

COMP284 Practical 6

MySQL

Introduction

- This worksheet is preparation for the final set of PHP exercises. We intend to explore how we can interact with a DBMS from within a PHP script. In order to be able to do so, we will first need to set up a database. We will do so using the departmental MySQL/MariaDB DBMS that you should already be familiar with.

While you work through the exercises below compare your results with those of your fellow students and ask for help and comments if required.

- You might proceed more quickly if you cut-and-paste code from the PDF file. Note that a cut-and-paste operation may introduce extra spaces into your code. It is important that those are removed and that your code exactly matches that shown in this worksheet.
- The exercises and instructions in this worksheet assume that you use the Department's Linux systems to experiment with PHP.
- To keep things simple, we will just use a text editor, a terminal, and a web browser. You can use whatever text editor and web browser you are most familiar or comfortable with.
- If you do not manage to get through all the exercises during this practical session, please complete them in your own time before the next practical takes place.

Exercises

1. The steps of this exercise are only necessary if you have never used the department MySQL DBMS before and have not set up a password for your account with it. If you already have, go to Exercise 2.

- a. Open a terminal and connect to the MySQL server using the command

```
mysql -h studdb.csc.liv.ac.uk <user>
```

where *<user>* should be replaced by your University (MWS) username (*<user>* indicates which database you want to use).

If successful, you should see the following output in the terminal, and you are now interacting with our MySQL/MariaDB server using its command line interface:

```
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 MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 71896
Server version: 10.5.16-MariaDB-log MariaDB Server
```

```
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
```

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

- b. By default, your database is not password protected, meaning that anyone can access it. To set a password for your database, use the following command at the MySQL prompt:

```
set password=password('<password>');
```

where *<password>* should be replaced by a password chosen by yourself. Note that you must be able to remember that password. It must also not coincide with your University (MWS) password.

If successful, you will see the output

```
Query OK, 0 rows affected (0.00 sec)
```

- c. We should now test whether your password is working. First, disconnect your connection to the MySQL server by using the command

```
quit;
```

You are now back at the shell command prompt and should reconnect to the MySQL server using **one** of the following two commands:

```
mysql -h studdb.csc.liv.ac.uk -p <user>
```

```
mysql -h studdb.csc.liv.ac.uk -u <user> -D <user> -p
```

where *<user>* should be replaced by your University (MWS) username (the option *-u <user>* specifies the MySQL user id that you want to use when connecting to the MySQL server, it defaults to your University user name).

You should now be asked to enter a password:

Enter password:

After entering the password that you have specified in Exercise 1b you should be back at the MySQL prompt.

2. Let us create our first database table for COMP284 in MySQL.

- a. Create a database table using the following command at the MySQL prompt:

```
create table meetings(slot INT NOT NULL, name VARCHAR(50),  
                      email VARCHAR(50), primary key(slot));
```

- b. If the command in Exercise 2a was successful, then

```
describe meetings;
```

should produce the following output

```
+-----+-----+-----+-----+-----+-----+  
| Field | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| slot  | int(11)       | NO   | PRI | NULL    |      |  
| name  | varchar(50)   | YES  |     | NULL    |      |  
| email | varchar(50)   | YES  |     | NULL    |      |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)
```

- c. We now want to fill the meetings table with the following data

slot	name	email
1	Michael North	M.North@student.liverpool.ac.uk
5	Jody Land	J.Land@student.liverpool.ac.uk
7	Trish Shelby	T.Shelby@student.liverpool.ac.uk
11	Wayne Ungood	W.Ungood@student.liverpool.ac.uk

- Devise four INSERT-statements that put this data into the meetings table and execute all four.
Hint: Refer to <http://dev.mysql.com/doc/refman/8.0/en/insert.html> if you have forgotten the syntax of INSERT-statements.
- Check that the data has been inserted correctly by using

```
select * from meetings;
```

The output should be

```
+-----+-----+-----+
| slot | name          | email                               |
+-----+-----+-----+
| 1    | Michael North | M.North@student.liverpool.ac.uk  |
| 5    | Jody Land     | J.Land@student.liverpool.ac.uk    |
| 7    | Trish Shelby  | T.Shelby@student.liverpool.ac.uk  |
| 11   | Wayne Ungood  | W.Ungood@student.liverpool.ac.uk  |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

- Devise a SELECT-statement that just returns the values stored for the slot attribute in the meetings table and execute it.
Hint: Refer to <http://dev.mysql.com/doc/refman/8.0/en/select.html> for information on SELECT-statements.
- Devise a DELETE-statement that removes the database entry for slot 11 and execute it.
Hint: DELETE-statements are described at <http://dev.mysql.com/doc/refman/8.0/en/delete.html>.
- Check that the deletion in the previous step has been successful.

d. Disconnect from the MySQL server by using the command

```
quit;
```

3. To make sure that your work is not lost or to transfer it to a different server, you should create a backup of your database, also called a *dump* of your database. You should have learned how to do so on the introductory database module. But in case there is a planet-size hole where that knowledge should be, here is what you need to do.

On a shell command line, execute the command

```
mysqldump -h studdb.csc.liv.ac.uk -u <user> -p <user> meetings > backup.sql
```

where *<user>* should be replaced by your University (MWS) username (the second occurrence of *<user>* is the name of your database, which happens to be your username). This will create a file backup.sql that contains all the SQL statements necessary to re-create the meetings table in your database and its current contents. In general, if you want to create a backup of several tables, then you just list them all after the second occurrence of your username in the command above. You can also omit the list of tables in which case your whole database with all its tables will be dumped.

References

- [1] Oracle. *MySQL 8.0 Reference Manual*. mysql.com. 27 January 2023. URL: <https://dev.mysql.com/doc/refman/8.0/en/> (accessed 29 January 2023).