Cardiff University
School of Computer Science & Informatics

Dr Natasha Edwards

CM1102 Web Applications

SPRING LABS - **WEEK 3**

Databases and mySQL

Attempt as many exercises as you can. If you do not manage to finish all the exercises in the lab, please continue doing them in the next lab or at home. If you find the first few exercises too easy — skip to the harder ones. Remember, the lab tutors are here to help. If you get stuck — do not be shy, raise your hand and ask for advice. It is also ok to discuss the solutions with your peers (these labs are not assessed!), however make sure you understand everything by yourself.

Good luck!

PRELIMINARIES

1. The school has a MySQL server available at `csmysql.cs.cf.ac.uk`. You should have received an email from the systems managers containing your username and password, with which you can access the MySQL server (Note: these are **not** the same as your normal network credentials. You can always reset your MySQL password by going to `https://docs.cs.cf.ac.uk/services/passwords/`.) Further information on MySQL in the School is found at `http://docs.cs.cf.ac.uk/notes/mysql-in-the-school/`.

   You can interact with the MySQL server in several ways, using:
   - terminal-based SQL command interface to MySQL (see quick guide: `https://docs.cs.cf.ac.uk/notes/accessing-mysql-from-windows/`);
   - web-based front end phpMyAdmin, available at `https://phpmyadmin.cs.cf.ac.uk`;
   - GUI front end called "MySQL Workbench" (see: `https://docs.cs.cf.ac.uk/notes/accessing-mysql-from-windows/`.

   For these exercises, we will be using the first option, i.e. the terminal-based SQL command interface.

2. The first thing to do is to make sure you can connect to the MySQL server.

   ```
   > mysql -u <username> -h csmysql.cs.cf.ac.uk -p
   ```

   The `-u` switch allows you to specify your username, the `-h` switch defines the address of the MySQL server to which to connect, and `-p` means the user is to be authenticated by password. When you enter your password it will not be echoed to the screen.

   You should then see the MySQL monitor prompt (`mysql>`), for example:

```
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is ...
....


%Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

%Oracle is a registered trademark of Oracle Corporation and/or its
%affiliates. Other names may be trademarks of their respective owners.

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

mysql>
```

You can now type your SQL commands. Remember to terminate them with a semicolon. Try:

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| sample             |
| your_username      |
+--------------------+
3 rows in set (0.03 sec)
```

You will be presented with a list of available databases. You have permissions to modify the database that has the the same name as your username. There is also a database called `sample`. First, you need to select the active database:

```
mysql> use sample;
```

All the subsequent commands will apply to this database. Let’s explore what tables are available in it:

```
mysql> show tables;
+------------------+
| Tables_in_sample |
+------------------+
| network          |
| opsystem         |
| people           |
| systems          |
+------------------+
4 rows in set (0.02 sec)
```

You will be presented with a list of available tables.

Let's run some SQL queries. For example, to show the entire table use this command:

```
mysql> select * from systems;
+-------+--------+-----------+
| name  | oscode | ownercode |
+-------+--------+-----------+
| blue  |      1 |         2 |
| red   |      2 |         2 |
| green |      3 |         1 |
| brown |      2 |         3 |
+-------+--------+-----------+
4 rows in set (0.03 sec)
```

If everything is working as in examples above, you may proceed to the remaining exercises. Otherwise ask for help.

Note: I strongly recommend that you take time to install a web server, python, and MySQL client/server on your home machine — this will make doing exercises and the coursework much easier. However, when you submit the coursework, make sure to test that your solution works on the school server!

For the subsequent exercises use ***your personal database*** - the one that has the same name as your **student number**! Do not use the `sample` database, as you do not have write permissions for it.

SQL QUERIES AND COMMANDS

3. Select your personal database. Create a table called `Staff`, similar to the one in the lecture notes:

   ```
   mysql> create table Staff (staffNo varchar(10), firstName varchar(30),
          lastName varchar(30), position varchar(20),
          salary int, branchNo varchar(10));
   ```

   Verify that the table has been created (albeit empty at the moment) and examine it:

   ```
   mysql> show tables;
   mysql> describe Staff;
   mysql> select * from Staff;
   ```

4. When creating a table, you need to consider which data types the attributes should have. For instance, in the above table we chose to store first names as strings of variable length up to 30 characters (i.e. **`varchar(30)`**), and salaries as integer numbers (i.e. **`int`**).

   Explore `http://www.w3schools.com/sql/sql_datatypes_general.asp` to find out more about the available data types. Further, when creating a table you may want to specify additional options and constraints for your attributes. Use the documentation at `http://www.w3schools.com/sql/sql_constraints.asp` to find out:

   (a) How to make sure that the `firstName` attribute always has a value (i.e. to disallow `NULL` values).
   (b) How to set the default value for salaries to be 10,000.

5. Delete the table using:

   ```
   mysql> drop table Staff;
   ```

   Ensure that the table has indeed been deleted:

   ```
   mysql> show tables;
   ```

   Then create the table again.

6. You should always define a primary key for your tables. In some cases, one of the existing attributes of the data that you are modelling may serve as the natural primary key (for example, unique student number), in other cases you may need to create a surrogate key just for the purposes of uniquely identifying the rows. Primary key may even consist of multiple attributes. Read http://www.w3schools.com/sql/sql_primarykey.asp to find out about the syntax for the primary key constraint.

   For the sake of example, let’s use `staffNo` as the primary key in our table `Staff`. We can declare it so:

   ```
   mysql> alter table Staff add primary key (staffNo);
   ```

   (Or alternatively you could have done it when mysqlcreateing the table.) Inspect your table now with:

   ```
   mysql> describe Staff;
   ```

   What has changed?

7. You can use the `alter` command to alter other properties of your tables after it has been created. For example, to make sure that the salaries are never unspecified and the default value is 10,000:

   ```
   mysql> alter table Staff modify salary int not null default 10000;
   ```

   Inspect your table again with:

   ```
   mysql> describe Staff;
   ```

   What has changed?

8. Let’s populate the table with some data. Download the file `staff.csv` from Learning Central and place it in your current folder[1]. Examine it: it is simply a comma-separated (CSV) human readable table, containing the data seen in the lecture notes. The following ’scary’ command[2] will import this CSV file into our database table:

   ```
   mysql> load data local infile ’staff.csv’ into table Staff
          fields terminated by ’,’ enclosed by ’"’;
   Query OK, 6 rows affected (0.05 sec)
   Records: 6  Deleted: 0  Skipped: 0  Warnings: 0
   ```

   ### If, when loading data from CSV, you got this error:

   ```
   ERROR 1148 (42000): The used command is not allowed with this MySQL version
   ```

   it means the security settings are too strict and loading data from local files is not allowed. To remedy this, restart `mysql` with the following options:

   ```
   > mysql -u username -h csmysql.cs.cf.ac.uk -p --local-infile
   ```

   (Alternatively, import the data using `phpMyAdmin`: *select your database and table, and then use the* Import *feature from the main menu. For the* Format *select* CSV.)

[1] The folder where you were when you started `mysql`.

[2] See https://dev.mysql.com/doc/refman/5.7/en/loading-tables.html

Verify that the command worked:

```
mysql> select * from Staff;
```

You should see:

```
+---------+-----------+----------+------------+--------+----------+
| staffNo | firstName | lastName | position   | salary | branchNo |
+---------+-----------+----------+------------+--------+----------+
| SL21    | John      | White    | Manager    |  40000 | B005     |
| SG37    | Ann       | Beech    | Assistant  |  18000 | B003     |
| SG14    | David     | Ford     | Supervisor |  25000 | B003     |
| SA9     | Mary      | Howe     | Assistant  |  15000 | B007     |
| SG5     | Susan     | Brand    | Manager    |  32000 | B003     |
| SL41    | Julie     | Lee      | Assistant  |  14000 | B005     |
+---------+-----------+----------+------------+--------+----------+
6 rows in set (0.02 sec)
```

9. Similarly, using SQL commands, create a table called `Branch` using the appropriate data types for the attributes (and do not forget to define a primary key). Populate this table from CSV file `branch.csv`.

   Your table should look like this:

   ```
   mysql> select * from Branch;
   +----------+--------------+----------+
   | branchNo | address      | city     |
   +----------+--------------+----------+
   | B005     | 22 Deer Rd   | London   |
   | B007     | 17 Argyll St | Aberdeen |
   | B003     | 163 Main St  | Glasgow  |
   | B004     | 32 Manse Rd  | Bristol  |
   | B002     | 56 Clover Dr | London   |
   +----------+--------------+----------+
   5 rows in set (0.02 sec)
   ```

10. Write an SQL command to add[3] another branch to table `Branch`, with number `B008` located at *12 Millbank, London*. Examine your table.

11. Assuming `staffNo` was declared as primary key in the `Staff` table, what would happen if you tried to add another record with the value of `staffNo` that already existed in the table?

    For example, try the the following query (check and confirm that `SL21` already exists before you run the query):

    ```
    mysql> insert into Staff values
    ("SL21", "Horatio", "Nelson", "Vice Admiral", 1337, "B001");
    ```

    What is the result of the above query?

12. Write an SQL command to modify[4] the address of the branch `B008` to be *PO Box 3255, London*.

[3] See http://www.w3schools.com/sql/sql_insert.asp

[4] See http://www.w3schools.com/sql/sql_update.asp

13. Write SQL queries to answer the following questions (or carry out operations).

    (a) In which branch does David Ford work?

    (b) Which employees work in branch `B003`?

    (c) Which branches are located in London?

    (d) What are the names of employees who have salary greater than $20,000$?

    (e) What is the average salary? The largest? The smallest?

    (f) What are the names of employees who have a below average salary? Above average?

        (Hint: Use a sub-query to find the average salary first, then filter the employees using the result of this sub-query.)

    (g) Produce the list of all employees sorted by salary in ascending order. Do the same in descending order.

    (h) Who are the three lowest paid employees?

        (Hint: use data selection limit , e.g. see: `http://www.w3schools.com/php/php_mysql_select_limit.asp`*.)*

    (i) Determine the set of all job titles (positions). That is, find the list of all job titles without duplication.

        (Hint: `http://www.w3schools.com/sql/sql_distinct.asp`*.)*

    (j) What are the names and salaries of employees who work in London?

        (Hint: since the information about names and salaries is in table `Staff` *while the information about cities is in table* `Branch`*, you need to do a* join*.*
        See `http://www.w3schools.com/sql/sql_join.asp`*.)*

    (k) Are there any assistants working in Aberdeen?

    (l) Employees in which city have the highest salary?

    (m) What is the top salary in Glasgow?

    (n) Which branches do not have a manager?

    (o) Which branches do not have any employees?

    (p) Find all last names ending with an *"e"*.

    (q) Reward Susan Brand who works in B003 as a manager by raising her salary to $36,000$.

    (r) In fact, let's reward all managers: raise their salaries by 10%.

    (s) Branch `B007` is closing. Remove all employees working there, and remove the branch itself.

    (t) Hire a new employee. Her name is Mary Smith and she will be working as an assistant in the Glasgow branch with the starting salary of $22,000$.

Useful addresses	
The COMSC database server:	`csmysql.cs.cf.ac.uk` (MySQL)
Upload your website to:	`websites.cs.cf.ac.uk`
View your website via:	`http://project.cs.cf.ac.uk/mailname` (where ’mailname’ is your user name)
COMSC `phpMyAdmin`:	`https://www.cs.cf.ac.uk/phpMyAdmin`
MySQL manual:	`https://dev.mysql.com/doc/refman/5.7/en/` *
SQL statement syntax:	`https://dev.mysql.com/doc/refman/5.7/en/sql-syntax.html` * * You can select documentation for other versions as well on that page.
SQL tutorial and reference:	`http://www.w3schools.com/sql/`
SQL and MySQL cheat sheets *(in no particular order)*	`[1]`, `[2]`