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_)
FLOOR(_X_)
ROUND(_X[,D]_)
DIV(_X_)
POW(_X,Y_)
SQRT(_X_)
PI()
RADIANS(_X_)
COS(_X_)
SIN(_X_)
TAN(_X_) **ATAN**(_X_)

SIGN(_X_)
CEILING(_X_)
EXP(_X_)
MOD(_N,M_)
POWER(_X,Y_)
RAND(_[seed]_)
DEGREES(_X_)
COT(_X_)
ACOS(_X_)
ASIN(_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'_)
ORD(_'str'_)
LENGTH(_'str'_)
BIT_LENGTH(_'str'_)
LCASE(_'str'_)
LPAD(_'str', len, 'padstr'_)
LEFT(_'str', length_)
LTRIM(_'str'_)
SPACE(_count_)
REPLACE(_'str', 'from', 'to'_)
INSTR(_'str', 'substr'_)

CONV(_number,from_base,to_base_)
CHAR(_number[USING charset],…_)
CHAR_LENGTH(_'str'_)
REVERSE(_'str'_)
UCASE(_'str'_)
RPAD(_'str', len, 'padstr'_)
RIGHT(_'str', length_)
RTRIM(_'str'_) **TRIM**(_'str'_)
REPEAT(_'str', count_)
INSERT(_'str', pos, length, 'newstr'_)
LOCATE(_'substr', 'str'[, pos]_)

BIN(_num_),**OCT**(_num_),**HEX**(_num_)
CONCAT(_'str'1, 'str1',…_)
CONCAT_WS(_'separator', 'str1', 'str2'_)
SOUNDEX(_'str'_)
QUOTE(_'str'_)
ELT(_number, 'str1', 'str2', 'str3',…_)
FIELD(_'str', 'str1', 'str2', 'str3',…_)
LOAD_FILE(_'filename'_)
SUBSTRING(_'str', pos[, length]_)
SUBSTRING_INDEX(_'str', 'del', count_)
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]_)
DAYOFYEAR(_'date'_)
QUARTER(_'date'_)
HOUR(_'date'_)
TO_DAYS(_'date'_)
SEC_TO_TIME(_seconds_)

WEEKDAY(_'date'_)
MONTH(_'date'_)
YEAR(_'date'_)
MINUTE(_'date'_)
FROM_DAYS(_number_)
TIME_TO_SEC(_'time'_)

CURTIME(),CURRENT_TIME(),CURRENT_TIME
CURDATE(),CURRENT_DATE(),CURRENT_DATE

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

grouping functions

AVG(*expr*)	**SUM**(*expr*)
MIN(*expr*)	**MAX**(*expr*)
VARIANCE(*expr*)	**STD**(*expr*)
BIT_AND(*expr*)	**BIT_OR**(*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'])

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