

Internal Assessment Resource

Digital Technologies Level 3

|  |
| --- |
| This resource supports assessment against:  Achievement Standard 91633v2  Implement complex procedures to develop a relational database embedded in a specified digital outcome  Assessed: Internally Credits: 6 |
| Assessment: Botany Library Database  Due Date: Friday, 1 September  Name:  **Candidate’s Statement**  This assessment has been done entirely by me. I have neither borrowed work from others nor lent my work to others to use (and I understand that doing so constitutes plagiarism).  Signed Date |

|  |  |  |
| --- | --- | --- |
| Achievement | Achievement with Merit | Achievement with Excellence |
| Implement complex procedures to develop a relational database embedded in a specified digital outcome. | Skilfully implement complex procedures to develop a relational database embedded in a specified digital outcome. | Efficiently implement complex procedures to develop a relational database embedded in a specified digital outcome. |

Student instructions

Contents

[2 Introduction 2](#_Toc454466115)

[3 Specifications 3](#_Toc454466116)

[4 Processes 4](#_Toc454466117)

[5 Logs 5](#_Toc454466118)

# Introduction

The Botany Downs library needs a new database and have asked you to help them. The database will record information about borrowers, staff, books/CDs/DVDs. It should also allow staff to record books issued to borrowers and ‘clear’ them once they are returned.

The library wishes to make the new database as simple to use as possible for both staff and borrowers. They have asked for the database to be stored on a web server and the interface to be through a web browser, designed in such a way to make it easy to read information (and record it where necessary).

The database will be expanded in the future as new borrowers and staff join and as new books are purchased so it is critical that the data is structured in such a way that there will be no anomalies when data is added, deleted or modified. The database should also store the data in the most efficient way possible to save space.

The library would like you to thoroughly test the database and its interface and send them the results of your testing.

# Specifications

The following is a list of requirements (specifications) the library has for their database and interface:

## The data in the database should:

* Be free from errors and inconsistencies
* Be split into multiple tables
* Be represented by appropriate data types
* Be stored efficiently (and in such a way that it prevents insert, update and delete anomalies)

## The database web interface should:

* Be attractive, following design principles.
* Be simple and intuitive reducing the chance for errors.
* Allow borrowers to search the database for titles by typing a search string (borrowers do not need to be logged in to do this) and see information about the titles they search for.
* Have an admin section that allows staff (once logged-in) to:
  + issue the books that a borrower takes out (and return them)
  + add new books to the database as they are purchased
  + add new borrowers to the database
  + edit the details of existing borrowers.
  + NOTE the username should be *admin* and the password should be *password*
* Allow for the input of a range of characters (including apostrophes in people’s names).

## Other features:

The library is open to you demonstrating other features to them that you think might be useful to include in their database system. This might include allowing users to check what items they currently have issued and the due dates of these items, allowing users to customise their search criteria or other things you think may increase functionality or ease of use of the system.

# Processes

**The process for developing this outcome will likely not be linear. You will find yourself implementing as you test and may choose to plan and implement a small section of the project before planning the next. That is fine as long as you have a plan for something before you work on it.**

## Plan

### Plan the Database

Before you begin any implementation of the database you must come up with a plan. Review the existing data and thing about any more data that will need to be held in the database.

You ***must*** submit your plan as part of your assessment. You can revise you plan as you move through the tasks if you find it necessary. The database plan should show all:

* Table names
* Field names
* Data types
* Attributes (data sizes etc.)
* Primary and foreign keys
* And relationships

*This is best done in the form of a database schema similar to practice tasks- Sports and DVD databases.*

### Plan the Queries

Plan out the sorts of queries you think you’ll need. Among others the database will require queries that can:



1 Image CC BY3.0 - [ThomasEagle](https://commons.wikimedia.org/wiki/User:Thomaseagle)

* show information about a particular library item or borrower
* Add information about items issued to a borrower
* Add new borrowers or items to the database
* Update current borrower information

### Plan the Pages and the Interface

Decide which pages the website will need and what the interface will look like. Draw these plans as wireframes and sitemaps. You may also wish to show which files are going to be linked to each other if you are going to send information between pages.

## Implement

### Implement the Database

Using your login to the school’s cPanel server, follow your plans to set up a database (and associated admin user). Once the database has been set up, add the supplied data (applying data integrity procedures to ensure it is correct and valid).

### Build the site

Following the plan you created for the site and planned queries build pages that will connect to the database, process user’s requests to display relevant information and modify data in the database.

## Test

### Test the Queries and Database

Before you use your queries on any pages, test them in the SQL tab of phpMyAdmin to ensure they return accurate results. If they don’t work or return unexpected results, adjust the queries or database to give you the results you need. Make sure you take screen clipped evidence of your testing of the database and queries to include in your log of testing.

### Test the website

As you develop your site you will naturally be testing the site to ensure you are making progress. You will likely be testing things like the connection to the database, the forms, the in-page queries, how the information is displayed, how the hyperlinks are working, how the queries work with a range of information (including information that may be risky to the data).

As you test each thing keep screen snipped evidence of what you are testing and describe your testing procedures in your testing log.

# Tempaltes

## Testing log

Use a log like this to keep a track log your testing

|  |  |  |
| --- | --- | --- |
| **Testing log** | | |
| Date, what was tested, and how it was tested | Evidence (screen shot) | What changes made if any? |
|  |  |  |
|  |  |  |

## Daily log of development

Keep a log of your development for each session you work on your database to use as evidence of your independence and efficiency. Use a similar format to the following:

|  |  |  |
| --- | --- | --- |
| **Record your database development at the end of each class** | | |
| Date / time | What have I done? | Screenshot if necessary |
|  |  |  |

# Plan

## Tables and Schema – (Download the data from OneNote)

Table name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| Field name | Datatype (and size if needed) | Keys / attributes |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Plan for linking data between applications

Explain how you will link the data that is stored in your phpMyAdmin to a webpage. i.e. which language and specifically which functions will you use to link these two things together.

|  |  |
| --- | --- |
| Questions | Your Answers |
| Language used to link your PHP MyAdmin database to your HTML webpage |  |
| Name and password of the user that you set up on your database |  |
| Which privileges does that user have |  |
| Function used in that language to connect your page to your database |  |
| Functions used in that language to query the database |  |

## Plan for how the website will look (site plan and wireframes)

|  |  |  |
| --- | --- | --- |
| Achieved: Implement complex procedures to develop a relational database embedded in a specified digital outcome | Achieved with Merit:  Skilfully implement complex procedures to develop a relational database embedded in a specified digital outcome | Achieved with Excellence: Efficiently implement complex procedures to develop a relational database embedded in a specified digital outcome |
| The student has:   * designed and constructed a database with a workable table structure   + Database plan (schema) has been produced that includes table names, datatypes, attributes.   + Plan shows how tables are linked together * created queries to ***retrieve*** and ***modify*** data   + some queries combine data from multiple tables   + some queries will insert and/or update data in tables * allowed data in at least one database table to be changed from another application   + update / insert queries run from the website successfully changes the tables in the database. * applied design elements and formatting techniques to customise data input and display   + pages are styled through CSS and make good use of design principles   + forms and data from the database have been customised with CSS * applied data integrity and testing procedures to ensure the outcome meets the specifications   + data has been checked before it is recorded in the database   + queries have been checked to ensure they work (should be documented)   + links between pages, and from forms to pages work correctly * applied data access permissions as appropriate to the outcome   + login system applied and only authorised staff members can record issues, update borrower information, add new items and borrowers | The student has:   * shown accuracy in designing and constructing the table structure, applying design elements, techniques and procedures   + tables are appropriately linked using foreign and primary keys   + datatypes are the most appropriate for the data and sizes/attributes are suitable   + table and field names are appropriate / consistent   + all pages and queries work as expected   + login system restricts non-staff members from editing the data   + pages show no PHP errors or warnings   + only minor errors may exist in relation to display or editing of information * shown independence with regard to decision making when producing the outcome   + student uses online resources and class notes / exercises effectively   + Test/Daily log shows independence in problem solving   + Student requires no (or only very minor assistance) | The student has:   * designed and constructed a database with a well-organised table structure (e.g. elimination of redundant data, effective data validation, effective use of data properties)   + organisation of data to third normal form means redundant data is eliminated   + form elements generated dynamically to reduce errors   + Thorough testing has been carried out to ensure that no errors are present in the display or editing of information   + Timely planning and production allows for the inclusion of extra features and/or thorough application of design principles * produced the outcome in a manner that economises the use of resources   + Datatype attributes allow for the most efficient storage of data   + Where code is used multiple times throughout the site ‘includes’ have been used to eliminate the need to type out the same code multiple times.   + CSS is externally linked not inline or internal |

Comments: N A M E