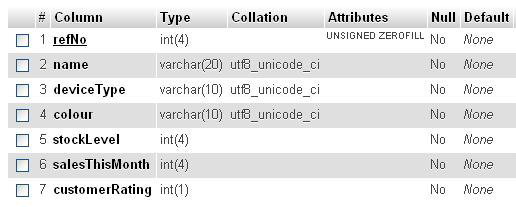
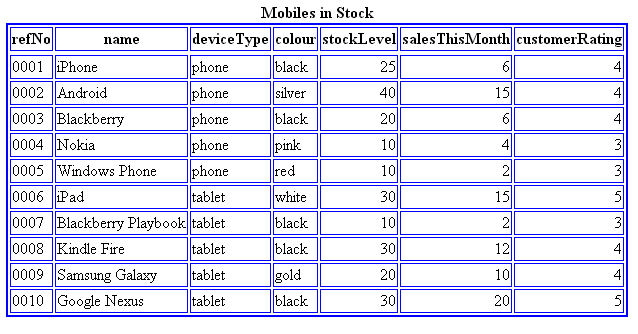
WEEK 6 NOTES

1. Introduction to MySQL
   1. The examples in these Lecture Notes will use a small database called *mobiles*, which is hosted on the Laureate server.
      1. Data structure of *devices* table:



* + - 1. Numbers in parenthesis specify maximum size.
      2. Varchar is used for text.
      3. UNSIGNED ZEROFILL means that the reference number (refNo) will be a positive value (or zero) and leading zeros will be inserted to pad the content up to the maximum of 4 digits.
         1. E.g., 2 becomes 0002.
      4. Primary key: refNo; it cannot be duplicated.
      5. “No” in Null column means that the field must be filled.
    1. Contents of the database:



* 1. MySQL has import and export functions:
     1. The simplest way of exporting the tables within a database is to select the Export command and then ask the database software to create a .sql file.
        1. This will contain Structured Query Language (SQL) commands to recreate a table structure and insert records on another copy of MySQL (using the Import command).
     2. When you are importing, create an empty database in MySQL (we shall give you details of how to do this elsewhere in Blackboard) and then select the Import command within that empty database.
        1. This will allow you to point at the .sql file you have exported.
  2. The devices.sql file enable you to recreate the devices table shown above on a version of MySQL running on your computer.

1. allDevices.php file
   1. Lists the database’s contents when we click on the ‘List all the devices’ link on the Mobiles Database main page
   2. Variables/objects in the first PHP section
      1. $dsn: The data source name consists of the name of the server hosting MySQL plus the name of the database.
      2. $dbc: Database connection object. When it is created, the following are required:
         1. Data source name: *$dsn*
         2. Username
         3. Password
      3. $query: self explanatory
      4. $results: Used to execute the SQL query stored in the variable $query, and saves all the data returned in $results
      5. $rows: Object used to establish how many rows of data have been extracted
   3. Logic:
      1. If there are no rows of data, echo “there are currently no records….”
      2. Else: echo $results
2. allDevices.css file:
   1. Helps with the formatting of the HTML table
3. Database Connections/visibility
   1. Users usually don’t see the source code $dsn=…, $dbc=new…
      1. One solution would be to place this connection code in a PHP *include* file which would be located in a directory that has *not* been made publicly visible by being nominated as some Web server’s home directory or one of its sub-directories.
         1. E.g. Move this sensitive code to a file named *connection.php*
            1. include("../../connection.php");
   2. Only the Web server’s home directory and its sub-directories are publicly visible on the Internet.
   3. Week 6 hand-in assignment
      1. You should utilize an *include* statement to connect to the database pointing at the information in the *include* file *connection.php*.
      2. Substitute $dsn = , $username = , $password = , $dbc = with:

Include(“../../connection.php”);

1. singleDevice.html file
   1. Used to view a single record in the database.
   2. Requires user input, so there are both HTML and PHP files.
2. updateDevice1.php file
   1. Checks to see if record given by the user exists. If it does, the program will display the current contents of that record in editable HTML text boxes using this code:

value="<?php echo($colour); ?>">

* 1. If the user clicks submit, then updateDevice2.php will run, which updates the database and report the outcome.
  2. Uses a hidden field:

<input type="hidden" name="refNo" value="<?php echo($refNo); ?>">

* + 1. Prevents the user from changing the reference number. Solves this by retrieving the reference number from the HTML form in updateDevice.html and store it in the PHP variable, $refNo. We then insert that variable in a hidden field which, as its name suggests, is not displayed on screen. However, the contents of the hidden field are returned to updateDevice2.php in the same way as other HTML form fields. This value is then used by updateDevice2.php in the SQL statement to identify the record which needs to be updated.

1. updateDevice2.php file
   1. Retrieves data from the HTML form contained in updateDevice1.php
   2. SQL statement
      1. "Update the devices table by setting each field to the contents of the HTML form in updateDevice1.php". Only where the user changed something will any difference be seen