CHEATSHEET – DATABASES

Table of Contents

INITIALIZE

To exploit databases, there are two choices:

- Command lines
- Databases client. There are two well-known, Datagrip which is from the JetBrains suite (paid account is needed though), or sequel pro (free and lighter)

Our DB scheme will be based on SQL as it is the only data structure we're using (even though there exist other ones as NoSQL).

The DB are hosted on the production and development server.
Development Server: forge@13.229.90.223
Production Server: forge@35.183.8.67

When connected by ssh, use the MySQL instance as such :

> *Mysql –u forge –p password*

Replace the password with the one in the keypass file (under MySQL instance)

CREATE

Before using it, it is important to create a database for the website to use.
To do so, connect to mysql and then
create database example
Make sure to create a new username and a password. Never use the default credentials, otherwise it would be a security breach.
Follow this tutorial : https://www.digitalocean.com/community/tutorials/how-to-create-a-new-user-and-grant-permissions-in-mysql

Once the user has been created, do:

> *GRANT ALL PRIVILEGES ON * . * TO 'newuser'@ 'localhost'*
>
> *FLUSH PRIVILEGES*

This last line is important as it tells the server to reload the grant tables

```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| ahoy               |
| cbd                |
| dl_script          |
| mysql              |
| performance_schema |
| rgc_lg2            |
| rgccomprod         |
| rgcprod            |
| sys                |
+--------------------+
```

MANAGE

Many commands are useful in MySQL, and lot of cheatsheets are out there, like this one from Cheatography.com (cf Annex 1)

For instance, to have the list of available databases, use

```
show databases
```

To handle a specific database:

```
use db_name
```

ANNEX 1

Cheatography

Essential MySQL Cheat Sheet
by guslong via cheatography.com/1345/cs/520/

MySQL Data Types

CHAR	String (0 - 255)
VARCHAR	String (0 - 255)
TINYTEXT	String (0 - 255)
TEXT	String (0 - 65535)
BLOB	String (0 - 65535)
MEDIUMTEXT	String (0 - 16777215)
MEDIUMBLOB	String (0 - 16777215)
LONGTEXT	String (0 - 4294967295)
LONGBLOB	String (0 - 4294967295)
TINYINT x	Integer (-128 to 127)
SMALLINT x	Integer (-32768 to 32767)
MEDIUMINT x	Integer (-8388608 to 8388607)
INT x	Integer (-2147483648 to 2147483647)
BIGINT x	Integer (-9223372036854775808 to 9223372036854775807)
FLOAT	Decimal (precise to 23 digits)
DOUBLE	Decimal (24 to 53 digits)
DECIMAL	"DOUBLE" stored as string
DATE	YYYY-MM-DD
DATETIME	YYYY-MM-DD HH:MM:SS
TIMESTAMP	YYYYMMDDHHMMSS
TIME	HH:MM:SS
ENUM	One of preset options
SET	Selection of preset options

Integers (marked x) that are "UNSIGNED" have the same range of values but start from 0 (i.e., an UNSIGNED TINYINT can have any value from 0 to 255).

Select queries

select all columns
```
SELECT * FROM tbl;
```

select some columns
```
SELECT col1, col2 FROM tbl;
```

select only unique records
```
SELECT DISTINCT FROM tbl WHERE condition;
```

column alias with AS
```
SELECT col FROM tbl AS newname;
```

order results
```
SELECT * FROM tbl ORDER BY col [ASC | DESC];
```

group results
```
SELECT col1, SUM(col2) FROM tbl GROUP BY col1;
```

Creating and modifying

create a database
```
CREATE DATABASE db_name;
```

select a database
```
USE db_name;
```

list the databases on the server
```
SHOW DATABASES;
```

show a table's fields
```
DESCRIBE tbl;
```

create a new table
```
CREATE TABLE tbl (field1, field2);
```

insert data into a table
```
INSERT INTO tbl VALUES ("val1", "val2");
```

delete a row
```
DELETE * FROM tbl WHERE condition;
```

add a column from a table
```
ALTER TABLE tbl ADD COLUMN col;
```

remove a column from a table
```
ALTER TABLE tbl DROP COLUMN col;
```

make a column a primary key
```
ALTER TABLE tbl ADD PRIMARY KEY (col);
```

Creating and modifying (cont)

return only 1 row matching query
```
... LIMIT = 1
```

amend the values of a column
```
UPDATE table SET column1="val1" WHERE ...
```

clear all the values, leaving the table structure
```
TRUNCATE TABLE tbl;
```

delete the table
```
DROP TABLE tbl;
```

delete the database
```
DROP DATABASE db_name;
```

Matching data

matching data using LIKE
```
SELECT * FROM tbl1 WHERE col LIKE '%value%'
```

matching data using REGEX
```
SELECT * FROM tbl1 WHERE col RLIKE 'regular_expression'
```

Joins

INNER JOIN	returns only where match in both tables
OUTER JOIN	also returns non-matching records from both tables
LEFT JOIN	also returns non-matching records from left table
RIGHT JOIN	also returns non-matching records in right table

JOIN syntax:

```
SELECT * FROM tbl1 INNER JOIN tbl2 ON tbl1.id = tbl2.id;
```

String functions mySQL

Compare strings	STRCMP("str1","str2")
Convert to lower case	LOWER("str")
Convert to upper case	UPPER("str")
Left trim	LTRIM("str")

By **guslong**
cheatography.com/guslong/

Published 13th August, 2012.
Last updated 29th June, 2014.
Page 1 of 2.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com