Introduction to MySQL

MySQL is a popular open-source database management system. The data is managed by a *server*, which in MySQL is called `mysqld`. Users access the data from *clients*. There are different clients for MySQL, for example `mysql` (a command line interface) and MySQL Workbench (a graphical interface). Both clients are available on a wide range of platforms. Here, we describe `mysql`, but you are free to use any client you wish. (On the student Linux computers, `mysql-workbench` is also installed.)

There is an enormous amount of information about MySQL on the web. The most important is the reference manual at `http://dev.mysql.com/doc/refman/5.6/en/`. The manual is searchable, but Google usually gives better results (if you prefix your search phrase with mysql).

There are many differences between MySQL and standard SQL. Some of these are pointed out during the lectures and in the lab assignments.

Connecting, Changing Password

When you start a client you must supply some connection parameters, either on the command line or in an option file (see below): the user account name, the password, the name of the host on which mysqld runs, and the database to use. On the command line, it looks like this:

```
% mysql -u <username> -p -h <hostname> <databasename>
```

Your user name is one of `db01`, `db02`, ...; `-p` tells `mysql` to ask for the password; the server runs on `puccini.cs.lth.se`; your database has the same name as your account. If your user name is `db01`, it should look like this:

```
% mysql -u db01 -p -h puccini.cs.lth.se db01
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 113
Server version: 5.6.14 MySQL Community Server (GPL)

Copyright ...

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

You can now type SQL commands:

```
mysql> select count(*) from Students;
+----------+
| count(*) |
+----------+
|       72 |
+----------+
1 row in set (0.01 sec)
```

You must change your password the first time that you use `mysql`, like this:

```
mysql> set password = password('newpassword');
Query OK, 0 rows affected (0.00 sec)
```

The connection parameters may be saved in an option file. Create a file *.my.cnf* (note the leading dot) in your home directory, with the following contents (for the user `db01`):

```
[client]
host = puccini.cs.lth.se
user = db01
database = db01
```

With this file in place, you can connect with `mysql -p`. (It is possible to save also the password in the option file, but it is not a good idea to do so.)

mysql Commands

There are two kinds of commands: SQL statements (`select`, `create table`, ...) and `mysql` commands. SQL statements are terminated by a semicolon (or `\G`, see below), `mysql` commands are terminated by carriage return.

NOTE: `Ctrl-C` interrupts a long-running query on the server. If no query is running, it terminates `mysql` and you must reconnect.

Useful commands:

`exit`	Exit `mysql`.
`help topic`	Help about `mysql` and SQL (e.g., help `create table`).
`nopager`	Disable pager (see `pager`).
`pager prog`	Pipe query output through `prog`, usually `less`.
`prompt newprompt`	Change the `mysql` prompt to `newprompt`.
`set names charset;`	Set the client character set and connection character to `charset` (should be `utf8` under Linux). Not necessary if you have `default-character-set` in the option file.
`set password = password('newpwd');`	Set password to `newpwd`.
`show x;`	Show information about `x`, e.g., `show tables`.
`source file`	Execute the commands in `file`. Do not use tabs for indentation (see `<tab>`).
`system cmd`	Execute the shell command `cmd`, e.g., `system ls`.
`use db`	Use another database `db`.
`\c`	Clear the input buffer.
`\G`	Terminate a command, display result vertically (useful for `select` commands which return long rows).
`<tab>`	Auto-completion of table names and attribute names.

There are many more commands; see the MySQL manual.

Your Account on Puccini

Your MySQL account has been created as follows (for the user `db01`):

```
create user db01 identified by 'somepassword';
create database db01;
grant all on db01.* to db01;
revoke create routine, alter routine, event on db01.* from db01;
```

You can connect to your account from any domain. You have all "simple" privileges on your own database, except the privileges to create stored routines or events. Stored routines are executed on the server, and a misbehaving routine could consume all CPU time on the server. Events can be recurring and also consume CPU time.

Don't save any important data in your database (save SQL commands in script files instead and copy-paste the commands or `source` the file). We don't take any backups at all of the databases.

MySQL at Home

MySQL is easy to install and is available on most platforms. We recommend that you install MySQL on your own computer if you have the possibility to do so; it opens the possibility for you to explore database constructs that you cannot use at the student computers, e.g., stored routines.

Some hints for the installation:

- Download the latest "recommended" release of MySQL Community Server for your platform from `http://dev.mysql.com/downloads/mysql/`. This is currently version 5.6.xx.
- Follow the installation instructions in section 2 of the manual. Don't forget section 2.11.3, Securing the Initial MySQL Accounts, where you assign a password to root and remove the anonymous accounts.
- From MySQL version 5.5, InnoDB is the default storage engine. If you use an earlier version of MySQL, you must create an option file for the server (*mysql/data/my.cnf* on Unix, `C:\my.cnf` on Windows) with the following contents (minimally):

  ```
  [mysqld]
  default_storage_engine = InnoDB
  ```

  If you don't do this all tables will be created as MyISAM tables, which don't support foreign keys and transactions.
- Create a user for normal use of the database (without administration privileges). Do this as root:

  ```
  mysql> create user db01@localhost identified by 'somepassword';
  mysql> create database db01;
  mysql> grant all on db01.* to db01@localhost;
  ```

  Omit the `@localhost` part from the username if you wish to connect to your server from other domains (or specify, for example, `@*.lth.se`).
- Now, you should be able to use MySQL exactly as you do at the student computers. Additionally, you have the privileges to create stored routines and events.

To use a MySQL database in a Java program you also need the JDBC driver Connector/J. It is available from `http://dev.mysql.com/downloads/connector/j/`.

You may also wish to download the graphical client and administrator program, the MySQL workbench, from `http://dev.mysql.com/downloads/tools/workbench/`.