COM1025 Web Publishing and Databases

Coursework 2

FULL NAME: Nithesh Koneswaran
USERNAME: NK00374
URN: 6474079

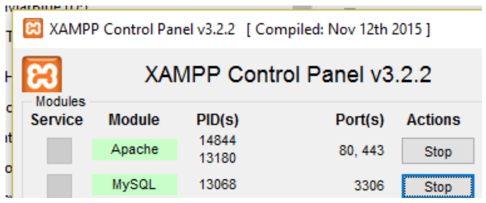
[illegible]

1 Developing and Testing Environment

DRAW.IO was used to draw my EER Diagram

I used Xamp portable to run my MySQL server.
The zip file of the Xamp has the name…
'xampp-portable-win32-5.6.24-1-VC11-min'
Its current version Is v3.2.2

I also used the same XAMP to test my web page.

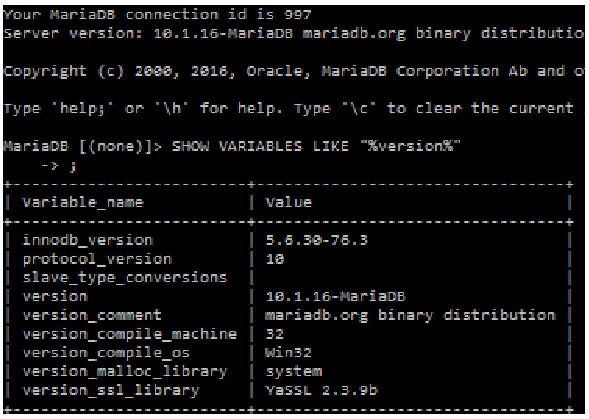


XAMPP Control Panel v3.2.2 [Compiled: Nov 12th 2015]

XAMPP Control Panel v3.2.2

Modules

Service	Module	PID(s)	Port(s)	Actions
	Apache	14844 13180	80, 443	Stop
	MySQL	13068	3306	Stop



```
Your MariaDB connection id is 997
Server version: 10.1.16-MariaDB mariadb.org binary distributio

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and o

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

MariaDB [(none)]> SHOW VARIABLES LIKE "%version%"
    -> ;
+-------------------------+---------------------------------+
| Variable_name           | Value                           |
+-------------------------+---------------------------------+
| innodb_version          | 5.6.30-76.3                     |
| protocol_version        | 10                              |
| slave_type_conversions  |                                 |
| version                 | 10.1.16-MariaDB                 |
| version_comment         | mariadb.org binary distribution |
| version_compile_machine | 32                              |
| version_compile_os      | Win32                           |
| version_malloc_library  | system                          |
| version_ssl_library     | YaSSL 2.3.9b                    |
+-------------------------+---------------------------------+
```

Operating system: Windows 10 PRO

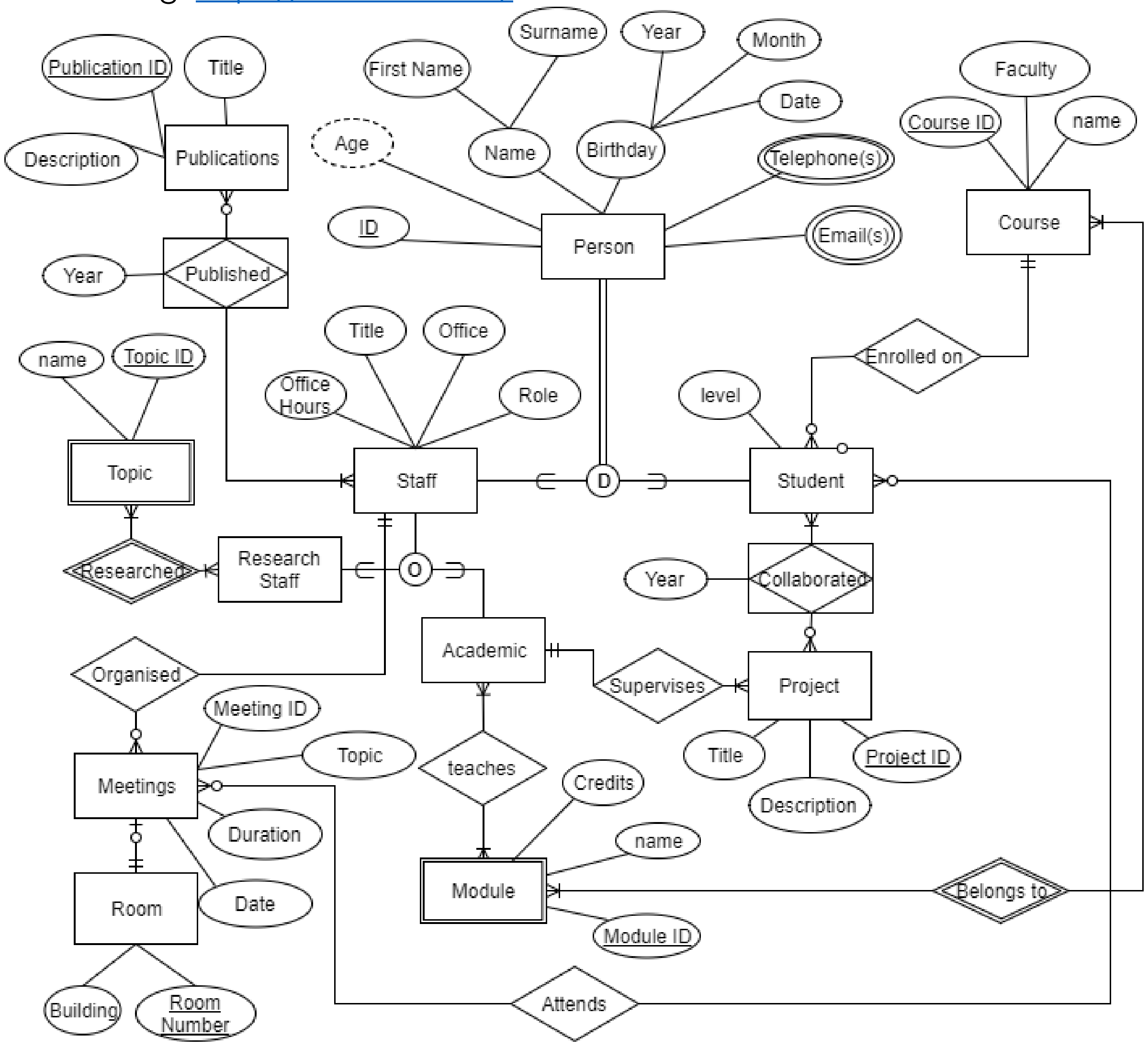
OS Build: 14393.1944

System type: 64-bit operating system

Google Chrome Version: Version 63.0.3239.132 (Official Build) (64-bit)

2 EER Data Model and Diagram

Made using: https://www.draw.io/



Attributes

The 'Person' Entity has different types of attributes. It has simple attributes that describes the entity and has two composite attributes that describes the user's name and birthday. In addition, the entity also has two multi-valued attributes as a person may have more than one telephone number or email and has the derived attribute 'Age' which would be derived (calculated) using the composite attribute 'Birthday'. The identifier attributes found in the entities will uniquely identify each person. All the other entities will also have attributes that describe the entity as well as identifier attributes just like the attributes in the 'Person' entity.

Entity Types

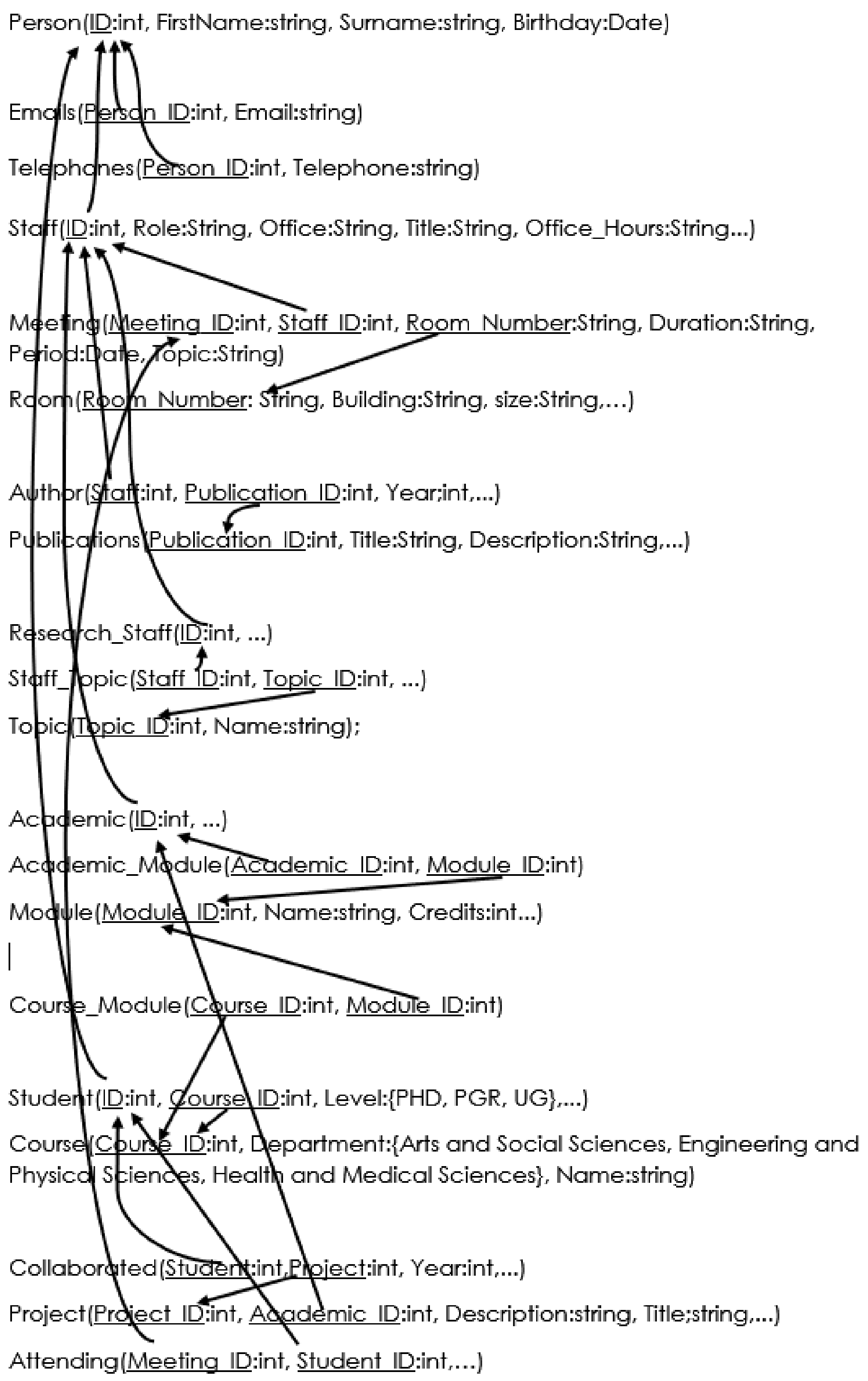
The 'Person' entity is the Super-type for the Sub-types 'Staff', 'Research Staff', 'Academic' and 'Student'. Each of the Sub-type will inherit the attributes of the 'Person' entity and will have unique attributes that are different to each other Sub-type. One additional entity type which I have included in my diagram is an associative entity type, which is a relationship that has attributes. Look at 'Published' and 'Collaborated' they both share the same attribute 'year'. The attribute 'year' for 'Published' will be the year the publication would have been published and for 'Collaborated' It will be the year the project would have begun. Lastly, I had included a weak entity called 'Module'. This is because the 'Module' entity depends on the course. I had also added the weak entity called 'Topic' because the entity depends on whether the research staff are researching a topic. A topic that exists that isn't currently being researched by a staff should not exist in the database. Therefore, I have included it as a weak entity.

Relationships
Between the **'Staff' and 'Publications' entity** we can see the relationship '**Published'**. The relation between the two is a many to many relationships. (Partial participation) One staff may publish 0 or more publications/ (total participation) one publication must belong to one or more staff. In addition to this, the relation 'Published' has attributes making it an associative entity type.
Between the '**Staff' and 'Meetings' entity** we can see the relationship '**Organised'**. The relation between the two is a one to many relationship. (Partial participation) One Staff can organise 0 or more Meetings. (total participation) A meeting must be organised by one staff member.
Between the **'Meetings' and 'Room' entity** there is a one to one relationship. (Total participation) A meeting can only happen in one room. (Partial participation) There may be a room that may or may not have a meeting.
Between the **'Meetings' and 'Student' entity** we can see the relationship '**Attends'**. The relation between the two is a Many to many relationship. (Partial participation) One or more Students may attend 0 or more Meetings. (Partial participation) A meeting may be attended by 0 or more students.
Between the **'Research Staff' and 'Topic' entity** we can see the relationship **'researched'**. The relation between the two is a many to many relationships. (total participation) One research staff must research 1 or more topics/ (total participation) one topic must belong to one or more research staff. In addition to this, the relation 'researched' is an identifying relationship meaning that the 'Topic' entity cannot be uniquely identified without the 'Research Staff' entity.
Between the **'Academic' and 'Module' entity** we can see the relationship **'teaches'**. The relation between the two is a many to many relationships. (total participation) One Academic must teach one or more modules/ (total participation) one module must be taught be one or more academic.
Between the **'Academic' and 'Project'** entity we can see the relationship **'supervises'**. The relation between the two is a one to many relationship. (total participation) One academic must supervise 1 or more projects/ (total participation) One Project must be supervised by an academic.
Between the **'Module' and 'Course' entity** we can see the relationship **'Belongs to'**. The relation between the two is a many to many relationships. (total participation) one module must belong to one or more course/ (total participation) one course must have 1 or more modules.
Between the **'Student' and 'Project'** entity we can see the relationship **'Collaborated'**. The relation between the two is a many to many relationships. (Partial participation) One Student may collaborate in 0 or more projects/ (total participation) one project must belong to one or more students.
Between the **'Student' and 'Course'** entity we can see the relationship **'Enrolled on'**. The relation between the two is a one to many relationships. (total participation) One student must be enrolled on one course/ (total participation) one course will be enrolled by one or more students.

Supertypes and subtypes
Looking at the entities 'Person', 'Staff' and 'Student' (ternary relationship). We can identify that the 'Person' entity is the super type whilst the others are subtypes. A Total participation constraint and a disjoint constraint are applied to the ternary relationship. This means that a person must be either a Staff or a Student. There shouldn't be a person that does not fall in those categories. Looking at the entities 'Staff', 'Research Staff' and 'Academic' (ternary relationship). We can again identify that the supertype is the 'Staff' Entity and the others are subtypes. A partial participation constraint and overlapping constraint is applied to the ternary relationship. This means that a staff may either be a research staff or an academic or both or neither because of the disjointness and constraints applied to the supertype- subtype relationships.

My Self-Assessment
I believe my diagram would gain 20+ marks as I have carefully followed the instructions and have fulfilled the criteria to achieve all the marks for the basic tasks. I have added one supertype and two subtypes along with their constraints (5 marks). I have added additional entity types which are weak & associative entity types (2 marks). I have added appropriate attributes that best describes each entity and would be needed that suit my research group (5 marks). I have included more than 2 one to many and many to many relationships with their constraints visually shown in my diagram (8 marks). I have created a complex EER diagram that uses a range of entity types and relationships. Therefore, the marks that should be allocated should be 20 marks or more.

3 Conceptual/Logical Relational Database Schema



Person(ID:int, FirstName:string, Surname:string, Birthday:Date)

Emails(Person_ID:int, Email:string)

Telephones(Person_ID:int, Telephone:string)

Staff(ID:int, Role:String, Office:String, Title:String, Office_Hours:String...)

Meeting(Meeting_ID:int, Staff_ID:int, Room_Number:String, Duration:String, Period:Date, Topic:String)

Room(Room_Number: String, Building:String, size:String,…)

Author(Staff:int, Publication_ID:int, Year;int,...)

Publications(Publication_ID:int, Title:String, Description:String,...)

Research_Staff(ID:int, ...)

Staff_Topic(Staff_ID:int, Topic_ID:int, ...)

Topic(Topic_ID:int, Name:string);

Academic(ID:int, ...)

Academic_Module(Academic_ID:int, Module_ID:int)

Module(Module_ID:int, Name:string, Credits:int...)

Course_Module(Course_ID:int, Module_ID:int)

Student(ID:int, Course_ID:int, Level:{PHD, PGR, UG},...)

Course(Course_ID:int, Department:{Arts and Social Sciences, Engineering and Physical Sciences, Health and Medical Sciences}, Name:string)

Collaborated(Student:int,Project:int, Year:int,...)

Project(Project_ID:int, Academic_ID:int, Description:string, Title;string,...)

Attending(Meeting_ID:int, Student_ID:int,…)

My Self-Assessment

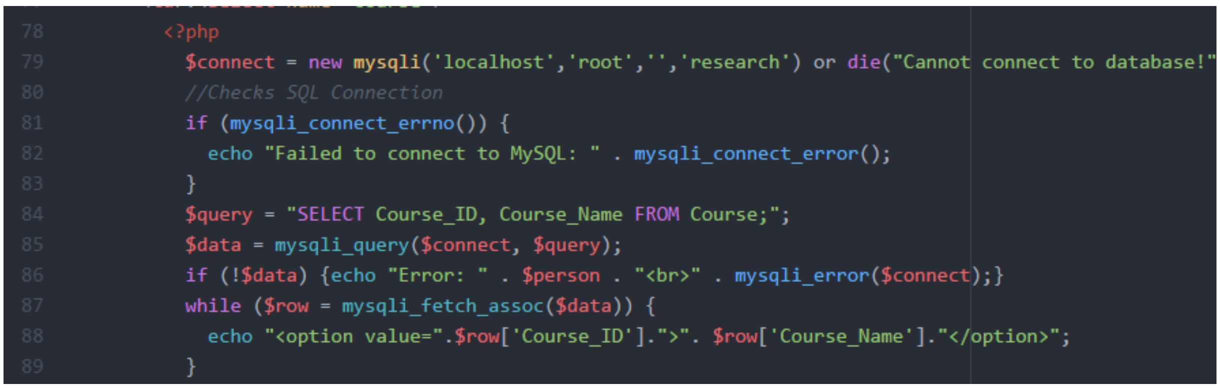
I have correctly translated all my supertypes and subtypes (3 marks), relationships (6 marks), attributes (3 marks) and additional entity types (1 mark). I have also included all the primary keys of all the relations (2 marks). Therefore, I should earn around 18 marks or more.

4 MySQL Code

Creating my data tables
I have created 14 tables in total. I decided to simplify the schema and remove the subtypes' academic' and 'research staff' that belong to the super type 'Staff'. I did this so that the queries would be easier to write and whether a staff is a researcher or academic can simply stated in an attribute that belongs to the table 'Staff'. Furthermore, it is not entirely important data to be outputted. I have translated almost all the tables this is so that I can gain additional mark (as it states in the additional mark scheme that creating "a more complicated MySQL database with more relations and/or foreign keys' would be considered for additional marks. Some data tables would not be included and will be ignored when constructing my website as it will require mostly the same SQL statement and hence will not gain any further marks.

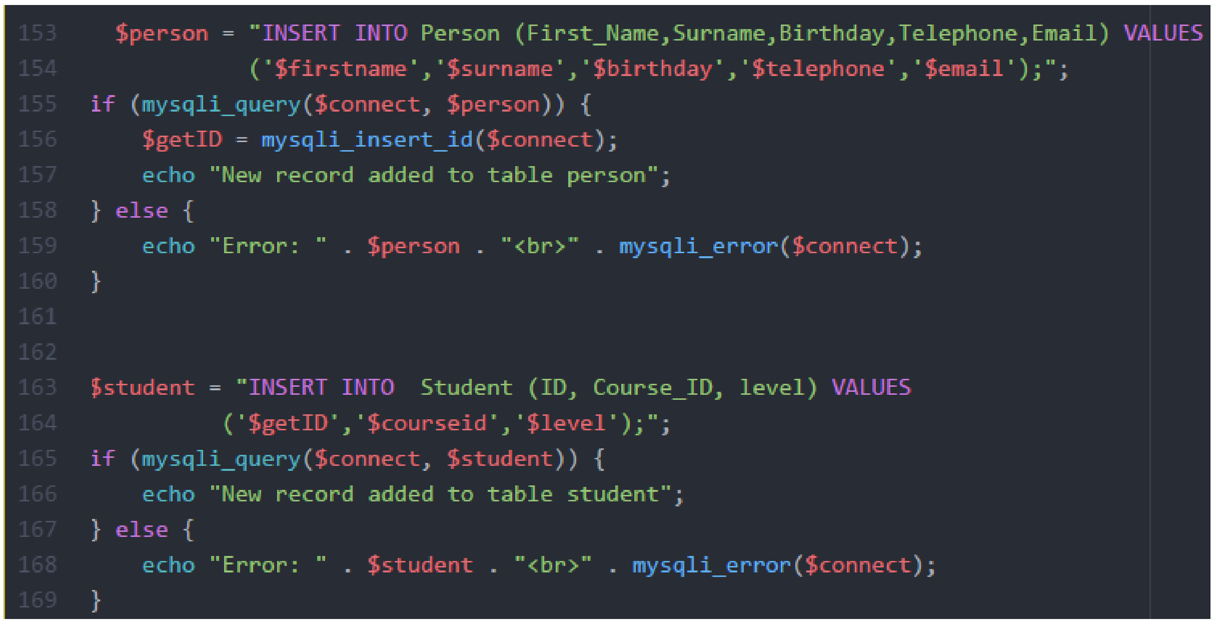
In my SQL file, I began by creating the database research. After that I decided to create my tables using my relational schema diagram as a guide. I began with the supertype table 'Person'. I ensured that each attribute is defined correctly with their correct data types. ID in 'Person' in this case is an integer, unsigned, unique, not null and auto incremented. For the other attributes I mostly used the data type VARCHAR() (for email, surname etc). For 'Birthday' I used the DATE format. I have also correctly assigned the attribute 'ID' as the primary key using the syntax PRIMARY KEY (ID). This was then repeated until all the tables were created. Some attributes had different data types, for example the field 'Faculty' in the table Course is defined as a ENUM. Meaning that its value is constricted to those that are stored inside the ENUM. With SQL I have correctly created relationships that are many to many or one to many. To get it right I used my relational schema as a guide, that way I knew where to put the foreign keys! Derived attributes such as AGE would be calculated using SQL code when the data needs to be displayed on to the website using PHP. To populate my data tables I simply used the INSERT INTO syntax to insert all my data. I had to ensure that I inserted data into the super types first before adding data into my other tables so that there are no missing links. There are not SQL syntax to simultaneously add rows into two tables at once, therefore you would need to require multiple queries to add a record to multiple tables.

SQL Code embedded into my PHP



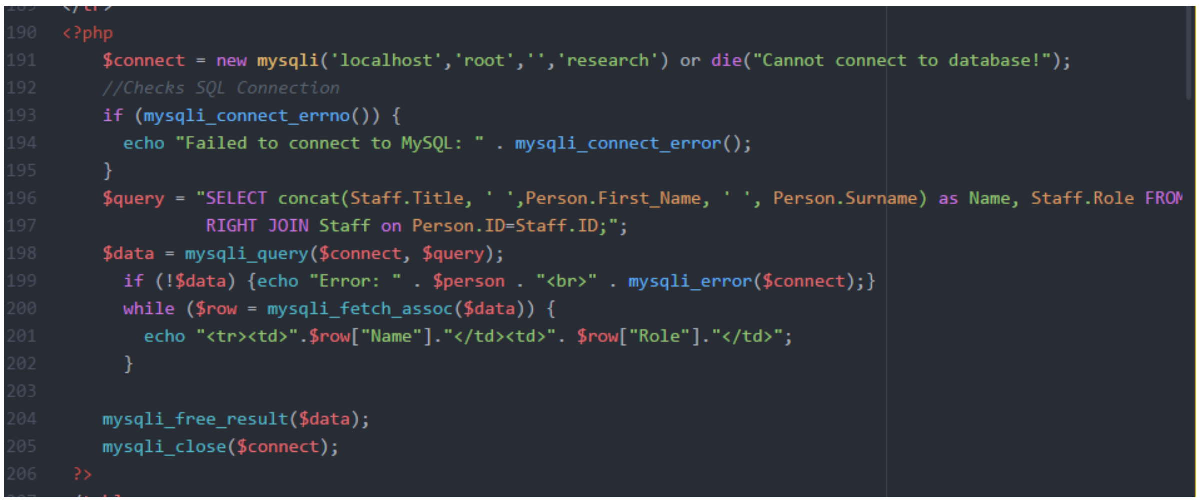
```
<?php
  $connect = new mysqli('localhost','root','','research') or die("Cannot connect to database!"
  //Checks SQL Connection
  if (mysqli_connect_errno()) {
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
  }
  $query = "SELECT Course_ID, Course_Name FROM Course;";
  $data = mysqli_query($connect, $query);
  if (!$data) {echo "Error: " . $person . "<br>" . mysqli_error($connect);}
  while ($row = mysqli_fetch_assoc($data)) {
    echo "<option value=".$row['Course_ID'].">". $row['Course_Name']."</option>";
  }
```

The following query will list all the course names onto a SELECT element on the Web page. Each item (course name) will have their value set as the course name's corresponding course id . This Select element will be used in the Add member section, it will determine which course the student studies when selected.



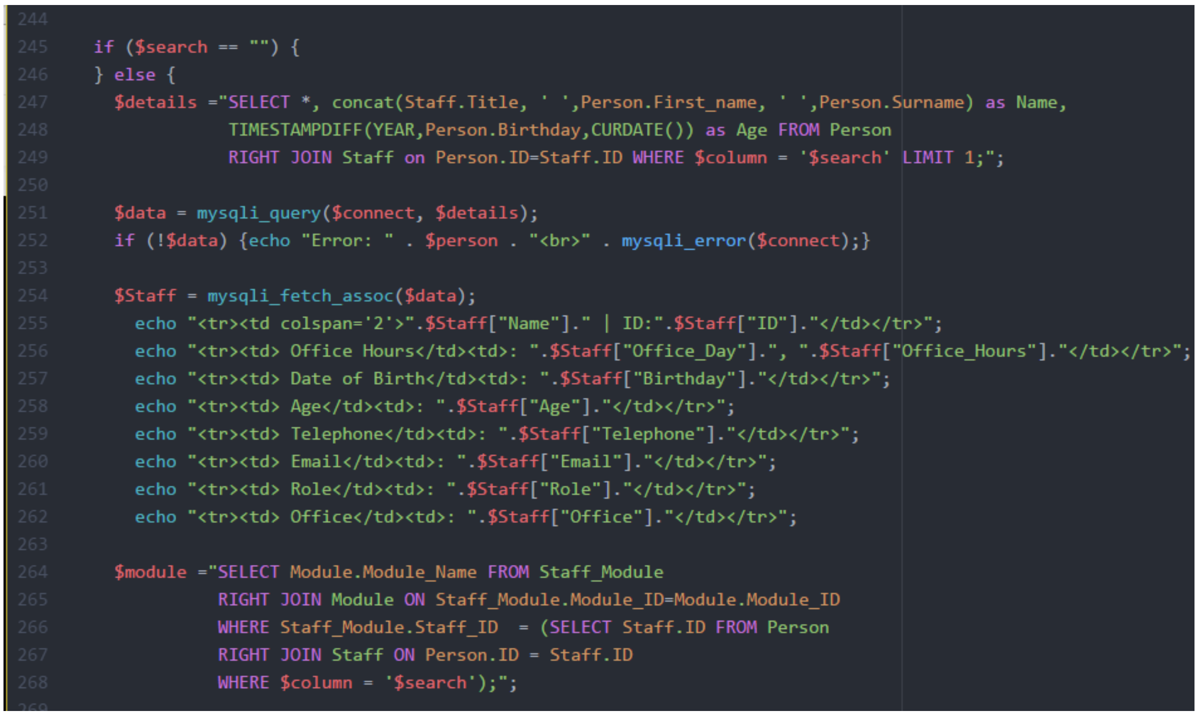
```php
  $person = "INSERT INTO Person (First_Name,Surname,Birthday,Telephone,Email) VALUES
            ('$firstname','$surname','$birthday','$telephone','$email');";
if (mysqli_query($connect, $person)) {
    $getID = mysqli_insert_id($connect);
    echo "New record added to table person";
} else {
    echo "Error: " . $person . "<br>" . mysqli_error($connect);
}


$student = "INSERT INTO  Student (ID, Course_ID, level) VALUES
            ('$getID','$courseid','$level');";
if (mysqli_query($connect, $student)) {
    echo "New record added to table student";
} else {
    echo "Error: " . $student . "<br>" . mysqli_error($connect);
}
```

The following dynamic query will insert a new person (which is also a student) into the tables Person and Students. Their values have been extracted from a xml document which have been generated from a HTML form once the submit button has been pressed. Because the ID of the person is auto generated I used the php function mysqli_insert_id($connect) to return the last used ID. This was then stored into a variable and used in the next SQL Statement to store the dynamic data into the Student table.



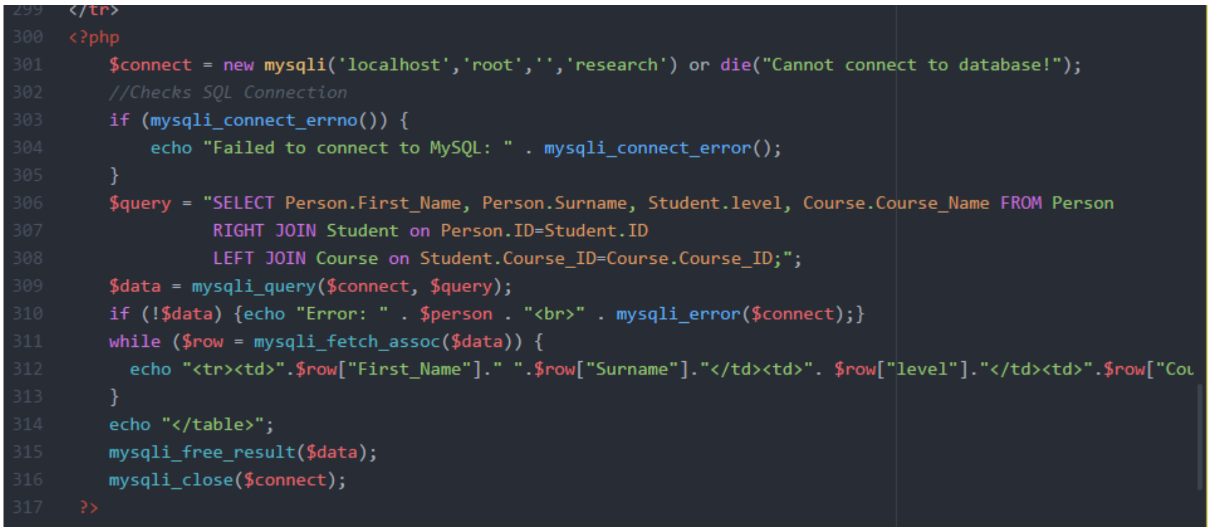
```php
<?php
    $connect = new mysqli('localhost','root','','research') or die("Cannot connect to database!");
    //Checks SQL Connection
    if (mysqli_connect_errno()) {
      echo "Failed to connect to MySQL: " . mysqli_connect_error();
    }
    $query = "SELECT concat(Staff.Title, ' ',Person.First_Name, ' ', Person.Surname) as Name, Staff.Role FROM
              RIGHT JOIN Staff on Person.ID=Staff.ID;";
    $data = mysqli_query($connect, $query);
      if (!$data) {echo "Error: " . $person . "<br>" . mysqli_error($connect);}
      while ($row = mysqli_fetch_assoc($data)) {
        echo "<tr><td>".$row["Name"]."</td><td>". $row["Role"]."</td>";
      }

    mysqli_free_result($data);
    mysqli_close($connect);
 ?>
```

The following query list all the staff members onto the webpage. I used the syntax concat to concatenate each staff's title, name and surname together and set the alias 'name' to it. I had also used the Join syntax so that I can access the title attribute from the staff table.



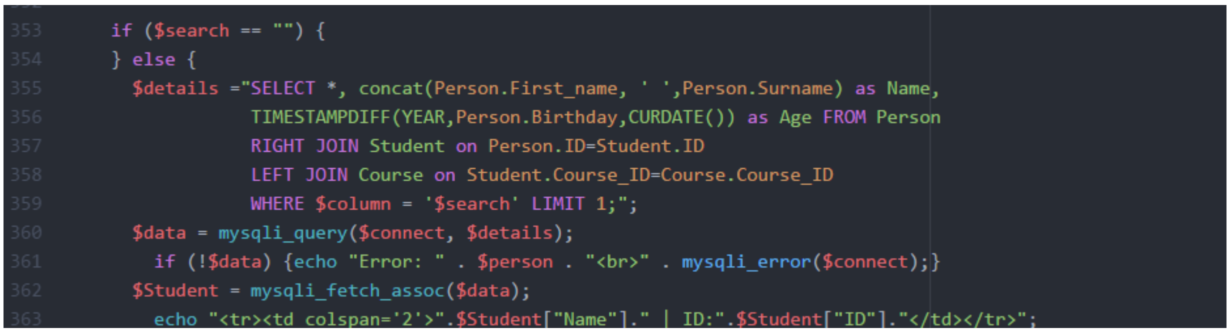
```
  if ($search == "") {
  } else {
    $details ="SELECT *, concat(Staff.Title, ' ',Person.First_name, ' ',Person.Surname) as Name,
               TIMESTAMPDIFF(YEAR,Person.Birthday,CURDATE()) as Age FROM Person
               RIGHT JOIN Staff on Person.ID=Staff.ID WHERE $column = '$search' LIMIT 1;";

    $data = mysqli_query($connect, $details);
    if (!$data) {echo "Error: " . $person . "<br>" . mysqli_error($connect);}

    $Staff = mysqli_fetch_assoc($data);
      echo "<tr><td colspan='2'>".$Staff["Name"]." | ID:".$Staff["ID"]."</td></tr>";
      echo "<tr><td> Office Hours</td><td>: ".$Staff["Office_Day"].", ".$Staff["Office_Hours"]."</td></tr>";
      echo "<tr><td> Date of Birth</td><td>: ".$Staff["Birthday"]."</td></tr>";
      echo "<tr><td> Age</td><td>: ".$Staff["Age"]."</td></tr>";
      echo "<tr><td> Telephone</td><td>: ".$Staff["Telephone"]."</td></tr>";
      echo "<tr><td> Email</td><td>: ".$Staff["Email"]."</td></tr>";
      echo "<tr><td> Role</td><td>: ".$Staff["Role"]."</td></tr>";
      echo "<tr><td> Office</td><td>: ".$Staff["Office"]."</td></tr>";

    $module ="SELECT Module.Module_Name FROM Staff_Module
              RIGHT JOIN Module ON Staff_Module.Module_ID=Module.Module_ID
              WHERE Staff_Module.Staff_ID  = (SELECT Staff.ID FROM Person
              RIGHT JOIN Staff ON Person.ID = Staff.ID
              WHERE $column = '$search');";
```

The first Query will list out the staff details from the tables Person and Staff. To do this, I used the JOIN syntax. I had also used the concat syntax again to concatenate the user's name. lastly, I used the TIMESTAMPDIFF syntax to calculate the staff's age (Age is one of the derived columns). This query is also dynamic as the user would be able to filter their results by setting which column they would like to search by and what they would like to search for. The next query will retrieve the Modules the staff teaches. I had to join multiple tables together so that the query can work dynamically and will retrieve the correct data. This query is also a subquery, this is to again allow the query to work dynamically. The subquery will return an ID depending on the values of the variable. The ID is then checked against the staff id in the table Staff_Module and then the corresponding module name will then be displayed.



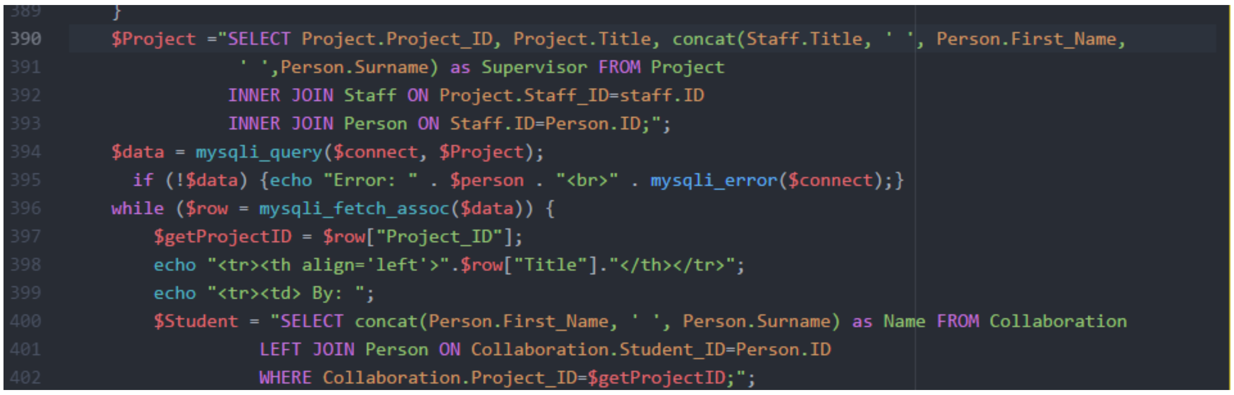
```
</tr>
<?php
    $connect = new mysqli('localhost','root','','research') or die("Cannot connect to database!");
    //Checks SQL Connection
    if (mysqli_connect_errno()) {
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
    }
    $query = "SELECT Person.First_Name, Person.Surname, Student.level, Course.Course_Name FROM Person
              RIGHT JOIN Student on Person.ID=Student.ID
              LEFT JOIN Course on Student.Course_ID=Course.Course_ID;";
    $data = mysqli_query($connect, $query);
    if (!$data) {echo "Error: " . $person . "<br>" . mysqli_error($connect);}
    while ($row = mysqli_fetch_assoc($data)) {
      echo "<tr><td>".$row["First_Name"]." ".$row["Surname"]."</td><td>". $row["level"]."</td><td>".$row["Cou
    }
    echo "</table>";
    mysqli_free_result($data);
    mysqli_close($connect);
  ?>
```

This query statement will list all the following columns in the person table that is joined with the student table as well as the course table. I have joined multiple columns together to retrieve the fields level and course name.



```php
    if ($search == "") {
    } else {
      $details ="SELECT *, concat(Person.First_name, ' ',Person.Surname) as Name,
                 TIMESTAMPDIFF(YEAR,Person.Birthday,CURDATE()) as Age FROM Person
                 RIGHT JOIN Student on Person.ID=Student.ID
                 LEFT JOIN Course on Student.Course_ID=Course.Course_ID
                 WHERE $column = '$search' LIMIT 1;";
      $data = mysqli_query($connect, $details);
        if (!$data) {echo "Error: " . $person . "<br>" . mysqli_error($connect);}
      $Student = mysqli_fetch_assoc($data);
        echo "<tr><td colspan='2'>".$Student["Name"]." | ID:".$Student["ID"]."</td></tr>";
```

The following Query will list out the Students details from the tables Person and Student. To do this, I used the JOIN syntax. I had also used the concat syntax again to concatenate the user's name. lastly, I used the TIMESTAMPDIFF syntax to calculate the Student's age (Age is one of the derived columns). This query is also dynamic as the user would be able to filter their results by setting which column they would like to search by and what they would like to search for. I also used the syntax LIMIT 1 so that only 1 data row can be searched. This is because the data would be displayed on to a list and can only display one row at a time.



```php
    }
    $Project ="SELECT Project.Project_ID, Project.Title, concat(Staff.Title, ' ', Person.First_Name,
                ' ',Person.Surname) as Supervisor FROM Project
               INNER JOIN Staff ON Project.Staff_ID=staff.ID
               INNER JOIN Person ON Staff.ID=Person.ID;";
    $data = mysqli_query($connect, $Project);
      if (!$data) {echo "Error: " . $person . "<br>" . mysqli_error($connect);}
    while ($row = mysqli_fetch_assoc($data)) {
        $getProjectID = $row["Project_ID"];
        echo "<tr><th align='left'>".$row["Title"]."</th></tr>";
        echo "<tr><td> By: ";
        $Student = "SELECT concat(Person.First_Name, ' ', Person.Surname) as Name FROM Collaboration
                  LEFT JOIN Person ON Collaboration.Student_ID=Person.ID
                  WHERE Collaboration.Project_ID=$getProjectID;";
```

This Query will list all the project's titles and their corresponding assigned supervisors out onto the web page. The Project's ID is stored as a PHP variable and is used in the next SQL statement to display all the students that have collaborated in the corresponding project.

<u>Self Assessment</u>
I have added all the relations with their corresponding primary keys and foreign keys. In 3 of my relations which are Staff_module, Attending, Author (and more relations such as Collaboration, Course_Module) I have included 2 foreign keys which allows a many to many relationships to be formed. I have populated my database using the SQL statement INSER INTO. In my php scrips I have used a SELECT statement, a sub query, a JOIN query multiple times. I should have achieved the criteria to gain the basic marks.

5 Website Working with MySQL Database

LIST OF FILES:

- COM1025_CW2_6474079.sql (this creates the database)
- Index.php (this is the only php file required for the website)

NOTE: When adding a new member through the web page an XML document named 'Students.XML' will be created.

The 'Index.php' will contain all the PHP Scripts to access the MySQL database.

- In order to test my web page and MySQL database, the SQL file should be placed into the xamp directory '…\xampp\mysql\bin'.
- The Index.php file should be placed into the xamp directory '…\xampp\htdocs'.
- After that you may begin and open the command window, typing mysql -u root to proceed into the database.
- After that type in Source COM1025_CW2_6474079.sql to load and create the database.
- You may then begin testing the database and proceed on to the website 'localhost/index.php'.

Self Assessment
I have successfully created PHP code that:

- Can connect to the MySQL database
- Can Query the MySQL database
- Can read data out of the result of a MySQL query
- Can free results properly from a MySQL query
- Can release a connection to the MySQL database
- Has more than one dynamically constructed MySQL statements that have a mixture of sub-query and uses the JOIN statement
- Has error handlining scrips around accessing the MySQL database.

I believe that my PHP code covers all the points to gain all basic marks in the criteria.

6 Advanced Tasks

MySQL code for creating, manipulating and reading from the MySQL database

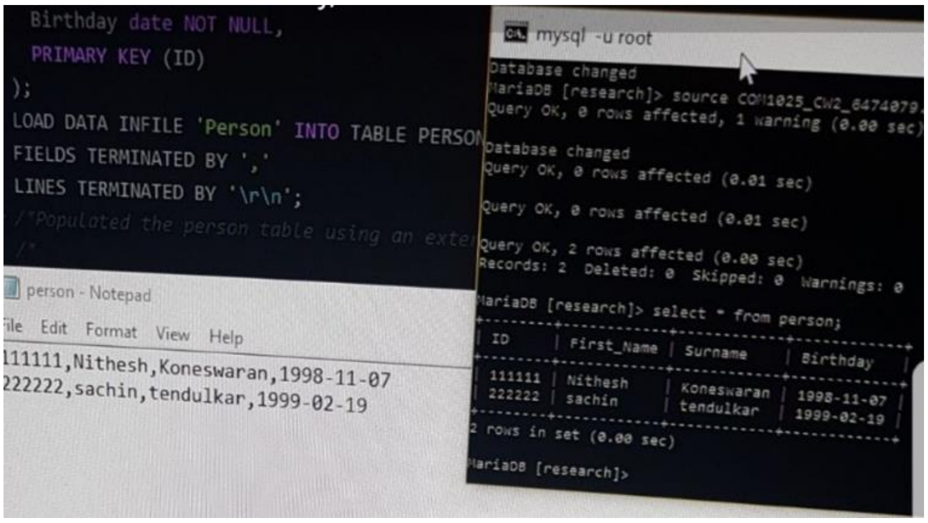
I have translated most of my relational schema into the database. The only thing that I did not translate was the research staff and academic which are subtypes of the super type staff. I did not translate them because they were unnecessary and would have required longer SQL code when querying the data. I've used a range of complex queries that are dynamic and are a mixture of sub-queries and uses multiple JOIN statement to get the results that the user would want (see part c to see the SQL code used). An example of advanced SQL code that I learned includes TIMESTAMPDIFF which calculates the derived attribute 'age' of a person.

When populating the data tables, a more powerful efficient way is to use the LOAD DATA INTO syntax. I began populating my data using external files. It's content was written like so…

2352,Kathrin,Ladosh,1988-05-08
5747,Simon,Evans,1989-05-24
1325,Peter,Klaver,1979-08-15
1764,Annette,Sterr,1983-01-19

The commas separate each field and at the end of each row would be signified with a value that does not have a comma at the end. This was done through the SQL syntax:

LOAD DATA INFILE 'Person' INTO TABLE PERSON
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\r\n'
'Person' is the file name and is in the database folder found on XAMP. The issue that had occurred was that the file must be in the database folder. This was an issue because after running the Source COM1027_CW2_6474079.sql errors occurred. These errors were generated because after creating the database the 'LOAD DATA INFILE' syntax will generate an error saying that the file does not exist! This is because after creating the database the external files holding the data of the data tables will then need to be transferred into the database folder (which was just created). Another method to work around this is to use the statement 'LOAD DATA LOCAL INFILE 'directory' ….' This statement will work however the directory of the external file must be stated in the directory bit. However, it is noted in the guidelines that "if the marker has to make (even minor) changes to your .sql file so that it can run to create the database needed for testing,"" marks will be deducted". Because of this I decided to avoid using the LOAD DATA INFILE in case marks were deducted, I had also avoided the use because the SQL command is not allowed with my version of MySQL. However, I included this in the report in case the marker may decide to consider this as additional marks I only have the image below to prove that I had used the 'LOAD DATA INFILE' syntax, the image below shows the SQL statement working on my command window. (Additional marks self assessment), in order for the command to work I had to use this statement 'mysql -u root –p –local-infile research'. In the end I used the simple INSERT INTO syntax to populate my database.



```
  Birthday date NOT NULL,
  PRIMARY KEY (ID)
);
LOAD DATA INFILE 'Person' INTO TABLE PERSON
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\r\n';
/*Populated the person table using an exte
```

person - Notepad

```
111111,Nithesh,Koneswaran,1998-11-07
222222,sachin,tendulkar,1999-02-19
```

mysql -u root

```
Database changed
MariaDB [research]> source COM1025_CW2_6474079.
Query OK, 0 rows affected, 1 warning (0.00 sec)

Database changed
Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 2 rows affected (0.00 sec)
Records: 2  Deleted: 0  Skipped: 0  Warnings: 0

MariaDB [research]> select * from person;
+--------+------------+------------+------------+
| ID     | First_Name | Surname    | Birthday   |
+--------+------------+------------+------------+
| 111111 | Nithesh    | Koneswaran | 1998-11-07 |
| 222222 | sachin     | tendulkar  | 1999-02-19 |
+--------+------------+------------+------------+
2 rows in set (0.00 sec)

MariaDB [research]>
```

EER Data modelling/Diagram

I have created a complex EER diagram that uses a range of entity types and relationships. Therefore, the marks that should be allocated should be 20 marks or more. The EER diagram is described in detail in the EER diagram section of the report. I have included weak entity types, associative entity types, identifying relationships, added more subtypes. I have also applied constraints between entities such as partial/total participation. I also have a range of attributes such as multi-valued attributes and derived attributes.

Conceptual/Logical relational database schema

I have taken my complex EER diagram and have completely translated it into a relational schema. Therefore, some additional marks should be considered. I believe I have correctly translated the diagram to the schema ensuring the data types and relations are valid by correctly placing the keys in the correct entities.
I believe my relational database is Third Normal Form (3NF). This is because I believe my data integrity has been achieved. My database does not hold any redundancy (there isn't any data duplication). My columns are also not dependent on the primary key. The primary key simply indexes each record and is not relied upon the other columns

PHP code bridging the MySQL database with your webpages

I have mixed XML with MySQL in my PHP code. My webpage has a form that has a series of textboxes. When the user fills the data, and presses the submit button an XML document is generated by a PHP script. The XML document temporarily contains the data that has been submitted. After this the XML file is loaded and each element from the XML is extracted and stored into variables which are then placed into the INSERT Query to add the data into the database.

I have also used many PHP functions that were not covered in the teaching material for handling MySQL databases such as the function mysqli_insert_id(), which retrieves the last ID used in a query statement. I also used the function mysqli_real_escape_string which adds special characters (such as escape character and backslash) on to the string passed from the text field form. This helps prevent SQL injection attacks., although a much safer alternative would be to use prepared statements.

When implementing the XML feature, I came across a lot of new PHP libraries and learnt a lot of new stuff. I used PHP Simple XML to load an xml file and extract the data from the file and upload it on to the database.

I also took advantage of the PHP function mysqli_num_rows(). I used it to check if more than 0 lines of code were displayed, if so then it'll go ahead and change some visual aspects of the webpage (add text).