A hands-on introduction to MySQL

The exercises below are aimed to bring you up-to-speed with using MySQL. You will not get points for completing them, but they will help you get used to the environment. For these exercises, the SQL statements shown in the classes so far are more than enough. You can refer to the course slides, or other online SQL documents on the web.

If you have any questions, do not hesitate to ask!

Connecting to the MySQL Database

Before starting you need a access to the MySQL server, for this course the name of the server is `mysql01.service.rug.` If you use access to another system with MySQL, this is also fine. You may need to adjust some of the instructions below.

All exercises can be run using either of the two methods:

- through the web interface phpMyAdmin at `http://siegfried.webhosting.rug.nl/phpmyadmin/` For documentation on phpMyAdmin, you can start from the official site at `http://www.phpmyadmin.net/`.
- `mysql` command-line client from the server `siegfried.webhosting.rug.nl`. For this you'll need an account on `siegfried.webhosting.rug.nl`, and this interface may look a bit less 'user friendly', it is a lot more flexible for some tasks.

  To be able to use `mysql` command-line client you need to connect to the server siegfried using ssh. You can normally connect to a remote MySQL server any computer, but connections to the MySQL server from other systems are not allowed.

The rest of the document explains the steps for using the command line client, you can also do all with phpMyAdmin, which should be easier to figure out.

To connect a MySQL server with the `mysql` command line tool, you need to type

```
mysql -p -u <db_username> -h mysql01.service.rug.nl
```

on Linux command line. The string '`<db_username>` should be replaced with your user name on the server. It will ask you to type your password. If you type it correctly, mysql will greet you and give you the prompt '`mysql>`'. You re now connected to the MySQL server.

If you are using phpMyAdmin, you should log in to using database username and password, and once you logged in you will be able to run the SQL commands below.

Now, you should have connected to the database server. At this point, you may want to change your password, if you want to change it, type

```
set password = PASSWORD('new_password');
```

on the mysql command line. Of course, you will need to replace `new_password` with the new password you'd like to use. Note that the password will be visible on the terminal screen.

So far, you are connected to the DBMS server, MySQL, but you did not select a database to work with yet. The command `show databases;` (the semicolon is needed), shows the databases available to you. If your account settings are in order, you should have (at least) two databases listed: one named same as your student ID, and one more with the name `information_schema` which is used by MySQL internally. `information_schema` database is for mysql internal use, we are interested in the other database. `connect <db_name>;` where `<db_name>` is the name of the database to select. Alternatively, you can

specify the name of the database on the command line like `mysql -p <db_name>` (Note that the space between the `-p` and the database name is significant). Once we are connected to the database server, and selected a database, we are ready to type some SQL commands.

The ultimate reference for MySQL commands and usage is the MySQL reference manual that can be browsed or downloaded here: `http://dev.mysql.com/doc/#manual`.

TIP: URLs in this document are clickable.

Exercises

For the exercises below, we will use the following table:

ISBN	title	author	year	pages	publisher
1857230744	The Left Hand of Darkness	Ursula K. LeGuin	1969	286	Orbit
055338256X	I, Robot	Isaac Asimov	1950	272	Spectra
0575070536	Roadside Picnic	Arkady & Boris Strugatsky	1972	145	Orbit
0553383043	A Wizard of Earthsea	Ursula K. LeGuin	1968	192	Spectra

Exercise 1: Creating and destroying tables

(a) Create the table presented above using the name 'books'. Only create the table with your choice of data types, do not populate it yet. And do not bother with assigning a primary key, but think ahead: which field(s) could be a good primary key?.

(b) You have just realized that an important attribute, the weight of the book is missing. Drop the table first, and re-create the table using this new field.

TIP: You can recall the previous commands using ↑ ('UP' key on your keyboard), and change them in `mysql` command line.

TIP: `show tables;` command shows the tables in the database you are connected. Similarly, you can display information about your newly created table using the command **`describe`** `books`.

Exercise 2: Manipulating Data

(a) Insert the data about books in the table above into the table you have crated. For now, we do not have information about weights of these books.

TIP: Remember that you can use the command **`select`** `*` **`from`** `books;` to retrieve everything in the books table.

(b) Being excited with all these new stuff, you typed the last insert command twice (please do it, I know you are not that excited). What is MySQL's reaction? Is it what you would expect?

(c) You realized that the initial table has listed one of the author names as 'Ursula K. LeGuin', it should have been 'Ursula K. Le Guin'. Write the query to fix that mistake.

Exercise 3: Primary Keys

(a) While creating the books table in Exercise 1 we could not decide for the primary key, so we created the table without a primary key. This is obviously a mistake. After thinking a while, we decided that ISBN is a good key for our purposes. We have doubts, of course. For example some old books do not have ISBN. But we hope that we wouldn't need to handle one of those, and go for ISBN as the primary key.

Create a new table with the same structure as books table, call it `books2`.

We do not want to drop the books table and recreate it, since we do not like the idea of inserting the data again (yes, command-line editing is great, but we do not want to do things twice: it is said that best programmers are the lazy ones).

(b) The command `insert into books2 select * from books;` allows us to transfer contents of the books table into the new books2 table. This command only works because both tables have the same structure. Run the command and populate books2. Did you get any errors? Are two tables different than each other? Explain why?

(c) We learned that there is another magic command:
`alter table books add primary key (ISBN);` which would have required a lot less work. Does this solution work in our case? If not why?

(d) Now that we have a copy of the data, we feel we can be experimental and try to see if we can fix the problem with the books table in another way. Can you think about a way to delete only one of the duplicate records?

(e) Some of us prefer to just be done with these exercises, so what we want is just get rid of the books table, which is obviously bad. But it bothers us that the table that keeps the records of books are called books2. Luckily, there is a command `alter table books2 rename to books;` that can rename the tables. Remove the table books and rename books2 as books. Is it possible to remove the table using command `delete from books;` statement instead of `drop` statement?

Exercise 4: Primary Keys

We do not have much to query for, but just for the sake of completeness

(a) Display all data about the books that were written by 'Isaac Asimov'.

(b) Find all authors that work with the publisher 'Spectra'.

(c) The choice of books seem to show the age of the person who put the list together. We feel like reading something modern from the list, well... as modern as it gets. Display the title and the year of the book(s) that were written after 1970.