List of all MySQL commands:

Note that all text commands must be first on line and end with ';'

(\?) Synonym for `help'.

clear (\c) Clear the current input statement.

connect (\r) Reconnect to the server. Optional arguments are db and host.

delimiter (\d) Set statement delimiter.

edit (\e) Edit command with $EDITOR.

ego (\G) Send command to mysql server, display result vertically.

exit (\q) Exit mysql. Same as quit.

go (\g) Send command to mysql server.

help (\h) Display this help.

nopager (\n) Disable pager, print to stdout.

notee (\t) Don't write into outfile.

pager (\P) Set PAGER [to_pager]. Print the query results via PAGER.

print (\p) Print current command.

prompt (\R) Change your mysql prompt.

quit (\q) Quit mysql.

rehash (\#) Rebuild completion hash.

source (\.) Execute an SQL script file. Takes a file name as an argument.

status (\s) Get status information from the server.

system (\!) Execute a system shell command.

tee (\T) Set outfile [to_outfile]. Append everything into given
outfile.

use (\u) Use another database. Takes database name as argument.

charset (\C) Switch to another charset. Might be needed for processing
binlog with multi-byte charsets.

warnings (\W) Show warnings after every statement.

nowarning (\w) Don't show warnings after every statement.

resetconnection(\x) Clean session context.

Executing SQL Statements from a Text File

Upload the text file, if upload was successful you press the execute button.

LITERAL VALUES

String Literals

A string is a sequence of bytes or characters, enclosed within either single quote (') or double quote (") characters. Examples:

NUMERIC LITERALS

Number literals include exact-value (integer and DECIMAL) literals and approximate-value (floating-point) literals.

DATE AND TIME LITERALS

Date and time values can be represented in several formats, such as quoted strings or as numbers, depending on the exact type of the value and other factors. For example, in contexts where MySQL expects a date, it interprets any of '2015-07-21', '20150721', and 20150721 as a date.

HEXADECIMAL LITERALS

Hexadecimal literal values are written using X'val' or 0xval notation, where val contains hexadecimal digits (0..9, A..F). Lettercase of the digits and of any leading X does not matter. A leading 0x is case-sensitive and cannot be written as 0X.

BIT-VALUE LITERALS

Bit-value literals are written using b'val' or 0bval notation. val is a binary value written using zeros and ones. Lettercase of any leading b does not matter. A leading 0b is case sensitive and cannot be written as 0B.

BOOLEAN LITERALS

The constants TRUE and FALSE evaluate to 1 and 0, respectively. The constant names can be written in any lettercase.

SQL STATEMENT SYNTAX

Atomic Data Definition Statement Support

1.ALTER TABLE Syntax

ALTER TABLE tbl_name

[alter_specification [, alter_specification] ...]

[partition_options]

ALTER TABLE changes the structure of a table. For example, you can add or delete columns, create or destroy indexes, change the type of existing columns, or rename columns or the table itself. You can also change characteristics such as the storage engine used for the table or the table comment.

To use ALTER TABLE, you need ALTER, CREATE, and INSERT privileges for the table. Renaming a table requires ALTER and DROP on the old table, ALTER, CREATE, and INSERT on the new table.

Following the table name, specify the alterations to be made. If none are given, ALTER TABLE does nothing.

ALTER FUNCTION Syntax

2. ALTER FUNCTION func_name [characteristic ...]

characteristic:

COMMENT 'string'

| LANGUAGE SQL

| { CONTAINS SQL | NO SQL | READS SQL DATA | MODIFIES SQL DATA }

| SQL SECURITY { DEFINER | INVOKER }

his statement can be used to change the characteristics of a stored function. More than one change may be specified in an ALTER FUNCTION statement. However, you cannot change the parameters or body of a stored function using this statement; to make such changes, you must drop and re-create the function using DROP FUNCTION and CREATE FUNCTION.

3 CREATE DATABASE Syntax

CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name

[create_specification] ...

CREATE DATABASE creates a database with the given name. To use this statement, you need the CREATE privilege for the database. CREATE SCHEMA is a synonym for CREATE DATABASE.

An error occurs if the database exists and you did not specify IF NOT EXISTS.

4 CREATE TABLE Syntax

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name

(create_definition,...)

[table_options]

[partition_options]

CREATE TABLE creates a table with the given name. You must have the CREATE privilege for the table.

By default, tables are created in the default database, using the InnoDB storage engine. An error occurs if the table exists, if there is no default database, or if the database does not exist.

ABLE CLONING AND COPYING

LIKE

Use CREATE TABLE ... LIKE to create an empty table based on the definition of another table, including any column attributes and indexes defined in the original table:

5 DROP DATABASE Syntax

DROP {DATABASE | SCHEMA} [IF EXISTS] db_name

DROP DATABASE drops all tables in the database and deletes the database. Be very careful with this statement! To use DROP DATABASE, you need the DROP privilege on the database. DROP SCHEMA is a synonym for DROP DATABASE.

DROP TABLE Syntax

6.DROP [TEMPORARY] TABLE [IF EXISTS]

tbl_name [, tbl_name] ...

[RESTRICT | CASCADE]

DROP TABLE removes one or more tables. You must have the DROP privilege for each table.

Be careful with this statement! For each table, it removes the table definition and all table data. If the table is partitioned, the statement removes the table definition, all its partitions, all data stored in those partitions, and all partition definitions associated with the dropped table.

Dropping a table also drops any triggers for the table.

7.RENAME TABLE Syntax

RENAME TABLE

tbl_name TO new_tbl_name

[, tbl_name2 TO new_tbl_name2] ...

RENAME TABLE renames one or more tables. You must have ALTER and DROP privileges for the original table, and CREATE and INSERT privileges for the new table.

8.TRUNCATE TABLE Syntax

TRUNCATE [TABLE] tbl_name

TRUNCATE TABLE empties a table completely. It requires the DROP privilege. Logically, TRUNCATE TABLE is similar to a DELETE statement that deletes all rows, or a sequence of DROP TABLE and CREATE TABLE statements.

DATA MANIPULATION STATEMENTS

https://dev.mysql.com/doc/refman/

8.0/en/sql-syntax-data-manipulation.html

DELETE SYNTAX

DELETE [LOW_PRIORITY] [QUICK] [IGNORE] FROM tbl_name [[AS] tbl_alias]

[PARTITION (partition_name [, partition_name] ...)]

[WHERE where_condition]

DELETE is a DML statement that removes rows from a table.

A DELETE statement can start with a WITH clause to define common table expressions accessible within the DELETE

The DELETE statement deletes rows from tbl_name and returns the number of deleted rows. To check the number of deleted rows, call the ROW_COUNT() function

INSERT + SELECT

INSERT [LOW_PRIORITY | HIGH_PRIORITY] [IGNORE]

[INTO] tbl_name

[PARTITION (partition_name [, partition_name] ...)]

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

SELECT ...

[ON DUPLICATE KEY UPDATE assignment_list]

value:

{expr | DEFAULT}

assignment:

col_name = value

assignment_list:

assignment [, assignment] .

EG :INSERT INTO tbl_temp2 (fld_id)

SELECT tbl_temp1.fld_order_id

FROM tbl_temp1 WHERE tbl_temp1.fld_order_id > 100;

[ORDER BY ...]

SELECT ... INTO Syntax

EG:SELECT id, data INTO @x, @y FROM test.t1 LIMIT 1;

[LIMIT row_count]

UPDATE Syntax

UPDATE [LOW_PRIORITY] [IGNORE] table_reference

SET assignment_list

[WHERE where_condition]

[ORDER BY ...]

[LIMIT row_count]

value:

{expr | DEFAULT}

assignment:

col_name = value

assignment_list:

assignment [, assignment] ...

UNION Syntax

SELECT ...

UNION [ALL | DISTINCT] SELECT ...

[UNION [ALL | DISTINCT] SELECT ...]

UNION is used to combine the result from multiple SELECT statements into a single result set.

The column names from the first SELECT statement are used as the column names for the results returned. Selected columns listed in corresponding positions of each SELECT statement should have the same data type. (For example, the first column selected by the first statement should have the same type as the first column selected by the other statements.)

If the data types of corresponding SELECT columns do not match, the types and lengths of the columns in the UNION result take into account the values retrieved by all of the SELECT statements.

START TRANSACTION, COMMIT, and ROLLBACK Syntax

START TRANSACTION

[transaction_characteristic [, transaction_characteristic] ...]

```
transaction_characteristic: {
    WITH CONSISTENT SNAPSHOT
  | READ WRITE
  | READ ONLY
}

BEGIN [WORK]
COMMIT [WORK] [AND [NO] CHAIN] [[NO] RELEASE]
ROLLBACK [WORK] [AND [NO] CHAIN] [[NO] RELEASE]
SET autocommit = {0 | 1}
```

START TRANSACTION or BEGIN start a new transaction.

COMMIT commits the current transaction, making its changes permanent.

ROLLBACK rolls back the current transaction, canceling its changes.

SET autocommit disables or enables the default autocommit mode for the current session.