# 4CS015 Fundamentals of Computing

**MySQL using Shell Scripts Assessment**

**DEADLINE IS 11/01/18**

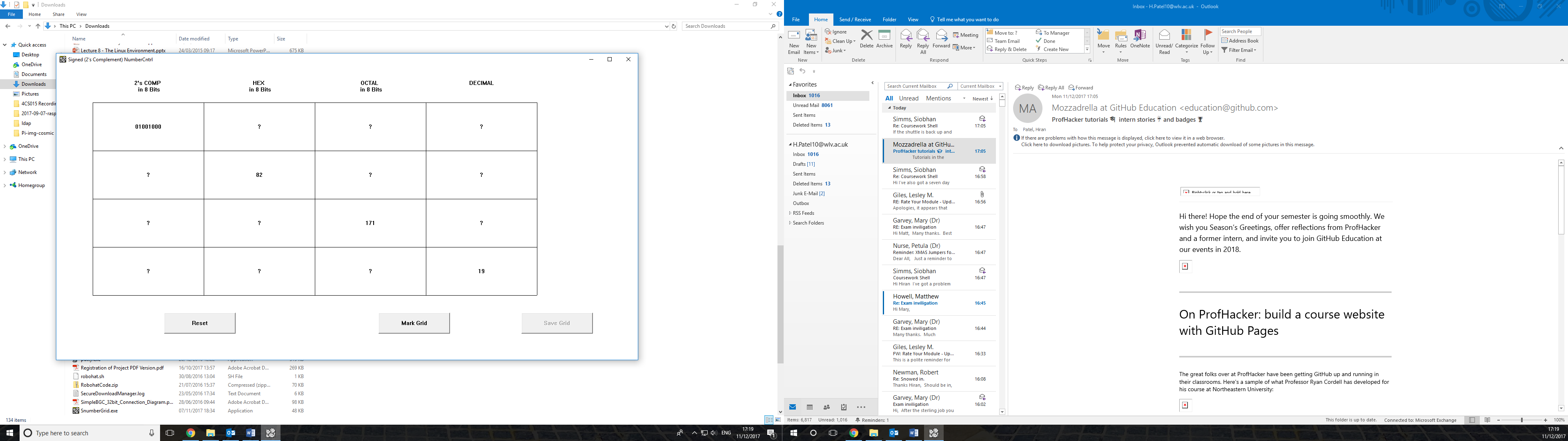
Before you start this assessment, refer to the workshop related to using MySQL using shell scripts. This document has all the source code and explanations needed to complete this assessment.

**Objective**

The objective of this assessment is to create a shell script to select, manipulate and store data in appropriate fields using MySQL. This assessment will test your knowledge of MySQL and shell scripts. In this task, you will convert a specific formatted number into three different values that you have learnt before. Instead of converting user inputted values, you will take a specific value from a table, manipulate it and then send it back in the appropriate column (similar to the NumberGrid assessment you first completed).

**Task 1.** The first shell script you need to create is to generate a table called NumberGrid which holds 5 fields:

1. NG\_ID – this should be an integer and should not contain NULL values (can’t be empty). The limit for this field should be just 1 digit.
2. NG\_Binary – this should be a varchar (acts as a string) which has a limit of 8 characters
3. NG\_Hex - this should be a varchar (acts as a string) which has a limit of 2 characters
4. NG\_Oct - this should be a varchar (acts as a string) which has a limit of 3 characters
5. NG\_Dec – this should be an varchar which has a limit of 4 digits

**Task 2.** The next shell script should partially fill up the table with values. Refer back to the NumberGrid software (screenshot below):

The question marks in the picture above should be empty values within the MySQL table. After creating a table with the appropriate fields, you now need to populate the table with appropriate values. To make life easier, use the values in the table shown above or below this paragraph, these should be your starting values for the table. Create a shell script which inserts these values appropriately into the table. NOTE – NG\_ID should be set to 1,2,3,4 working from top to bottom. Your table should now look like the one below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 01001000 |  |  |  |
| 2 |  | 82 |  |  |
| 3 |  |  | 171 |  |
| 4 |  |  |  | 19 |

The next part of the assessment is to dynamically fill in the remaining sections of the rows.

IMPORTANT NOTE – You should always fill in the remaining data on row 1 depending on the Binary value, row 2 is dependent on the Hex value, row 3 is dependent on the Octal value and row 4 is dependent on the decimal value. This is important because to test your program, you will change the binary value (on row 1) and update the remaining three columns. The next test will be to change the binary value on row 2 but this should not make a difference to the remaining three columns as the Hex is the dominant value in the second row.

**Task 3.** This next task requires you to store all/parts of the data into a shell script variable. Using the algorithms and programs you have already written, convert these values to the remaining three that you need. For example, when working on row 1, you will need to convert the binary value to Hex, Oct and Decimal. NOTE – you will not be penalised if you cannot convert a negative decimal number. Once you have stored all three values in separate shell script variables, you will then pass those arguments into a query. You will then repeat this step for row two but remember, this time, you are taking the HEX value and converting it into Binary, Octal and Decimal. This needs to be repeated for the remaining rows too.

If the binary number in any row has a 1 at the beginning, count this bit as 128. For example, if row 1 has a binary value of 11001100, the value in the decimal column can either be 204 or -52. Ideally the decimal value should be -52 for this example however, this task does not assess number system conversions, it assesses whether you can manipulate a value and store it within the appropriate column using SQL.

**Task 4.** The final task is to store the converted values within the appropriate fields in the table.

Marking criteria:

1. Creating table with appropriate fields and limits (15 marks)
2. Fill up table with dummy data (15 marks)
3. Requesting values from database and storing them into shell script variables (30 marks)
4. Manipulation of values and sending them back into the database within the appropriate columns (50 marks)