MySQL REFERENCE SHEET

versions 3.23, 4.0, 4.1

numeric

TINYINT*[(digits)] [unsigned|zerofill]*	*256*
BIT,BOOL,BOOLEAN	*synonyms for tinyint(1)*
SMALLINT*[(digits)] [unsigned|zerofill]*	*65,536*
MEDIUMINT*[(digits)] [unsigned|zerofill]*	*16,777,216*
INT,INTEGER*[(digits)] [unsigned|zerofill]*	*4,294,967,296*
BIGINT*[(digits)] [unsigned|zerofill]*	*18,446,744,073,709,551,616*
FLOAT*[(digits, digits after decimal)] [unsigned|zerofill]*	*23 digits*
DOUBLE*[(digits, digits after decimal)] [unsigned|zerofill]*	*24…53 digits*
DECIMAL*[(digits, digits after decimal)] [unsigned|zerofill]*	*a type of DOUBLE stored as a string*

functions

ABS(*X*) **SIGN**(*X*)
FLOOR(*X*) **CEILING**(*X*)
ROUND(*X[,D]*) **EXP**(*X*)
DIV(*X*) **MOD**(*N,M*)
POW(*X,Y*) **POWER**(*X,Y*)
SQRT(*X*) **RAND**(*[seed]*)
PI() **DEGREES**(*X*)
RADIANS(*X*) **COT**(*X*)
COS(*X*) **ACOS**(*X*)
SIN(*X*) **ASIN**(*X*)
TAN(*X*) **ATAN**(*X*) **ATAN2**(*X*)
LOG(*X*), **LOG2**(*X*), **LOG10**(*X*) **LN**(*X*)
TRUNCATE(*X, D*)

strings

CHAR*[(length)]*	*0…255 – fixed length, right-padded with spaces*
VARCHAR*[(length)]*	*0…255 – variable length (trailing spaces removed)*
BINARY,VARBINARY*[(length)]*	*0…255 – stores bytes instead of character strings*
TINYTEXT|TINYBLOB	*0…255 – text stores strings, blob stores bytes*
TEXT|BLOB	*0…65,535 – text stores strings, blob stores bytes*
MEDIUMTEXT|MEDIUMBLOB	*0…16,777,215 – text stores strings, blob stores bytes*
LONGTEXT|LONGBLOB	*0…4,294,967,295 – text stores strings, blob stores bytes*
ENUM*('value1','value2',…)*	*list of up to 65,535 members, can have only one value*
SET*('value1','value2',…)*	*list of up to 64 members, can have zero or more values*

REGEXP *'expression'*

functions

ASCII(*'str'*) **CONV**(*number,from_base,to_base*) **BIN**(*num*),**OCT**(*num*),**HEX**(*num*)
ORD(*'str'*) **CHAR**(*number[USING charset],…*) **CONCAT**(*'str'1, 'str1',…*)
LENGTH(*'str'*) **CHAR_LENGTH**(*'str'*) **CONCAT_WS**(*'separator','str1','str2'*)
BIT_LENGTH(*'str'*) **REVERSE**(*'str'*) **SOUNDEX**(*'str'*)
LCASE(*'str'*) **UCASE**(*'str'*) **QUOTE**(*'str'*)
LPAD(*'str', len, 'padstr'*) **RPAD**(*'str', len, 'padstr'*) **ELT**(*number, 'str1','str2','str3',…*)
LEFT(*'str', length*) **RIGHT**(*'str', length*) **FIELD**(*'str','str1','str2','str3',…*)
LTRIM(*'str'*) **RTRIM**(*'str'*) **TRIM**(*'str'*) **LOAD_FILE**(*'filename'*)
SPACE(*count*) **REPEAT**(*'str', count*) **SUBSTRING**(*'str', pos[, length]*)
REPLACE(*'str', 'from', 'to'*) **INSERT**(*'str', pos, length, 'newstr'*) **SUBSTRING_INDEX**(*'str', 'del', count*)
INSTR(*'str', 'substr'*) **LOCATE**(*'substr', 'str'[, pos]*) **STRCMP**(*'str1', 'str2'*)

date & time

DATE	*'YYYY-MM-DD'*
DATETIME	*'YYYY-MM-DD HH:MM:SS'*
TIMESTAMP*[(display width)]*	*'YYYY-MM-DD HH:MM:SS' – display widths: 6, 8, 12 or 14*
TIME	*'HH:MM:SS'*
YEAR*[(2|4)]*	*'YYYY' – a year in 2-digit or 4-digit format*

functions

WEEK(*'date'[, mode]*) **WEEKDAY**(*'date'*) **DAYOFWEEK**(*'date'*)
DAYOFYEAR(*'date'*) **MONTH**(*'date'*) **MONTHNAME**(*'date'*)
QUARTER(*'date'*) **YEAR**(*'date'*) **YEARWEEK**(*'date'[, mode]*)
HOUR(*'date'*) **MINUTE**(*'date'*) **SECOND**(*'date'*)
TO_DAYS(*'date'*) **FROM_DAYS**(*number*) **LAST_DAY**(*'date'*)
SEC_TO_TIME(*seconds*) **TIME_TO_SEC**(*'time'*) **SYSDATE**()
CURTIME(),**CURRENT_TIME**(),**CURRENT_TIME** **TIME_FORMAT**(*'date','format'*)
CURDATE(),**CURRENT_DATE**(),**CURRENT_DATE** **DATE_FORMAT**(*'date','format'*)

NOW(),**CURRENT_TIMESTAMP**(),**CURRENT_TIMESTAMP**,**LOCALTIME**(),**LOCALTIME**

UNIX_TIMESTAMP(*['date']*) **FROM_UNIXTIME**(*'unix_timestamp'[, 'format']*)
PERIOD_ADD(*'period', num*) **PERIOD_DIFF**(*'period', num*) **EXTRACT**(*unit FROM 'date'*)

ADDDATE(*'date', days*) | **ADDDATE**(*'date', INTERVAL expr unit*),**DATE_ADD**(*'date', INTERVAL expr unit*)
SUBDATE(*'date', days*) | **SUBDATE**(*'date', INTERVAL expr unit*),**DATE_SUB**(*'date', INTERVAL expr unit*)

commands

connecting to a database
 # mysql [-h hostname] [-u username] [-ppassword] [dbname]

importing data
 # mysql dbname < dbdumpfile.sql

backup a database
 # mysqldump [-options] dbname [> dumpfile.sql]

syntax & examples

Create a database
mysql> *CREATE DATABASE dbname;*

Select a database
mysql> *USE dbname;*

Delete a database
mysql> *DROP DATABASE dbname;*

Add a user to a database
mysql> *GRANT ALL [PRIVILEGES] ON database.* TO [username]@'hostname' [IDENTIFIED BY 'password'];*

List tables in a database
mysql> *SHOW TABLES;*

Show table format
mysql> *DESCRIBE table;*

Delete records in a table
mysql> *DELETE FROM TABLE table [WHERE conditions];*

Create a table
mysql> *CREATE TABLE table (column definition,. . .) [options. . .];*

Show create table syntax
mysql> *SHOW CREATE TABLE table;*

Change a column definition in a table
mysql> *ALTER TABLE table CHANGE column definition;*

Add a column to a table
mysql> *ALTER TABLE table ADD column definition [AFTER col];*

Change auto_increment value
mysql> *ALTER TABLE table AUTO_INCREMENT=value;*

Alter table syntax
mysql> *ALTER TABLE table change specs[, change specs. . .];*

Add a new record
mysql> *INSERT table (column1, column2,. . .) VALUES (expr1, expr2. . .);*

or Add a new record
mysql> *INSERT table SET column=expr[, column=expr. . .);*

Update a record in a single table
mysql> *UPDATE table SET column=expr[, column=expr. . .] [WHERE conditions] [ORDER BY . . .] [LIMIT count]*

Retrieve information from a table
mysql> *SELECT {*|expr|column,. . .} [FROM table,. . .] [WHERE conditions] [GROUP BY . . .] [HAVING conditions] [ORDER BY . . .] [LIMIT count]*

miscellaneous functions

DATABASE()
USER()
FOUND_ROWS()
BIT_COUNT(*number***)**
CAST(*expr AS type***)**
INET_NTOA(*expr***)**
GET_LOCK(*'lock',timeout***)**
ENCRYPT(*'str'[, 'salt']***)**
MD5(*'string'***)**
COMPRESS(*'string'***)**
DES_ENCRYPT(*'str'[, {keynum|keystr}]***)**
VERSION()
CURRENT_USER()
ROW_COUNT()
FORMAT(*number,digits***)**
CONVERT(*expr, type***)**
INET_ATON(*expr***)**
RELEASE_LOCK(*'lock'***)**
DECODE(*'crypt', 'pass'***)**
SHA1(*'string'***)**
UNCOMPRESS(*'string'***)**
DES_DECRYPT(*'string'[, 'key']***)**
CONNECTION_ID()
PASSWORD(*'string'***)**
LAST_INSERT_ID(*[expr]***)**
BENCHMARK(*count, expr***)**
CHARSET(*'str'***)**
LEAST(*val1,val2,. . .***)**
GREATEST(*val1,val2,. . .***)**
ENCODE(*'str', 'password'***)**
AES_ENCRYPT(*'str', 'key'***)**
AES_DECRYPT(*'str', 'key'***)**

grouping functions

AVG(*expr***)**
MIN(*expr***)**
VARIANCE(*expr***)**
BIT_AND(*expr***)**
COUNT(*expr***)**
COUNT(*DISTINCT expr[, expr. . .]***)**
GROUP_CONCAT(*expr***)**
GROUP_CONCAT(*[DISTINCT] expr[, expr. . .]*
[ORDER BY {int|column|expr}
[ASC | DESC] [, column . . .]
*[SEPARATOR 'string']***)**
SUM(*expr***)**
MAX(*expr***)**
STD(*expr***)**
BIT_OR(*expr***)**

operators

AND, &&	*Logical AND*
||, OR	*Logical OR*
XOR	*Logical XOR*
BINARY	*Cast a string to binary string*
&	*Bitwise AND*
|	*Bitwise OR*
^	*Bitwise XOR*
<<	*Left shift*
>>	*Right shift*
-	*Invert bits*
-	*Change sign of value*
-	*Minus*
+	*Addition*
*****	*Multiplication*
%	*Modulo*
DIV, /	*Integer division, division*
<=>	*NULL-safe equal to*
=	*Equal operator*
>=	*Greater than or equal to*
>	*Greater than*
<=	*Less than or equal to*
<	*Less than*
IS	*Boolean test*
LIKE	*Simple pattern matching*
!=, <>	*Not equal to*
NOT LIKE	*Negative simple match*
NOT RGEXP	*Negative regular expression*
NOT, !	*Negates value*
REGEXP	*Match on regular expression*
RLIKE	*Synonym for REGEXP*
SOUNDS LIKE	*Compare sounds*

control flow

IF(*expression,true_result,false_result***)**
IFNULL(*expression,result***)**
NULLIF(*expression1,expression2***)**
CASE *[value] WHEN [comparison] THEN [result]*
[WHEN [comparison] THEN result. . .]
[ELSE result] END