User Documentation for the DBMS Mysql

From: Sebastian Klinger, Benno Rodehack

In this Document you will find everything you need to know about the DBMS Mysql. How to set it up, how to use it and how to let others in your Network access it.

Initial Setup

For this step, it is assumed that you have an Ubuntu Server 22.04 LTS Jammy Jellyfish setup and ready to go. Don't worry, we will keep the Command Line stuff to a minimum.

Update the System

Use the bellow command to Update the System, to ensure that you will get the newest Version of mysql.

```
$ sudo apt-get update && sudo apt-get upgrade
```

Now we can install the Mysql Server:

```
$ sudo apt-get install mysql-server
```

That is actually all of it! You now have a Mysql Server installed! The installation process should have started the Server, but you can verify that by using:

```
$ sudo systemctl status mysql
```

If there is a green Dot and it says active, then everything is allright.

If it is not started, use:

```
$ sudo systemctl enable --now mysql
```

or:

```
$ sudo systemctl restart mysql
```

Configuring Network access

If you want others in your Network to access the Server, you will need to do a bit more work.

Start with editing your mysql config:

```
$ sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

There should be a Line that says "bind-address". Change the Number behind the = sign to the ip address of the server:

```
...
bind-address            = 127.0.0.1
mysqlx-bind-address     = 127.0.0.1
...
```

to

```
...
bind-address            = <your-ip-address>
mysqlx-bind-address     = 127.0.0.1
...
```

Now you need to restart the Server:

```
$ sudo systemctl restart mysql
```

You are good to go!

Using the DBMS

First things first: Accessing the DBMS

You have multiple Ways of doing this:

1. Using an External Programm (e.g Data Grip from Jetbrains)

    - You can use an External Programm (like Data Grip) to access the Server, that might be more convenient for you, but it also **requires** you to enable Networking on the Server, wich is a thing you might not want to do depending on your environment

2. Using the Command Line Interface (preffered way if you cannot or do not want to enable Networking)
    - This is the Method we will be using
    - You can login to your DBMS by using `$ sudo mysql -u root` on the Command Line Interface on the Ubuntu Machine

Operating the DBMS

Running Scripts

To run scripts (that your software department gave you or something like that) just use:

```
SOURCE path/to/file.sql;
```

Listing all Databases

```
SHOW DATABASES;
```

This would return something like this:

Databases
information_schema
izanami
monitor
mysql
performance_schema
sys
statping

excerpt from my database server

Change into a Database

```
USE <your_db_name>;
```

This will change your CLI into the CLI of the Database Name that you provided

Listing all Tables

```
SHOW TABLES;
```

This would return something like this (your table names are obviously different):

Tables_in_izanami
CategoryItemDefinition
CollectiblePresentationNode
DamageItemDefinition
DestinyAchievementDefinition
DestinyActivityDefinition
...

Creating databases and Tables

To Create a Database you can use the `CREATE DATABASE` statement:

```
CREATE DATABASE database_name;
```

To create a Table you can issue the `CREATE TABLE` statement. You must provide at least one column for the Table, else the command will fail.

```
CREATE TABLE table_name {
    column_name datatype,
    column_name_2 datatype,
    column_name_3 datatype,
...
};
```

Deleting a Database or Table

The `DROP` keyword is used to Delete tables and Databases. **BE CAREFULL THOUGH:** This will delete every bit of Information from these Tables or Databases.

Database

```
DROP DATABASE database_name;
```

Table:

```
DROP TABLE table_name;
```

Altering Tables

The `ALTER TABLE` statement is commonly used to either delete, add or modify a column

Add Column:

```
ALTER TABLE table_name
ADD column_name datatype;
```

Drop Column:

```
ALTER TABLE table_name
DROP COLUMN column_name;
```

Modify Column:

```
ALTER TABLE table_name
MODIFY COLUMN column_name datatype;
```

Inserting records

The `INSERT` keyword is used to add new Datasets to a Table. Normally an Administrator shouldn't be required to add new Records to tables (because scripts or applications normally do that), but it is allways good to have the knowledge should the need arrise.

```
INSERT INTO table_name (column_name1, column_name2, column_name3, ...) VALUES (value1, value2, value3);
```

You do not have to insert into every column of the table. You can also insert into specific columns:

```
INSERT INTO table_name (column_name1, column_name3) VALUES (value1, value3);
```

This will insert into column 1 and 3 but not into 2.

Getting Specific Records

The `SELECT` keyword and `WHERE` Statements are used to Filter Records from Tables

Get every Record

To get every record of a Table you can use:

```
SELECT * FROM table_name
```

This is an Example if you want to get every record form a table, from one of my tables

Statement:

```
SELECT * FROM EnumItemType
```

Result:

ID	Name
0	None
1	Currency
2	Armor
3	Weapon
7	Message
...	...

Filtering Records

To only get Specific Records from a Table you can use the `WHERE` Statement.

```
SELECT * FROM table_name WHERE condition
```

A "condition" could be a comparision between a column and a Value that you provide.

Example:

Statement:

```
SELECT * FROM EnumItemType WHERE ID = 2
```

Result:

ID	Name
2	Armor

Editing Records

You can update Records after you inserted them.

```
UPDATE table_name
SET column_name1 = value1, column_name2 = value2
WHERE condition;
```

You can use the condition of the Filtering Records Section to update specific records.

Note: You should always be carefull when you update existing records manually. It could seriously break an application or the user experience.
