MySQL Reference Card

Version: 0.1

Author: ProgM4c

Attribute Types

Numbers

Name	Coded on	Name	Coded on
TINYINT	1 byte	FLOAT(W, D)	4 bytes
SMALLINT	2 bytes	DOUBLE(W, D)	8 bytes
MEDIUMINT	3 bytes	W: width(number of digits with the '.') D: number of decimals	
INT	4 bytes		
BIGINT	8 bytes		

Parameters:

- UNSIGNED
- ZEROFILL

Coded on:

- SIGNED : $-2^{8*CODED-1} \leqslant val \leqslant 2^{8*CODED-1}-1$
- UNSIGNED: $0 \leqslant val \leqslant 2^{8*CODED}-1$

Strings (between ' ')

Name	Size
CHAR(M)	String with fixed size, 1 <= M <= 255
VARCHAR(M)	String with variable size, 1 <= M <= 255
TINYTEXT	Max length = 255
TEXT	Max length = 65535
MEDIUMTEXT	Max length = 16777215
LONGTEXT	Max length = 4294967295
DECIMAL(M, D)	Simulate a floating point number in a string format

Date and Time

Name	Format
DATE	AAAA-MM-JJ
DATETIME	AAAA-MM-JJ HH:MM:SS
TIMESTAMP	AAAAMMJJHHMMSS
TIMESTAMP(M)	First M characters of a TIMESTAMP
TIME	HH:MM:SS
YEAR	AAAA

ENUM: take one value in the defined list (can be NULL)
syntax:

```
attr_name ENUM('value1', 'value2', …) {NULL | NOT NULL}
```

Database queries

create a database
```
CREATE DATABASE [IF NOT EXISTS] <db_name>;
```

delete a database
```
DROP DATABSE [IF EXISTS] <db_name>;
```

rename a database
```
ALTER DATABASE <db_name> RENAME <db_new_name>;
```

list databases
```
SHOW DATABASES;
```

select a database
```
USE <db_name>;
```

Table queries

show a table
```
SHOW TABLES;
```

rename a table
```
ALTER TABLE <table_name> RENAME <new_name>;
```

describe a table
```
DESCRIBE <table>;
```

delete a table
```
DROP TABLE <table_name>;
```

type of constraints

- NOT NULL
- UNIQUE
- PRIMARY KEY = NOT NULL + UNIQUE
- FOREIGN KEY
- CHECK
- DEFAULT
- AUTO_INCREMENT

create a table
```
CREATE TABLE <table_name> (
      <attr1> <datatype1>(size) <constraints>,
      <attr1> <datatype1>(size) <constraints>,
      …
      PRIMARY KEY(<pk>)
);
```

add / delete a constraints
```
ALTER TABLE <table_name> ADD CONSTRAINT <const_name> TYPEOFCONSTRAINT (<attr1>, …)
ALTER TABLE <table_name> DROP [CONSTRAINT <const_name> | TYPEOFCONSTRAINT <const_name>];
```

Modify table structure

add / delete attribute

```
ALTER TABLE <table_name> ADD <attr> <type> [FIRST|AFTER <other_attr>];
ALTER TABLE <table_name> DROP <attr> ;
```

add / delete default value to an column

```
ALTER TABLE <table_name> ALTER <attr> {SET DEFAULT <default_value>|DROP DEFAULT};
```

change definition of an attribute without/with renaming it

```
ALTER TABLE <table_name> MODIFY <attr_name> <new_type>;
ALTER TABLE <table_name> CHANGE <attr_name> <attr_new_name> <new_type>;
```

Inserting data

```
INSERT INTO <table_name>(<col1>, <col2>, …) VALUES (<val1>, <val2>, …);
```

Modifying data

```
UPDATE <table_name>
SET <field1> = <val1>, <field2> = <val2>, …
WHERE <condition>;
```

Deleting data

```
DELETE FROM <table_name> WHERE <condition>;
```

Retrieving data

Select statement

```
SELECT [ DISTINCT ] attributs
[ INTO OUTFILE fichier ]
[ FROM relation ]
[ WHERE condition ]
[ GROUP BY attributs [ ASC | DESC ] ]
[ HAVING condition ]
[ ORDER BY attributs ]
[ LIMIT [a,] b ]
```

operators in a where clause

=	Equal
<>	Not equal. **Note:** In some versions of SQL this operator may be written as !=
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
BETWEEN	Between an inclusive range
LIKE	Search for a pattern ('%' any sequence of characters '_' any character)

[NOT] IN	To specify multiple possible values for a column
IS [NOT] NULL	To check if the value of a column is NULL or not
AND OR NOT	Filter records based on more than once condition

Sub-requests

```
SELECT * FROM <table> WHERE prix > (SELECT MIN(prix) FROM tab2)
SELECT * FROM <table> WHERE nom NOT IN (SELECT nom FROM tab2)
SELECT * FROM <table> WHERE prix > ALL (SELECT prix FROM tab2) (sup. à ttes les valeurs)
SELECT * FROM <table> WHERE prix > ANY (SELECT prix FROM tab2) (sup. à au moins 1)
```

SQL aliases on column / table

```
SELECT <column> AS <col_alias> FROM <table>          (alias a result)
SELECT <column> FROM <table> AS <new_name>           (alias a table name)
```

SQL functions

```
AVG()         - (moyenne)
COUNT()       - (nombre d'élément)
MAX()         - (maximum)
MIN()         - (minimum)
SUM()         - (somme)

UCASE()
LCASE()
LEN()
NOW()
FORMAT()
```