MySQL cheatsheet

Browsing

```
SHOW DATABASES;
SHOW TABLES;
SHOW FIELDS FROM table / DESCRIBE table;
SHOW CREATE TABLE table;
SHOW PROCESSLIST;
KILL process_number;
```

Select - Join

```
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 ...
```

Create / Open / Delete Database

```
CREATE DATABASE DatabaseName;
CREATE DATABASE DatabaseName CHARACTER SET utf8;
USE DatabaseName;
DROP DATABASE DatabaseName;
ALTER DATABASE DatabaseName CHARACTER SET utf8;
```

Backup Database to SQL File

```
mysqldump -u Username -p dbNameYouWant > databasename_backup.sql
```

Repair Tables After Unclean Shutdown

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

Update

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

Create / Delete / Modify Table

Create

```
CREATE TABLE table (field1 type1, field2 type2);
CREATE TABLE table (field1 type1, field2 type2, INDEX (field));
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|ON DELETE] [CASCADE|SET NULL]
```

```
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 TEMPORARY TABLE table;
```

Drop

```
DROP TABLE table;
DROP TABLE IF EXISTS table;
DROP TABLE table1, table2, ...
```

Alter

```
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);
```

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
```

Reset Root Password

```
$ /etc/init.d/mysql stop
```

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

```
$ mysql # on another terminal
mysql> UPDATE mysql.user SET password=PASSWORD('new_pass') WHERE user='roo
```

```
## Switch back to the mysqld_safe terminal and kill the process using Cont
$ /etc/init.d/mysql start
```

Your commands may vary depending on your OS.

Select

```
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 DESC;
SELECT ... FROM ... WHERE condition LIMIT 10;
SELECT DISTINCT field1 FROM ...
SELECT DISTINCT field1, field2 FROM ...
```

Conditions

```
field1 = value1
field1 <> value1
field1 LIKE 'value _ %'
field1 IS NULL
field1 IS NOT NULL
field1 IS IN (value1, value2)
field1 IS NOT IN (value1, value2)
condition1 AND condition2
condition1 OR condition2
```

Restore from backup SQL File

```
mysql -u Username -p dbNameYouWant < databasename_backup.sql;
```

Insert

```
INSERT INTO table1 (field1, field2) VALUES (value1, value2);
```

Delete

```
DELETE FROM table1 / TRUNCATE table1
DELETE FROM table1 WHERE condition
DELETE FROM table1, table2 WHERE table1.id1 =
  table2.id2 AND condition
```

Keys

```
CREATE TABLE table (..., PRIMARY KEY (field1, field2))
CREATE TABLE table (..., FOREIGN KEY (field1, field2) REFERENCES table2
(t2_field1, t2_field2))
```

Users and Privileges

```
CREATE USER 'user'@'localhost';
GRANT ALL PRIVILEGES ON base.* TO 'user'@'localhost' IDENTIFIED BY 'passwo
GRANT SELECT, INSERT, DELETE ON base.* TO 'user'@'localhost' IDENTIFIED BY
REVOKE ALL PRIVILEGES ON base.* FROM 'user'@'host'; -- one permission only
REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'user'@'host'; -- all permissions
FLUSH PRIVILEGES;
```

```
SET PASSWORD = PASSWORD('new_pass');
SET PASSWORD FOR 'user'@'host' = PASSWORD('new_pass');
SET PASSWORD = OLD_PASSWORD('new_pass');
```

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

Host '%' indicates any host.

Main Data Types

```
TINYINT (1o: -128 to +127)
SMALLINT (2o: +-65 000)
MEDIUMINT (3o: +-16 000 000)
INT (4o: +- 2 000 000 000)
BIGINT (8o: +-9.10^18)
```

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

⚠ INT(2) = "2 digits displayed" – NOT "number with 2 digits max"

```
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), and `LONG` (max=4Go).
Ex: `VARCHAR(32)`, `TINYTEXT`, `LONGBLOB`, `MEDIUMTEXT`

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