqladmin -u[username] -p[password] drop [database]
```

◦

DROP SQLDROP

```
DROP DATABASE database_name
```

```
DROP SCHEMA database_name
```

Atomic RENAME

```
RENAME TABLE t TO t_old, t_copy TO t;
```

RENAME TABLE◦

Atomic RenameDELETE

```
CREATE TABLE new LIKE real;  
load `new` by whatever means - LOAD DATA, INSERT, whatever  
RENAME TABLE real TO old, new TO real;  
DROP TABLE old;
```

MySQL <https://riptutorial.com/zh-TW/mysql/topic/2991/mysql>

10: MySQL

- `SELECT column_namesFROM table1 UNION SELECT column_namesFROM table2;`
- `SELECT column_namesFROM table1 UNION ALL SELECT column_namesFROM table2;`
- `SELECT column_namesFROM table1 WHERE col_name =“XYZ”UNION ALL SELECT column_namesFROM table2 WHERE col_name =“XYZ”;`

`UNION DISTINCTUNION;UNION ALL° DISTINCTALL °`

Examples

`UNIONSELECT °`

`:(“”“”`

```
SELECT City FROM Customers
UNION
SELECT City FROM Suppliers
ORDER BY City;
```

Number of Records: 10

```
City
-----
Aachen
Albuquerque
Anchorage
Annecy
Barcelona
Barquisimeto
Bend
Bergamo
Berlin
Bern
```

ALL

`UNION ALL “”“” °`

```
SELECT City FROM Customers
UNION ALL
SELECT City FROM Suppliers
ORDER BY City;
```

Number of Records: 12

```
City
-----
Aachen
Albuquerque
```

Anchorage
Ann Arbor
Annecy
Barcelona
Barquisimeto
Bend
Bergamo
Berlin
Berlin
Bern

UNION ALLWHERE

UNION ALL "" Country="Germany"where

```
SELECT City, Country FROM Customers  
WHERE Country='Germany'  
UNION ALL  
SELECT City, Country FROM Suppliers  
WHERE Country='Germany'  
ORDER BY City;
```

Number of Records: 14

aM	

MySQL <https://riptutorial.com/zh-TW/mysql/topic/5376/mysql>

11: SSL

Examples

Debian

MySQL_{sudo} ◦

CASSL

OpenSSL

```
apt-get -y install openssl
apt-get -y install libssl-dev
```

SSL

```
mkdir /home/ubuntu/mysqlcerts
cd /home/ubuntu/mysqlcerts
```

CA

```
openssl genrsa 2048 > ca-key.pem
openssl req -new -x509 -nodes -days 3600 -key ca-key.pem -out ca.pem
```

◦ CA

```
openssl req -newkey rsa:2048 -days 3600 -nodes -keyout server-key.pem -out server-req.pem
openssl rsa -in server-key.pem -out server-key.pem
```

```
openssl x509 -req -in server-req.pem -days 3600 -CA ca.pem -CAkey ca-key.pem -set_serial 01 -
out server-cert.pem
```

```
openssl req -newkey rsa:2048 -days 3600 -nodes -keyout client-key.pem -out client-req.pem
openssl rsa -in client-key.pem -out client-key.pem
openssl x509 -req -in client-req.pem -days 3600 -CA ca.pem -CAkey ca-key.pem -set_serial 01 -
out client-cert.pem
```

```
openssl verify -CAfile ca.pem server-cert.pem client-cert.pem
```

MySQL

MySQL ◦

```
vim /etc/mysql/mysql.conf.d/mysqld.cnf
```

```
[mysqld]
```

```
ssl-ca = /home/ubuntu/mysqlcerts/ca.pem  
ssl-cert = /home/ubuntu/mysqlcerts/server-cert.pem  
ssl-key = /home/ubuntu/mysqlcerts/server-key.pem
```

MySQL

```
service mysql restart
```

SSL

```
ssl-ca ssl-certssl-key key° cd /home/ubuntu/mysqlcerts
```

```
mysql --ssl-ca=ca.pem --ssl-cert=client-cert.pem --ssl-key=client-key.pem -h 127.0.0.1 -u  
superman -p
```

```
superman@127.0.0.1 [None]> SHOW VARIABLES LIKE '%ssl%';  
+-----+-----+  
| Variable_name | Value |  
+-----+-----+  
| have_openssl  | YES   |  
| have_ssl     | YES   |  
| ssl_ca       | /home/ubuntu/mysqlcerts/ca.pem |  
| ssl_capath   |       |  
| ssl_cert     | /home/ubuntu/mysqlcerts/server-cert.pem |  
| ssl_cipher   |       |  
| ssl_crl      |       |  
| ssl_crlpath  |       |  
| ssl_key      | /home/ubuntu/mysqlcerts/server-key.pem |  
+-----+-----+
```

```
superman@127.0.0.1 [None]> STATUS;  
...  
SSL:                Cipher in use is DHE-RSA-AES256-SHA  
...
```

SSL

```
GRANT REQUIRE SSL
```

```
GRANT ALL PRIVILEGES ON *.* TO 'superman'@'127.0.0.1' IDENTIFIED BY 'pass' REQUIRE SSL;  
FLUSH PRIVILEGES;
```

```
superman SSL°
```

◦ MySQL

```
vim /etc/mysql/mysql.conf.d/mysqld.cnf
```

```
[client]
```

```
ssl-ca = /home/ubuntu/mysqlcerts/ca.pem  
ssl-cert = /home/ubuntu/mysqlcerts/client-cert.pem  
ssl-key = /home/ubuntu/mysqlcerts/client-key.pem
```

superman**SSL**

```
mysql -h 127.0.0.1 -u superman -p
```

Python◦ Python

```
import MySQLdb  
ssl = {'cert': '/home/ubuntu/mysqlcerts/client-cert.pem', 'key':  
      '/home/ubuntu/mysqlcerts/client-key.pem'}  
conn = MySQLdb.connect(host='127.0.0.1', user='superman', passwd='imsoawesome', ssl=ssl)
```

- <https://www.percona.com/blog/2013/06/22/setting-up-mysql-ssl-and-secure-connections/>
- <https://lowendbox.com/blog/getting-started-with-mysql-over-ssl/>
- <http://xmodulo.com/enable-ssl-mysql-server-client.html>
- <https://ubuntuforums.org/showthread.php?t=1121458>

CentOS7 / RHEL7

1. dbserver
2. appclient

FWIWSELinux◦

dbserver

◦

```
mkdir /root/certs/mysql/ && cd /root/certs/mysql/
```

```
openssl genrsa 2048 > ca-key.pem  
openssl req -sha1 -new -x509 -nodes -days 3650 -key ca-key.pem > ca-cert.pem  
openssl req -sha1 -newkey rsa:2048 -days 730 -nodes -keyout server-key.pem > server-req.pem  
openssl rsa -in server-key.pem -out server-key.pem  
openssl x509 -sha1 -req -in server-req.pem -days 730 -CA ca-cert.pem -CAkey ca-key.pem -  
set_serial 01 > server-cert.pem
```

/ etc / pki / tls / certs / mysql /

CentOS/RHEL

```
mkdir /etc/pki/tls/certs/mysql/
```

◦ mysql◦

```
chown -R mysql:mysql /etc/pki/tls/certs/mysql
```

MySQL / MariaDB

```
# vi /etc/my.cnf
# i
[mysqld]
bind-address=*
ssl-ca=/etc/pki/tls/certs/ca-cert.pem
ssl-cert=/etc/pki/tls/certs/server-cert.pem
ssl-key=/etc/pki/tls/certs/server-key.pem
# :wq
```

```
systemctl restart mariadb
```

appclientIP 1.2.3.4

```
firewall-cmd --zone=drop --permanent --add-rich-rule 'rule family="ipv4" source
address="1.2.3.4" service name="mysql" accept'
# I force everything to the drop zone. Season the above command to taste.
```

firewalld

```
service firewalld restart
```

dbservermysql

```
mysql -uroot -p
```

◦ GRANTREQUIRE SSL◦

```
GRANT ALL PRIVILEGES ON *.* TO 'iamsecure'@'appclient' IDENTIFIED BY 'dingdingding' REQUIRE
SSL;
FLUSH PRIVILEGES;
# quit mysql
```

/ root / certs / mysql◦ ◦

```
openssl req -sha1 -newkey rsa:2048 -days 730 -nodes -keyout client-key.pem > client-req.pem
openssl rsa -in client-key.pem -out client-key.pem
openssl x509 -sha1 -req -in client-req.pem -days 730 -CA ca-cert.pem -CAkey ca-key.pem -
set_serial 01 > client-cert.pem
```

◦ ◦

/root/certs/mysql/

CA

```
cat server-cert.pem client-cert.pem > ca.pem
```

```
cat ca.pem
```

◦

```
ssh appclient
```

```
mkdir /etc/pki/tls/certs/mysql/
```

dbserverappclient ◦ scp ◦

```
scp dbserver  
# copy files from dbserver to appclient  
# exit scp
```

◦ mysql ◦

```
chown -R mysql:mysql /etc/pki/tls/certs/mysql
```

mysql

```
/etc/pki/tls/certs/mysql/ca.pem  
/etc/pki/tls/certs/mysql/client-cert.pem  
/etc/pki/tls/certs/mysql/client-key.pem
```

[client] appclientMariaDB / MySQL ◦

```
vi /etc/my.cnf  
# i  
[client]  
ssl-ca=/etc/pki/tls/certs/mysql/ca.pem  
ssl-cert=/etc/pki/tls/certs/mysql/client-cert.pem  
ssl-key=/etc/pki/tls/certs/mysql/client-key.pem  
# :wq
```

appclientmariadb

```
systemctl restart mariadb
```

ssl TRUE

```
mysql --ssl --help
```

appclientmysql

```
mysql -uroot -p
```

YES

```
show variables LIKE '%ssl';
have_openssl      YES
have_ssl          YES
```

```
have_openssl NO
```

mariadb.log

SSL'/etc/pki/tls/certs/mysql/client-cert.pem'

rootclient-cert.pem◦ / etc / pki / tls / certs / mysql /mysql◦

```
chown -R mysql:mysql /etc/pki/tls/certs/mysql
```

mariadb

appclient

dbservermysql◦

```
mysql -h dbserver -u iamsecure -p
# enter password dingdingding (hopefully you changed that to something else)
```

◦

SSLMariaDB / MySQL

```
\s
```

```
Connection id:          4
Current database:
Current user:           iamsecure@appclient
SSL:                   Cipher in use is DHE-RSA-AES256-GCM-SHA384
Current pager:         stdout
Using outfile:         ''
Using delimiter:       ;
```



```
Server:           MariaDB
Server version:   5.X.X-MariaDB MariaDB Server
Protocol version: 10
Connection:       dbserver via TCP/IP
Server charsetset: latin1
Db charsetset:    latin1
Client charsetset: utf8
Conn. charsetset: utf8
TCP port:         3306
Uptime:          42 min 13 sec
```

GRANT'。

SSL。

RHEL7CentOS7。。

。

SSL <https://riptutorial.com/zh-TW/mysql/topic/7563/ssl>

12: TRIGGERS

- CREATE [DEFINER = {user | CURRENT_USER}] TRIGGER trigger_name trigger_time trigger_event ON tbl_name FOR EACH ROW [trigger_order] trigger_body
- trigger_time{BEFORE |}
- trigger_event{INSERT |}
- trigger_order{FOLLOWS | PRECEDES} other_trigger_name

DB

FOR EACH ROW

Oracle

MySQLCREATE OR REPLACE

MySQL

```
DELIMITER $$

DROP TRIGGER IF EXISTS myTrigger;
$$
CREATE TRIGGER myTrigger
-- ...

$$
DELIMITER ;
```

- CREATE
- DROPCREATELOCK TABLES myTable WRITE;UNLOCK TABLES;CREATE

Examples

```
mysql> CREATE TABLE account (acct_num INT, amount DECIMAL(10,2));
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> CREATE TRIGGER ins_sum BEFORE INSERT ON account
-> FOR EACH ROW SET @sum = @sum + NEW.amount;
Query OK, 0 rows affected (0.06 sec)
```

CREATE TRIGGERins_sum

@sumINSERT

```
mysql> SET @sum = 0;
mysql> INSERT INTO account VALUES (137,14.98), (141,1937.50), (97,-100.00);
```

```
mysql> SELECT @sum AS 'Total amount inserted';
+-----+
| Total amount inserted |
+-----+
| 1852.48                |
+-----+
```

INSERT@sum14.98 + 1937.50 - 1001852.48。

```
mysql> DROP TRIGGER test.ins_sum;
```

。

- BEFOREBEFORE
- AFTERAFTER。

- INSERT
- UPDATE
- DELETE

Insert

```
DELIMITER $$

CREATE TRIGGER insert_date
  BEFORE INSERT ON stack
  FOR EACH ROW
BEGIN
  -- set the insert_date field in the request before the insert
  SET NEW.insert_date = NOW();
END;

$$
DELIMITER ;
```

Update

```
DELIMITER $$

CREATE TRIGGER update_date
  BEFORE UPDATE ON stack
  FOR EACH ROW
BEGIN
  -- set the update_date field in the request before the update
  SET NEW.update_date = NOW();
END;

$$
DELIMITER ;
```

```
DELIMITER $$

CREATE TRIGGER deletion_date
  AFTER DELETE ON stack
  FOR EACH ROW
BEGIN
  -- add a log entry after a successful delete
  INSERT INTO log_action(stack_id, deleted_date) VALUES(OLD.id, NOW());
END;

$$
DELIMITER ;
```

TRIGGERS <https://riptutorial.com/zh-TW/mysql/topic/3069/triggers>

13: UPDATE

- UPDATE [LOW_PRIORITY] [IGNORE] tableName SET column1 = expression1column2 = expression2... [WHERE conditions]; //
- UPDATE [LOW_PRIORITY] [IGNORE] tableName SET column1 = expression1column2 = expression2... [WHERE conditions] [ORDER BY[ASC | DESC]] [LIMIT row_count]; //order by limit
- UPDATE [LOW_PRIORITY] [IGNORE] table1table2... SET column1 = expression1column2 = expression2... [WHERE conditions]; //

Examples

```
UPDATE customers SET email='luke_smith@email.com' WHERE id=1
```

customersemailluke_smith@email.com id1.

customers			
id	firstname	lastname	email
1	Luke	Smith	luke@example.com
2	Anna	Carey	anna@example.com
3	Todd	Winters	todd@example.com

customers			
id	firstname	lastname	email
1	Luke	Smith	luke_smith@email.com
2	Anna	Carey	anna@example.com
3	Todd	Winters	todd@example.com

```
UPDATE customers SET lastname='smith'
```

customerslastname◦

customers			
id	firstname	lastname	email
1	Luke	Smith	luke@example.com
2	Anna	Carey	anna@example.com
3	Todd	Winters	todd@example.com

customers			
id	firstname	lastname	email
1	Luke	Smith	luke@example.com
2	Anna	Smith	anna@example.com
3	Todd	Smith	todd@example.com

UPDATEWHERE◦ ◦ customerslastnameSmith◦

Join Pattern

questions_mysql iwtQuestions LOAD DATA INFILECSV◦ ◦

◦

```
UPDATE questions_mysql q -- our real table for production
join iwtQuestions i -- imported worktable
ON i.qId = q.qId
```

```
SET q.closeVotes = i.closeVotes,  
q.votes = i.votes,  
q.answers = i.answers,  
q.views = i.views;
```

qi° °

qId **StackoverflowID**° °

ORDER BYLIMIT

SQLORDER BY ORDER BY°

SQLLIMIT° LIMIT°

ORDER BYLIMIT°

ORDER BYLIMIT**MySQL** UPDATE

```
UPDATE [ LOW_PRIORITY ] [ IGNORE ]  
tableName  
SET column1 = expression1,  
    column2 = expression2,  
    ...  
[WHERE conditions]  
[ORDER BY expression [ ASC | DESC ]]  
[LIMIT row_count];  
  
---> Example  
UPDATE employees SET isConfirmed=1 ORDER BY joiningDate LIMIT 10
```

joiningDate**10**°

UPDATE ° °

UPDATE ORDER BYLIMIT °

UPDATE

```
UPDATE [LOW_PRIORITY] [IGNORE]  
table1, table2, ...  
    SET column1 = expression1,  
        column2 = expression2,  
        ...  
[WHERE conditions]
```

productssalesOrders ° ° products° **SQL**°

```
UPDATE products, salesOrders  
    SET salesOrders.Quantity = salesOrders.Quantity - 5,  
        products.availableStock = products.availableStock + 5  
WHERE products.productId = salesOrders.productId  
    AND salesOrders.orderId = 100 AND salesOrders.productId = 20;
```

“5”salesOrdersWHEREproducts◦

◦

```
UPDATE people
SET name =
  (CASE id WHEN 1 THEN 'Karl'
        WHEN 2 THEN 'Tom'
        WHEN 3 THEN 'Mary'
        END)
WHERE id IN (1,2,3);
```

◦ WHERE◦

UPDATE <https://riptutorial.com/zh-TW/mysql/topic/2738/update>

14:

1M。

1M1M'one"many"many'""。

1M 。

- EMP_ID。 EMP_ID。
- EMPLOYEESMGR_ID。 。

Examples

◦

EMP_ID		MGR_ID
E01		M02
E02	Macklemore	M01
E03		M03
E04	Sonswan	M01

MGR_ID		
M01		
M02		
M03		

```
SELECT e.emp_id , e.first_name , e.last_name FROM employees e INNER JOIN managers m ON m.mgr_id = e.mgr_id WHERE m.mgr_id = 'M01' ;
```

EMP_ID	
E02	Macklemore
E04	Sonswan

1。

◦


```
SELECT m.mgr_id , m.first_name , m.last_name FROM managers m INNER JOIN employees e ON e.mgr_id = m.mgr_id WHERE e.emp_id = 'E03' ;
```

MGR_ID
M03

◦

<https://riptutorial.com/zh-TW/mysql/topic/9600/>

15:

Examples

SQLselectinsertupdatedelete.

- ◦
- ◦ SQL
- 1.
- 2.
- 3.

ACID

- ;◦
- ◦
- ◦
- ◦

START TRANSACTIONBEGIN WORKCOMMITROLLBACK◦ SQL◦

```
START TRANSACTION;
SET @transAmt = '500';
SELECT @availableAmt:=ledgerAmt FROM accTable WHERE customerId=1 FOR UPDATE;
UPDATE accTable SET ledgerAmt=ledgerAmt-@transAmt WHERE customerId=1;
UPDATE accTable SET ledgerAmt=ledgerAmt+@transAmt WHERE customerId=2;
COMMIT;
```

START TRANSACTION COMMITROLLBACK◦ ◦

FOR UPDATE◦

-
- SQLBEGIN WORKSTART TRANSACTION ◦
- SQL◦
- ◦
- COMMITROLLBACK◦
- Galera / PXC◦

COMMITROLLBACKAUTOCOMMIT

AUTOCOMMIT

MySQL◦ UPDATE DELETEINSERTBEGINSTART TRANSACTION◦

AUTOCOMMIT *true*◦

```
--->To make autocommit false
SET AUTOCOMMIT=false;
--or
SET AUTOCOMMIT=0;

--->To make autocommit true
SET AUTOCOMMIT=true;
--or
SET AUTOCOMMIT=1;
```

AUTOCOMMIT

```
SELECT @@autocommit;
```

AUTOCOMMIT *false*◦

COMMIT◦

1

```
--->Before making autocommit false one row added in a new table
mysql> INSERT INTO testTable VALUES (1);

--->Making autocommit = false
mysql> SET autocommit=0;

mysql> INSERT INTO testTable VALUES (2), (3);
mysql> SELECT * FROM testTable;
+-----+
| tId |
+-----+
|  1  |
|  2  |
|  3  |
+-----+
```

2

```
mysql> SELECT * FROM testTable;
+-----+
| tId |
+-----+
|  1  |
+-----+
---> Row inserted before autocommit=false only visible here
```

1

```
mysql> COMMIT;
--->Now COMMIT is executed in connection 1
```

```
mysql> SELECT * FROM testTable;
+-----+
| tId |
+-----+
|  1 |
|  2 |
|  3 |
+-----+
```

2

```
mysql> SELECT * FROM testTable;
+-----+
| tId |
+-----+
|  1 |
|  2 |
|  3 |
+-----+
--->Now all the three rows are visible here
```

ROLLBACK

ROLLBACK^o

```
--->Before making autocommit false one row added in a new table
mysql> INSERT INTO testTable VALUES (1);
```

```
--->Making autocommit = false
mysql> SET autocommit=0;
```

```
mysql> INSERT INTO testTable VALUES (2), (3);
mysql> SELECT * FROM testTable;
+-----+
| tId |
+-----+
|  1 |
|  2 |
|  3 |
+-----+
```

ROLLBACK

```
--->Rollback executed now
mysql> ROLLBACK;

mysql> SELECT * FROM testTable;
+-----+
| tId |
+-----+
|  1 |
+-----+
--->Rollback removed all rows which all are not committed
```

COMMIT ROLLBACK

```

mysql> INSERT INTO testTable VALUES (2), (3);
mysql> SELECT * FROM testTable;
mysql> COMMIT;
+-----+
| tId |
+-----+
|  1 |
|  2 |
|  3 |
+-----+

--->Rollback executed now
mysql> ROLLBACK;

mysql> SELECT * FROM testTable;
+-----+
| tId |
+-----+
|  1 |
|  2 |
|  3 |
+-----+

--->Rollback not removed any rows

```

AUTOCOMMIT *true* COMMITROLLBACK

JDBC

JDBC。JDBCMySQL

MySQLJDBC

JDBC

```

Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection(DB_CONNECTION_URL,DB_USER,USER_PASSWORD);
--->Example for connection url "jdbc:mysql://localhost:3306/testDB";

```

SQL。。

。

```

jdbc:mysql://localhost:3306/testDB?useUnicode=true&characterEncoding=utf8

```

AUTOCOMMIT *true false*。

```

con.setAutoCommit(false);

```

setAutoCommit()。

START TRANSACTIONBEGIN WORK。 START TRANSACTIONBEGIN WORK AUTOCOMMIT *false* 。

。 JDBC。

```

package jdbcTest;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class accTrans {

    public static void doTransfer(double transAmount,int customerIdFrom,int customerIdTo) {

        Connection con = null;
        PreparedStatement pstmt = null;
        ResultSet rs = null;

        try {
            String DB_CONNECTION_URL =
"jdbc:mysql://localhost:3306/testDB?useUnicode=true&characterEncoding=utf8";

            Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection(DB_CONNECTION_URL,DB_USER,USER_PASSWORD);

            --->set auto commit to false
            con.setAutoCommit(false);
            ---> or use con.START TRANSACTION / con.BEGIN WORK

            --->Start SQL Statements for transaction
            --->Checking availability of amount
            double availableAmt    = 0;
            pstmt = con.prepareStatement("SELECT ledgerAmt FROM accTable WHERE customerId=?
FOR UPDATE");
            pstmt.setInt(1, customerIdFrom);
            rs = pstmt.executeQuery();
            if(rs.next())
                availableAmt    = rs.getDouble(1);

            if(availableAmt >= transAmount)
            {
                ---> Do Transfer
                ---> taking amount from cutomerIdFrom
                pstmt = con.prepareStatement("UPDATE accTable SET ledgerAmt=ledgerAmt-? WHERE
customerId=?");
                pstmt.setDouble(1, transAmount);
                pstmt.setInt(2, customerIdFrom);
                pstmt.executeUpdate();

                ---> depositing amount in cutomerIdTo
                pstmt = con.prepareStatement("UPDATE accTable SET ledgerAmt=ledgerAmt+? WHERE
customerId=?");
                pstmt.setDouble(1, transAmount);
                pstmt.setInt(2, customerIdTo);
                pstmt.executeUpdate();

                con.commit();
            }
            --->If you performed any insert,update or delete operations before
            ----> this availability check, then include this else part
            /*else { --->Rollback the transaction if availability is less than required
                con.rollback();
            }*/
        }
    }
}

```

```
    } catch (SQLException ex) {
        --> Rollback the transaction in case of any error
        con.rollback();
    } finally {
        try {
            if(rs != null) rs.close();
            if(pstmt != null) pstmt.close();
            if(con != null) con.close();
        }
    }
}

public static void main(String[] args) {
    doTransfer(500, 1020, 1021);
    -->doTransfer(transAmount, customerIdFrom, customerIdTo);
}
}
```

JDBCSQLSQL。

<https://riptutorial.com/zh-TW/mysql/topic/5771/>

16: Docker-ComposeMysql

Examples

docker-compose

dockermysql

1.-docker-compose.yml

HOME_PATHPATH。 **docker。**

```
version: '2'
services:
  cabin_db:
    image: mysql:latest
    volumes:
      - "./.mysql-data/db:/var/lib/mysql"
    restart: always
    ports:
      - 3306:3306
    environment:
      MYSQL_ROOT_PASSWORD: rootpw
      MYSQL_DATABASE: cabin
      MYSQL_USER: cabin
      MYSQL_PASSWORD: cabinpw
```

2.-

```
cd PATH_TO_DOCKER-COMPOSE.YML
docker-compose up -d
```

3.-

```
mysql -h 127.0.0.1 -u root -P 3306 -p rootpw
```

4.-

```
docker-compose stop
```

Docker-ComposeMysql <https://riptutorial.com/zh-TW/mysql/topic/4458/docker-composemysql>

17: mysqldump

- `mysqldump -u [username] -p [password] [] db_name> dumpFileName.sql ///`
- `mysqldump -u [username] -p [password] [] db_name [tbl_name1 tbl_name2 tbl_name2 ...]> dumpFileName.sql ///`
- `mysqldump -u [username] -p [password] [] --databases db_name1 db_name2 db_name3 ...> dumpFileName.sql ///`
- `mysqldump -u [username] -p [password] [] --all-databases> dumpFileName.sql ///MySQL`

-	#Server
-h -- --host	IP◦ localhost 127.0.0.1 -h localhost
-u -- --user	MySQL
-p - --password	MySQL◦ -p ◦ -pMyPassword
-	
--add-drop-database	CREATE DATABASEDROP DATABASE CREATE DATABASE◦ ◦
--add-drop-table	CREATE TABLEDROP TABLE CREATE TABLE◦ ◦
--no-create-db	CREATE DATABASE◦ ◦
-t -- --no-create-info	CREATE TABLE◦ /◦
-d -- --no-data	◦ CREATE TABLE◦ “”
-R - --routines	/◦
-K - --disable-keys	◦ MyISAM◦

`mysqldump MySQLSQL◦ mysqldump ed◦`

- DROP
- CREATE
- --no-data
 - LOCK
 - INSERT
- UNLOCK TABLES
-
- DROP
- CREATE
-

`CREATEDROPmysqldump◦`

Examples

```
mysqldump [options] db_name > filename.sql
```

```
mysqldump [options] --databases db_name1 db_name2 ... > filename.sql  
mysqldump [options] --all-databases > filename.sql
```

```
mysqldump [options] db_name table_name... > filename.sql
```

```
mysqldump [options] db_name --ignore-table=tbl1 --ignore-table=tbl2 ... > filename.sql
```

.sql° °

```
> mysqldump -u username -p [other options]  
Enter password:
```

-p

```
> mysqldump -u username -ppassword [other options]
```

shell /°

```
> mysqldump --user=username --password=password [other options]
```

°

```
mysql [options] db_name < filename.sql
```

- db_name;
- filename.sql;
- .sql° °
- ° filename.sql°

MySQL source

```
source filename.sql
```

```
\. filename.sql
```

mysqldump

--compress--compressmysqldump °

```
mysqldump -h db.example.com -u username -p --compress dbname > dbname.sql
```

--lock-tables=false ◦ ◦

gzip ◦

```
mysqldump -h db.example.com -u username -p --compress dbname | gzip --stdout > dbname.sql.gz
```

gzipmysqldump

```
gunzip -c dbname.sql.gz | mysql dbname -u username -p
```

-cstdout◦

Amazon S3

MySqlAmazon S3◦ DB

```
mysqldump -u root -p --host=localhost --opt --skip-lock-tables --single-transaction \  
--verbose --hex-blob --routines --triggers --all-databases |  
gzip -9 | s3cmd put - s3://s3-bucket/db-server-name.sql.gz
```

◦

MySQLMySQL

1

- 1.
- 2.
- 3.

```
mysqldump [options] > dump.sql
```

```
mysql [options] < dump.sql
```

2

```
mysqldump [options to connect to the source server] | mysql [options]
```

◦ 1◦

mysqldump--routines -R

```
mysqldump -u username -p -R db_name > dump.sql
```

--routinesmysql.proc◦

[mysqldump https://riptutorial.com/zh-TW/mysql/topic/604/mysqldump](https://riptutorial.com/zh-TW/mysql/topic/604/mysqldump)

18: UTF-8。

Examples

12utf8

```
# -*- coding: utf-8 -*-
```

```
db = MySQLdb.connect(host=DB_HOST, user=DB_USER, passwd=DB_PASS, db=DB_NAME,  
    charset="utf8mb4", use_unicode=True)
```

```
<meta charset="utf-8" />  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

PHP

php.iniPHP 5.6

```
default_charset UTF-8
```

```
header('Content-type: text/plain; charset=UTF-8');
```

MySQL

```
(for mysql:)    Do not use the mysql_* API!  
(for mysqli:)  $mysqli_obj->set_charset('utf8mb4');  
(for PDO:)     $db = new PDO('dblib:host=host;dbname=db;charset=utf8', $user, $pwd);
```

。

```
<form accept-charset="UTF-8">
```

JSON\uxxxx

```
$t = json_encode($s, JSON_UNESCAPED_UNICODE);
```

UTF-8。 <https://riptutorial.com/zh-TW/mysql/topic/7332/utf-8->

19: Un-Pivot

Examples

BI /。

。

```
create table rawdata
(
  PersonId VARCHAR(255)
,Question1Id INT(11)
,Question2Id INT(11)
,Question3Id INT(11)
)
```

rawdataETL。 。

rawdata

	PersonId	Question1Id	Question2Id	Question3Id
	Giannaros	1	3	1
	Patra	2	4	3

MYSQLUNION ALL

```
create table unpivoteddata
(
  PersonId VARCHAR(255)
,QuestionId VARCHAR(255)
,QuestionValue INT(11)
);

INSERT INTO unpivoteddata SELECT PersonId, 'Question1Id' col, Question1Id
FROM rawdata
UNION ALL
SELECT PersonId, 'Question2Id' col, Question2Id
FROM rawdata
UNION ALL
SELECT PersonId, 'Question3Id' col, Question3Id
FROM rawdata;
```

QuestionId。 。

GROUP_CONCAT“int”。 GROUP_CONCATSELECT。

```
set @temp2 = null;

SELECT GROUP_CONCAT(' SELECT ', 'PersonId',' ','',COLUMN_NAME,'', ' col
',' ',COLUMN_NAME,' FROM rawdata' separator ' UNION ALL' ) FROM INFORMATION_SCHEMA.COLUMNS
```

```
WHERE table_name = 'rawdata' AND DATA_TYPE = 'Int' INTO @temp2;

select @temp2;
```

```
DATA_TYPE = 'Int'
```

```
COLUMN_NAME LIKE 'Question%'
```

ETL。

```
set @temp3 = null;

select concat('INSERT INTO unpivoteddata',@temp2) INTO @temp3;

select @temp3;

prepare stmt FROM @temp3;
execute stmt;
deallocate prepare stmt;
```

unpivoteddata

```
SELECT * FROM unpivoteddata
```

PersonId	QuestionId	QuestionValue
Giannaros	Question1Id	1
Patra	Question1Id	2
Giannaros	Question2Id	3
Patra	Question2Id	4
Giannaros	Question3Id	1
Patra	Question3Id	3

。

Un-Pivot <https://riptutorial.com/zh-TW/mysql/topic/6491/un-pivot>

20:

Examples

1. SET

```
EXSET @var_string='my_var'; SET @var_num='2' SET @var_date='2015-07-20';
```

2. =select

```
EX@var='123'; =SETselectupdate ...“=”“...”“”SET“。
```

3. INTOselect

```
EXSET @start_date='2015-07-20'; SET @end_date='2016-01-31';
```

```
#this gets the year month value to use as the partition names
SET @start_yearmonth = (SELECT EXTRACT(YEAR_MONTH FROM @start_date));
SET @end_yearmonth = (SELECT EXTRACT(YEAR_MONTH FROM @end_date));

#put the partitions into a variable
SELECT GROUP_CONCAT(partition_name)
FROM information_schema.partitions p
WHERE table_name = 'partitioned_table'
AND SUBSTRING_INDEX(partition_name, 'P', -1) BETWEEN @start_yearmonth AND @end_yearmonth
INTO @partitions;

#put the query in a variable. You need to do this, because mysql did not recognize my variable
as a variable in that position. You need to concat the value of the variable together with the
rest of the query and then execute it as a stmt.
SET @query =
CONCAT('CREATE TABLE part_of_partitioned_table (PRIMARY KEY(id))
SELECT partitioned_table.*
FROM partitioned_table PARTITION(', @partitions,')
JOIN users u USING(user_id)
WHERE date(partitioned_table.date) BETWEEN ', @start_date, ' AND ', @end_date);

#prepare the statement from @query
PREPARE stmt FROM @query;
#drop table
DROP TABLE IF EXISTS tech.part_of_partitioned_table;
#create table using statement
EXECUTE stmt;
```

Select

table team_person

```
+=====+=====+
| team | person |
+=====+=====+
| A | John |
+-----+-----+
```

```

| B | Smith |
+-----+-----+
| A | Walter |
+-----+-----+
| A | Louis |
+-----+-----+
| C | Elizabeth |
+-----+-----+
| B | Wayne |
+-----+-----+

```

```

CREATE TABLE team_person AS SELECT 'A' team, 'John' person
UNION ALL SELECT 'B' team, 'Smith' person
UNION ALL SELECT 'A' team, 'Walter' person
UNION ALL SELECT 'A' team, 'Louis' person
UNION ALL SELECT 'C' team, 'Elizabeth' person
UNION ALL SELECT 'B' team, 'Wayne' person;

```

row_numberteam_person

```

SELECT @row_no := @row_no+1 AS row_number, team, person
FROM team_person, (SELECT @row_no := 0) t;

```

```

SET @row_no := 0;
SELECT @row_no := @row_no + 1 AS row_number, team, person
FROM team_person;

```

```

+-----+-----+-----+
| row_number | team | person |
+-----+-----+-----+
| 1 | A | John |
+-----+-----+-----+
| 2 | B | Smith |
+-----+-----+-----+
| 3 | A | Walter |
+-----+-----+-----+
| 4 | A | Louis |
+-----+-----+-----+
| 5 | C | Elizabeth |
+-----+-----+-----+
| 6 | B | Wayne |
+-----+-----+-----+

```

teamrow_number

```

SELECT @row_no := IF(@prev_val = t.team, @row_no + 1, 1) AS row_number
, @prev_val := t.team AS team
, t.person
FROM team_person t,
(SELECT @row_no := 0) x,
(SELECT @prev_val := '') y
ORDER BY t.team ASC, t.person DESC;

```

```

+-----+-----+-----+
| row_number | team | person |
+-----+-----+-----+

```



```
|          1 | A | Walter |
+-----+-----+-----+
|          2 | A | Louis  |
+-----+-----+-----+
|          3 | A | John   |
+-----+-----+-----+
|          1 | B | Wayne  |
+-----+-----+-----+
|          2 | B | Smith  |
+-----+-----+-----+
|          1 | C | Elizabeth |
+-----+-----+-----+
```

<https://riptutorial.com/zh-TW/mysql/topic/5013/>

21:

MySQL。

。

-
-
-
-
-
-
-
-
-
-
- ASC
- ASENSITIVE
-
-
- BIGINT
- BINARY
- BLOB
-
-
-
-
-
-
-
- CHAR
-
-
- COLLATE
-
-
-
-
-
-
-
-
-
-
- CURRENT_TIMESTAMP
-
-
-
-
- DATABASES
- DAY_HOUR
- DAY_MICROSECOND
- DAY_MINUTE
- DAY_SECOND
- DEC
- DECIMAL
-

-
-
-
- DESC
-
-
-
- DISTINCTROW
- DIV
-
-
-
-
-
- ELSEIF
- ENCLOSED
- ESCAPED
- EXISTS
-
-
-
-
-
-
- FLOAT4
- float8
-
-
-
-
-
-
- GENERATED
-
- GRANT
-
- HAVING
- HIGH_PRIORITY
- HOUR_MICROSECOND
- HOUR_MINUTE
- HOUR_SECOND
-
-
-
-
- INFILE
-
-
- INSENSITIVE
-
- INT
- INT1
- INT2

- INT3
- INT4
- INT8
-
-
- INTO
- IO_AFTER_GTIDS
- IO_BEFORE_GTIDS
- IS
-
-
-
- KEYS
-
-
-
-
-
-
- LINEAR
- LINES
-
-
- LOCALTIMESTAMP
-
-
- LONGBLOB
- LONGTEXT
-
-
- MASTER_BIND
- MASTER_SSL_VERIFY_SERVER_CERT
-
- MAXVALUE
- MEDIUMBLOB
- MEDIUMINT
- MEDIUMTEXT
- MIDDLEINT
- MINUTE_MICROSECOND
- MINUTE_SECOND
- MOD
- MODIFIES
-
-
- NO_WRITE_TO_BINLOG
-
-
-
- OPTIMIZE
- OPTIMIZER_COSTS
-

- OPTIONALLY
-
-
- OUT
-
- OUTFILE
-
-
-
-
-
-
-
-
-
-
- READS
- READ_WRITE
-
-
- REGEXP
-
-
-
-
-
-
- RESIGNAL
-
-
-
-
-
- RLIKE
- SCHEMA
- SCHEMAS
- SECOND_MICROSECOND
-
-
-
-
-
-
-
- SMALLINT
-
-
- SQL
- SQLEXCEPTION
- SQLSTATE
- SQLWARNING
- SQL_BIG_RESULT
- SQL_CALC_FOUND_ROWS
- SQL_SMALL_RESULT
- SSL
-
- STORED

- STRAIGHT_JOIN
-
- TERMINATED
-
- TINYBLOB
- TINYINT
- TINYTEXT
-
- TRAILING
-
-
- UNDO
-
-
-
- UNSIGNED
- UPDATE
-
-
-
- UTC_DATE
- UTC_TIME
- UTC_TIMESTAMP
- VALUES
- VARBINARY
- VARCHAR
- VARCHARACTER
-
-
-
-
-
-
- WITH
-
- XOR
- YEAR_MONTH
- ZEROFILL
- GENERATED
- OPTIMIZER_COSTS
- STORED
-

Examples

order

```
select * from order
```

1064。 SQL;MySQL1“order”

MySQL `

```
select * from `order`
```

。

MySQL 。

<https://riptutorial.com/zh-TW/mysql/topic/1398/>

22:

MySQL FULLTEXT

FULLTEXT MySQL

FULLTEXT WHERE column LIKE 'text%'

FULLTEXT MyISAM MySQL 5.6.4 InnoDB

Examples

FULLTEXT

```
SET @searchTerm= 'Database Programming';
SELECT MATCH (Title) AGAINST (@searchTerm IN NATURAL LANGUAGE MODE) Score,
       ISBN, Author, Title
FROM book
WHERE MATCH (Title) AGAINST (@searchTerm IN NATURAL LANGUAGE MODE)
ORDER BY MATCH (Title) AGAINST (@searchTerm IN NATURAL LANGUAGE MODE) DESC;
```

book ISBN 'Title' Author 'Database Programming'

Title

```
ALTER TABLE book ADD FULLTEXT INDEX Fulltext_title_index (Title);
```

BOOLEAN

```
SET @searchTerm= 'Database Programming -Java';
SELECT MATCH (Title) AGAINST (@searchTerm IN BOOLEAN MODE) Score,
       ISBN, Author, Title
FROM book
WHERE MATCH (Title) AGAINST (@searchTerm IN BOOLEAN MODE)
ORDER BY MATCH (Title) AGAINST (@searchTerm IN BOOLEAN MODE) DESC;
```

book ISBN Title Author 'Database' 'Programming' 'Database' 'Java'

Title

```
ALTER TABLE book ADD FULLTEXT INDEX Fulltext_title_index (Title);
```

FULLTEXT

```
SET @searchTerm= 'Date Database Programming';
SELECT MATCH (Title, Author) AGAINST (@searchTerm IN NATURAL LANGUAGE MODE) Score,
       ISBN, Author, Title
```



```
FROM book
WHERE MATCH (Title, Author) AGAINST (@searchTerm IN NATURAL LANGUAGE MODE)
ORDER BY MATCH (Title, Author) AGAINST (@searchTerm IN NATURAL LANGUAGE MODE) DESC;
```

book ISBN Title Author 'Date Database Programming'。 。 CJ Date。

The Date Doctor。 FULLTEXT。

Title Author

```
ALTER TABLE book ADD FULLTEXT INDEX Fulltext_title_author_index (Title, Author);
```

<https://riptutorial.com/zh-TW/mysql/topic/8759/>

23:

- **RANGE** ◦ ◦
- ◦ RANGE◦
- **HASH** ◦ ◦ MySQL◦ LINEAR HASH ◦
- **KEY** ◦ HASHMySQL◦ MySQL◦ LINEAR KEY ◦

Examples

- VALUES LESS THAN◦ 20120

```
CREATE TABLE employees (  
  id INT NOT NULL,  
  fname VARCHAR(30),  
  lname VARCHAR(30),  
  hired DATE NOT NULL DEFAULT '1970-01-01',  
  separated DATE NOT NULL DEFAULT '9999-12-31',  
  job_code INT NOT NULL,  
  store_id INT NOT NULL  
);
```

- store_id◦ PARTITION BY RANGE⁴PARTITION BY RANGE

```
ALTER TABLE employees PARTITION BY RANGE (store_id) (  
  PARTITION p0 VALUES LESS THAN (6),  
  PARTITION p1 VALUES LESS THAN (11),  
  PARTITION p2 VALUES LESS THAN (16),  
  PARTITION p3 VALUES LESS THAN MAXVALUE  
);
```

MAXVALUE◦

MySQL ◦

- **RANGE**◦ ◦ ◦ PARTITION BY LIST(expr)exprVALUES IN (value_list)value_list◦

CREATE TABLE

```
CREATE TABLE employees (  
  id INT NOT NULL,  
  fname VARCHAR(30),  
  lname VARCHAR(30),  
  hired DATE NOT NULL DEFAULT '1970-01-01',  
  separated DATE NOT NULL DEFAULT '9999-12-31',  
  job_code INT,  
  store_id INT  
);
```

420。

ID
3,5,6,9,17
1,2,10,11,19,20
4,12,13,14,18
7,8,15,16

```
ALTER TABLE employees PARTITION BY LIST(store_id) (  
    PARTITION pNorth VALUES IN (3,5,6,9,17),  
    PARTITION pEast VALUES IN (1,2,10,11,19,20),  
    PARTITION pWest VALUES IN (4,12,13,14,18),  
    PARTITION pCentral VALUES IN (7,8,15,16)  
);
```

MySQL。

HASH

HASH。;MySQL。

store_id4

```
CREATE TABLE employees (  
    id INT NOT NULL,  
    fname VARCHAR(30),  
    lname VARCHAR(30),  
    hired DATE NOT NULL DEFAULT '1970-01-01',  
    separated DATE NOT NULL DEFAULT '9999-12-31',  
    job_code INT,  
    store_id INT  
)  
PARTITION BY HASH(store_id)  
PARTITIONS 4;
```

PARTITIONS1。

MySQL。

<https://riptutorial.com/zh-TW/mysql/topic/5128/>

24:

- [LOW_PRIORITY] [QUICK] [IGNORE] FROM[WHERE] [ORDER BY[ASC | DESC]] [LIMIT number_rows]; ///

	LOW_PRIORITY
	IGNORE
	◦
ORDER BY	ORDER BY
	◦ number_rows◦

Examples

Where

```
DELETE FROM `table_name` WHERE `field_one` = 'value_one'
```

field_one'value_one'

WHEREselect> < <>LIKE◦

WHERELIKE◦ ◦

```
DELETE FROM table_name ;
```

◦ ◦ DELETEWHERE◦

```
DELETE FROM `table_name` WHERE `field_one` = 'value_one' LIMIT 1
```

“Where”。

◦ ◦

MySQLDELETEJOIN◦ ◦

```
create table people
(
  id int primary key,
  name varchar(100) not null,
  gender char(1) not null
);
```

```

insert people (id,name,gender) values
(1,'Kathy','f'), (2,'John','m'), (3,'Paul','m'), (4,'Kim','f');

create table pets
(
  id int auto_increment primary key,
  ownerId int not null,
  name varchar(100) not null,
  color varchar(100) not null
);
insert pets(ownerId,name,color) values
(1,'Rover','beige'), (2,'Bubbles','purple'), (3,'Spot','black and white'),
(1,'Rover2','white');

```

ID		
1	F	
2		
3		
4	F	

ID	OWNERID		
1	1		
2	2		
4	1	Rover2	

```

DELETE p2
FROM pets p2
WHERE p2.ownerId in (
  SELECT p1.id
  FROM people p1
  WHERE p1.name = 'Paul');

```

```

DELETE p2 -- remove only rows from pets
FROM people p1
JOIN pets p2
ON p2.ownerId = p1.id
WHERE p1.name = 'Paul';

```

1 SpotPets

p1p2◦

```

DELETE p1, p2 -- remove rows from both tables
FROM people p1
JOIN pets p2
ON p2.ownerId = p1.id

```

```
WHERE p1.name = 'Paul';
```

2

SpotPets

DELETEforeing◦ pets

```
ALTER TABLE pets ADD CONSTRAINT `fk_pets_2_people` FOREIGN KEY (ownerId) references people(id)
ON DELETE CASCADE;
```

petspeople

```
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails
(`test`.`pets`, CONSTRAINT `pets_ibfk_1` FOREIGN KEY (`ownerId`) REFERENCES `people` (`id`))
```

peopleInnoDBON DELETE

```
DELETE FROM people
WHERE name = 'Paul';
```

2

Spot

```
SET foreign_key_checks = 0;
DELETE p1, p2 FROM people p1 JOIN pets p2 ON p2.ownerId = p1.id WHERE p1.name = 'Paul';
SET foreign_key_checks = 1;
```

```
DELETE FROM `myTable` WHERE `someColumn` = 'something'
```

WHERE◦

DELETETRUNCATE

```
TRUNCATE tableName;
```

AUTO_INCREMENT◦ DELETE FROM tableName◦ /◦

SQL Server◦ innodb_file_per_table=OFF◦

MySQL

```
-- remove only the employees
DELETE e
FROM Employees e JOIN Department d ON e.department_id = d.department_id
WHERE d.name = 'Sales'
```

```
-- remove employees and department
DELETE e, d
FROM Employees e JOIN Department d ON e.department_id = d.department_id
WHERE d.name = 'Sales'
```

```
-- remove from all tables (in this case same as previous)
DELETE
FROM Employees e JOIN Department d ON e.department_id = d.department_id
WHERE d.name = 'Sales'
```

<https://riptutorial.com/zh-TW/mysql/topic/1487/>

25:

- `{| SCHEMA} [IF NOT EXISTS] db_name [create_specification] ///`
- `DROP {DATABASE | SCHEMA} [IF EXISTS] db_name ///`

	CREATE DATABASE
create_specification	create_specificationCHARACTER SETCOLLATE

Examples

◦ SCHEMA ◦

```
CREATE DATABASE Baseball; -- creates a database named Baseball
```

1007 ◦

```
CREATE DATABASE IF NOT EXISTS Baseball;
```

```
DROP DATABASE IF EXISTS Baseball; -- Drops a database if it exists, avoids Error 1008  
DROP DATABASE xyz; -- If xyz does not exist, ERROR 1008 will occur
```

DDL IF EXISTS ◦

CHARACTER SET ◦

```
CREATE DATABASE Baseball CHARACTER SET utf8 COLLATE utf8_general_ci;
```

```
SHOW CREATE DATABASE Baseball;
```

```
+-----+-----+  
| Database | Create Database |  
+-----+-----+  
| Baseball | CREATE DATABASE `Baseball` /*!40100 DEFAULT CHARACTER SET utf8 */ |  
+-----+-----+
```

```
SHOW DATABASES;
```

```
+-----+  
| Database |  
+-----+  
| information_schema |  
| ajax_stuff |  
| Baseball |  
+-----+
```

```
USE Baseball; -- set it as the current database
```



```
SELECT @@character_set_database as cset, @@collation_database as col;
+-----+-----+
| cset | col |
+-----+-----+
| utf8 | utf8_general_ci |
+-----+-----+
```

CHARACTER SET Collation.

```
CREATE USER 'John123'@'%' IDENTIFIED BY 'OpenSesame';
```

John123%。 'OpenSesame'.

```
CREATE USER 'John456'@'%' IDENTIFIED BY 'somePassword';
```

mysql

```
SELECT user, host, password from mysql.user where user in ('John123', 'John456');
+-----+-----+-----+
| user | host | password |
+-----+-----+-----+
| John123 | % | *E6531C342ED87 ..... |
| John456 | % | *B04E11FAAAE9A ..... |
+-----+-----+-----+
```

Baseball.

。 John123BaseballSELECT

```
GRANT ALL ON Baseball.* TO 'John123'@'%;
GRANT SELECT ON Baseball.* TO 'John456'@'%;
```

```
SHOW GRANTS FOR 'John123'@'%;
```

```
+-----+
-----+
| Grants for John123@%
|
+-----+
-----+
| GRANT USAGE ON *.* TO 'John123'@'%' IDENTIFIED BY PASSWORD '*E6531C342ED87
..... |
| GRANT ALL PRIVILEGES ON `baseball`.* TO 'John123'@'%'
|
+-----+
-----+
```

```
SHOW GRANTS FOR 'John456'@'%;
```

```
+-----+
-----+
| Grants for John456@%
|
+-----+
-----+
| GRANT USAGE ON *.* TO 'John456'@'%' IDENTIFIED BY PASSWORD '*B04E11FAAAE9A
..... |
```

```
| GRANT SELECT ON `baseball`.* TO 'John456'@'%'  
|  
+-----+  
-----+
```

GRANT USAGE° °

MyDatabase

write° °

```
CREATE DATABASE my_db;  
USE my_db;  
CREATE TABLE some_table;  
INSERT INTO some_table ...;
```

my_db.some_table °

MySQL° SELECT INSERT / UPDATE / DELETE ° °

- mysql - GRANT°
- information_schema - ""° °
- performance_schema - ?? []
- MariaDBGaleraTokuDB

°

```
mysql> CREATE DATABASE menagerie;
```

UnixSQLmenagerieMenagerieMENAGERIE° ° Windows° ° °

;

```
mysql> USE menagerie  
Database changed
```

mysql° USE° mysql° °

```
shell> mysql -h host -u user -p menagerie  
Enter password: *****
```

<https://riptutorial.com/zh-TW/mysql/topic/600/>

26:

MySQL

```
SELECT User,Host FROM mysql.user;
```

Examples

MySQL

1 rootMySQL

```
$ mysql -u root -p
```

2mysql

```
mysql> CREATE USER 'my_new_user'@'localhost' IDENTIFIED BY 'test_password';
```

permissions permissions

```
mysql> GRANT ALL PRIVILEGES ON my_db.* TO 'my_new_user'@'localhost' identified by 'my_password';
```

```
mysql> CREATE USER 'my_new_user'@'localhost' IDENTIFIED BY 'test_password';
```

PASSWORDPASSWORD()

```
mysql> select PASSWORD('test_password'); -- returns *4414E26EDED6D661B5386813EBBA95065DBC4728
mysql> CREATE USER 'my_new_user'@'localhost' IDENTIFIED BY PASSWORD '*4414E26EDED6D661B5386813EBBA95065DBC4728';
```

```
grant all privileges on schema_name.* to 'new_user_name'@'%' identified by 'newpassword';
```

root

```
rename user 'user'@'%' to 'new_name'@'%';
```

<https://riptutorial.com/zh-TW/mysql/topic/3508/>

27:

- INNER OUTER ◦
- MySQL FULL ◦
- “commajoin” FROM a,b WHERE ax=by ;FROM a JOIN b ON ax=by ◦
- FROM a JOIN b ON ax = by ◦
- FROM LEFT JOIN b ON ax = by abNULLs ◦

Examples

db

```
CREATE TABLE `user` (  
  `id` smallint(5) unsigned NOT NULL AUTO_INCREMENT,  
  `name` varchar(30) NOT NULL,  
  `course` smallint(5) unsigned DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB;
```

```
CREATE TABLE `course` (  
  `id` smallint(5) unsigned NOT NULL AUTO_INCREMENT,  
  `name` varchar(50) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB;
```

InnoDBuser.coursecourse.id

```
ALTER TABLE `user`  
ADD CONSTRAINT `FK_course`  
FOREIGN KEY (`course`) REFERENCES `course` (`id`)  
ON UPDATE CASCADE;
```

```
SELECT user.name, course.name  
FROM `user`  
INNER JOIN `course` on user.course = course.id;
```

“”

```
SELECT x, ...  
  FROM ( SELECT y, ... FROM ... ) AS a  
  JOIN tbl ON tbl.x = a.y  
  WHERE ...
```

JOINtbl ◦

5.6.

```
SELECT ...
  FROM ( SELECT y, ... FROM ... ) AS a
  JOIN ( SELECT x, ... FROM ... ) AS b ON b.x = a.y
  WHERE ...
```

5.6. "。

```
SELECT
  @n := @n + 1,
  ...
FROM ( SELECT @n := 0 ) AS initialize
JOIN the_real_table
ORDER BY ...
```

CROSS JOIN ON ° the_real_table n the_real_table °

-

```
SELECT c.CustomerName, o.OrderID
  FROM Customers AS c
  INNER JOIN Orders AS o
    ON c.CustomerID = o.CustomerID
  ORDER BY c.CustomerName, o.OrderID;
```

```
SELECT c.CustomerName, COUNT(*) AS 'Order Count'
  FROM Customers AS c
  INNER JOIN Orders AS o
    ON c.CustomerID = o.CustomerID
  GROUP BY c.CustomerID;
  ORDER BY c.CustomerName;
```

```
SELECT c.CustomerName,
  ( SELECT COUNT(*) FROM Orders WHERE CustomerID = c.CustomerID ) AS 'Order Count'
  FROM Customers AS c
  ORDER BY c.CustomerName;
```

°

```
SELECT c.CustomerName,
  FROM Customers AS c
  WHERE EXISTS ( SELECT * FROM Orders WHERE CustomerID = c.CustomerID )
  ORDER BY c.CustomerName;
```

MySQL FULL OUTER JOIN °

```
-----
-- Table structure for `owners`
-----
DROP TABLE IF EXISTS `owners`;
CREATE TABLE `owners` (
```

```

`owner_id` int(11) NOT NULL AUTO_INCREMENT,
`owner` varchar(30) DEFAULT NULL,
PRIMARY KEY (`owner_id`)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;

-- -----
-- Records of owners
-- -----
INSERT INTO `owners` VALUES ('1', 'Ben');
INSERT INTO `owners` VALUES ('2', 'Jim');
INSERT INTO `owners` VALUES ('3', 'Harry');
INSERT INTO `owners` VALUES ('6', 'John');
INSERT INTO `owners` VALUES ('9', 'Ellie');
-- -----
-- Table structure for `tools`
-- -----
DROP TABLE IF EXISTS `tools`;
CREATE TABLE `tools` (
`tool_id` int(11) NOT NULL AUTO_INCREMENT,
`tool` varchar(30) DEFAULT NULL,
`owner_id` int(11) DEFAULT NULL,
PRIMARY KEY (`tool_id`)
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=latin1;
-- -----
-- Records of tools
-- -----
INSERT INTO `tools` VALUES ('1', 'Hammer', '9');
INSERT INTO `tools` VALUES ('2', 'Pliers', '1');
INSERT INTO `tools` VALUES ('3', 'Knife', '1');
INSERT INTO `tools` VALUES ('4', 'Chisel', '2');
INSERT INTO `tools` VALUES ('5', 'Hacksaw', '1');
INSERT INTO `tools` VALUES ('6', 'Level', null);
INSERT INTO `tools` VALUES ('7', 'Wrench', null);
INSERT INTO `tools` VALUES ('8', 'Tape Measure', '9');
INSERT INTO `tools` VALUES ('9', 'Screwdriver', null);
INSERT INTO `tools` VALUES ('10', 'Clamp', null);

```

o

UNIONo LEFT JOINo o

RIGHT JOINo NULLo WHERE **-clause** owners.owner_id IS NULLo

UNION ALLo

```

SELECT `owners`.`owner`, tools.tool
FROM `owners`
LEFT JOIN `tools` ON `owners`.`owner_id` = `tools`.`owner_id`
UNION ALL
SELECT `owners`.`owner`, tools.tool
FROM `owners`
RIGHT JOIN `tools` ON `owners`.`owner_id` = `tools`.`owner_id`
WHERE `owners`.`owner_id` IS NULL;

```

```

+-----+-----+
| owner | tool      |
+-----+-----+
| Ben   | Pliers    |
| Ben   | Knife     |

```

```

| Ben      | Hacksaw      |
| Jim      | Chisel       |
| Harry    | NULL         |
| John     | NULL         |
| Ellie    | Hammer      |
| Ellie    | Tape Measure |
| NULL     | Level        |
| NULL     | Wrench       |
| NULL     | Screwdriver  |
| NULL     | Clamp        |
+-----+-----+
12 rows in set (0.00 sec)

```

3

o

- o
-
-

“”

ID	reg_date	
1	201688

“”

ID	
1	yennefer
2	

“tags_meta”

POST_ID	TAG_ID
1	2

```

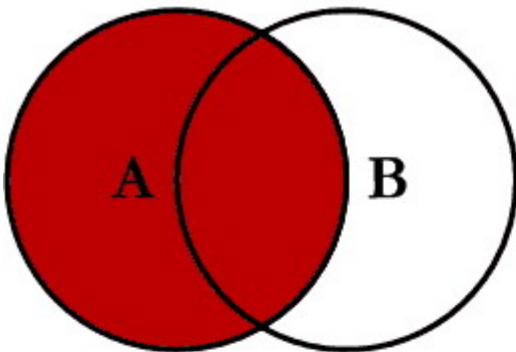
SELECT videogame.id,
       videogame.title,
       videogame.reg_date,
       tags.name,
       tags_meta.post_id
FROM tags_meta
INNER JOIN videogame ON videogame.id = tags_meta.post_id
INNER JOIN tags ON tags.id = tags_meta.tag_id
WHERE tags.name = "elizabeth"
ORDER BY videogame.reg_date

```

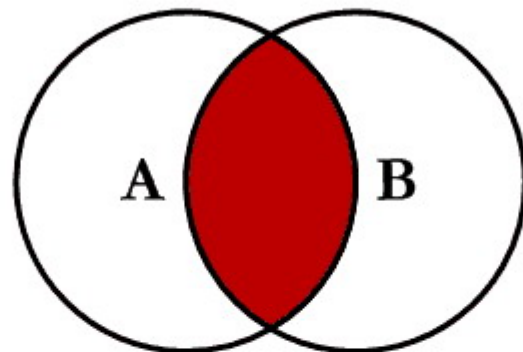
“#elizabeth”

MySQLJOIN ◦

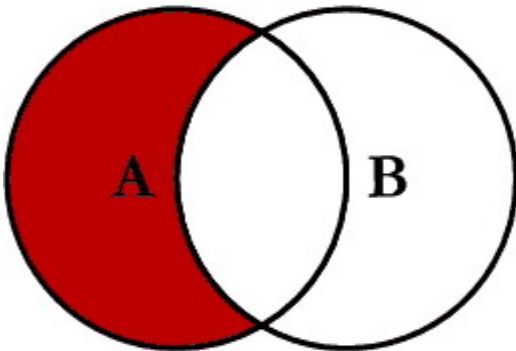
SQL JOINS



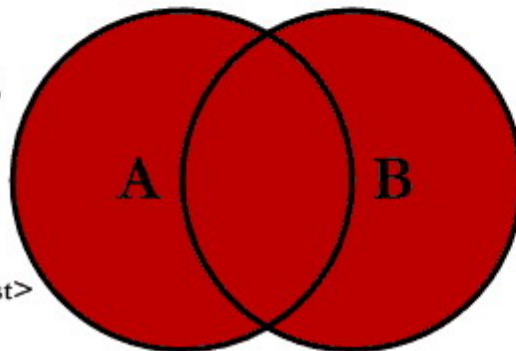
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



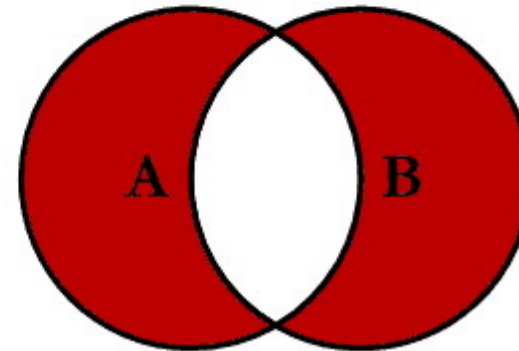
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



© C.L. Moffatt, 2008

<https://riptutorial.com/zh-TW/mysql/topic/2736/>

28:

1. [LOW_PRIORITY | CONCURRENT] [LOCAL] INFILE'file_name'
2. INTO TABLE tbl_name
3. [CHARACTER SET charset]
4. [{FIELDS | COLUMNS} ['string'] [[OPTIONALL]'char']]
5. [LINES [STARTING BY'string'] [TERMINATED BY'string']]
6. [IGNORE{LINES | ROWS}]
7. [col_name_or_user_var...]
8. [SET col_name = expr...]

Examples

LOAD DATA INFILE

';' - CSV。

```
1;max;male;manager;12-7-1985
2;jack;male;executive;21-8-1990
.
.
.
1000000;marta;female;accountant;15-6-1992
```

```
CREATE TABLE `employee` ( `id` INT NOT NULL ,
                           `name` VARCHAR NOT NULL,
                           `sex` VARCHAR NOT NULL ,
                           `designation` VARCHAR NOT NULL ,
                           `dob` VARCHAR NOT NULL );
```

```
LOAD DATA INFILE 'path of the file/file_name.txt'
INTO TABLE employee
FIELDS TERMINATED BY ';' //specify the delimiter separating the values
LINES TERMINATED BY '\r\n'
(id,name,sex,designation,dob)
```

```
1;max;male;manager;17-Jan-1985
2;jack;male;executive;01-Feb-1992
.
.
.
1000000;marta;female;accountant;25-Apr-1993
```

dob◦

```
LOAD DATA INFILE 'path of the file/file_name.txt'  
INTO TABLE employee  
FIELDS TERMINATED BY ';' //specify the delimiter separating the values  
LINES TERMINATED BY '\r\n'  
(id,name,sex,designation,@dob)  
SET date = STR_TO_DATE(@date, '%d-%b-%Y');
```

LOAD DATA INFILE◦

LOAD DATA INFILE◦

CSVMySQL

CSVMySQLCSV◦

```
load data infile '/tmp/file.csv'  
into table my_table  
fields terminated by ','  
optionally enclosed by '"'  
escaped by '"'  
lines terminated by '\n'  
ignore 1 lines; -- skip the header row
```

LOAD DATA INFILE◦ ◦

◦ ◦

```
LOAD DATA LOCAL INFILE 'path of the file/file_name.txt'  
INTO TABLE employee
```

LOAD DATA INFILE'fname'

replace

```
LOAD DATA INFILE 'path of the file/file_name.txt'  
REPLACE INTO TABLE employee
```

LOAD DATA INFILE'fname'IGNORE

REPLACE◦ LOCAL◦ ◦

```
LOAD DATA INFILE 'path of the file/file_name.txt'  
IGNORE INTO TABLE employee
```

◦ ◦ ◦

```
INSERT INTO employee SELECT * FROM intermediary WHERE ...
```

```
SELECT a,b,c INTO OUTFILE 'result.txt' FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'  
LINES TERMINATED BY '\n' FROM table;
```

```
LOAD DATA INFILE 'result.txt' INTO TABLE table;
```

<https://riptutorial.com/zh-TW/mysql/topic/2356/>

29:

Examples

``。

“MySQL”。PHPmyAdmin“MySQL”。

“group”。

```
SELECT student_name, AVG(test_score) FROM student GROUP BY group
```

```
SELECT student_name, AVG(test_score) FROM student GROUP BY `group`
```

◦ JOIN◦

```
SELECT `users`.`username`, `groups`.`group` FROM `users`
```

◦

```
select student_name, AVG(test_score) from student group by group  
select `student_name`, AVG(`test_score`) from `student` group by `group`
```

MySQL。R。◦◦◦

<https://riptutorial.com/zh-TW/mysql/topic/5208/>

30:

- AUTO_INCREMENT ID - PK"PK;◦
- MEDIUMINT - INTs⇒◦ ◦
- UNSIGNED - INT
- NOT NULL -
- InnoDB - MyISAMPRIIMARY KEYInnoDB◦
- INDEX(y_id, x_id) - PRIMARY KEY;◦ UNIQUE ;INSERTs◦
- INDEX(y_id)x_id◦ ""◦

;◦ ◦

FOREIGN KEY◦

Examples

```
CREATE TABLE XtoY (  
  # No surrogate id for this table  
  x_id MEDIUMINT UNSIGNED NOT NULL,    -- For JOINing to one table  
  y_id MEDIUMINT UNSIGNED NOT NULL,    -- For JOINing to the other table  
  # Include other fields specific to the 'relation'  
  PRIMARY KEY(x_id, y_id),             -- When starting with X  
  INDEX      (y_id, x_id)               -- When starting with Y  
) ENGINE=InnoDB;
```

◦

<https://riptutorial.com/zh-TW/mysql/topic/4857/>

31:

ASCII	
BIN	
BIT_LENGTH	
CHAR	
CHAR_LENGTH	
CHARACTER_LENGTH	CHAR_LENGTH
CONCAT	
CONCAT_WS	
ELT	
EXPORT_SET	on
FIND_IN_SET	
FROM_BASE64	base-64
HEX	
INSTR	
LCASE	LOWER
LOAD_FILE	
LPAD	
LTRIM	
MAKE_SET	

REGEXP	REGEXP
OCTET_LENGTH	LENGTH
ORD	
	LOCATE
	SQL
REGEXP	
RLIKE	REGEXP
RPAD	
RTRIM	
SOUNDEX	soundex
STRCMP	
SUBSTR	
SUBSTRING	
SUBSTRING_INDEX	
TO_BASE64	base-64
UCASE	UPPER
UNHEX	
WEIGHT_STRING	

Examples

```
SELECT FIND_IN_SET('b', 'a,b,c');
```

```
SELECT FIND_IN_SET('d','a,b,c');
```

0

STR_TO_DATE -

my_date_field [string] 07/25/2016 STR_TO_DATE

```
SELECT STR_TO_DATE(my_date_field, '%m/%d/%Y') FROM my_table;
```

WHERE◊

LOWER/ LCASE

LOWERstr

```
LOWER('fOoBar') -- 'foobar'  
LCASE('fOoBar') -- 'foobar'
```

REPLACEstrfrom_strto_str

```
REPLACE('foobarbaz', 'bar', 'BAR') -- 'fooBARbaz'  
REPLACE('foobarbaz', 'zzz', 'ZZZ') -- 'foobarbaz'
```

SUBSTRING

SUBSTRINGSUBSTR

SUBSTRING(str, start_position)

```
SELECT SUBSTRING('foobarbaz', 4); -- 'barbaz'  
SELECT SUBSTRING('foobarbaz' FROM 4); -- 'barbaz'  
  
-- using negative indexing  
SELECT SUBSTRING('foobarbaz', -6); -- 'barbaz'  
SELECT SUBSTRING('foobarbaz' FROM -6); -- 'barbaz'
```

SUBSTRING(str, start_position, length)

```
SELECT SUBSTRING('foobarbaz', 4, 3); -- 'bar'  
SELECT SUBSTRING('foobarbaz', FROM 4 FOR 3); -- 'bar'  
  
-- using negative indexing  
SELECT SUBSTRING('foobarbaz', -6, 3); -- 'bar'  
SELECT SUBSTRING('foobarbaz' FROM -6 FOR 3); -- 'bar'
```

UPPER/ UCASE

UPPERstr


```
UPPER('fOoBar') -- 'FOOBAR'  
UCASE('fOoBar') -- 'FOOBAR'
```

◦ CHAR_LENGTH

LENGTHstr

```
LENGTH('foobar') -- 6  
LENGTH('fööbar') -- 8 -- contrast with CHAR_LENGTH(...) = 6
```

CHAR_LENGTH

CHAR_LENGTHstr

```
CHAR_LENGTH('foobar') -- 6  
CHAR_LENGTH('fööbar') -- 6 -- contrast with LENGTH(...) = 8
```

HEXSTR

◦ ◦

```
HEX('fööbar') -- 66F6F6626172 -- in "CHARACTER SET latin1" because "F6" is hex for ö  
HEX('fööbar') -- 66C3B6C3B6626172 -- in "CHARACTER SET utf8 or utf8mb4" because "C3B6" is hex  
for ö
```

<https://riptutorial.com/zh-TW/mysql/topic/1399/>

32:

Examples

```
CREATE TABLE foo ( ...
  name CHARACTER SET utf8mb4
  ... );
```

MySQL

```
SET NAMES utf8mb4;
```

PHP Python Java ... SET NAMES

```
SET NAMES utf8mb4 CHARACTER SET latin1 -latin1 utf8mb4 INSERTing SELECTing
```

```
SHOW COLLATION; SHOW COLLATION;
```

4 CHARACTER SETs

```
ascii -- basic 7-bit codes.
latin1 -- ascii, plus most characters needed for Western European languages.
utf8 -- the 1-, 2-, and 3-byte subset of utf8. This excludes Emoji and some of Chinese.
utf8mb4 -- the full set of UTF8 characters, covering all current languages.
```

- utf8 utf8mb4

...

- utf8mb4 TEXT VARCHAR
- UUID MD5 country_code postal_code ascii latin1

utf8mb4 5.3 utf8

MySQL "UTF8" MySQL utf8mb4 MySQL utf8

charset_ci "case and accent insensitive" _bin

"utf8mb4 utf8mb4_unicode_520_ci Unicode 5.20 utf8mb4_polish_ci

CHARACTER SET CHARSET

```
CREATE TABLE Address (
  `AddressID` INTEGER NOT NULL PRIMARY KEY,
  `Street` VARCHAR(80) CHARACTER SET ASCII,
  `City` VARCHAR(80),
  `Country` VARCHAR(80) DEFAULT "United States",
  `Active` BOOLEAN DEFAULT 1,
```

```
) Engine=InnoDB default charset=UTF8;
```

CityCountryUTF8 ◦ StreetASCII ◦

◦

<https://riptutorial.com/zh-TW/mysql/topic/4569/>

33:



◦

CALL ◦

◦

Examples

INT12 ◦

```
DELIMITER ||
CREATE FUNCTION functionname()
RETURNS INT
BEGIN
    RETURN 12;
END;
||
DELIMITER ;
```

DELIMITER || ;;CREATE ◦

CREATE FUNCTION; DELIMITER ; ◦

```
SELECT functionname();
+-----+
| functionname() |
+-----+
|           12 |
+-----+
```

```
DELIMITER $$
CREATE FUNCTION add_2 ( my_arg INT )
RETURNS INT
BEGIN
    RETURN (my_arg + 2);
END;
$$
DELIMITER ;
```

```
SELECT add_2(12);
+-----+
| add_2(12) |
+-----+
```

```
|          14 |
+-----+
```

DELIMITER ° CREATE \\ || \$\$ °

GUI °

```
DROP PROCEDURE if exists displayNext100WithName;
DELIMITER $$
CREATE PROCEDURE displayNext100WithName
(
    nStart int,
    tblName varchar(100)
)
BEGIN
    DECLARE thesql varchar(500); -- holds the constructed sql string to execute

    -- expands the sizing of the output buffer to accomodate the output (Max value is at least
    4GB)
    SET session group_concat_max_len = 4096; -- prevents group_concat from barfing with error
    1160 or whatever it is

    SET @thesql=CONCAT("select group_concat(qid order by qid SEPARATOR '%3B') as nums ","from
    (
        select qid from ");
    SET @thesql=CONCAT(@thesql,tblName," where qid>? order by qid limit 100 )xDerived");
    PREPARE stmt1 FROM @thesql; -- create a statement object from the construct sql string to
    execute
    SET @p1 = nStart; -- transfers parameter passed into a User Variable compatible with the
    below EXECUTE
    EXECUTE stmt1 USING @p1;

    DEALLOCATE PREPARE stmt1; -- deallocate the statement object when finished
END $$
DELIMITER ;
```

DELIMITER °

```
call displayNext100WithName(1,"questions_mysql");
```



%3B

INOUT INOUT

```
DELIMITER $$
DROP PROCEDURE IF EXISTS sp_nested_loop $$
CREATE PROCEDURE sp_nested_loop(IN i INT, IN j INT, OUT x INT, OUT y INT, INOUT z INT)
BEGIN
    DECLARE a INTEGER DEFAULT 0;
    DECLARE b INTEGER DEFAULT 0;
    DECLARE c INTEGER DEFAULT 0;
    WHILE a < i DO
        WHILE b < j DO
            SET c = c + 1;
```

```

        SET b = b + 1;
    END WHILE;
    SET a = a + 1;
    SET b = 0;
END WHILE;
SET x = a, y = c;
SET z = x + y + z;
END $$
DELIMITER ;

```

CALL

```

SET @z = 30;
call sp_nested_loop(10, 20, @x, @y, @z);
SELECT @x, @y, @z;

```

```

+-----+-----+-----+
| @x | @y | @z |
+-----+-----+-----+
| 10 | 200 | 240 |
+-----+-----+-----+

```

IN◦ ◦

OUT◦ **NULL**◦

INOUT◦

[http //dev.mysql.com/doc/refman/5.7/en/create-procedure.html](http://dev.mysql.com/doc/refman/5.7/en/create-procedure.html)

◦ **DECLARE SQL**

```

DECLARE student CURSOR FOR SELECT name FROM student;

```

◦ ◦

```

CREATE TABLE product
(
    id INT(10) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    type VARCHAR(50) NOT NULL,
    name VARCHAR(255) NOT NULL
);
CREATE TABLE product_type
(
    name VARCHAR(50) NOT NULL PRIMARY KEY
);
CREATE TABLE product_type_count
(
    type VARCHAR(50) NOT NULL PRIMARY KEY,
    count INT(10) UNSIGNED NOT NULL DEFAULT 0
);
INSERT INTO product_type (name) VALUES

```

```

('dress'),
('food');

INSERT INTO product (type, name) VALUES
('dress', 'T-shirt'),
('dress', 'Trousers'),
('food', 'Apple'),
('food', 'Tomatoes'),
('food', 'Meat');

```

```

DELIMITER //
DROP PROCEDURE IF EXISTS product_count;
CREATE PROCEDURE product_count()
BEGIN
    DECLARE p_type VARCHAR(255);
    DECLARE p_count INT(10) UNSIGNED;
    DECLARE done INT DEFAULT 0;
    DECLARE product CURSOR FOR
        SELECT
            type,
            COUNT(*)
        FROM product
        GROUP BY type;
    DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET done = 1;

    TRUNCATE product_type;

    OPEN product;

    REPEAT
        FETCH product
        INTO p_type, p_count;
        IF NOT done
        THEN
            INSERT INTO product_type_count
            SET
                type = p_type,
                count = p_count;
        END IF;
    UNTIL done
    END REPEAT;

    CLOSE product;
END //
DELIMITER ;

```

```
CALL product_count();
```

product_type_count

type	count
dress	2
food	3

CURSOR

```
INSERT INTO product_type_count
    (type, count)
SELECT type, COUNT(*)
FROM product
GROUP BY type;
```

◦

ResultSet

SELECT Stored Procedure ◦ **PerIPHP**CALL◦

```
DELIMITER $$
CREATE
    DEFINER=`db_username`@`hostname_or_IP`
    FUNCTION `function_name` (optional_param data_type (length_if_applicable))
    RETURNS data_type
BEGIN
    /*
    SQL Statements goes here
    */
END$$
DELIMITER ;
```

RETURNS data_typeMySQL◦

<https://riptutorial.com/zh-TW/mysql/topic/1351/-->

34:

◦

Examples

REGEXP / RLIKE

REGEXP RLIKE ◦

employee

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER	SALARY
100	Steven	King	515.123.4567	24000.00
101	Neena	Kochhar	515.123.4568	17000.00
102	Lex	De Haan	515.123.4569	17000.00
103	Alexander	Hunold	590.423.4567	9000.00
104	Bruce	Ernst	590.423.4568	6000.00
105	David	Austin	590.423.4569	4800.00
106	Valli	Pataballa	590.423.4560	4800.00
107	Diana	Lorentz	590.423.5567	4200.00
108	Nancy	Greenberg	515.124.4569	12000.00
109	Daniel	Faviet	515.124.4169	9000.00
110	John	Chen	515.124.4269	8200.00

^

FIRST_NAME **N** ◦

```
SELECT * FROM employees WHERE FIRST_NAME REGEXP '^N'  
-- Pattern start with-----^
```

\$ **

PHONE_NUMBER **4569** ◦

```
SELECT * FROM employees WHERE PHONE_NUMBER REGEXP '4569$'  
-- Pattern end with-----^
```

REGEXP

FIRST_NAME **N** ◦

```
SELECT * FROM employees WHERE FIRST_NAME NOT REGEXP '^N'
```

```
-- Pattern does not start with-----^
```

LAST_NAME FIRST_NAMEa ◦

```
SELECT * FROM employees WHERE FIRST_NAME REGEXP 'a' AND LAST_NAME REGEXP 'in'  
-- No ^ or $, pattern can be anywhere -----^
```

[

FIRST_NAME**ABC**◦

```
SELECT * FROM employees WHERE FIRST_NAME REGEXP '^[ABC]'  
-----^^---
```

|

FIRST_NAME**ABC**r ei◦

```
SELECT * FROM employees WHERE FIRST_NAME REGEXP '^[ABC]|[rei]$'  
-----^^-----^^-----^^
```

```
SELECT FIRST_NAME, FIRST_NAME REGEXP '^N' as matching FROM employees
```

FIRST_NAME REGEXP '^N' **10** FIRST_NAME^N◦

```
SELECT  
FIRST_NAME,  
IF(FIRST_NAME REGEXP '^N', 'matches ^N', 'does not match ^N') as matching  
FROM employees
```

```
SELECT  
IF(FIRST_NAME REGEXP '^N', 'matches ^N', 'does not match ^N') as matching,  
COUNT(*)  
FROM employees  
GROUP BY matching
```

<https://riptutorial.com/zh-TW/mysql/topic/9444/>

35: JSON

MySQL 5.7.8+JSON。 json。

JSON_EXTRACT -->>>。

- JSON_EXTRACTjson_doc[...]
- JSON_EXTRACTjson_doc
- JSON_EXTRACTjson_doc12

json_doc	JSON

MySQL 5.7

- - ◦
- NULL
 - argemuntNULL
 -

NULLNULL。

Examples

JSON

@myjsonJSON

```
SET @myjson = CAST('["A","B",{ "id":1,"label":"C"}]' as JSON) ;
```

SELECT

```
SELECT
  JSON_EXTRACT( @myjson , '$[1]' ) ,
  JSON_EXTRACT( @myjson , '$[*].label' ) ,
  JSON_EXTRACT( @myjson , '$[1].*' ) ,
  JSON_EXTRACT( @myjson , '$[2].*' )
;
-- result values:
'\"B\"' , '[\"C\"]' , NULL , '[1, \"C\"]'
-- visually:
"B" , ["C"] , NULL , [1, "C"]
```

JSON

->->>path ->>UNQUOTED

```
SELECT
  myjson_col->>'${1}' , myjson_col->'${1}' ,
  myjson_col->>'${*}.label' ,
  myjson_col->>'${1}.*' ,
  myjson_col->>'${2}.*'
FROM tablename ;
-- visuall:
  B, "B" , ["C"], NULL, [1, "C"]
--^^^ ^^^
```

col->>pathJSON_UNQUOTE(JSON_EXTRACT(col,path))

->->>EXPLAIN

```
mysql> EXPLAIN SELECT c->>'$.name' AS name
->      FROM jemp WHERE g > 2\G
***** 1. row *****
      id: 1
      select_type: SIMPLE
      table: jemp
      partitions: NULL
      type: range
possible_keys: i
      key: i
      key_len: 5
      ref: NULL
      rows: 2
      filtered: 100.00
      Extra: Using where
1 row in set, 1 warning (0.00 sec)

mysql> SHOW WARNINGS\G
***** 1. row *****
      Level: Note
      Code: 1003
      Message: /* select#1 */ select
      json_unquote(json_extract(`jtest`.`jemp`.`c`, '$.name')) AS `name` from
      `jtest`.`jemp` where (`jtest`.`jemp`.`g` > 2)
1 row in set (0.00 sec)
```

+

JSON <https://riptutorial.com/zh-TW/mysql/topic/9042/json>

36: MyISAMInnoDB

Examples

```
ALTER TABLE foo ENGINE=InnoDB;
```

◦ ◦ ◦

```
SET @DB_NAME = DATABASE();

SELECT CONCAT('ALTER TABLE `', table_name, '` ENGINE=InnoDB;') AS sql_statements
FROM information_schema.tables
WHERE table_schema = @DB_NAME
AND `ENGINE` = 'MyISAM'
AND `TABLE_TYPE` = 'BASE TABLE';
```

DATABASE() NULL ◦ **mysql**◦

SQL_{MyISAM}◦

SQL◦

MyISAMInnoDB <https://riptutorial.com/zh-TW/mysql/topic/3135/myisaminnodb>

37: root

Examples

rootrootsockethttp

rootmySQL

```
sudo systemctl stop mysql
```

mySQL

```
sudo mysqld_safe --skip-grant-tables
```

```
mysql -u root
```

SQL shell

```
select User, password,plugin FROM mysql.user ;
```

null

```
update mysql.user set password=PASSWORD('mypassword'), plugin = NULL WHERE User = 'root';  
exit;
```

Unix shellmySQL

```
sudo service mysql stop  
sudo service mysql start
```

root <https://riptutorial.com/zh-TW/mysql/topic/9973/root>

38:

- SELECT DISTINCT GROUP BY.
- OFFSET "".
- WHERE ab=22,33.
- UNION ALL DISTINCT - ALL DISTINCT.
- SELECT * TEXT BLOB. tmp.
- GROUP BY ORDER BY.
- FORCE INDEX;

ORDER BY LIKE REGEXP. ◦

◦

Examples

""

""

```
INDEX(last_name, first_name)
```

```
WHERE last_name = '...'
WHERE first_name = '...' AND last_name = '...' -- (order in WHERE does not matter)
```

```
WHERE first_name = '...' -- order in INDEX _does_ matter
WHERE last_name = '...' OR first_name = '...' -- "OR" is a killer
```

innodb_buffer_pool_size 70 ◦

```
x IN ( SELECT ... )
```

JOIN

OR ◦

"" WHERE DATE(x) = ... ; WHERE x = ...

WHERE LCASE(name1) = LCASE(name2) ◦

OFFSET ""。

SELECT * ...。

Maria DelevaBarrankaBatsu; /。

。

- InnoDBMyISAM。
- PARTITIONing;。
- query_cache_size 100M。
- my.cnf ""。
- ""INDEX(foo(20))。
- OPTIMIZE TABLE。。

。

```
WHERE a = 12 --> INDEX(a)
WHERE a > 12 --> INDEX(a)

WHERE a = 12 AND b > 78 --> INDEX(a,b) is more useful than INDEX(b,a)
WHERE a > 12 AND b > 78 --> INDEX(a) or INDEX(b); no way to handle both ranges

ORDER BY x --> INDEX(x)
ORDER BY x, y --> INDEX(x,y) in that order
ORDER BY x DESC, y ASC --> No index helps - because of mixing ASC and DESC
```

。

```
WHERE DATE(dt) = '2000-01-01'
```

INDEX(dt)

```
WHERE dt = '2000-01-01' -- if `dt` is datatype `DATE`
```

DATE DATETIME TIMESTAMP DATETIME(6)

```
WHERE dt >= '2000-01-01'
AND dt < '2000-01-01' + INTERVAL 1 DAY
```

OR。

```
WHERE a = 12 OR b = 78
```

INDEX(a,b) ""INDEX(a), INDEX(b)。。

```
WHERE x = 3 OR x = 5
```



```
WHERE x IN (3, 5)
```

x°

◦ **“”** ◦ ◦ ◦

◦ **1**◦ LEFT JOIN◦

```
SELECT a, b, ( SELECT ... FROM t WHERE t.x = u.x ) AS c
      FROM u ...
SELECT a, b, ( SELECT MAX(x) ... ) AS c
      FROM u ...
SELECT a, b, ( SELECT x FROM t ORDER BY ... LIMIT 1 ) AS c
      FROM u ...
```

```
SELECT ...
      FROM ( SELECT ... ) AS a
      JOIN b ON ...
```

FROM-SELECT

- **1**◦
- **“1”**(SELECT @n := 0) `@variable`◦
- JOIN(SELECT ...)◦ **5.6**CROSS JOIN ; **5.6+**SELECT◦

+ GROUP BY

```
SELECT ...
      FROM a
      JOIN b ON ...
      WHERE ...
      GROUP BY a.id
```

JOIN;GROUP BY **whittles**a ◦

◦ JOINSELECT◦ GROUP BY ◦

<https://riptutorial.com/zh-TW/mysql/topic/4292/>

39: MySQL 5.7+root

MySQL 5.7MySQLrootroot mysql.log

rootMySQL 5.7+

Examples

-
- SSL
- validate_password
- "root"@"localhost"

root

"root"

```
shell> sudo grep 'temporary password' /var/log/mysql.log
```

root

```
shell> mysql -uroot -p
```

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'MyNewPass5!';
```

MySQLvalidate_password 8

UNIX"/ var / run / mysqld"root

.

```
$ mysql -u root -p
```

ERROR 104528000'root'@'localhost'YES

```
$ systemctl status mysql.service
```

mysql.service - MySQL/lib/systemd/system/mysql.service;;2017-06-08 14:31:33 IST; 38s ago

```
mysqld_safe --skip-grant-tables &
```

mysqld_safe UNIX"/ var / run / mysqld'.

```
$ systemctl stop mysql.service
```

```
$ ps -eaf|grep mysql
```

```
$ mysqld_safe --skip-grant-tables &
```

```
$ mkdir -p /var/run/mysqld
$ chown mysql:mysql /var/run/mysqld
```

```
mysqld_safe --skip-grant-tables &
```

```
mysqld_safe/ var / lib / mysqlmysqld
```

```
$ mysql -u root
```

5.7.18-0ubuntu0.16.04.1Ubuntu

©2000,2017Oracle/。。

OracleOracle Corporation/。。

'help;'\ h'。 \ c'。

MySQL>

```
mysql> use mysql
mysql> describe user;
```

-A

```
mysql> FLUSH PRIVILEGES;
mysql> SET PASSWORD FOR root@'localhost' = PASSWORD('newpwd');
```

mysql root

```
UPDATE mysql.user SET Password=PASSWORD('newpwd') WHERE User='root';
```

```
USE mysql
UPDATE user SET Password = PASSWORD('newpwd')
WHERE Host = 'localhost' AND User = 'root';
```

root

```
USE mysql
UPDATE user SET Password = PASSWORD('newpwd')
WHERE Host = '%' AND User = 'root';`enter code here
```

quit mysql/

```
FLUSH PRIVILEGES;
sudo /etc/init.d/mysql stop
sudo /etc/init.d/mysql start
```

`mysql -u root -p'

MySQL>

MySQL 5.7+root <https://riptutorial.com/zh-TW/mysql/topic/9563/mysql-5-7plusroot>

40:

- DROP TABLE table_name;
- DROP TABLE IF EXISTS table_name; -
- DROP TABLE t1t2t3; - DROP
- DROP TEMPORARY TABLE t; - CREATE TEMPORARY TABLE...

◦ DROP TABLE ◦
◦ DROP TABLE ◦

Examples

Drop Table◦

tbl

```
CREATE TABLE tbl (  
  id INT NOT NULL AUTO_INCREMENT,  
  title VARCHAR(100) NOT NULL,  
  author VARCHAR(40) NOT NULL,  
  submission_date DATE,  
  PRIMARY KEY (id)  
);
```

```
DROP TABLE tbl;
```

◦

DROP TABLE Database.table_name

<https://riptutorial.com/zh-TW/mysql/topic/4123/>

41:

1. INSERT [LOW_PRIORITY || HIGH_PRIORITY] [IGNORE] [INTO] tbl_name [PARTITION partition_name...] [col_name...] {VALUES | VALUE}{expr | DEFAULT}..... [ON DUPLICATE KEY UPDATE col_name = expr [col_name = expr] ...]
2. INSERT [LOW_PRIORITY || HIGH_PRIORITY] [IGNORE] [INTO] tbl_name [PARTITION partition_name...] SET col_name = {expr | DEFAULT}... [ON DUPLICATE KEY UPDATE col_name = expr [col_name = expr] ...]
3. INSERT [LOW_PRIORITY | HIGH_PRIORITY] [IGNORE] [INTO] tbl_name [PARTITION partition_name...] [col_name...] SELECT ... [ON DUPLICATE KEY UPDATE col_name = expr [col_name = expr] ...]
4. expr. col2col1
INSERT INTO tbl_name col1 col2 VALUES 15 col1 * 2;
5. VALUESINSERT. ◦
INSERT INTO tbl_name abc VALUES 1,2,3,4,5,6,7,8,9;
6. ◦
INSERT INTO tbl_name abc VALUES 1,2,3,4,5,6,7,8,9;
7. **INSERT ... SELECT**
INSERT [LOW_PRIORITY | HIGH_PRIORITY] [IGNORE] [INTO] tbl_name [PARTITION partition_name...] [col_name...] SELECT ... [ON DUPLICATE KEY UPDATE col_name = expr...]
8. INSERT ... SELECT.
INSERT INTO tbl_temp2 fld_id SELECT tbl_temp1.fld_order_id FROM tbl_temp1 WHERE tbl_temp1.fld_order_id > 100;

INSERT

Examples

```
INSERT INTO `table_name` (`field_one`, `field_two`) VALUES ('value_one', 'value_two');
```

```
table_name field_one field_two value_one value_two field_one field_two ◦
```

INSERT ON DUPLICATE KEY UPDATE

```
INSERT INTO `table_name`  
  (`index_field`, `other_field_1`, `other_field_2`)  
VALUES  
  ('index_value', 'insert_value', 'other_value')  
ON DUPLICATE KEY UPDATE
```

```
`other_field_1` = 'update_value',
`other_field_2` = VALUES(`other_field_2`);
```

INSERT table_name other_field_1

VALUES () INSERT UPDATE other_field_1 insert_value INSERT update_value UPDATE other_field_2
other_value

IODKU . . .

```
INSERT INTO `my_table` (`field_1`, `field_2`) VALUES
('data_1', 'data_2'),
('data_1', 'data_3'),
('data_4', 'data_5');
```

INSERT

“”。 10010。

◦ INSERT IGNORE

```
SELECT * FROM `people`;
--- Produces:
+----+-----+
| id | name |
+----+-----+
| 1  | john |
| 2  | anna |
+----+-----+
```

```
INSERT IGNORE INTO `people` (`id`, `name`) VALUES
('2', 'anna'), --- Without the IGNORE keyword, this record would produce an error
('3', 'mike');
```

```
SELECT * FROM `people`;
--- Produces:
+----+-----+
| id | name |
+----+-----+
| 1  | john |
| 2  | anna |
| 3  | mike |
+----+-----+
```

INSERT IGNORE Mysql

IGNORE. IGNORE>;

- ◦

INSERT

```
INSERT INTO `my_table` VALUES
('data_1', 'data_2'),
```

```
('data_1', 'data_3'),
('data_4', 'data_5');
```

INSERT SELECT

SELECT。

```
INSERT INTO `tableA` (`field_one`, `field_two`)
  SELECT `tableB`.`field_one`, `tableB`.`field_two`
  FROM `tableB`
  WHERE `tableB`.clmn <> 'someValue'
  ORDER BY `tableB`.`sorting_clmn`;
```

SELECT * FROM tableA tableB。

AUTO_INCREMENT INSERT with VALUES。

。

AUTO_INCREMENT + LAST_INSERT_ID INSERT

AUTO_INCREMENT PRIMARY KEY ID。

```
CREATE TABLE t (
  id SMALLINT UNSIGNED AUTO_INCREMENT NOT NULL,
  this ...,
  that ...,
  PRIMARY KEY(id) );

INSERT INTO t (this, that) VALUES (... , ...);
SELECT LAST_INSERT_ID() INTO @id;
INSERT INTO another_table (... , t_id, ...) VALUES (... , @id, ...);
```

LAST_INSERT_ID() ID。

API LAST_INSERT_ID() SELECT @variable MySQL @variable 。

IODKU "" UNIQUE AUTO_INCREMENT PRIMARY KEY "" 。

```
CREATE TABLE iodku (
  id INT AUTO_INCREMENT NOT NULL,
  name VARCHAR(99) NOT NULL,
  misc INT NOT NULL,
  PRIMARY KEY(id),
  UNIQUE(name)
) ENGINE=InnoDB;

INSERT INTO iodku (name, misc)
VALUES
  ('Leslie', 123),
  ('Sally', 456);

Query OK, 2 rows affected (0.00 sec)
Records: 2 Duplicates: 0 Warnings: 0
```



```

+----+-----+-----+
| id | name  | misc |
+----+-----+-----+
|  1 | Leslie | 123  |
|  2 | Sally  | 456  |
+----+-----+-----+

```

IODKU "" LAST_INSERT_ID() id

```

INSERT INTO iodku (name, misc)
VALUES
('Sally', 3333)           -- should update
ON DUPLICATE KEY UPDATE  -- `name` will trigger "duplicate key"
id = LAST_INSERT_ID(id),
misc = VALUES(misc);
SELECT LAST_INSERT_ID();  -- picking up existing value
+-----+
| LAST_INSERT_ID() |
+-----+
|                2 |
+-----+

```

IODKU "" LAST_INSERT_ID() id

```

INSERT INTO iodku (name, misc)
VALUES
('Dana', 789)           -- Should insert
ON DUPLICATE KEY UPDATE
id = LAST_INSERT_ID(id),
misc = VALUES(misc);
SELECT LAST_INSERT_ID();  -- picking up new value
+-----+
| LAST_INSERT_ID() |
+-----+
|                3 |
+-----+

```

```

SELECT * FROM iodku;
+----+-----+-----+
| id | name  | misc |
+----+-----+-----+
|  1 | Leslie | 123  |
|  2 | Sally  | 3333 | -- IODKU changed this
|  3 | Dana   | 789  | -- IODKU added this
+----+-----+-----+

```

AUTO_INCREMENT ID

"" ID. InnoDB

```

CREATE TABLE Burn (
  id SMALLINT UNSIGNED AUTO_INCREMENT NOT NULL,
  name VARCHAR(99) NOT NULL,
  PRIMARY KEY(id),
  UNIQUE(name)

```

```

) ENGINE=InnoDB;

INSERT IGNORE INTO Burn (name) VALUES ('first'), ('second');
SELECT LAST_INSERT_ID();           -- 1
SELECT * FROM Burn ORDER BY id;
+----+-----+
|  1 | first |
|  2 | second|
+----+-----+

INSERT IGNORE INTO Burn (name) VALUES ('second'); -- dup 'IGNOREd', but id=3 is burned
SELECT LAST_INSERT_ID();           -- Still "1" -- can't trust in this situation
SELECT * FROM Burn ORDER BY id;
+----+-----+
|  1 | first |
|  2 | second|
+----+-----+

INSERT IGNORE INTO Burn (name) VALUES ('third');
SELECT LAST_INSERT_ID();           -- now "4"
SELECT * FROM Burn ORDER BY id;   -- note that id=3 was skipped over
+----+-----+
|  1 | first |
|  2 | second|
|  4 | third |    -- notice that id=3 has been 'burned'
+----+-----+

```

- `auto_increment` ◦ `ID` ◦
- `MAX(id)` ◦ `ID` ◦ `DELETE` ◦ `ID` ◦

`INSERT REPLACE DELETE + INSERT ID` ◦ InnoDB `innodb_autoinc_lock_mode` ◦

“” `AUTO INCREMENT id` ◦ `INT` ◦

<https://riptutorial.com/zh-TW/mysql/topic/866/>

42:

Examples

/

```
select '123' * 2;
```

2 MySQL 123°

246

° 0

```
select '123ABC' * 2
```

246

```
select 'ABC123' * 2
```

0

VARCHAR255 -

len

ASCII° CHARACTER SET ascii latin1

```
UUID CHAR(36) CHARACTER SET ascii -- or pack into BINARY(16)
country_code CHAR(2) CHARACTER SET ascii
ip_address CHAR(39) CHARACTER SET ascii -- or pack into BINARY(16)
phone VARCHAR(20) CHARACTER SET ascii -- probably enough to handle extension
postal_code VARCHAR(20) CHARACTER SET ascii -- (not 'zip_code') (don't know the max

city VARCHAR(100) -- This Russian town needs 91:
    Poselok Uchebnogo Khozyaystva Srednego Professionalno-Tekhnicheskoye Uchilishche Nomer
    Odin
country VARCHAR(50) -- probably enough
name VARCHAR(64) -- probably adequate; more than some government agencies allow
```

255255°

- SELECT UNION GROUP BY MEMORY RAM° VARCHARsCHAR ° VARCHAR(255) CHARACTER SET utf8mb4 1020°
- °
- InnoDB CREATE TABLE °

VARCHARTEXT

*TEXT CHARVARCHAR

- TINYTEXT ◦
- CHAR - ;CHARACTER SET utf8mb44/◦
- CHAR CHARACTER SET ascii ◦
- VARCHAR (n) n ; TEXT◦
- *TEXT *TEXT SELECTs ◦

INTAUTO_INCREMENT

INTAUTO_INCREMENT ◦ UNSIGNED◦

“” AUTO_INCREMENT ID◦ ◦ INSERT IGNOREREPLACE ◦ id◦ InnoDB◦

“FLOATDOUBLEDECIMAL”“ENUM”◦ - “”“”

- INT
- FLOATDOUBLEDECIMAL
- CHARTEXT
- BINARYBLOB
- DATETIMETIMESTAMP
- ENUMSET
-
- **JSON** MySQL 5.7.8+
- Money“”
- ALTERing
-
- MySQL
- PRIMARY KEY
-
-

“”◦

MySQL◦

	INTEGER INT SMALLINT TINYINT MEDIUMINT BIGINT
	DECIMAL NUMERIC
	FLOAT DOUBLE
	BIT

0◦

TINYINT	1	-2^7 -128	$2^7 - 1$ 127	$2^8 - 1$ 255
SMALLINT	2	-2^{15} -32,768	$2^{15} - 1$ 32,767	$2^{16} - 1$ 65,535
MEDIUMINT	3	-2^{23} -8,388,608	$2^{23} - 1$ 8,388,607	$2^{24} - 1$ 16,777,215
INT	4	-2^{31} -2,147,483,648	$2^{31} - 1$ 2,147,483,647	$2^{32} - 1$ 4,294,967,295
BIGINT	8	-2^{63} -9,223,372,036,854,775,808	$2^{63} - 1$ 9,223,372,036,854,775,807	$2^{64} - 1$ 18,446,744,073,709,551,615

MySQL DECIMAL NUMERIC

.

Precision

```
salary DECIMAL(5,2)
```

5precision 2scale -999.99 to 999.99

scale 0

65

DECIMAL(M,N) M/2

FLOAT DOUBLE

4	23/ ~7	$10^{+/-} 38$
8	53/ ~16	$10^{+/-} 308$

REAL FLOAT DOUBLE PRECISION DOUBLE

MySQL MD MD ;

FLOAT DOUBLE

BIT BIT(M) M 1 to 64

bit value◦

```
b'111'      -> 7
b'1000000' -> 128
```

'shift'(1 << 7) 128◦

NDB BIT4096◦

CHARN

CHAR(n)n ◦ CHARACTER SET utf8mb4 4*n ◦

CHAR(n)CHARACTER SET ascii ◦ latin1◦

```
country_code CHAR(2) CHARACTER SET ascii,
postal_code  CHAR(6) CHARACTER SET ascii,
uuid        CHAR(39) CHARACTER SET ascii, -- more discussion elsewhere
```

DATEDATETIMETIMESTAMPYEARTIME

DATE◦ 'YYYY-MM-DD' '1000-01-01'9999-12-31'◦

DATETIME"YYYY-MM-DD HHMMSS"◦ '1000-01-01 00:00:00'9999-12-31 23:59:59'◦

TIMESTAMP"1970-01-01 00:00:01"UTC"2038-01-19 03:14:07"UTC◦

YEAR19012155◦

TIME'HHMMSS'-8385959'8385959'◦

Data Type	Before MySQL 5.6.4	as of MySQL 5.6.4
YEAR	1 byte	1 byte
DATE	3 bytes	3 bytes
TIME	3 bytes	3 bytes + fractional seconds storage
DATETIME	8 bytes	5 bytes + fractional seconds storage
TIMESTAMP	4 bytes	4 bytes + fractional seconds storage

5.6.4

Fractional Seconds Precision	Storage Required
0	0 bytes
1,2	1 byte
3,4	2 byte
5,6	3 byte

MySQLDATEDATETIMETIMESTAMP ◦

<https://riptutorial.com/zh-TW/mysql/topic/4137/>

43:

Examples

```
Select Now();
```

◦

```
Update `footable` set mydatefield = Now();
```

mydatefield

```
'2016-07-21 12:00:00'
```

```
NOW() + INTERVAL 1 DAY -- This time tomorrow
```

```
CURDATE() - INTERVAL 4 DAY -- Midnight 4 mornings ago
```

310180600mysql

```
SELECT qId,askDate,minuteDiff
FROM
(
  SELECT qId,askDate,
    TIMESTAMPDIFF(MINUTE,askDate,now()) as minuteDiff
  FROM questions_mysql
) xDerived
WHERE minuteDiff BETWEEN 180 AND 600
ORDER BY qId DESC
LIMIT 50;
```

```
+-----+-----+-----+
| qId      | askDate                | minuteDiff |
+-----+-----+-----+
| 38546828 | 2016-07-23 22:06:50 |      182 |
| 38546733 | 2016-07-23 21:53:26 |      195 |
| 38546707 | 2016-07-23 21:48:46 |      200 |
| 38546687 | 2016-07-23 21:45:26 |      203 |
| ...      |                        |           |
+-----+-----+-----+
```

[TIMESTAMPDIFF\(\)](#) MySQL◦

[CURDATE\(\) + 1](#) MySQL◦ [Oracle](#)◦ [CURDATE\(\) + INTERVAL 1 DAY](#)◦

[BETWEEN ... AND ...](#)◦

```
WHERE x >= '2016-02-25'
AND x < '2016-02-25' + INTERVAL 5 DAY
```


- BETWEEN"
- DATETIME23 23:59:59°
- °
- xDATE DATETIMETIMESTAMP °

SYSDATENOWCURDATE

```
SELECT SYSDATE();
```

'YYYY-MM-DD HH:MM:SS'YYYYMMDDHHMMSS° °

```
SELECT NOW();
```

SYSDATE()°

```
SELECT CURDATE();
```

'YYYY-MM-DD'YYYYMMDD° °

```
SELECT DATE('2003-12-31 01:02:03');
```

2003-12-31

DATETIMETIMESTAMP° WHERE° 201691°

```
WHERE DATE(x) = '2016-09-01' /* slow! */
```

DATE() - ° MySQLx°

```
WHERE x >= '2016-09-01'
AND x < '2016-09-01' + INTERVAL 1 DAY
```

x < °

x° X° DATE(x) = '2016-09-01°

°

```
WHERE x BETWEEN '2016-09-01' AND '2016-09-01' + INTERVAL 1 DAY /* wrong! */
```

x201692°

<https://riptutorial.com/zh-TW/mysql/topic/1882/>

44:

Examples

- - - VARIABLE general_log
- - long_query_time - slow_query_log_file
- Binlog - - log_bin_basename
- -
- - mysqld.err
- / - mysql.log - log_error
- InnoDB - iblog *

basedir datadir

VARIABLES/。

。

Windows* nix。

long_query_time 10。

```
SELECT @@long_query_time;
+-----+
| @@long_query_time |
+-----+
|          10.000000 |
+-----+
```

my.cnf my.ini GLOBAL。 010。

- 10;
- 2;
- 0.5;
- 0;。

。

```
SELECT @@slow_query_log; -- Is capture currently active? (1=On, 0=Off)
SELECT @@slow_query_log_file; -- filename for capture. Resides in datadir
SELECT @@datadir; -- to see current value of the location for capture file

SET GLOBAL slow_query_log=0; -- Turn Off
-- make a backup of the Slow Query Log capture file. Then delete it.
SET GLOBAL slow_query_log=1; -- Turn it back On (new empty file is created)
```

MySQL

/slowlog5.6;。

“

```

long_query_time=...
turn on the slowlog
run for a few hours
turn off the slowlog (or raise the cutoff)
run pt-query-digest to find the 'worst' couple of queries. Or mysqldumpslow -s t

```

◦ ◦

```

36 Query insert questions_c23(qId,ownerId,title,votes,answers,isClosed,closeVotes,views,owne
comments,answeredAccepted,askDate,closeDate,lastScanDate,ign,bn,pvtc,
mainTagForImport,prepStatus,touches,status,status_bef_change,cv_bef_change,max_cv_r
values(38666373, 1322183, 'How to post a numeric value in c#', 0, 1, 0, 0, 50, 1,
0, 0, '2016-07-29 19:40:32', null, now(), 0, 0, 0,
'c%23',0,1,'0','',0,0)
on duplicate key update title='How to post a numeric value in c#', votes=0, answers
answeredAccepted=0,lastScanDate=now(), touches=touches+1,status='0'

```

```
SELECT @@general_log; -- 1 = Capture is active; 0 = It is not.
```

```
SELECT @@general_log_file; -- Full path to capture file
```

datadir ◦

Windows

```

+-----+
| @@general_log_file |
+-----+
| C:\ProgramData\MySQL\MySQL Server 5.7\Data\GuySmiley.log |
+-----+

```

Linux

```

+-----+
| @@general_log_file |
+-----+
| /var/lib/mysql/ip-ww-xx-yy-zz.log |
+-----+

```

general_log_file GLOBALdatadir ◦ ◦

general_log_filedatadir@@hostname .log◦

◦ /◦ ◦ ◦ ◦ ON◦

```
/LogBackup/GeneralLog_20160802_1520_to_20160802_1815.log
```

◦

Windows

```
SELECT @@general_log; -- 0. Not being captured
SELECT @@general_log_file; -- C:\ProgramData\MySQL\MySQL Server 5.6\Data\GuySmiley.log
SELECT @@datadir; -- C:\ProgramData\MySQL\MySQL Server 5.7\Data\
SET GLOBAL general_log_file='GeneralLogBegin_20160803_1420.log'; -- datetime clue
SET GLOBAL general_log=1; -- Turns on actual log capture. File is created under `datadir`
SET GLOBAL general_log=0; -- Turn logging off
```

Linux

```
[mysqld]
general_log_file = /path/to/currentquery.log
general_log      = 1
```

TABLElog_output FILE

MySQL

o

```
2016-08-02 20:40:39 2420 [Note] Shutting down plugin 'binlog'
2016-08-02 20:40:39 2420 [Note] mysqld: Shutdown complete

2016-08-02 20:43:11 2888 [Note] Plugin 'FEDERATED' is disabled.
2016-08-02 20:43:11 2888 [Note] InnoDB: Using atomics to ref count buffer pool pages
2016-08-02 20:43:11 2888 [Note] InnoDB: The InnoDB memory heap is disabled
```

log_error

log_errordatadir@hostname .err log_error cnf Manual Page

o o o

GLOBAL log_warnings

```
SELECT @@log_warnings; -- make a note of your prior setting
SET GLOBAL log_warnings=2; -- setting above 1 increases output (see server version)
```

log_warnings

cnf

```
[mysqld]
log_error      = /path/to/CurrentError.log
log_warnings   = 2
```

MySQL 5.7.23 GLOBAL log_error_verbosity 5.7.2 cnf

MySQL 5.7.2

```
[mysqld]
log_error          = /path/to/CurrentError.log
log_warnings       = 2
log_error_verbosity = 3
```

MySQLlog_warningserror_log_verbosity°

<https://riptutorial.com/zh-TW/mysql/topic/5102/>

45:

MySQL 5.6.4.

DATETIME (3) TIMESTAMP (6) * nix.

<http://dev.mysql.com/doc/refman/5.7/en/fractional-seconds.html>

NOW(3) MySQL.

MySQL* 0.001IEEE754.

Examples

```
SELECT NOW(3)
```

o

Javascript.

JavascriptUNIX time_t1970-01-01 00:00:00 UTC.

Javascript. time_zone.

```
ROUND (UNIX_TIMESTAMP (NOW(3)) * 1000.0, 0)
```

TIMESTAMPUNIX_TIMESTAMPJavascript.

```
SELECT ROUND (UNIX_TIMESTAMP (column) * 1000.0, 0)
```

DATETIMEJavascript.

o

```
CREATE TABLE times (  
  dt DATETIME(3),  
  ts TIMESTAMP(3)  
);
```

/o

```
INSERT INTO times VALUES (NOW(3), NOW(3));
```

NOW().

```
INSERT INTO times VALUES ('2015-01-01 16:34:00.123','2015-01-01 16:34:00.128');
```

◦

NOW(3)NOW() ◦

/◦

%f **DATE_FORMAT**◦

```
SELECT DATE_FORMAT(NOW(3), '%Y-%m-%d %H:%i:%s.%f')
```

2016-11-19 09:52:53.248000 ◦ NOW(3) 0◦

JavascriptTIMESTAMP

Javascript1478960868932 MySQL

```
FROM_UNIXTIME(1478960868932 * 0.001)
```

JavascriptMySQL◦

```
INSERT INTO table (col) VALUES (FROM_UNIXTIME(1478960868932 * 0.001))
```

◦

<https://riptutorial.com/zh-TW/mysql/topic/7850/>

46:

Examples

LinuxMySQL root

MySQLroot

1MySQL。

- **UbuntuDebian**
sudo /etc/init.d/mysql stop
- **CentOSFedoraRed Hat Enterprise Linux**
sudo /etc/init.d/mysqld stop

2MySQL。

```
sudo mysqld_safe --skip-grant-tables &
```

mysqld_safe

```
sudo mysqld --skip-grant-tables &
```

3MySQL。

```
mysql -u root
```

4root。

5.7

```
FLUSH PRIVILEGES;  
ALTER USER 'root'@'localhost' IDENTIFIED BY 'new_password';  
FLUSH PRIVILEGES;  
exit;
```

5.7

```
FLUSH PRIVILEGES;  
SET PASSWORD FOR 'root'@'localhost' = PASSWORD('new_password');  
FLUSH PRIVILEGES;  
exit;
```

ALTER USERMySQL 5.7.6。

5MySQL。

- **UbuntuDebian**
sudo /etc/init.d/mysql stop


```
sudo /etc/init.d/mysql start
```

- **CentOS/Fedora/Red Hat Enterprise Linux**

```
sudo /etc/init.d/mysqld stop
```

```
sudo /etc/init.d/mysqld start
```

Windows MySQL root

Windows root

1

Press **Ctrl+R** Goto Start Menu > Run cmd **Enter**

2 MySQL

```
C:\> cd C:\mysql\bin
```

3 mysql

```
C:\mysql\bin> mysql -u root mysql
```

4 root

```
mysql> SET PASSWORD FOR root@localhost=PASSWORD('my_new_password');
```

1. MySQL `mysqld/`.

2. MySQL `--skip-grant-tables` `mysqld_safe --skip-grant-tables &`

3. root MySQL `mysql -u root`

4.

- **5.7.6** `ALTER USER 'root'@'localhost' IDENTIFIED BY 'new-password';`

- **5.7.5 MariaDB** `SET PASSWORD FOR 'root'@'localhost' = PASSWORD('new-password'); flush privileges; quit;`

5. MySQL.

o

<http://dev.mysql.com/doc/refman/5.7/en/resetting-permissions.html>

<https://riptutorial.com/zh-TW/mysql/topic/2761/>

47:

- ALTER [IGNORE] TABLE tbl_name [alter_specification [alter_specification] ...] [partition_options]

```
alter_specification: table_options
| ADD [COLUMN] col_name column_definition [FIRST | AFTER col_name ]
| ADD [COLUMN] (col_name column_definition,...)
| ADD {INDEX|KEY} [index_name] [index_type] (index_col_name,...) [index_option] ...
| ADD [CONSTRAINT [symbol]] PRIMARY KEY [index_type] (index_col_name,...) [index_option]
...
| ADD [CONSTRAINT [symbol]] UNIQUE [INDEX|KEY] [index_name] [index_type]
(index_col_name,...) [index_option] ...
| ADD FULLTEXT [INDEX|KEY] [index_name] (index_col_name,...) [index_option] ...
| ADD SPATIAL [INDEX|KEY] [index_name] (index_col_name,...) [index_option] ...
| ADD [CONSTRAINT [symbol]] FOREIGN KEY [index_name] (index_col_name,...)
reference_definition
| ALGORITHM [=] {DEFAULT|INPLACE|COPY}
| ALTER [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}
| CHANGE [COLUMN] old_col_name new_col_name column_definition [FIRST|AFTER col_name]
| LOCK [=] {DEFAULT|NONE|SHARED|EXCLUSIVE}
| MODIFY [COLUMN] col_name column_definition [FIRST | AFTER col_name]
| DROP [COLUMN] col_name
| DROP PRIMARY KEY
| DROP {INDEX|KEY} index_name
| DROP FOREIGN KEY fk_symbol
| DISABLE KEYS
| ENABLE KEYS
| RENAME [TO|AS] new_tbl_name
| RENAME {INDEX|KEY} old_index_name TO new_index_name
| ORDER BY col_name [, col_name] ...
| CONVERT TO CHARACTER SET charset_name [COLLATE collation_name]
| [DEFAULT] CHARACTER SET [=] charset_name [COLLATE [=] collation_name]
| DISCARD TABLESPACE
| IMPORT TABLESPACE
| FORCE
| {WITHOUT|WITH} VALIDATION
| ADD PARTITION (partition_definition)
| DROP PARTITION partition_names
| DISCARD PARTITION {partition_names | ALL} TABLESPACE
| IMPORT PARTITION {partition_names | ALL} TABLESPACE
| TRUNCATE PARTITION {partition_names | ALL}
| COALESCE PARTITION number
| REORGANIZE PARTITION partition_names INTO (partition_definitions)
| EXCHANGE PARTITION partition_name WITH TABLE tbl_name [{WITH|WITHOUT} VALIDATION]
| ANALYZE PARTITION {partition_names | ALL}
| CHECK PARTITION {partition_names | ALL}
| OPTIMIZE PARTITION {partition_names | ALL}
| REBUILD PARTITION {partition_names | ALL}
| REPAIR PARTITION {partition_names | ALL}
| REMOVE PARTITIONING
| UPGRADE PARTITIONING
index_col_name: col_name [(length)] [ASC | DESC]
index_type: USING {BTREE | HASH}
index_option: KEY_BLOCK_SIZE [=] value
| index_type
| WITH PARSE parser_name
| COMMENT 'string'
```

table_options: table_option [[,] table_option] ... (see [CREATE TABLE options](#))

partition_options: (see [CREATE TABLE options](#))

MySQL 5.7/ ... / ALTER TABLE/ 14.1.8 ALTER TABLE

Examples

;;file_per_table

t1InnoDBInnoDB

```
ALTER TABLE t1 ENGINE = InnoDB;
```

InnoDBOPTIMIZE TABLE◦ ◦

innodb_file_per_tablet1file_per_table◦

```
CREATE DATABASE stackoverflow;

USE stackoverflow;

Create table stack(
    id_user int NOT NULL,
    username varchar(30) NOT NULL,
    password varchar(30) NOT NULL
);

ALTER TABLE stack ADD COLUMN submit date NOT NULL; -- add new column
ALTER TABLE stack DROP COLUMN submit; -- drop column
ALTER TABLE stack MODIFY submit DATETIME NOT NULL; -- modify type column
ALTER TABLE stack CHANGE submit submit_date DATETIME NOT NULL; -- change type and name of column
ALTER TABLE stack ADD COLUMN mod_id INT NOT NULL AFTER id_user; -- add new column after existing column
```

ALTERINDEX

```
ALTER TABLE TABLE_NAME ADD INDEX `index_name` (`column_name`)
```

```
ALTER TABLE TABLE_NAME ADD INDEX `index_name` (`col1`,`col2`)
```

AUTO_INCREMENT◦

◦ AUTO_INCREMENTMAX100◦ ◦

```
ALTER TABLE your_table_name AUTO_INCREMENT = 101;
```

```
ALTER TABLE fish_data.fish DROP PRIMARY KEY;
```

```
ALTER TABLE fish_data.fish MODIFY COLUMN fish_id DECIMAL(20,0) NOT NULL PRIMARY KEY;
```

◦

dbdb

```
users (  
  firstname varchar(20),  
  lastname varchar(20),  
  age char(2)  
)
```

agecharint

```
ALTER TABLE users CHANGE age age tinyint UNSIGNED NOT NULL;
```

```
ALTER TABLE table_name CHANGE column_name new_column_definition
```

MySQL

MySQL

```
mysqladmin -uroot -p<password> create <new name>  
mysqldump -uroot -p<password> --routines <old name> | mysql -uroot -pmypassword <new name>  
mysqladmin -uroot -p<password> drop <old name>
```

1. ◦
2. <old name> <new name><password>+root◦
3. MySQL“bin”“y”◦

dbdb◦

```
RENAME TABLE `<old db>`.`<name>` TO `<new db>`.`<name>`;
```

```
SELECT CONCAT('RENAME TABLE old_db.', table_name, ' TO ',  
              'new_db.', table_name)  
FROM information_schema.TABLES  
WHERE table_schema = 'old_db';
```

◦ ◦ MyISAMInnoDB◦ “InnoDB◦ DROpping InnoDBPARTITION”◦ ◦

MySQL

MySQL <db1><db2>

```
mysqladmin -uroot -p<password> create swaptmp  
mysqldump -uroot -p<password> --routines <db1> | mysql -uroot -p<password> swaptmp  
mysqladmin -uroot -p<password> drop <db1>
```

```
mysqladmin -uroot -p<password> create <db1>
mysqldump -uroot -p<password> --routines <db2> | mysql -uroot -p<password> <db1>
mysqladmin -uroot -p<password> drop <db2>
mysqladmin -uroot -p<password> create <db2>
mysqldump -uroot -p<password> --routines swaptmp | mysql -uroot -p<password> <db2>
mysqladmin -uroot -p<password> drop swaptmp
```

1. ◦
2. <db1> <db2><password>+root◦
3. MySQL“bin”“y”◦

MySQL

```
RENAME TABLE `<old name>` TO `<new name>`;
```

```
ALTER TABLE `<old name>` RENAME TO `<new name>`;
```

ALTER TABLE◦

1. <old name><new name>◦ *dbname*◦ *tablename<old name>/<new name>*◦
2. MySQLMySQL Workbench◦ **ALTERDROPCREATEINSERT**◦

MySQL

“”◦

```
ALTER TABLE `<table name>` CHANGE `<old name>` `<new name>` <column definition>;
```

1. MySQLMySQL Workbench◦
2. SHOW CREATE TABLE <table name>; <table name>◦
3. ◦
4. <old name> <new name><column definition>◦

<https://riptutorial.com/zh-TW/mysql/topic/2627/>

48:

	o o
SESSION	o o

Examples

SHOW VARIABLES

SQLPHPMysqlAdminMySQL CLI

```
SHOW VARIABLES;
```

```
SHOW SESSION VARIABLES;
```

```
SHOW GLOBAL VARIABLES;
```

SQLLIKE

```
SHOW [GLOBAL | SESSION] VARIABLES LIKE 'max_join_size';
```

```
SHOW [GLOBAL | SESSION] VARIABLES LIKE '%size%';
```

WHERE SHOW

```
SHOW [GLOBAL | SESSION] VARIABLES WHERE VALUE > 0;
```

PHPMyAdminSQLMySQL CLI

```
SHOW STATUS;
```

SESSION GLOBAL

```
SHOW SESSION STATUS;
```

```
SHOW GLOBAL STATUS;
```

SQLLIKE

```
SHOW [GLOBAL | SESSION] STATUS LIKE 'Key%';
```

Where

```
SHOW [GLOBAL | SESSION] STATUS WHERE VALUE > 0;
```

GLOBALSESSIONGLOBALSESSION。

<https://riptutorial.com/zh-TW/mysql/topic/9924/>

49:

Examples

SQLMySQLEVENTcron。 。 。

。

```
SHOW VARIABLES WHERE variable_name='event_scheduler';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| event_scheduler | OFF |
+-----+-----+
```

。

```
SET GLOBAL event_scheduler = ON;
```

```
create table theMessages
(
  id INT AUTO_INCREMENT PRIMARY KEY,
  userId INT NOT NULL,
  message VARCHAR(255) NOT NULL,
  updateDt DATETIME NOT NULL,
  KEY(updateDt)
);

INSERT theMessages(userId,message,updateDt) VALUES (1,'message 123','2015-08-24 11:10:09');
INSERT theMessages(userId,message,updateDt) VALUES (7,'message 124','2015-08-29');
INSERT theMessages(userId,message,updateDt) VALUES (1,'message 125','2015-09-03 12:00:00');
INSERT theMessages(userId,message,updateDt) VALUES (1,'message 126','2015-09-03 14:00:00');
```

。 2。

21102

。 INTERVAL。

```
DROP EVENT IF EXISTS `delete7DayOldMessages`;
DELIMITER $$
CREATE EVENT `delete7DayOldMessages`
  ON SCHEDULE EVERY 1 DAY STARTS '2015-09-01 00:00:00'
  ON COMPLETION PRESERVE
DO BEGIN
  DELETE FROM theMessages
  WHERE datediff(now(),updateDt)>6; -- not terribly exact, yesterday but <24hrs is still 1
  day

  -- Other code here
```



```
END$$
DELIMITER ;
```

...

```
DROP EVENT IF EXISTS `Every_10_Minutes_Cleanup`;
DELIMITER $$
CREATE EVENT `Every_10_Minutes_Cleanup`
  ON SCHEDULE EVERY 10 MINUTE STARTS '2015-09-01 00:00:00'
  ON COMPLETION PRESERVE
DO BEGIN
  DELETE FROM theMessages
  WHERE TIMESTAMPDIFF(HOUR, updatedDt, now())>168; -- messages over 1 week old (168 hours)

  -- Other code here
END$$
DELIMITER ;
```

```
SHOW EVENTS FROM my_db_name; -- List all events by schema name (db name)
SHOW EVENTS;
SHOW EVENTS\G; -- <----- I like this one from mysql> prompt
```

```
***** 1. row *****
      Db: my_db_name
      Name: delete7DayOldMessages
      Definer: root@localhost
      Time zone: SYSTEM
      Type: RECURRING
      Execute at: NULL
      Interval value: 1
      Interval field: DAY
      Starts: 2015-09-01 00:00:00
      Ends: NULL
      Status: ENABLED
      Originator: 1
character_set_client: utf8
collation_connection: utf8_general_ci
  Database Collation: utf8_general_ci
***** 2. row *****
      Db: my_db_name
      Name: Every_10_Minutes_Cleanup
      Definer: root@localhost
      Time zone: SYSTEM
      Type: RECURRING
      Execute at: NULL
      Interval value: 10
      Interval field: MINUTE
      Starts: 2015-09-01 00:00:00
      Ends: NULL
      Status: ENABLED
      Originator: 1
character_set_client: utf8
collation_connection: utf8_general_ci
  Database Collation: utf8_general_ci
2 rows in set (0.06 sec)
```

```
DROP EVENT someEventName; -
```

ON COMPLETION PRESERVE - ON COMPLETION PRESERVE ◦ ◦

◦ ◦ ◦ ◦

```
quantity {YEAR | QUARTER | MONTH | DAY | HOUR | MINUTE |  
         WEEK | SECOND | YEAR_MONTH | DAY_HOUR | DAY_MINUTE |  
         DAY_SECOND | HOUR_MINUTE | HOUR_SECOND | MINUTE_SECOND}
```

◦ DDL DML. “MySQL”.

<https://riptutorial.com/zh-TW/mysql/topic/4319/>

50:

- PREPARE stmt_name FROM preparable_stmt
- EXECUTE stmt_name [@var_name [@ var_name] ...]
- {DEALLOCATE | DROP} PREPARE stmt_name

Examples

DEALLOCATE PREPARE

PREPARE

EXECUTE

DEALLOCATE PREPARE

```
SET @s = 'SELECT SQRT(POW(?,2) + POW(?,2)) AS hypotenuse';
PREPARE stmt2 FROM @s;
SET @a = 6;
SET @b = 8;
EXECUTE stmt2 USING @a, @b;
```

```
+-----+
| hypotenuse |
+-----+
|          10 |
+-----+
```

```
DEALLOCATE PREPARE stmt2;
```

- @variablesFROM @s @sDECLARED
- Prepare“”。

CONCAT SELECT +。 @variablesDECLARED - 。

```
SET v_column_definition := CONCAT(
    v_column_name
    , ' ', v_column_type
    , ' ', v_column_options
);

SET @stmt := CONCAT('ALTER TABLE ADD COLUMN ', v_column_definition);

PREPARE stmt FROM @stmt;
EXECUTE stmt;
DEALLOCATE PREPARE stmt;
```

<https://riptutorial.com/zh-TW/mysql/topic/2603/>

51: mysqlimport

<code>--delete -D</code>	
<code>--fields-optionally-enclosed-by</code>	
<code>--fields-terminated-by</code>	
<code>--ignore -i</code>	
<code>--lines-terminated-by</code>	
<code>--password -p</code>	
<code>--port -P</code>	
<code>--replace -r</code>	
<code>--user -u</code>	
<code>--where -w</code>	

mysqlimport^o

Examples

employee.txt

```
1 \t
2 \t
3 \t Zaphod Beeblebrox
```

```
$ mysql --user=user --password=password mycompany -e 'CREATE TABLE employee(id INT, name
VARCHAR(100), PRIMARY KEY (id))'
```

```
$ mysqlimport --user=user --password=password mycompany employee.txt
```

employee.txt

```
1 | Arthur Dent
2 |
3 | Zaphod Beeblebrox
```

```
$ mysqlimport --fields-terminated-by='|' mycompany employee.txt
```

Windows

```
$ mysqlimport --lines-terminated-by='\r\n' mycompany employee.txt
```

Employee

ID	
3	Yooden Vranx

employee.txt

```
1 \t
2 \t
3 \t Zaphod Beeblebrox
```

--ignore

```
$ mysqlimport --ignore mycompany employee.txt
```

ID	
1	
2	
3	Yooden Vranx

--replace

```
$ mysqlimport --replace mycompany employee.txt
```

ID	
1	
2	
3	Zaphod Beeblebrox

```
$ mysqlimport --where="id>2" mycompany employee.txt
```

CSV

```
$ mysqlimport
  --fields-optionally-enclosed-by='"'
  --fields-terminated-by=,
  --lines-terminated-by="\r\n"
mycompany employee.csv
```

mysqlimport <https://riptutorial.com/zh-TW/mysql/topic/5215/mysqlimport>

52:

Examples

NULL

- - end_date rating
- - middle_initial
- 0/0 - °
- NULL ""0°
-

NULL

- IS NULL / IS NOT NULL - = NULL°
- x <=> y ""°

LEFT JOINab °

```
SELECT ...
  FROM a
 LEFT JOIN b ON ...
 WHERE b.id IS NULL
```

<https://riptutorial.com/zh-TW/mysql/topic/6757/>

53:

MySQL [64IEEE 754](#).

◦

- RAND () ◦

Examples

MySQL

		SELECT 3+5; -> 8
+		SELECT 3.5+2.5; -> 6.0
		SELECT 3.5+2; -> 5.5
-		SELECT 3-5; -> -2
*		SELECT 3 * 5; -> 15
/		SELECT 20 / 4; -> 5
		SELECT 355 / 113; -> 3.1416
		SELECT 10.0 / 0; -> NULL
DIV		SELECT 5 DIV 2; -> 2
%MOD		SELECT 7 % 3; -> 1
		SELECT 15 MOD 4 -> 3
		SELECT 15 MOD -4 -> 3
		SELECT -15 MOD 4 -> -3
		SELECT -15 MOD -4 -> -3
		SELECT 3 MOD 2.5 -> 0.5

BIGINT

MySQL [BIGINT 64](#).

```
select (1024 * 1024 * 1024 * 1024 *1024 * 1024) + 1 -> 1,152,921,504,606,846,977
```

```
select (1024 * 1024 * 1024 * 1024 *1024 * 1024 * 1024 -> BIGINT
```

MySQL [64IEEE 754](#) ◦ ◦

PI [6](#) ◦ DOUBLE;


```
SELECT PI(); -> 3.141593
```

SINCOS

- IEEE 754 64◦ epsilon ◦ ;◦

DECIMAL◦

X

```
SELECT SIN(PI()); -> 1.2246063538224e-16
```

XX

```
SELECT COS(PI()); -> -1
```

X◦ ◦ ε◦

```
SELECT TAN(PI()); -> -1.2246063538224e-16
```

X₋₁ to 1X

```
SELECT ACOS(1); -> 0
SELECT ACOS(1.01); -> NULL
```

X₋₁ to 1X

```
SELECT ASIN(0.2); -> 0.20135792079033
```

ATAN(x)◦

```
SELECT ATAN(2); -> 1.1071487177941
```

ATAN2(X, Y)XY◦ Y / X◦ Xt◦

ATAN2() ATAN()◦

```
ATAN2(1,1); -> 0.7853981633974483 (45 degrees)
ATAN2(1,-1); -> 2.356194490192345 (135 degrees)
ATAN2(0, -1); -> PI (180 degrees) don't try ATAN(-1 / 0)... it won't work
```

X

```
SELECT COT(12); -> -1.5726734063977
```

```
SELECT RADIANS(90) -> 1.5707963267948966
SELECT SIN(RADIANS(90)) -> 1
SELECT DEGREES(1), DEGREES(PI()) -> 57.29577951308232, 180
```

ROUNDFLOORCEIL

DECIMAL 5。 4。

```
SELECT ROUND(4.51) -> 5
SELECT ROUND(4.49) -> 4
SELECT ROUND(-4.51) -> -5
```

DOUBLE ROUND() **C**; ROUND()

```
SELECT ROUND(45e-1) -> 4 -- The nearest even value is 4
SELECT ROUND(55e-1) -> 6 -- The nearest even value is 6
```

CEIL() CEILING()

```
SELECT CEIL(1.23) -> 2
SELECT CEILING(4.83) -> 5
```

FLOOR()

```
SELECT FLOOR(1.99) -> 1
```

FLOORCEIL/-infinity

```
SELECT FLOOR(-1.01), CEIL(-1.01) -> -2 and -1
SELECT FLOOR(-1.99), CEIL(-1.99) -> -2 and -1
```

o

```
SELECT ROUND(1234.987, 2) -> 1234.99
SELECT ROUND(1234.987, -2) -> 1200
```

“5”。

POW

xy POW() POWER()

```
SELECT POW(2,2); => 4
SELECT POW(4,2); => 16
```

SQRT

SQRT() ◦ NULL

```
SELECT SQRT(16); -> 4
SELECT SQRT(-3); -> NULL
```

01RAND()

```
SELECT i, RAND() FROM t;
```

	RAND
1	0.6191438870682
2	0.93845168309142
3	0.83482678498591

$a \leq n \leq b$

```
FLOOR(a + RAND() * (b - a + 1))
```

712

```
SELECT FLOOR(7 + (RAND() * 6));
```

```
SELECT * FROM tbl ORDER BY RAND();
```

◦

MySQL ◦ MySQLMySQL ◦

ABS

```
SELECT ABS(2); -> 2
SELECT ABS(-46); -> 46
```

sign0 ◦

-1	$n < 0$	SELECT SIGN(42); -> 1
0	$n = 0$	SELECT SIGN(0); -> 0
1	$n > 0$	SELECT SIGN(-3); -> -1

```
SELECT SIGN(-423421); -> -1
```

<https://riptutorial.com/zh-TW/mysql/topic/4516/>

54:

- -

```
CREATE INDEX index_name ON table_name column_name1 [ column_name2 ...]
```

- -

```
CREATE UNIQUE INDEX index_name ON table_name column_name1 [ column_name2 ...]
```

- -

```
DROP INDEX index_name ON tbl_name [ algorithm_option | lock_option ] .....
```

```
algorithm_option ALGORITHM [=] {DEFAULT | INPLACE | COPY}
```

```
lock_option LOCK [=] {DEFAULT | NONE | SHARED | EXCLUSIVE}
```

MySQL。

◦ “” 113-120,231354。 ◦

-

- “”1-59,61-290292-400。 ◦ “”。
- 10105 - 10。
- - “L”。

Examples

```
-- Create an index for column 'name' in table 'my_table'  
CREATE INDEX idx_name ON my_table(name);
```

◦ NULL NULLNULL

```
-- Creates a unique index for column 'name' in table 'my_table'  
CREATE UNIQUE INDEX idx_name ON my_table(name);
```

```
-- Drop an index for column 'name' in table 'my_table'  
DROP INDEX idx_name ON my_table;
```

key mystringmydatetimeWHERE◦

```
CREATE INDEX idx_mycol_myothercol ON my_table(mycol, myothercol)
```

WHERE

◦ WHEREmycolWHEREmyothercol mycol ◦ ◦

BTREE◦ DATETIMEWHERE datecol > '2016-01-01 00:00:00' ◦ **BTREE**◦

AUTO_INCREMENT

```
CREATE TABLE (  
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,  
    ...  
    PRIMARY KEY(id),  
    ... );
```

- INSERT 11 NULL ◦
- ID.....
- id◦
- 'next'"MAX(id)+1 ◦
- id id◦ *auto_increments* ;◦
- auto_increment_offsetauto_increment_increment ◦
- PRIMARY KEYINDEX(id) ◦ ◦
- AUTO_INCREMENT“ PARTITION” ;◦
- “”◦ INSERT IGNORE **dup** REPLACE DELETEINSERT ◦ ROLLBACKID◦
- ID◦ idInnoDBCOMMIT◦

<https://riptutorial.com/zh-TW/mysql/topic/1748/>

55:

Examples

“MySQL Cluster”.....

- NDB - . . .
- Galera Cluster Percona XtraDB PXCGariraMariaDB. - MySQL; .

“” .

“” PRIMARY KEY .

<https://riptutorial.com/zh-TW/mysql/topic/5130/>

56:

- UNION DISTINCT - SELECT
- UNION ALL -
- UNION - DISTINCT
- SELECT ... UNION SELECT ... - ORDER BY
- SELECT ...UNIONSELECT ...ORDER BY ... -

UNIONCPU。

UNION*。 *5.7.3 / MariaDB 10.1UNIONtmp。

Examples

SELECTUNION

UNION。

authorseditorsUNION

```
select name, email, phone_number
from authors

union

select name, email, phone_number
from editors
```

union。 ALL UNION ALL 。

UNION

```
( SELECT ... )
UNION
( SELECT ... )
ORDER BY
```

ORDER BYSELECT。

OFFSET

UNIONLIMIT

```
( SELECT ... ORDER BY x LIMIT 10 )
UNION
( SELECT ... ORDER BY x LIMIT 10 )
ORDER BY x LIMIT 10
```


“10”SELECT10ORDER BYLIMIT。

104

```
( SELECT ... ORDER BY x LIMIT 40 )  
UNION  
( SELECT ... ORDER BY x LIMIT 40 )  
ORDER BY x LIMIT 30, 10
```

SELECT4UNIONOFFSET。

```
SELECT name, caption as title, year, pages FROM books  
UNION  
SELECT name, title, year, 0 as pages FROM movies
```

2。

UNION ALLUNION

1,22,44 UNION SELECT 2,33,55

信息	结果1	概况	状态
1	22	44	
▶ 1	22	44	
2	33	55	

SELECT 1,22,44 UNION SELECT 2,33,55 UNION SELECT 2,33,55

。

UNION ALL

SELECT 1,22,44 UNION SELECT 2,33,55 UNION ALL SELECT 2,33,55

信息	结果1	概况	状态
1	22	44	
▶ 1	22	44	
2	33	55	
2	33	55	

MySQL

UNION ALL

```
SELECT YEAR(date_time_column), MONTH(date_time_column), MIN(DATE(date_time_column)),  
MAX(DATE(date_time_column)), COUNT(DISTINCT (ip)), COUNT(ip), (COUNT(ip) / COUNT(DISTINCT  
(ip))) AS Ratio
```

```
FROM (  
    (SELECT date_time_column, ip FROM server_log_1 WHERE state = 'action' AND log_id = 150)  
UNION ALL  
    (SELECT date_time_column, ip FROM server_log_2 WHERE state = 'action' AND log_id = 150)  
UNION ALL  
    (SELECT date_time_column, ip FROM server_log_3 WHERE state = 'action' AND log_id = 150)  
UNION ALL  
    (SELECT date_time_column, ip FROM server_log WHERE state = 'action' AND log_id = 150)  
 ) AS table_all  
GROUP BY YEAR(date_time_column), MONTH(date_time_column);
```

<https://riptutorial.com/zh-TW/mysql/topic/3847/>

57:

Examples

- MySQL3.23◦

- ◦

-

```
--->Basic temporary table creation
CREATE TEMPORARY TABLE tempTable1(
    id INT NOT NULL AUTO_INCREMENT,
    title VARCHAR(100) NOT NULL,
    PRIMARY KEY ( id )
);
```

```
--->Temporary table creation from select query
CREATE TEMPORARY TABLE tempTable1
    SELECT ColumnName1,ColumnName2,... FROM table1;
```

```
CREATE TEMPORARY TABLE tempTable1
    ( PRIMARY KEY(ColumnName2) )
    SELECT ColumnName1,ColumnName2,... FROM table1;
```

IF NOT EXISTS ""◦ ◦

```
CREATE TEMPORARY TABLE IF NOT EXISTS tempTable1
    SELECT ColumnName1,ColumnName2,... FROM table1;
```

Drop Temporary Table◦

```
DROP TEMPORARY TABLE tempTable1
```

```
DROP TEMPORARY TABLE IF EXISTS tempTable1
```

IF EXISTS

<https://riptutorial.com/zh-TW/mysql/topic/5757/>

58: PS1

Examples

MySQL PS1

.bashrc.bash_profile

```
export MYSQL_PS1="\u@\h [\d]>"
```

MySQLPROMPT@[]。

```
✓ [23:06:51] wenzhong@musicforever:~
$ mysql -uroot data
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.6.23 Homebrew

Copyright (c) 2000, 2015, 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.

root@localhost [data]>|
```

MySQLPS1

mysqld.cnf

```
[mysql]
prompt = '\u@\h [\d]> '
```

.bashrc

PS1 <https://riptutorial.com/zh-TW/mysql/topic/5795/ps1>

59:

MySQLTIMESTAMP。

◦ VARCHAR64 ◦ America/Edmonton ◦ Asia/KolkataAustralia/NSW ◦ WordPress.org。

JavaphpDBMSSQL

```
SET SESSION time_zone='(whatever tz string the user gave you)'
```

◦ TIMESTAMP。

UTC。NOWCURDATE。TIMESTAMPDATETIMEDATE。

MySQLUTC。◦ MySQL。

Examples

◦

NOW() NOW() NOW() **UTC**。

```
SELECT NOW();
SET time_zone='Asia/Kolkata';
SELECT NOW();
SET time_zone='UTC';
SELECT NOW();
```

DATE DATETIME。

DATEDATETIME MySQL。◦ CONVERT_TZ()。◦

```
SELECT CONVERT_TZ(date_sold, 'UTC', 'America/Los_Angeles') date_sold_local
FROM sales
WHERE state_sold = 'CA'
```

TIMESTAMP

◦ TIMESTAMptime_zone。

```
SET SESSION time_zone='America/Los_Angeles';
SELECT timestamp_sold
FROM sales
WHERE state_sold = 'CA'
```

TIMESTAMP **UNIX** time_t。 **UNIX**1970-01-01 00:00:00 **UTC**1970-01-01 00:00:00。

TIMESTAMP ◦ DATEDATETIME ◦

time_zone ◦

```
SELECT @@time_zone
```

SYSTEM MySQL ◦

UTC ◦

```
CREATE TEMPORARY TABLE times (dt DATETIME, ts TIMESTAMP);
SET time_zone = 'UTC';
INSERT INTO times VALUES (NOW(), NOW());
SET time_zone = 'SYSTEM';
SELECT dt, ts, TIMESTAMPDIFF(MINUTE, dt, ts)offset FROM times;
DROP TEMPORARY TABLE times;
```

◦ DATETIME TIMESTAMP UTC ◦ time_zone UTC INSERT / ◦

time_zone SELECT ◦ TIMESTAMP UTC SELECT ◦ DATETIME ◦ [TIMESTAMPDIFF \(MINUTE...\)](#) ◦

time_zone

MySQL time_zone ◦

```
SELECT mysql.time_zone_name.name
```

Paul Eggert [ZoneInfo](#) ◦ 600 ◦

Unix Linux BSD Mac OS ◦ MySQL / ◦

Windows ◦ ZoneInfo ◦

<https://riptutorial.com/zh-TW/mysql/topic/7849/>

60:

Examples

NULL

MySQLSQL NULL。

- NULL

```
CREATE TABLE example
(`applicant_id` INT, `company_name` VARCHAR(255), `end_date` DATE);
```

applicant_id	company_name	end_date
1	Google	NULL
1	Initech	2013-01-31
2	Woodworking.com	2016-08-25
2	NY Times	2013-11-10
3	NFL.com	2014-04-13

2016-01-01NULL。

```
SELECT * FROM example WHERE end_date > '2016-01-01';
```

NULL

applicant_id	company_name	end_date
2	Woodworking.com	2016-08-25

MySQL <>=<>NULLTRUEFALSE。 NULL end_date2016-01-012016-01-01。

IS NULL

```
SELECT * FROM example WHERE end_date > '2016-01-01' OR end_date IS NULL;
```

applicant_id	company_name	end_date
1	Google	NULL
2	Woodworking.com	2016-08-25

MAX() GROUP BYNULL。 applicant_id

```
SELECT applicant_id, MAX(end_date) FROM example GROUP BY applicant_id;
```

```
+-----+-----+
| applicant_id | MAX(end_date) |
+-----+-----+
|           1 | 2013-01-31   |
|           2 | 2016-08-25   |
|           3 | 2014-04-13   |
+-----+-----+
```

NULL◦ CASE WHENNULL

```
SELECT
  applicant_id,
  CASE WHEN MAX(end_date is null) = 1 THEN 'present' ELSE MAX(end_date) END
  max_date
FROM example
GROUP BY applicant_id;
```

```
+-----+-----+
| applicant_id | max_date      |
+-----+-----+
|           1 | present       |
|           2 | 2016-08-25   |
|           3 | 2014-04-13   |
+-----+-----+
```

example

```
SELECT
  data.applicant_id,
  data.company_name,
  data.max_date
FROM (
  SELECT
    *,
    CASE WHEN end_date is null THEN 'present' ELSE end_date END max_date
  FROM example
) data
INNER JOIN (
  SELECT
    applicant_id,
    CASE WHEN MAX(end_date is null) = 1 THEN 'present' ELSE MAX(end_date) END max_date
  FROM
    example
  GROUP BY applicant_id
) j
ON data.applicant_id = j.applicant_id AND data.max_date = j.max_date;
```

```
+-----+-----+-----+
| applicant_id | company_name  | max_date  |
+-----+-----+-----+
|           1 | Google        | present   |
|           2 | Woodworking.com | 2016-08-25 |
|           3 | NFL.com       | 2014-04-13 |
+-----+-----+-----+
```

MySQLNULL◦

<https://riptutorial.com/zh-TW/mysql/topic/5866/>

61:

- CREATE TABLE table_name column_name1 data_type size column_name2 data_type size column_name3 data_type size....; //
- CREATE TABLE table_name [IF NOT EXISTS] column_name1 data_type size column_name2 data_type size column_name3 data_type size....; //
- CREATE [TEMPORARY] TABLE table_name [IF NOT EXISTS] column_name1 data_type size column_name2 data_type size column_name3 data_type size....; //
- CREATE TABLE new_tbl [AS] SELECT * FROM orig_tbl; //SELECT

CREATE TABLE ENGINE

```
CREATE TABLE table_name ( column_definitions ) ENGINE=engine;
```

- InnoDB : (5.5.5 ACID) ◦ ◦
- MyISAM : (5.5.5) ◦
- Memory RAM ◦

◦

Examples

CREATE TABLE MySQL ◦

```
CREATE TABLE Person (
  `PersonID`      INTEGER NOT NULL PRIMARY KEY,
  `LastName`      VARCHAR(80),
  `FirstName`     VARCHAR(80),
  `Address`       TEXT,
  `City`          VARCHAR(100)
) Engine=InnoDB;
```

1. ◦ ` -chars ◦ fieldname space -chars ◦
2. ◦ [] CHAR VARCHAR ◦
3. ◦ NULL | NOT NULL NOT NULL NULL ◦
4. ◦

Engine=... ◦ InnoDB MyISAM ◦

DEFAULT

```
CREATE TABLE Address (
  `AddressID`     INTEGER NOT NULL PRIMARY KEY,
  `Street`        VARCHAR(80),
```

```

`City`          VARCHAR(80),
`Country`      VARCHAR(80) DEFAULT "United States",
`Active`      BOOLEAN DEFAULT 1,
) Engine=InnoDB;

```

StreetNULL ◦ Country “” ◦

BLOB TEXT GEOMETRYJSON ◦

```

CREATE TABLE Person (
  PersonID      INT UNSIGNED NOT NULL,
  LastName      VARCHAR(66) NOT NULL,
  FirstName     VARCHAR(66),
  Address       VARCHAR(255),
  City          VARCHAR(66),
  PRIMARY KEY (PersonID)
);

```

NOT NULL ◦ NOT NULL MySQL ◦

PRIMARY KEY ◦ InnoDBMySQL ◦

AUTO_INCREMENT INT “” ◦ 11 ◦

◦

INT TRUNCATE TABLE ◦

PRIMARY KEY

```

CREATE TABLE Person (
  PersonID      INT UNSIGNED NOT NULL PRIMARY KEY,
  LastName      VARCHAR(66) NOT NULL,
  FirstName     VARCHAR(66),
  Address       VARCHAR(255),
  City          VARCHAR(66)
);

```

◦

◦ ◦ PRIMARY KEY ◦ PRIMARY KEY inline ◦

```

CREATE TABLE invoice_line_items (
  LineNum       SMALLINT UNSIGNED NOT NULL,
  InvoiceNum     INT UNSIGNED NOT NULL,
  -- Other columns go here
  PRIMARY KEY (InvoiceNum, LineNum),
  FOREIGN KEY (InvoiceNum) REFERENCES -- references to an attribute of a table
);

```

◦

I/O. InnoDB PRIMARY KEY

```
CREATE TABLE Account (
  AccountID INT UNSIGNED NOT NULL,
  AccountNo INT UNSIGNED NOT NULL,
  PersonID INT UNSIGNED,
  PRIMARY KEY (AccountID),
  FOREIGN KEY (PersonID) REFERENCES Person (PersonID)
) ENGINE=InnoDB;
```

FK ◦ FK ◦ FK ◦

◦ FOREIGN KEY ◦ TEMPORARY

◦ ◦ ◦ ◦

InnoDB MyISAM MEMORY ◦ CREATE TABLE ◦ MySQL InnoDB ◦

```
CREATE TABLE ClonedPersons LIKE Persons;
```

◦

```
CREATE TABLE ClonedPersons SELECT * FROM Persons;
```

SELECT

```
CREATE TABLE ModifiedPersons
SELECT PersonID, FirstName + LastName AS FullName FROM Persons
WHERE LastName IS NOT NULL;
```

SELECT

```
CREATE TABLE ModifiedPersons (PRIMARY KEY (PersonID))
SELECT PersonID, FirstName + LastName AS FullName FROM Persons
WHERE LastName IS NOT NULL;
```

SELECT

CREATE TABLE SELECT CREATE TABLE

```
CREATE TABLE stack (
  id_user INT,
  username VARCHAR(30),
  password VARCHAR(30)
);
```

```
-- create a table from another table in the same database with all attributes
CREATE TABLE stack2 AS SELECT * FROM stack;
```

```
-- create a table from another table in the same database with some attributes
CREATE TABLE stack3 AS SELECT username, password FROM stack;
```

```
-- create a table from another table from another database with all attributes
CREATE TABLE stack2 AS SELECT * FROM second_db.stack;
```

```
-- create a table from another table from another database with some attributes
CREATE TABLE stack3 AS SELECT username, password FROM second_db.stack;
```

NB

```
FROM NAME_DATABASE.name_table
```

```
SHOW CREATE TABLE child; -- Option 1
```

```
CREATE TABLE `child` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `fullName` varchar(100) NOT NULL,
  `myParent` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `mommy_daddy` (`myParent`),
  CONSTRAINT `mommy_daddy` FOREIGN KEY (`myParent`) REFERENCES `parent` (`id`)
  ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

mysql

```
SHOW CREATE TABLE child \G
```

```
mysql> CREATE TABLE Tab1(id int, name varchar(30));
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> DESCRIBE Tab1; -- Option 2
```

```
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)       | YES  |     | NULL    |       |
| name  | varchar(30)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

DESCRIBEDESC。

DESCRIBE ◦

TimeStamp

TIMESTAMP。

```
CREATE TABLE `TestLastUpdate` (
  `ID` INT NULL,
```

```
`Name` VARCHAR(50) NULL,  
`Address` VARCHAR(50) NULL,  
`LastUpdate` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP  
)  
COMMENT='Last Update'  
;
```

<https://riptutorial.com/zh-TW/mysql/topic/795/>

62:

[Backup]MySQLMySQL。

Master - MySQL

Slave - MySQLMaster

MySQL。 。 。

。

SBR - SQL。 masterSQL。 SQL。

RBR - 。 。 。

MBR。 。 。

5.7.7MySQL。 MySQL 5.7.7。

Examples

-

2MySQLMasterSlave。

Master。 SlaveMaster。

Master。 SlaveMaster。

```
CREATE USER 'user_name'@'%' IDENTIFIED BY 'user_password';
GRANT REPLICATION SLAVE ON *.* TO 'user_name'@'%';
FLUSH PRIVILEGES;
```

user_nameuser_password。

my.inf Linuxmy.cnf。 [mysqld]。

```
server-id = 1
log-bin = mysql-bin.log
binlog-do-db = your_database
```

MySQLID。

MySQL。 Linuxlog-bin = /home/mysql/logs/mysql-bin.log。 MySQL。

。 your_database。

skip-networking MySQLMaster

my.infSlave [mysqld]

```
server-id = 2
master-host = master_ip_address
master-connect-retry = 60

master-user = user_name
master-password = user_password
replicate-do-db = your_database

relay-log = slave-relay.log
relay-log-index = slave-relay-log.index
```

MySQLID. ID.

IP. IP

◦

SlaveMaster.

◦

relay-logrelay-log-index◦

skip-networking MySQLSlave

MasterMaster. Master.

```
FLUSH TABLES WITH READ LOCK;
```

◦

mysqldumpMaster

```
mysqldump your_database -u root -p > D://Backup/backup.sql;
```

your_database◦ backup.sql◦

```
CREATE DATABASE `your_database`;
```

Slave MySQL.

```
mysql -u root -p your_database <D://Backup/backup.sql
--->Change `your_database` and backup directory according to your setup
```

Master. Master


```
SHOW MASTER STATUS;
```

```
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000001 | 130      | your_database |                   |
+-----+-----+-----+-----+
```

Slave

```
SLAVE STOP;
CHANGE MASTER TO MASTER_HOST='master_ip_address', MASTER_USER='user_name',
  MASTER_PASSWORD='user_password', MASTER_LOG_FILE='mysql-bin.000001', MASTER_LOG_POS=130;
SLAVE START;
```

◦ **Master** ◦ MASTER_LOG_FILE MASTER_LOG_POS SHOW MASTER STATUS ◦

MASTER_HOST IP ◦

◦

```
SHOW SLAVE STATUS;
```

Master FLUSH TABLES WITH READ LOCK

```
UNLOCK TABLES;
```

Master Slave Master ◦ Master Slave ◦

MySQL ◦ ◦

```
SET GLOBAL sql_slave_skip_counter = N;
```

N ◦ ◦

```
STOP SLAVE;
SET GLOBAL sql_slave_skip_counter=1;
START SLAVE;
```

◦ ◦

my.cnf ◦

```
1062 | Error 'Duplicate entry 'xyz' for key 1' on query
```

my.cnf

```
slave-skip-errors = 1062
```

o

```
slave-skip-errors=[err_code1,err_code2,...|all]
```

```
slave-skip-errors=1062,1053
```

```
slave-skip-errors=all
```

```
slave-skip-errors=ddl_exist_errors
```

<https://riptutorial.com/zh-TW/mysql/topic/7218/>

63:

- CREATE VIEW view_name AS SELECT column_names FROM table_name WHERE condition; ///
- CREATE [OR REPLACE] [ALGORITHM = {UNDEFINED || TEMPTABLE}] [DEFINER = {user | CURRENT_USER}] [SQL SECURITY {DEFINER | INVOKER}] VIEW view_name [column_list] AS select_statement [WITH [CASCADED | LOCAL] CHECK OPTION]; ///
- DROP VIEW [IF EXISTS] [db_name.] view_name; ///

VIEW_NAME	
SELECT	SQL. SELECT.

- ◦
- SELECT. SELECT FROM.
- SELECT.
-
- "Sales" - "Sales". ◦ ◦
- VIEWS. ◦ "" MySQL 5.7.6 Optimizer. VIEW.

Examples

CREATE VIEW
CREATE VIEW
SELECT. SELECT
SELECT. OR REPLACE
DROP. CREATE
VIEWS
SUPERDEFINER.

db_name.view_name

```
mysql> CREATE VIEW test.v AS SELECT * FROM t;
```

- ◦
- SELECT
-
- UNION
- SELECT

```
mysql> CREATE TABLE t (qty INT, price INT);
mysql> INSERT INTO t VALUES(3, 50);
mysql> CREATE VIEW v AS SELECT qty, price, qty*price AS value FROM t;
mysql> SELECT * FROM v;
+-----+-----+-----+
| qty | price | value |
+-----+-----+-----+
| 3 | 50 | 150 |
+-----+-----+-----+
```

- MySQL 5.7.7 SELECT FROM.
- SELECT.
- SELECT.
- SELECT.
- .
- . . CHECK TABLE.
- TEMPORARY TEMPORARY.
- .
- 64 SELECT 256.
- VIEW SELECT . .

```
CREATE VIEW myview AS
SELECT a.*, b.extra_data FROM main_table a
LEFT OUTER JOIN other_table b
ON a.id = b.id
```

mysql> SELECT * FROM myview mysql LEFT JOIN.

VIEW

VIEW UPDATE SELECT UPDATE .

GROUP BY UNION HAVING DISTINCT .

- .

```
CREATE VIEW few_rows_from_t1 AS SELECT * FROM t1 LIMIT 10;
DROP VIEW few_rows_from_t1;
```

- .

```
CREATE VIEW table_from_other_db AS SELECT x FROM db1.foo WHERE x IS NOT NULL;
DROP VIEW table_from_other_db;
```

<https://riptutorial.com/zh-TW/mysql/topic/1489/>

64:

Examples

SELECT

```
SELECT ... FROM ... WHERE ... GROUP BY ... HAVING ...
    ORDER BY ... -- goes here
    LIMIT ... OFFSET ...;

( SELECT ... ) UNION ( SELECT ... ) ORDER BY ... -- for ordering the result of the UNION.

SELECT ... GROUP_CONCAT(DISTINCT x ORDER BY ... SEPARATOR ...) ...

ALTER TABLE ... ORDER BY ... -- probably useful only for MyISAM; not for InnoDB
```

X

x°

- NULLsNULLs°
- ASC
- VARCHARCOLLATION
- ENUMs°

```
ORDER BY x ASC -- same as default
ORDER BY x DESC -- highest to lowest
ORDER BY lastname, firstname -- typical name sorting; using two columns
ORDER BY submit_date DESC -- latest first
ORDER BY submit_date DESC, id ASC -- latest first, but fully specifying order.
```

- ASC = ASCENDING DESC = DESCENDING
- DESC NULLs°
- INDEX(x) INDEX(lastname, firstname) INDEX(submit_date)°

.....ASCDESC° INDEX(submit_date DESC, id ASC) - “DESC”INDEX°

```
ORDER BY FIND_IN_SET(card_type, "MASTER-CARD,VISA,DISCOVER") -- sort 'MASTER-CARD' first.
ORDER BY x IS NULL, x -- order by `x`, but put `NULLs` last.
```

```
SELECT * FROM some_table WHERE id IN (118, 17, 113, 23, 72)
ORDER BY FIELD(id, 118, 17, 113, 23, 72);
```

id°

ID	...
118	...

ID	...
17	...
113	...
23	...
72	...

id。

<https://riptutorial.com/zh-TW/mysql/topic/5469/>

65: Mysql

-- SQL ◦

Examples

```
# This comment continues to the end of line

-- This comment continues to the end of line

/* This is an in-line comment */

/*
This is a
multiple-line comment
*/
```

```
SELECT * FROM t1; -- this is comment
```

```
CREATE TABLE stack(
  /*id_user int,
  username varchar(30),
  password varchar(30)
  */
  id int
);
```

----◦

```
#This comment works
/*This comment works.*/
--This comment does not.
```

```
CREATE TABLE menagerie.bird (
  bird_id INT NOT NULL AUTO_INCREMENT,
  species VARCHAR(300) DEFAULT NULL COMMENT 'You can include genus, but never subspecies.',
  INDEX idx_species (species) COMMENT 'We must search on species often.',
  PRIMARY KEY (bird_id)
) ENGINE=InnoDB COMMENT 'This table was inaugurated on February 10th.';
```

COMMENT=◦

SHOW CREATE TABLE information_schema◦

Mysql <https://riptutorial.com/zh-TW/mysql/topic/2337/mysql>

66:

MySQL GROUP_CONCAT() ◦ pivot group_concat_max_len

```
set session group_concat_max_len = 1024 * 1024; -- This should be enough for most cases
```

Examples

MySQL ◦ ◦

tbl_values

ID			
1			10
2			20
3			10

Name Value; Group Name ◦

```
-- 1. Create an expression that builds the columns
set @sql = (
  select group_concat(distinct
    concat(
      "sum(case when `Group`='", Group, "' then `Value` end) as `", `Group`, "`"
    )
  )
  from tbl_values
);

-- 2. Complete the SQL instruction
set @sql = concat("select Name, ", @sql, " from tbl_values group by `Name`");

-- 3. Create a prepared statement
prepare stmt from @sql;

-- 4. Execute the prepared statement
execute stmt;
```

		10
	10	20

```
deallocate prepare stmt;
```

SQL

<https://riptutorial.com/zh-TW/mysql/topic/3074/>

67: ...

1. SELECT expression1expression2... expression_n
2. aggregate_function
3. FROM
4. []
5. GROUP BY expression1expression2... expression_n;

expression1expression2... expression_n	GROUP BY.
aggregate_function	SUMCOUNTMINMAXAVG.
	◦ FROM.
	◦ ◦

MySQL GROUP BYSELECT.

[ONLY_FULL_GROUP_BY](#) ◦ SELECT ◦ [5.7.5](#) ◦ DBMS.

Examples

GROUP BYSUM

```
SELECT product, SUM(quantity) AS "Total quantity"
FROM order_details
GROUP BY product;
```

MIN

name departmentsalary◦

```
SELECT department, MIN(salary) AS "Lowest salary"
FROM employees
GROUP BY department;
```

- name◦ "groupwise max"◦

GROUP BYCOUNT

```
SELECT department, COUNT(*) AS "Man_Power"
FROM employees
GROUP BY department;
```

GROUP BYHAVING

```
SELECT department, COUNT(*) AS "Man_Power"  
FROM employees  
GROUP BY department  
HAVING COUNT(*) >= 10;
```

GROUP BY ... HAVINGSELECT ... WHERE°

HAVING Man_Power >= 10HAVING“”°

Group Concat

Group ConcatMySQL° Name(1):Score(*)

A +	
-	
C +	
D-	
-	

```
SELECT Name, GROUP_CONCAT(Score ORDER BY Score desc SEPERATOR ' ') AS Grades  
FROM Grade  
GROUP BY Name
```

```
+-----+-----+  
| Name | Grades |  
+-----+-----+  
| Adam | C+ B A- A+ |  
| Bill | D- |  
| John | A- |  
+-----+-----+
```

AGGREGATEGROUP BY

```
+-----+-----+-----+-----+-----+  
|orderid|customerid|customer|total|items|  
+-----+-----+-----+-----+-----+  
| 1 | 1 | Bob | 1300 | 10 |  
| 2 | 3 | Fred | 500 | 2 |  
| 3 | 5 | Tess | 2500 | 8 |  
| 4 | 1 | Bob | 300 | 6 |  
| 5 | 2 | Carly | 800 | 3 |  
| 6 | 2 | Carly | 1000 | 12 |  
| 7 | 3 | Fred | 100 | 1 |
```

8	5	Tess	11500	50
9	4	Jenny	200	2
10	1	Bob	500	15

-

WHERE ◦

◦

```
SELECT customer, COUNT(*) as orders
FROM orders
GROUP BY customer
ORDER BY customer
```

customer	orders
Bob	3
Carly	2
Fred	2
Jenny	1
Tess	2

-

◦

◦

```
SELECT customer, SUM(total) as sum_total, SUM(items) as sum_items
FROM orders
GROUP BY customer
ORDER BY customer
```

customer	sum_total	sum_items
Bob	2100	31
Carly	1800	15
Fred	600	3
Jenny	200	2
Tess	14000	58

• AVG

◦

◦

```
SELECT customer, AVG(total) as avg_total
FROM orders
GROUP BY customer
ORDER BY customer
```

```
+-----+-----+
| customer | avg_total |
+-----+-----+
| Bob      |      700 |
| Carly    |      900 |
| Fred     |      300 |
| Jenny    |      200 |
| Tess     |     7000 |
+-----+-----+
```

- **MAX**

◦

◦

```
SELECT customer, MAX(total) as max_total
FROM orders
GROUP BY customer
ORDER BY customer
```

```
+-----+-----+
| customer | max_total |
+-----+-----+
| Bob      |     1300 |
| Carly    |     1000 |
| Fred     |      500 |
| Jenny    |      200 |
| Tess     |    11500 |
+-----+-----+
```

- **MIN**

◦

◦

```
SELECT customer, MIN(total) as min_total
FROM orders
GROUP BY customer
ORDER BY customer
```

```
+-----+-----+
| customer | min_total |
+-----+-----+
| Bob      |      300 |
| Carly    |      800 |
| Fred     |      100 |
| Jenny    |      200 |
+-----+-----+
```

| Tess | 2500 |
+-----+-----+

... <https://riptutorial.com/zh-TW/mysql/topic/3523/--->

68: GRANTS

Examples

rootSUPER

```
GRANT ... TO root@localhost ...
```

◦ SUPER. SUPER.◦

```
GRANT ... ON dbname.* ...
```

dbname◦

```
GRANT SELECT ON dbname.* ... -- "read only"  
GRANT ... ON dbname.tblname ... -- "just one table"
```

readonly"

```
GRANT SELECT, CREATE TEMPORARY TABLE ON dbname.* ... -- "read only"
```

◦ ◦ ◦

root◦ SECURITY DEFINER root""◦ SP◦

@

""IP◦ ◦

```
GRANT SELECT ON db.* TO sam@'my.domain.com' IDENTIFIED BY 'foo';
```

```
localhost -- the same machine as mysqld  
'my.domain.com' -- a specific domain; this involves a lookup  
'11.22.33.44' -- a specific IP address  
'192.168.1.%' -- wild card for trailing part of IP address. (192.168.% and 10.% and 11.% are  
"internal" ip addresses.)
```

localhost◦ localhostroot◦ 0.0.0.1::1◦

GRANTS <https://riptutorial.com/zh-TW/mysql/topic/5131/grants>

69:

SELECT°

- SELECT DISTINCT [] FROM TableName [WHERE]; ///
- SELECT DISTINCTab ...SELECT DISTINCT ab...
- [[] DISTINCT | DISTINCTROW] [HIGH_PRIORITY] [STRAIGHT_JOIN] [SQL_SMALL_RESULT | SQL_BIG_RESULT] [SQL_BUFFER_RESULT] [SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]FROM[WHERE] [GROUP BY] [HAVING] [ORDER BY[ASC | DESC]] [LIMIT [offset_value] number_rows | LIMIT number_rows OFFSET offset_value] [PROCEDURE procedure_name] [INTO [OUTFILE'file_name'options | DUMPFILE'file_name'] @ variable1 @ variable2... @variable_n] [FOR UPDATE []]; ///

MySQLSELECT [MySQL Docs](#) °

Examples

```
CREATE TABLE stack(  
  id INT,  
  username VARCHAR(30) NOT NULL,  
  password VARCHAR(30) NOT NULL  
);  
  
INSERT INTO stack (`id`, `username`, `password`) VALUES (1, 'Foo', 'hiddenGem');  
INSERT INTO stack (`id`, `username`, `password`) VALUES (2, 'Baa', 'verySecret');
```

```
SELECT id FROM stack;
```

```
+-----+  
| id   |  
+-----+  
|  1  |  
|  2  |  
+-----+
```

*

```
SELECT * FROM stack;
```

```
+-----+-----+-----+  
| id   | username | password |  
+-----+-----+-----+  
|  1  | admin   | admin   |  
|  2  | stack   | stack   |  
+-----+-----+-----+  
2 rows in set (0.00 sec)
```

```
SELECT stack.* FROM stack JOIN Overflow ON stack.id = Overflow.id;
```

* ADD / DROP / MySQL.

1. /SELECT *
- 2.
3. SELECT * -usage

1. 200kVARBINARY. - SELECT *102MB
- 2.
- 3.
4. - @vinodadhikary
- 5.
6. SELECT *
- 7.
8. TEXT

WHERE

```
SELECT * FROM stack WHERE username = "admin" AND password = "admin";
```

```
+-----+-----+-----+
| id    | username | password |
+-----+-----+-----+
| 1     | admin    | admin    |
+-----+-----+-----+
1 row in set (0.00 sec)
```

WHERESELECT

WHERESELECT " "

o

```
SELECT title FROM books WHERE author_id = (SELECT id FROM authors WHERE last_name = 'Bar' AND first_name = 'Foo');
```

```
SELECT * FROM stack WHERE username IN (SELECT username FROM signups WHERE email IS NULL);
```

o

SELECT with LIKE

```
CREATE TABLE stack
( id int AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(100) NOT NULL
);
```

```
INSERT stack(username) VALUES
('admin'),('k admin'),('adm'),('a adm b'),('b XadmY c'), ('adm now'), ('not here');
```

“adm”

```
SELECT * FROM stack WHERE username LIKE "%adm%";
+----+-----+
| id | username |
+----+-----+
| 1 | admin    |
| 2 | k admin  |
| 3 | adm      |
| 4 | a adm b  |
| 5 | b XadmY c|
| 6 | adm now  |
+----+-----+
```

“adm”

```
SELECT * FROM stack WHERE username LIKE "adm%";
+----+-----+
| id | username |
+----+-----+
| 1 | admin    |
| 3 | adm      |
| 6 | adm now  |
+----+-----+
```

“adm”

```
SELECT * FROM stack WHERE username LIKE "%adm";
+----+-----+
| id | username |
+----+-----+
| 3 | adm      |
+----+-----+
```

LIKE% _°

```
SELECT * FROM stack WHERE username LIKE "adm_n";
+----+-----+
| id | username |
+----+-----+
| 1 | admin    |
+----+-----+
```

username

- LIKE 'adm'`='adm' LIKE 'adm'
- BETWEEN..AND.. LIKE 'adm%'“BETWEEN..AND..°
- LIKE '%adm' ° ° °
- RLIKE REGEXP LIKE°
- MySQLFULLTEXTFULLTEXTLIKE°

SELECT with AliasAS

SQL. ◦

```
SELECT username AS val FROM stack;  
SELECT username val FROM stack;
```

AS◦

```
+-----+  
| val  |  
+-----+  
| admin |  
| stack |  
+-----+  
2 rows in set (0.00 sec)
```

LIMITSELECT

```
SELECT *  
  FROM Customers  
 ORDER BY CustomerID  
LIMIT 3;
```

ID				
1	Alfreds Futterkiste	Obere Str. 57		12209
2	Ana Trujillo Emparedadoshelados	AVDA. delaConstitución2222	DF	05021
3	AntonioMorenoTaquería	Mataderos 2312	DF	05023

LIMITORDER BY ;◦

```
SELECT *  
  FROM Customers  
 ORDER BY CustomerID  
LIMIT 2,1;
```

LIMITLIMIT offset,count ◦ ◦

ID				
3	AntonioMorenoTaquería	Mataderos 2312	DF	05023

LIMIT;◦

DISTINCT

SELECTDISTINCT◦

```
CREATE TABLE `car`
(
  `car_id` INT UNSIGNED NOT NULL PRIMARY KEY,
  `name` VARCHAR(20),
  `price` DECIMAL(8,2)
);

INSERT INTO CAR (`car_id`, `name`, `price`) VALUES (1, 'Audi A1', '20000');
INSERT INTO CAR (`car_id`, `name`, `price`) VALUES (2, 'Audi A1', '15000');
INSERT INTO CAR (`car_id`, `name`, `price`) VALUES (3, 'Audi A2', '40000');
INSERT INTO CAR (`car_id`, `name`, `price`) VALUES (4, 'Audi A2', '40000');

SELECT DISTINCT `name`, `price` FROM CAR;
+-----+-----+
| name   | price   |
+-----+-----+
| Audi A1 | 20000.00 |
| Audi A1 | 15000.00 |
| Audi A2 | 40000.00 |
+-----+-----+
```

DISTINCT◦ SQL◦ ◦ “Audi A1”◦

MySQL DISTINCTORDER BY◦ ONLY_FULL_GROUP_BYMySQLMySQL GROUP OF GROUP BY◦

LIKE_

LIKE_◦

```
SELECT username FROM users WHERE users LIKE 'admin_';
```

```
+-----+
| username |
+-----+
| admin1   |
| admin2   |
| admin-   |
| adminA   |
+-----+
```

CASEIF

```
SELECT st.name,
       st.percentage,
       CASE WHEN st.percentage >= 35 THEN 'Pass' ELSE 'Fail' END AS `Remark`
FROM student AS st ;
```

```
+-----+-----+-----+
| name   | percentage | Remark |
+-----+-----+-----+
```

Isha	67	Pass
Rucha	28	Fail
Het	35	Pass
Ansh	92	Pass

IF

```
SELECT st.name,
       st.percentage,
       IF(st.percentage >= 35, 'Pass', 'Fail') AS `Remark`
FROM student AS st ;
```

NB

```
IF(st.percentage >= 35, 'Pass', 'Fail')
```

st.percentage >= 35 TRUE 'Pass' Pass ELSE 'Fail'

BETWEEN

BETWEEN ""。

id	username
1	admin
2	root
3	toor
4	mysql
5	thanks
6	java

```
SELECT * FROM stack WHERE id >= 2 and id <= 5;
```

BETWEEN

```
SELECT * FROM stack WHERE id BETWEEN 2 and 5;
```

id	username
2	root
3	toor
4	mysql
5	thanks

4 rows in set (0.00 sec)

BETWEEN >= <= >< 。

NOT BETWEEN

NOT ◦

```
SELECT * FROM stack WHERE id NOT BETWEEN 2 and 5;
```

```
+----+-----+
| id | username |
+----+-----+
|  1 | admin   |
|  6 | java    |
+----+-----+
2 rows in set (0.00 sec)
```

NOT BETWEEN><>=<= WHERE id NOT BETWEEN 2 and 5WHERE (id < 2 OR id > 5) ◦

BETWEENMySQL◦

```
SELECT ... WHERE dt >= '2017-02-01'
                AND dt < '2017-02-01' + INTERVAL 1 MONTH
```

BETWEEN23:59:59◦

-
- BETWEEN'235959'◦
- DATE TIMESTAMP DATETIME DATETIME (6) ◦
- ◦
- BETWEEN◦

<https://riptutorial.com/zh-TW/mysql/topic/3307/>

70:

3

-
- my.cnf
-

mysqld --long-parameter-name=value --another-parameter my.cnf MySQL

_
_ = K M G

ON 1OFF0

my.cnf [mysqld] mysqlmy.cnf

Examples

InnoDB

my.cnf MySQL'lite'

```
innodb_buffer_pool_size
```

RAM704GBRAM;VM InnoDB ENGINE InnoDB

max_allowed_packet = 10M

MMbGGbKKb

group_concat

group_concatgroup group_concat_max_len

```
SET [GLOBAL | SESSION] group_concat_max_len = val;
```

GLOBALSESSION

InnoDB

InnoDBMySQL InnoDB DROP SSDSDD

```
default_storage_engine = InnoDB  
query_cache_type = 0  
innodb_file_per_table = 1  
innodb_flush_neighbors = 0
```


innodb_thread_concurrency=infinity;InnoDB。

```
innodb_thread_concurrency = 0
innodb_read_io_threads = 64
innodb_write_io_threads = 64
```

MySQLIOPScapacity_max。 200HDDSSDIOPS。 IOPS。 MySQL 。

```
innodb_io_capacity = 2500
innodb_io_capacity_max = 3000
```

RAM

MySQLRAM。 70-80MySQLRAM。 RAM。

```
innodb_buffer_pool_size = 10G
```

MySQL

aes-128-ecbECB。

```
block_encryption_mode = aes-256-cbc
```

<https://riptutorial.com/zh-TW/mysql/topic/3134/>

71: 1055ONLY_FULL_GROUP_BYGROUP BY

...

MySQL1055 ◦ MySQLGROUP BY◦

MySQLGROUP BY◦ GROUP BY◦

GROUP BYSELECT *GROUP BY◦ ◦

◦ <https://dev.mysql.com/doc/refman/5.7/en/group-by-handling.html>

MySQL◦ 5.7.5ONLY_FULL_GROUP_BYsql_mode◦ ◦ MySQL5.7.14◦

1055

1. SQL◦
2. MySQL◦
3. [sql_modeONLY_FULL_GROUP_BY](#)◦

SET◦

```
SET sql_mode =  
'STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_EN
```

MySQL◦

[MySQLinit](#) sql_mode=ONLY_FULL_GROUP_BY ◦

Examples

GROUP BY

```
SELECT item.item_id, item.name,      /* not SQL-92 */  
       COUNT(*) number_of_uses  
FROM item  
JOIN uses ON item.item_id, uses.item_id  
GROUP BY item.item_id
```

itemuses◦ SQL-92◦

GROUP BYSELECTORDER BY

1. GROUP BY
2. COUNT() MIN()◦

SELECTitem.name ◦ [SQLONLY_FULL_GROUP_BY](#) MySQL 5.6◦

GROUP BY [SQL-92](#).

```
SELECT item.item_id, item.name,
       COUNT(*) number_of_uses
FROM item
JOIN uses ON item.item_id, uses.item_id
GROUP BY item.item_id, item.name
```

[DBMS SQL-99](#) `SELECT item.name, item.item_id` [SQL-99](#) [MySQL 5.7](#) `ONLY_FULL_GROUP_BY`
`ONLY_FULL_GROUP_BY`.

GROUP BY

```
SELECT item.item_id, uses.category, /* nonstandard */
       COUNT(*) number_of_uses
FROM item
JOIN uses ON item.item_id, uses.item_id
GROUP BY item.item_id
```

`item.uses` `uses.category`

[MySQL](#) `ONLY_FULL_GROUP_BY` [MySQL](#) `GROUP BY`.

`uses` `JOIN ON` [MySQL](#) `category` `MySQL`.

`MySQL` `MySQL`.

[MySQL 5.7](#) `ONLY_FULL_GROUP_BY` `sql_mode` [MySQL 1055](#).

GROUP BY `SELECT *`

`SELECT *`.

```
SELECT item.*, /* nonstandard */
       COUNT(*) number_of_uses
FROM item
JOIN uses ON item.item_id, uses.item_id
GROUP BY item.item_id
```

`ONLY_FULL_GROUP_BY`.

`GROUP BY` `item_id` `number_of_uses` `uses`.

```
SELECT item_id, COUNT(*) number_of_uses
FROM uses
GROUP BY item_id
```

`item`.

```
SELECT item.*, usecount.number_of_uses
```

```

FROM item
JOIN (
        SELECT item_id, COUNT(*) number_of_uses
        FROM uses
        GROUP BY item_id
    ) usecount ON item.item_id = usecount.item_id

```

GROUP BY*

* .

ANY_VALUE

```

SELECT item.item_id, ANY_VALUE(uses.tag) tag,
       COUNT(*) number_of_uses
FROM item
JOIN uses ON item.item_id, uses.item_id
GROUP BY item.item_id

```

itemuses

[ANY_VALUE\(\)](#) . [MySQL](#) . [1055](#) .

[ANY_VALUE\(\)](#) .

[SURPRISE_ME\(\)](#) . [GROUP BY](#) . [MySQL](#) .

. [RAM](#) .

.

[1055ONLY_FULL_GROUP_BYGROUP BY...](https://riptutorial.com/zh-TW/mysql/topic/8245/1055-only-full-group-by-group-by---) <https://riptutorial.com/zh-TW/mysql/topic/8245/1055-only-full-group-by-group-by--->

72:

Examples

1064

```
select LastName, FirstName,  
from Person
```

1064. SQL;MySQL2“”。

MySQL“1064”。

MySQL。 MySQLfrom Person。 from Person。 MySQLSELECT

... near '...'。

... near ''; MySQL。 '”。

DELIMITER。

1064。

1064。

1175

KEYWHERE。

-

```
SET SQL_SAFE_UPDATES = 0;
```

-

```
SET SQL_SAFE_UPDATES = 1;
```

1215

FK。

```
CREATE TABLE `gtType` (  
  `type` char(2) NOT NULL,  
  `description` varchar(1000) NOT NULL,  
  PRIMARY KEY (`type`)  
) ENGINE=InnoDB;
```

```
CREATE TABLE `getTogethers` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `type` char(2) NOT NULL,
  `eventDT` datetime NOT NULL,
  `location` varchar(1000) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `fk_gt2type` (`type`), -- see Note1 below
  CONSTRAINT `gettogethers_ibfk_1` FOREIGN KEY (`type`) REFERENCES `gtType` (`type`)
) ENGINE=InnoDB;
```

1FKKEY。KEY。someOther。

。

```
CREATE TABLE `someOther` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `someDT` datetime NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `someOther_dt` FOREIGN KEY (`someDT`) REFERENCES `getTogethers` (`eventDT`)
) ENGINE=InnoDB;
```

1215。

getTogethersgetTogetherseventDT。。

```
CREATE INDEX `gt_eventdt` ON getTogethers (`eventDT`);
```

getTogetherssomeOther。

MySQL FOREIGN KEY

MySQL。。

。

InnoDB。。

。

```
SHOW CREATE TABLE someOther;
```

- INTINT UNSIGNED。
- KEYS。

1045

“GRANT”“root”。

1236“”

Master_{sync_binlog}OFF◦ SlavebinlogMasterCHANGE MASTER to POS=0◦

binlog_{sync_binlog}=OFF MasterSlave◦ Slavebinlog◦ MasterbinlogCHANGEingbinlog◦

I / O_{sync_binlog}=ON◦

GTID.....

20022003

3306◦

/

-
- “service firewalld stop”“systemctl disable firewalld”
- telnet master 3306
- bind-address
- skip-name-resolve
- ◦

1067,1292,1366,1411 -

1067_{TIMESTAMP}◦ _{TIMESTAMP} defaults◦

1292/1366 DOUBLE / Integer◦ ;_{VARCHAR}◦

1292 DATETIME◦ 23◦ +00◦

1292 VARIABLE_{SETVARIABLE}◦

1292 LOAD DATA“”◦ ◦ ◦

1411 STR_TO_DATE

126,127,134,144,145

MySQL◦ MySQL◦

```
MySQL error code 126 = Index file is crashed
MySQL error code 127 = Record-file is crashed
MySQL error code 134 = Record was already deleted (or record file crashed)
MySQL error code 144 = Table is crashed and last repair failed
MySQL error code 145 = Table was marked as crashed and should be repaired
```

MySQL◦ ◦ ◦ MySQL◦

MyISAM CHECK TABLE REPAIR TABLE◦

InnoDB◦

```
CHECK TABLE <table name> ////To check the extent of database corruption
REPAIR TABLE <table name> ////To repair table
```

139

139°

-
-
-

1366

°

126,1054,1146,1062,24

450°

"°

24

open_files_limit OS° table_open_cache°

- DEALLOCATE PREPARE °
- innodb_file_per_table = ONPARTITIONed° 50° “Native Partitions”°

ulimit/etc/security/limits.conf sysctl.conf kern.maxfiles kern.maxfilesperproc° open_files_limit
table_open_cache °

5.6.8 open_files_limit max_connections°

1062 -

1. - Error Code: 1062. Duplicate entry '12' for key 'PRIMARY'

° °

AUTO_INCREMENT ° NULL°

```
CREATE TABLE userDetails(
  userId INT(10) NOT NULL AUTO_INCREMENT,
  firstName VARCHAR(50),
  lastName VARCHAR(50),
  isActive INT(1) DEFAULT 0,
  PRIMARY KEY (userId) );

--->and now while inserting
INSERT INTO userDetails VALUES (NULL , 'John', 'Doe', 1);
```


2. - Error Code: 1062. Duplicate entry 'A' for key 'code'

◦

INSERT IGNOREINSERT ◦ **MySQL** ◦ IGNORE ◦

```
INSERT IGNORE INTO userDetails VALUES (NULL , 'John', 'Doe', 1);
```

<https://riptutorial.com/zh-TW/mysql/topic/895/>

73:

- `SELECT column_1 [column_2]`
`table_1`
`ORDER BY order_column`
`LIMIT row_count [OFFSET row_offset]`
- `SELECT column_1 [column_2]`
`table_1`
`ORDER BY order_column`
`LIMIT [row_offset] row_count`

“”。

“”
row

Examples

users

ID	
1	1
2	2
3	3
4	4
	5

`SELECT LIMIT`。

LIMIT

```
SELECT * FROM users ORDER BY id ASC LIMIT 2
```

ID	
1	1
2	2

0 。

`ORDER BY`。

LIMIT

LIMIT

- - 0
- 0

```
SELECT * FROM users ORDER BY id ASC LIMIT 2, 3
```

ID	
3	3
4	4
	5

offset LIMIT 2

```
SELECT * FROM users ORDER BY id ASC LIMIT 0, 2
```

```
SELECT * FROM users ORDER BY id ASC LIMIT 2
```

ID	
1	1
2	2

OFFSET

LIMIT OFFSET

```
SELECT * FROM users ORDER BY id ASC LIMIT 2 OFFSET 3
```

ID	
3	3
4	4

- ;
- 0

<https://riptutorial.com/zh-TW/mysql/topic/548/>

S. No		Contributors
1	MySQL	A. Raza , Aman Dhanda , Andy , Athafoud , CodeWarrior , Community , Configure , Dipen Shah , e4c5 , Epodax , Giacomo Garabello , greatwolf , inetphantom , JayRizzo , juergen d , Lahiru Ashan , Lambda Ninja , Magisch , Marek Skiba , Md. Nahiduzzaman Rose , moopet , msohng , Noah van der Aa , O. Jones , OverCoder , Panda , Parth Patel , rap-2-h , rhavendc , Romain Vincent , YCF_L
2	ENUM	Philipp , Rick James
3	JOINS3id	FMashiro
4	JSON	A. Raza , Ben , Drew , e4c5 , Manatax , Mark Amery , MohaMad , phatfingers , Rick James , sunkuet02
5	MyISAM	Rick James
6	MySQL LOCK TABLE	Ponnarasu , Rick James , vijeeshin
7	MySQL	Batsu , Nathaniel Ford , Rick James
8	Mysql	arushi , RamenChef , Rick James , Rodrigo Darti da Costa
9	MySQL	Florian Genser , Matas Vaitkevicius , RationalDev , Rick James
10	MySQL	Ani Menon , Rick James
11	SSL	4444 , a coder , Eugene
12	TRIGGERS	Blag , e4c5 , Matas Vaitkevicius , ratchet , WAF , YCF_L
13	UPDATE	4thfloorstudios , Chris , Drew , Khurram , Ponnarasu , Rick James , Sevle
14		falsefive
15		Ponnarasu , Rick James
16	Docker-ComposeMysql	Marc Alff , molavec
17	mysqldump	agold , Asaph , Barranka , Batsu , KalenGi , Mark Amery , Matthew , mnoronha , Ponnarasu , RamenChef , Rick James , still_learning , strangeqargo , Sumit Gupta , Timothy , WAF

18	UTF-8。	Epodax , Rick James
19	Un-Pivot	rpd
20		kolunar , user6655061
21		juergen d , user2314737
22		O. Jones
23		Majid , Rick James
24		Batsu , Drew , e4c5 , ForguesR , gabe3886 , Khurram , Parth Patel , Ponnarasu , Rick James , strangeqargo , WAF , whrrgarbl , ypercube , Илья Плотников
25		Daniel Käfer , Drew , Ponnarasu , R.K123 , Rick James , still_learning
26		Aminadav , Batsu , Hardik Kanjariya 🙄 , josinalvo , Rick James , WAF
27		Artisan72 , Batsu , Benvorth , Bikash P , Drew , Matt , Philipp , Rick , Rick James , user3617558
28		aries12 , Asaph , bhrached , CGritton , e4c5 , RamenChef , Rick James , WAF
29		Drew , SuperDJ
30		Rick James
31		Abubakkar , Batsu , juergen d , kolunar , Rick James , uruloke , WAF
32		frlan , Rick , Rick James
33		Abhishek Aggrawal , Abubakkar , Darwin von Corax , Dinidu , Drew , e4c5 , juergen d , kolunar , llanato , Rick James , userlond
34		user2314737 , YCF_L
35	JSON	MohaMad
36	MyISAMInnoDB	Ponnarasu , Rick James , yukoff
37	root	BacLuc , Jen R
38		e4c5 , RamenChef , Rick James

39	MySQL 5.7+root	Lahiru, ParthaSen
40		Noah van der Aa, Parth Patel, Ponnarasu, R.K123, Rick James, trf, Tushar patel, YCF_L
41		0x49D1, AbcAeffchen, Abubakkar, Aukhan, CGritton, Dinidu, Dreamer, Drew, e4c5, fnkr, gabe3886, Horen, Hugo Buff, Ian Kenney, Johan, Magisch, NEER, Parth Patel, Philipp, Rick James, Riho, strangeqargo, Thuta Aung, zeppelin
42		Batsu, dakab, Drew, Dylan Vander Berg, e4c5, juergen d, MohaMad, Richard Hamilton, Rick James
43		Abhishek Aggrawal, Drew, Matt S, O. Jones, Rick James, Sumit Gupta
44		Drew, Rick James
45		O. Jones
46		e4c5, Hardik Kanjariya `ヾ`, Rick James, Viktor, ydaetskcoR
47		e4c5, JohnLBevan, kolunar, LiuYan , Matas Vaitkevicius, mayojava, Rick James, Steve Chambers, Thuta Aung, WAF, YCF_L
48		FMashiro
49		Drew, rene
50		kolunar, Rick James, winter
51	mysqlexport	Batsu
52		Rick James, Sumit Gupta
53		Barranka, Dinidu, Drew, JonMark Perry, O. Jones, RamenChef, Richard Hamilton, Rick James
54		Alex Recarey, Barranka, Ben Visness, Drew, kolunar, Rick James, Sanjeev kumar
55		Drew, Rick James
56		Matthew Whitt, Rick James, Riho, Tarik, wangengzheng
57		Ponnarasu, Rick James
58	PS1	Eugene, Wenzhong

59		O. Jones
60		Batsu , Nate Vaughan
61		4444 , Alex Shesterov , alex9311 , andygeers , Aryo , Asaph , Barranka , Benvorth , Brad Larson , CPHPython , Darwin von Corax , Dinidu , Drew , fedorqui , HCarrasko , Jean Vitor , John M , Matt , Misa Lazovic , Panda , Parth Patel , Paulo Freitas , Přemysl Šťastný , Rick , Rick James , Ronnie Wang , Saroj Sasmal , Sebastian Brosch , skytreader , Stefan Rogin , Strawberry , Timothy , ultrajohn , user6655061 , vijaykumar , Vini.g.fer , Vladimir Kovpak , WAF , YCF_L , Yury Fedorov
62		Ponnarasu
63		Abhishek Aggrawal , Divya , e4c5 , Marina K. , Nikita Kurtin , Ponnarasu , R.K123 , ratchet , Rick James , WAF , Yury Fedorov , Илья Плотников
64		Florian Genser , Rick James
65	Mysql	Franck Dernoncourt , Rick James , WAF , YCF_L
66		Barranka
67	...	Adam , Filipe Martins , Lijo , Rick James , Thuta Aung , WAF , whrrgarbl
68	GRANTS	Rick James
69		Ani Menon , Asjad Athick , Benvorth , Bhavin Solanki , Chip , Drew , greatwolf , Inzimam Tariq IT , julienc , KartikKannapur , Kruti Patel , Matthis Kohli , O. Jones , Ponnarasu , Rick James , SeeuD1 , ThisIsImpossible , timmyRS , YCF_L , ypercube
70		ChintaMoney , CodeWarrior , Epodax , Eugene , jan_kiran , Rick James
71	1055 ONLY_FULL_GROUP_BY GROUP BY...	Damian Yerrick , O. Jones
72		Drew , e4c5 , juergen d , Lucas Paolillo , O. Jones , Ponnarasu , Rick James , WAF , Wojciech Kazior
73		Alvaro Flaño Larrondo , Ani Menon , animuson , ChaoticTwist , Chris Rasys , CPHPython , Ian Gregory , Matt S , Rick James , Sumit Gupta , WAF