Name: Daniel Wilkinson

Course: Computer Science

University Code: Bg69tg

University Number: 1111863434

Phase 1: Database Design and Storyboard

Contents

[E-R Diagram 2](#_Toc436061287)

[Appendix / Data Dictionary 3](#_Toc436061288)

[Justification 4](#_Toc436061289)

[Story Board / flowchart 5](#_Toc436061290)

[Flow Chart Form Description 5](#_Toc436061291)

[SQL included separately in this zip file. 6](#_Toc436061292)

## E-R Diagram



My E-R Diagram shows that the module leader manages all of his/her modules. A module has both module lecturers and module assessments, I have implemented these tables to minimize data redundancy. For example in assessments will be a list of possible assessments and in the module assessments will be the ID’s of both assessments and modules this allows for a module leader to easily assign assessments to a selected module. The same implementation has been given to the lecturers at Sunderland University, I believe this is the best way to assign lecturers on a selected module.

## Appendix / Data Dictionary

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Entity | Attribute Name | Data Type | Length | | Validation | | Format |
| Assessment | Assessment\_ID | Number | 6 |  | | Primary key | |
| Assessment | Assessment\_Name | Varchar2 | 25 |  | |  | |
| Assessment | Assessment\_Length | Varchar2 | 25 |  | |  | |
| Assessment | Assessment\_Date | Date |  |  | |  | |
| Module | Module\_ID | Number | 6 |  | | Primary Key | |
| Module | Model\_Name | Varchar2 | 25 |  | |  | |
| Module | Module\_Desc | Varchar2 | 255 |  | |  | |
| Administrator | Admin\_ID | Number | 6 |  | | Primary Key | |
| Administrator | Admin\_Forename | Varchar2 | 25 |  | |  | |
| Administrator | Admin\_Surname | Varchar2 | 25 |  | |  | |
| Administrator | Admin\_Address | Varchar2 | 60 |  | |  | |
| Administrator | Admin\_DOB | Date |  |  | |  | |
| Administrator | Admin\_Approved | Char | 1 | ‘Y’ or ’N’ | |  | |
| Administrator | M\_ID | Number | 6 |  | | References(Module:M\_ID) | |
| External Markers | EM\_ID | Numeric | 6 |  | | Primary Key | |
| External Markers | EM\_Address | Varchar2 | 60 |  | |  | |
| External Markers | EM\_Postcode | Varchar2 | 25 |  | |  | |
| External Markers | EM\_Phoneno | Varchar2 | 15 |  | |  | |
| External Markers | EM\_Approved | Char | 1 | ‘Y’ or ’N’ | |  | |
| External Markers | EM\_Company | Varchar2 | 25 |  | |  | |
| External Markers | M\_ID | Number | 6 |  | | References(Module:M\_ID) | |
| Module Leaders | ML\_ID | Number | 6 | 000001-999999 | | Primary Key | |
| Module Leaders | MLForename | Varchar2 | 25 |  | |  | |
| Module Leaders | MLSurname | Varchar2 | 25 |  | |  | |
| Module Leaders | MLAddress | Varchar2 | 60 |  | |  | |
| Module Leaders | MLDOB | Date |  |  | |  | |
| Module Leaders | M\_ID | Number | 6 |  | | References(Module:M\_ID) | |
| Module Assessments | MA\_ID | Number | 6 |  | | Primary Key | |
| Module Assessments | Assessment\_ID | Number | 6 |  | | References(Assessment:Assessment\_ID) | |
| Module Assessments | M\_ID | Number | 6 |  | | References(Modules:M\_ID) | |
| Lecturers | L\_ID | Number | 6 |  | | Primary Key | |
| Lecturers | L\_Forename | Varchar2 | 25 |  | |  | |
| Lecturers | L\_Surname | Varchar2 | 25 |  | |  | |
| Lecturers | L\_Address | Varchar2 | 60 |  | |  | |
| Lecturers | L\_DOB | Date |  |  | |  | |
| Module Lecturers | MOL\_ID | Number | 6 |  | | Primary Key | |
| Module Lecturers0 | L\_ID | Number | 6 |  | | References(Lecturers:L\_ID) | |
| Module Lecturers | M\_ID | Number | 6 |  | | References(Modules:M\_ID) | |

## Justification

Below I will explain the rules I have stood by in my E-R Diagram and Data Dictionary to make it easy to understand and for clarity.

* All fields that are primary keys are integers and of the length of 6, I decided upon integer because it takes up considerably less space in comparison to Varchar2 which suggests that we are storing real data. I’ve also read that if an integer is between 0-1000000 the size will only ever be 4 bytes however it will more than double with a varchar, this proves for better performance in a database, the length I have chosen is 6 characters long and this is because each year the university will update modules, hire new staff etc. there will be plenty of room with the currently selected length.
* All fields that are either Forename or Surname are Varchar2 as I have read that it really doesn’t matter what data type you use out of the following “Text, Varchar & Char” however the Advantages of Varchar2 is that you can preset the length of data, SQL also converts the text data type to varchar in the long run anyway so it would be more efficient to just use varchar2 and finally it is said that Text still has a very small amount of bugs. In my design all forenames and surnames are 25 Characters long as I think this is an appropriate length.
* Addresses are a little longer and are of the type Varchar 2 with a length of 60 characters, this is because a place may have considerably longer name then a surname or forename meaning more space has to be given to the addresses, varchar2 also lets us use both text and integers which will come in handy for the house number alongside the street name. In my opinion 60 characters is a big enough address length.
* In the external marker I have used a phone number data type of Varchar2 and a length of 15, considering the University this system is for and they specify in the brief they will use an external examiner, sometimes this is out of the country and an international standard telephone number can be up to 15 numbers long, of course this may never get used however better to have the maximum then have to change this later if it ever pops up.
* The only large data field is the module description which is of data type varchar2 with a length of 255, this is because a module description can be quite long as this is what attracts a person to a course in the first place, therefor I have given a big length off 255.
* I have used a Char to signal the approval of both the administrator and the external examiner of the modules, I believe this is simple and effective way show when a module is approved. I have done this through references to the module ID inside of both admin and the external examiner, this way it is simple to get the modules that the admin or external marker has approved.
* I could have just had a User’s table instead of an admin, module leader and staff members, however the problem is that not all of these users have clearance to approve of a module, only an admin and external marker. How was I to incorporate this into my database? I decided upon giving each of these Users their own table but however with good reason, a column in External Markers names Company will not be in any other table there for making this table separate, another reason is that clearance and I needed to add a column with approval for both admin and external marker which left Module leader separate.

## Story Board / flowchart



The flowchart above shows how we reach the end goal of getting a module assessed by an external marker before getting the feedback and sending a notification of that feedback back into the database.

Firstly either an Administrator or a Module Leader will be required to login, once they have the system will know where you are supposed to be based upon the username and password entered. If you are an administrator you will have accesses everywhere including the final say in sending a module off, however if you are a module leader you will be restricted to editing the module which includes the features of creating and updating the database and saying either the module is ready to be checked or not. When all needed work is complete simply exit this menu.

## Flow Chart Form Description

|  |  |
| --- | --- |
| Entity | Description Of Form |
| Login | On the login screen will be 2 textboxes and a button to click to go to a page based upon what is entered into the fields. |
| All Modules | The all Modules form is the result of an admin longing in to the system, from the login they will arrive at a page that displays all of the modules. There will be a menu that gives access to all other forms. |
| Selected Modules | The selected Modules page is the result of a module leader logging in to the system. This will display the modules this module leader is leader of or on. There will be a more restricted menu to other forms. |
| Update Module | On the update form it is possible to update a module, there will be a search bar, sort & order fields and the menu will still be at the top as a form of good navigation. |
| Insert Module | On the insert module form it is possible to insert a new module, this will include all of the information from the module entity in my data dictionary, consisting of text boxes, drop down boxes and other input methods preferably to reduce user error. |
| Delete Module | Just like insert form this form will have the same feature but however allows an admin to remove a selected module. |

## SQL included separately in this zip file.