F21DF MySQL Guide

This guide provides additional support for working with MySQL in the context of the F21DF Database and Information Systems course at Heriot Watt University.

These instructions assume that you have access to the mysql client, either from a MACS lab machine or the MACS virtual machine. Please see the relevant instructions for achieving this.

The examples and screenshots included in this document were constructed using the MACS VM.

MySQL Documentation

The official MySQL documentation is available from https://dev.mysql.com/doc/refman/5.6/en/. The most useful part for getting started is the tutorial.

Naming Conventions

You should use descriptive names, but avoid overly long names for your tables and columns.

Table Names

Table names are case sensitive on Linux and Unix based systems, this applies to both the MACS VM and the MACS database server.

The convention used for table names in this course is PascalCase, i.e. they start with a capital letter and multiple words are concatenated together with each word starting with a capital letter.

It is not always possible to have multiple databases, e.g. the MACS MySQL server restricts users to a single database corresponding to their username. In these situations, it is useful to use a prefix for your table names to indicate which database they are a part of, e.g. the following tables are all part of a Bank database `BankAccount`, `BankEmployee`, and `BankBranch`. The other advantage of this approach is that the table names are listed together when you list all available tables.

Column Names

The convention used for column names in this course is camelCase, i.e. they start with a lowercase letter and multiple words are concatenated together with each word starting with a capital letter.

SQL Conventions

SQL commands are space and case insensitive, except for table names on Linux/Unix based systems (see Table Names). This means that it doesn't matter if you add in extra spaces between words or use lowercase for SQL commands.

The convention used in this course is to use UPPERCASE for SQL commands, with each part of a query on a new line and appropriately indented, as shown in this example query which gets a list of employee names with their department name who earn more than 30000.

```
SELECT E.firstname, E.surname, D.dName
FROM   Employee AS E, Department AS D
WHERE  E.deptNo = D.deptNo AND
       E.salary >= 30000;
```

Connecting to MySQL

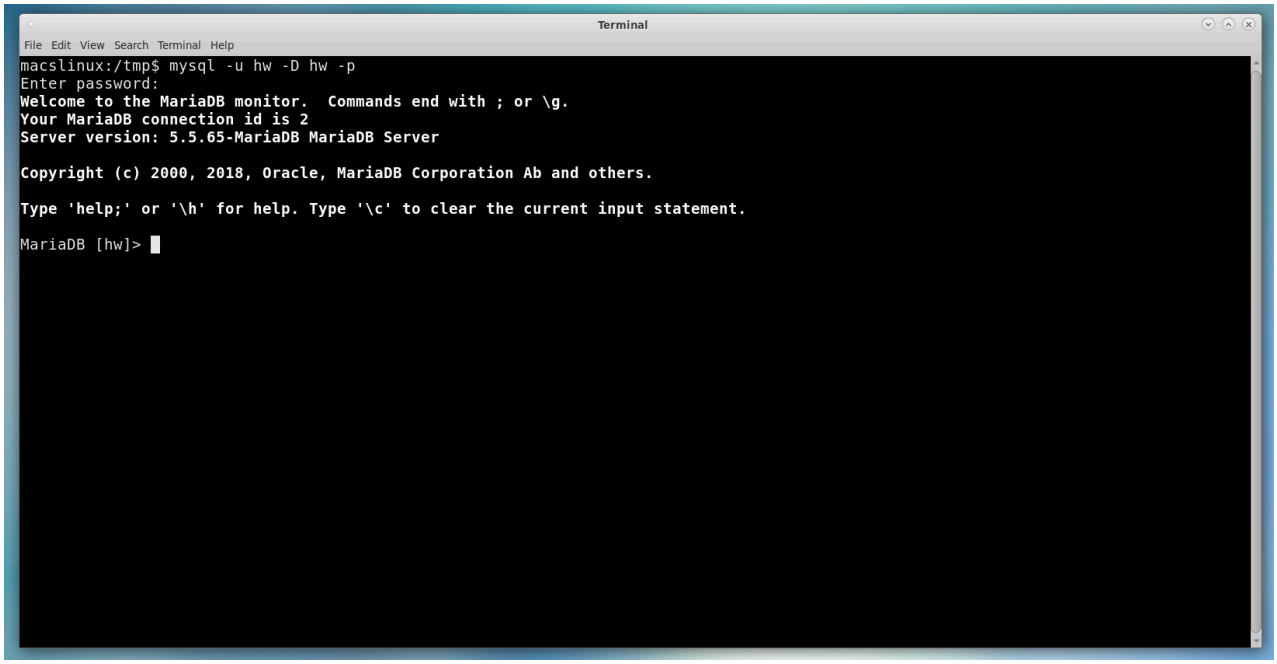
To start the MySQL client, you first need to open a Terminal window and then enter the following command.

```
mysql -u hw -D hw -p
```

The command uses three flags to pass in arguments to the mysql application. The meaning of these flags are:

- `-u` set the username, in the above example the username is set to `hw`
- `-D` set the database to be used, in the above example the database is `hw`
- `-p` instructs mysql to use a password to authenticate the user, the application will prompt you to enter your password; this flag should always be the last flag

You should then be logged in and see something like the following screenshot.



To end your session, enter the following command

```
quit;
```

Changing your Password

You can change your password with the following command. Replace `xxx` with your new password.

```
SET PASSWORD = PASSWORD('xxx');
```

Working with SQL

You can run SQL commands directly from the mysql client. Commands can span multiple lines. They only get executed when a `;` is entered.

It is helpful to develop and save SQL commands in a separate file. These can be conveniently edited in a code editor such as Atom, Emacs, or Visual Studio Code. All these applications are available on the VM and support syntax highlighting for SQL.

Typically files containing SQL commands are given a `.sql` suffix.

Individual SQL commands can be copied and pasted into the mysql client. Alternatively, if you want to run all the commands in a file, you can issue the following command from a terminal window

```
mysql -u hw -D hw -p -t < filename.sql > result.txt
```

The `-t` flag informs mysql to print the results as a table, the `< filename.sql` instructs the client to read in and execute the content in the `filename.sql` file, while `> result.txt` will store the results of the commands in the file `result.txt`. Note that the command needs to be issued in the same directory as the file, or you need to provide a full path to the file.

Bulk Loading Data

It is possible to load large amounts of data that is stored in a file into a database table. This is useful for loading CSV data into a database. The following command can be used from within the mysql client. (Note that the full path to the file needs to be given since the command is run by the server.)

```
LOAD DATA LOCAL INFILE '/home/hw/MySQL/Bank/AccountData.csv'
REPLACE INTO TABLE BankAccount
FIELDS TERMINATED BY ',';
```

The table `BankAccount` must already have been created in the database and the columns correspond to the data. The content of the `AccountData.csv` would look something like the following.

```
23501,current,2350,1999-01-12,1
23502,current,150,1999-07-12,1
```

When inserting data, use the mysql command 'warnings' to enable you to view warnings about invalid input data which has been changed to a default.

Help with Error Messages

From time to time, your commands will not work. Generally this is due to an error in your syntax or the command you are issuing is in violation of the constraints defined in the database. A useful source of help is to copy the error code and first part of the error message into a search engine; it is unlikely that you will be the first to have encountered the problem.

The following error is commonly encountered when creating a new table that involves a foreign key reference.

```
ERROR 1005 (HY000): Can't create table './hw/SpyWithSkill.frm' (errno: 150)
```

The meaning of the error is that the foreign key referenced doesn't exist yet, so you have either not created the corresponding table and column, or you have a typing mistake in your SQL. I would put the following into a search engine to get more details.

```
ERROR 1005 (HY000): Can't create table
```

Useful MySQL Commands

To set a database to use for subsequent commands

```
USE Company
```

where `Company` is the name of the database that you want to use.

To display a list of tables and views

```
SHOW FULL TABLES;
```

To display the columns for a given table, e.g. `BankAccount`

```
SHOW COLUMNS FROM BankAccount;
```

or in MySQL you can use

```
DESCRIBE BankAccount;
```

To find out the create table statement for a table, e.g. to see constraints, use the following command, replace `BankAccount` with the relevant table name.

```
SHOW CREATE TABLE BankAccount;
```