# MySQL/CheatSheet

## **Contents**

- 1 Connect/Disconnect
- 2 Query
- 3 Conditionals
- 4 Data Manipulation
- 5 Browsing
- 6 Create / delete / select / alter database
- 7 Create/delete/modify table
- 8 Keys
- 9 create/modify/drop view
- 10 Privileges
- 11 Main data types
- 12 Forgot root password?
  - 12.1 Repair tables after unclean shutdown
  - 12.2 Loading data

#### **Connect/Disconnect**

```
mysql -h <host> -u <user> -p<passwd>
mysql -h <host> -u <user> -p
Enter password: *******
mysql -u user -p
mysql
mysql -h <host> -u <user> -p <Database>
```

## Query

```
SELECT * FROM table
SELECT * FROM table1, table2, ...
SELECT field1, field2, ... FROM table1, table2, ...
SELECT ... FROM ... WHERE condition
SELECT ... FROM ... WHERE condition GROUP BY field
SELECT ... FROM ... WHERE condition GROUP BY field HAVING condition2
SELECT ... FROM ... WHERE condition ORDER BY field1, field2
SELECT ... FROM ... WHERE condition ORDER BY field1, field2
SELECT ... FROM ... WHERE condition LIMIT 10
SELECT DISTINCT field1 FROM ...
SELECT DISTINCT field1, field2 FROM ...
SELECT ... FROM t1 JOIN t2 ON t1.id1 = t2.id2 WHERE condition
SELECT ... FROM t1 LEFT JOIN t2 ON t1.id1 = t2.id2 WHERE condition
SELECT ... FROM t1 JOIN (t2 JOIN t3 ON ...) ON ...
SELECT ... FROM t1 JOIN (t2 JOIN t3 ON ...) ON ...
SELECT ... FROM t1 JOIN t2 USING(id) WHERE condition
```

#### **Conditionals**

```
field1 = value1
field1 <> value1
field1 LIKE 'value _ %'
field1 IS NULL
field1 IS NOT NULL
field1 IN (value1, value2)
field1 NOT IN (value1, value2)
```

```
condition1 AND condition2
condition1 OR condition2
```

#### **Data Manipulation**

```
INSERT INTO table1 (field1, field2, ...) VALUES (value1, value2, ...)
INSERT table1 SET field1=value_1, field2=value_2 ...

LOAD DATA INFILE '/tmp/mydata.txt' INTO TABLE table1
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"' ESCAPED BY '\\'

DELETE FROM table1 / TRUNCATE table1
DELETE FROM table1 WHERE condition
-- join:
DELETE FROM table1, table2 WHERE table1.id1 = table2.id2 AND condition

UPDATE table1 SET field1=new_value1 WHERE condition
-- join:
UPDATE table1, table2 SET field1=new_value1, field2=new_value2, ...
WHERE table1.id1 = table2.id2 AND condition
```

#### **Browsing**

```
SHOW DATABASES
SHOW TABLES
SHOW FIELDS FROM table / SHOW COLUMNS FROM table / DESCRIBE table / DESC table / EXPLAIN table
SHOW CREATE TABLE table
SHOW CREATE TRIGGER trigger
SHOW TRIGGERS LIKE '%update%'
SHOW PROCESSLIST
KILL process_number
SELECT table_name, table_rows FROM INFORMATION_SCHEMA .TABLES WHERE TABLE_SCHEMA = '**yourdbname**';
$ mysqlshow
$ mysqlshow database
```

#### Create / delete / select / alter database

```
CREATE DATABASE [IF NOT EXISTS] mabase [CHARACTER SET charset] [COLLATE collation]
CREATE DATABASE mabase CHARACTER SET utf8
DROP DATABASE mabase
USE mabase
ALTER DATABASE mabase CHARACTER SET utf8
```

## Create/delete/modify table

```
CREATE TABLE table (field1 type1, field2 type2, ...)
CREATE TABLE table (field1 type1 unsigned not null auto_increment, field2 type2, ...)
CREATE TABLE table (field1 type1, field2 type2, ..., INDEX (field1))
CREATE TABLE table (field1 type1, field2 type2, ..., PRIMARY KEY (field1))
CREATE TABLE table (field1 type1, field2 type2, ..., PRIMARY KEY (field1, field2))
CREATE TABLE table1 (fk_field1 type1, field2 type2, ...,
FOREIGN KEY (fk_field1) REFERENCES table2 (t2_fieldA)

[ON UPDATE] [CASCADE | SET NULL | RESTRICT]
[ON DELETE] [CASCADE | SET NULL | RESTRICT])
CREATE TABLE table1 (fk_field1 type1, fk_field2 type2, ...,
FOREIGN KEY (fk_field1, fk_field2) REFERENCES table2 (t2_fieldA, t2_fieldB))
CREATE TABLE table IF NOT EXISTS (...)

CREATE TABLE new_tbl_name LIKE tbl_name
[SELECT ... FROM tbl_name ...]

CREATE TEMPORARY TABLE table (...)

CREATE table new_table_name as SELECT [ *|column1, column2 ] FROM table_name
```

```
DROP TABLE table
DROP TABLE IF EXISTS table
DROP TABLE table1, table2,
DROP TEMPORARY TABLE table
ALTER TABLE table MODIFY field1 type1
ALTER TABLE table MODIFY field1 type1 NOT NULL ...
ALTER TABLE table CHANGE old_name_field1 new_name_field1 type1
ALTER TABLE table CHANGE old_name_field1 new_name_field1 type1 NOT NULL ... ALTER TABLE table ALTER field1 SET DEFAULT ...
ALTER TABLE table ALTER field1 DROP DEFAULT
ALTER TABLE table ADD new_name_field1 type1
ALTER TABLE table ADD new_name_field1 type1 FIRST
ALTER TABLE table ADD new_name_field1 type1 AFTER another_field
ALTER TABLE table DROP field1
ALTER TABLE table ADD INDEX (field);
ALTER TABLE table ADD PRIMARY KEY (field);
-- Change field order:
ALTER TABLE table MODIFY field1 type1 FIRST ALTER TABLE table MODIFY field1 type1 AFTER another_field
ALTER TABLE table CHANGE old_name_field1 new_name_field1 type1 FIRST
ALTER TABLE table CHANGE old_name_field1 new_name_field1 type1 AFTER another_field
ALTER TABLE old_name RENAME new_name;
```

## Keys

```
CREATE TABLE table (..., PRIMARY KEY (field1, field2))
CREATE TABLE table (..., FOREIGN KEY (field1, field2) REFERENCES table2 (t2_field1, t2_field2))
ALTER TABLE table ADD PRIMARY KEY (field);
ALTER TABLE table ADD CONSTRAINT constraint_name PRIMARY KEY (field, field2);
```

#### create/modify/drop view

```
CREATE VIEW view AS SELECT ... FROM table WHERE ...
```

## **Privileges**

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

GRANT ALL PRIVILEGES ON base.* TO 'user'@'localhost' IDENTIFIED BY 'password';

GRANT SELECT, INSERT, DELETE ON base.* TO 'user'@'localhost' IDENTIFIED BY 'password';

REVOKE ALL PRIVILEGES ON base.* FROM 'user'@'host'; -- one permission only

REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'user'@'host'; -- all permissions

SET PASSWORD = PASSWORD('new_pass')

SET PASSWORD FOR 'user'@'host' = PASSWORD('new_pass')

SET PASSWORD = OLD_PASSWORD('new_pass')

DROP USER 'user'@'host'
```

#### Main data types

```
TINYINT (10: -127+128)

SMALLINT (20: +-65 000)

MEDIUMINT (30: +-16 000 000)

INT (40: +-2 000 000 000)

BIGINT (80: +-9.10^18)

Precise interval: -(2^(8*N-1)) -> (2^8*N)-1

/!\ INT(2) = "2 digits displayed" -- NOT "number with 2 digits max"

INT NOT NULL auto_increment PRIMARY KEY -- auto-counter for PK
```

```
FLOAT(M,D) DOUBLE(M,D) FLOAT(D=0->53)

/!\ 8,3 -> 12345,678 -- NOT 12345678,123!

TIME (HH:MM) YEAR (AAAA) DATE (AAAA-MM-JJ) DATETIME (AAAA-MM-JJ HH:MM; années 1000->9999)

TIMESTAMP (like DATETIME, but 1970->2038, compatible with Unix)

VARCHAR (single-line; explicit size)
TEXT (multi-lines; max size=65535)
BLOB (binary; max size=65535)
Variants for TEXT&BLOB: TINY (max=255) MEDIUM (max=~16000) LONG (max=4Go)
Ex: VARCHAR(32), TINYTEXT, LONGBLOB, MEDIUMTEXT

ENUM ('value1', 'value2', ...) -- (default NULL, or '' if NOT NULL)
```

## Forgot root password?

```
$ /etc/init.d/mysql stop
$ mysqld_safe --skip-grant-tables &
$ mysql # on another terminal
mysql> UPDATE mysql.user SET password=PASSWORD('nouveau') WHERE user='root';
## Kill mysqld_safe from the terminal, using Control + \
$ /etc/init.d/mysql start
```

### Repair tables after unclean shutdown

```
mysqlcheck --all-databases
mysqlcheck --all-databases --fast
```

## Loading data

```
mysql> SOURCE input_file
$ mysql database < filename-20120201.sql
$ cat filename-20120201.sql | mysql database
```

Retrieved from "https://en.wikibooks.org/w/index.php?title=MySQL/CheatSheet&oldid=3174496"

- This page was last edited on 10 January 2017, at 18:07.
- Text is available under the Creative Commons Attribution-ShareAlike License.; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy.