Working with MySQL

Prof. Dorothee Koch, Hochschule für Technik Stuttgart

1 Introduction to MySQL

1.1 Background Information

MySQL is a relational database system that is free software licensed under the GNU General Public Licence (http://www.gnu.org) and available for a variety of operating systems, including various Windows and Unix/Linux versions. You can download your own version for your home computer from http://www.mysql.com.
For instance here:
https://dev.mysql.com/downloads/mysql/8.0.html

ODBC and JDBC drivers are available from there as well.

MySQL generally uses ANSI SQL92 with some non-standard extensions.

Online Documentation is available at http://dev.mysql.com/doc/.

We have found the documentation well-written and understandable. Installing the system from scratch was no problem with the documentation. Even if you do not plan to install your own MySQL, browsing through the documentation can be a valuable add-on to the concepts we studied in class.

A package by the name of XAMPP (x = many operating systems, a = Apache Webserver, m = MariaDB, p = PHP, p = Perl) is very easily configurable and installs itself without trouble. It can be downloaded safely for instance here:
https://www.apachefriends.org/de/download.html
MariaDB was developed from MySQL and is very similar.

It is good for testing or for doing your pre-exam project, but don't use it for real internet applications! It was developed just as a test system and contains security gaps.

1.2 Logging in, Starting and Exiting from MySQL

MySQL is installed on a server associated with the LIDA[1] with the IP address **193.196.143.2**. The server can also be accessed under the name **c-143-002.fkc.hft-stuttgart.de**.

1. LIDA = Lab for Linux and Data Technologies

In order to use the database system, you need to login to this computer.

See the separate document called ServerAccess.pdf for a description how to log in to the server.

To start MySQL you need additionally a username and a password for MySQL. The username is dk1s_youremailusername (e.g. dk1s_71deka1mst). The password will be told in class. Please note that there is the prefix dk1s_ in front of your general user names. Also note, that this is dk"one"s, so NOT the letter „l" in the alphabet!

The university database of our example is called **koch_universitydb** in MySQL. Every student in our class can access it. For every student, there is additionally a database with the name dk1s_youremailusernamedb (e.g. dk1s_71deka1mstdb). The other students have no access to your private database, so you can work there without effects from others, whereas in the koch_universitydb, everyone can see the effects of the others' operations.

In order to start and exit from MySQL, take the following steps:

From Unix (in the LIDA, room 2/501):

- Log in to the LIDA server and open a shell window. Putty will directly log you into a shell. If you use ThinLinc, open a shell terminal by rightclicking on the desktop background and selecting „Terminal öffnen" (open terminal) or by using the menu in the upper left corner of the xfce desktop.
- Connect to MySQL by typing to the shell prompt:
  mysql -h 193.196.143.168 -u username -p
  (do make sure to use the database username, i.e. the one with the prefix!)
- The password will be announced in class and should be changed.
- Now you can enter SQL and MySQL commands
- You can exit from MySQL with the command **exit**.
- In the very end, also log off from your work station in the LIDA.

Additional hints for working in the LIDA:

This is the Ubuntu operating system, a Linux version.

The IDEs Netbeans, Eclipse, and IntelliJ are availble on the LIDA server.

If you want to use Netbeans for developing the programs, execute the following steps:

Type into the shell window:
`module load ide/netbeans` (then hit return) followed by:
`netbeans`

If you prefer Eclipse:
`module load ide/eclipse/4.3`
`eclipse`

Alternatively, you can enter directly the path of the commands:

`/opt/hft/ide/netbeans/8.0.1/bin/netbeans`
or
`/opt/hft/ide/eclipse/4.3/eclipse`

- Opening a shell window:
  Click the right mouse button on the screen background and select "Terminal öffnen". You can open as many shell windows as you wish. (Well, don't open hundreds, or the system performance will suffer eventually :-)).
- Starting a web browser:
  Probably there is a menu for starting a web browser somewhere, but you can simply start Firefox by typing
  `firefox &`
  in a shell window.
- Shell Configuration:
  Your login shell in the Ubuntu system here is a bash. It is recommended to configure a shell by a configuration file. For a bash, this file is called .bashrc
  Copy Prof. Koch's **`.bashrc`** by entering the following commands, AFTER having made a backup copy of your standard .bashrc by typing:
  `cd ~`
  `cp .bashrc .bashrcbackup`
  `cp ~koch/.bashrc .`
  (yes, that includes the dot (**`.`**) after the blank at the end!)
  Then enter
  `source .bashrc`
  These commands need to be entered only once. From the next login, the configuration will be executed automatically.
- Text Editor
  You can edit plain text by calling the emacs text editor. Just type
  `emacs filename`
  where filename is the name (new or existing) of the file you want to edit.
  You can also use a standard text editor available in the start menu in the lower left desktop corner. Nano for instance is also available.
- Netbeans
  For an easier start of Netbeans, you can adjust your path variable in the .bashrc.
- Define aliases and paths
  The absolute paths of calling MySQL and Netbeans don't have to be entered if you define the appropriate paths in the .bashrc or define an alias in the .bashrc. See examples in Prof. Koch's .bashrc.

After starting MySQL the following text is shown:

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8 to server version: 5.1
Type „help“ for help.
mysql>
```

Now you can enter SQL commands or MySQL commands.

1.3 Changing the MySQL Password

You can use the following command to change your password for your MySQL account:
mysql> SET PASSWORD = 'your_new_password';

Please be careful, as the password is clearly visible while you type this. You may want to close the terminal window in which you typed this, afterwards.

1.4 Exporting and Importing SQL To and From MySQL

Your MySQL session can be logged to a file. You start the logging session by
`mysql>tee outfile;`
where outfile is the name of the file to which you want to write the log. You can chose any valid file name.

You end the logging sessin by entering:
`mysql>notee;`

For queries that you want to execute frequently, it is convenient to define a script file (similar to a Unix shell script). A SQL script is a simple text file which contains SQL commands. By loading the file into MySQL, the commands in the file can be executed.

You can import the script file by typing to the MySQL prompt:
mysql>source infile;
where infile is the name of your SQL script file. You can chose any valid filename.

An alternative way of importing and exporting information between MySQL and operating system is doing it from a Unix shell without even directly logging into MySQL:

You can load a SQL script from a shell by using the following command:

shell> mysql -u <username> -p <password> < <filename>

To view the output screen after screen you can pipe the result through more.

Example:

```
shell> mysql < batch-file | more
```

You can also catch the output in a file for further processing by redirecting stdout to a file.
Example:

shell> mysql < batch-file > mysql.out

Comments in SQL Scripts

In a script, comments are preceded by double dashes.

Example:

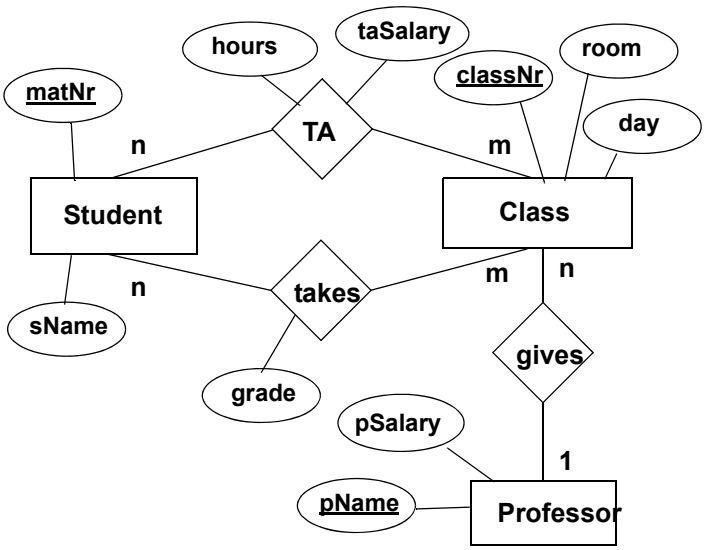
-- This is a script file

2 The University Database Schema

The example used here is the one you have seen in class. You can create and recreate it easily by executing the SQL scripts with the create table statements and insert statements that you find in the Moodle course.

2.1 ER Model

The underlined attributes constitute the primary key of each entity.



2.2 Tables

From the ER model the following tables are derived:

Class (classNr, room, day, pName)
Student (matNr, name)
Professor (pName, pSalary)

Takes (matNr, classNr, grade)
TA (matNr, classNr, hours, taSalary)

If you are using this for the class Business Intelligence, the schema looks slightly different (see the class notes as well as the SQL scripts).

3 Notes about the Syntax in this Document

In the following chapters, SQL commands as well as special MySQL commands are explained.

The SQL commands are largely identical with standard SQL. In addition, MySQL accepts commands that are specific only for this system.

Reserved words are capitalized in these class notes to make them more readable. However, MySQL is not case sensitive, so you can enter them in small letters into the system, too.

Expressions in [ ] are optional.
Expressions in { } contain several options. The vertical bar | means "or".

4 Data Definition and Using the Data Dictionary

The data dictionary, called INFORMATION_SCHEMA in MySQL, contains information about the databases that you can use, in particular, you can find out from it which tables are available, what are their attributes with which data types.

MySQL provides specific commands with an SQL-like syntax to use the data dictionary. Some examples are given here. See the documentation for more details.
The following commands can be typed after the MySQL prompt.

4.1 CREATE DATABASE [IF NOT EXISTS] db_name;

Syntax:
```
CREATE DATABASE [IF NOT EXISTS] db_name
```

Description:
Creates a database with the given name. An error occurs if the database already exists and you did not specify IF NOT EXISTS.

Databases in MySQL are implemented as directories containing files that correspond to tables in the database. Because there are no tables in a database when it is initially created, the CREATE DATABASE statement only creates a directory under the MySQL data directory.

You can ask your MySQL administrator for permission to use a database of your own. Suppose you want to call yours menagerie. The administrator needs to execute a command like this:

```
mysql> GRANT ALL ON menagerie.* TO your_mysql_name;
```

where your_mysql_name is the MySQL user name assigned to you.

4.2. SHOW DATABASES;

Syntax:
SHOW DATABASES;

Description: Lists the databases on the MySQL host for which the user has access rights.

4.3 SELECT DATABASE();

Syntax:
```
SELECT DATABASE();
```

Description:
Returns the name of the current database.

4.4 USE database_name;

Syntax:
USE database_name;

Description:
After execution of this command, the current database is the one with the name database_name. It must be an existing database.

If you want to use a database that is not the current database, you need to put the name of the database in front of the table names in your SQL-statements.

Example:

```
SELECT * FROM universityDB.Student;
```

4.5 SHOW TABLES;

Syntax:
SHOW TABLES;

Description:
Lists the tables in the current database. If a user does not have any privileges for a table, the table will not be listed in the output of this command.

4.6 SHOW COLUMNS FROM table_name;

Syntax:
SHOW COLUMNS FROM table_name;

Description: Lists the columns from the table table_name and some additional information, e.g.:
whether the value may be NULL or not, if the column is part of the primary key of the table the default value.

Example:
SHOW COLUMNS FROM Student;

Result:

```
Field | Type        | Null | Key | Default | Extra
------+-------------+------+-----+---------+-------
 matNr | int(11)    |      | PRI | 0       |
 sName | varchar(30)|      |     |         |
------+-------------+------+-----+---------+-------
```

4.7 DESCRIBE table_name

An alternate way of inspecting the definition of a table is using the MySQL command describe.

Syntax:
{DESCRIBE | DESC} tbl_name {col_name | wild}

Description:
Describe provides information about a table's columns. col_name may be a column name or a string containing the SQL '%' and '_' wildcard characters.

5 Spelling and Syntactic Formats in MySQL

Some rules concerning the spelling and syntactic format must be obeyed in MySQL.

5.1 SQL and MySQL commands:

- MySQL is NOT case-sensitive, i.e. SQL commands can be typed in upper case or lower case
- At the end of a SQL command, a semicolon(;) is required.

- At the end of a MySQL command usually (but not always!) a semicolon is required. If a semicolon is used where none is required, this does no harm.
- A command need not be given all on a single line, so lengthy commands that require several lines are not a problem. MySQL determines where your statement ends by looking for the terminating semicolon , not by looking for the end of the input line. (In other words, mysql accepts free-format input: it collects input lines but does not execute them until it sees the semicolon.)

5.2 Names (Identifiers)

Names of tables, columns, databases etc.
- are case-sensitive,
- must not be reserved words.

Reserved words are reserved SQL words or MySQL commands like ‘use’ etc.
Please note that ALTER is a reserved word. It is also the German word for age. So if you need an attribute named Alter, give it a different name.

5.3 Names (Identifiers) in Quotation Marks (’’)

- These are not case-sensitive.
- You can use all available characters (e.g. also blanks)
  (e.g.: *"Mat-Nr"* VARCHAR(9) CONSTRAINT *"nn_Mat-Nr"* NOT NULL ...
- Using also reserved words is allowed.

5.4 Data of the Data types CHAR and VARCHAR

- Values of such columns must be enclosed in single quotes (’)
  e.g.: ... WHERE Student = ’Schmidt’ ...
- You can use all available characters (e.g. also blanks).