ECM2419 Database and Theory Design

SID: 710046731

Assignment

Report

***Part 1***

In this section of the report, there is a description of how the database system for the Department of Computer Science at the University of Exeter looks like. To show this database system in a way to make it more understandable there is a design of the Entity Relationship Diagram(ERD).

The reason for creating this type of database is for the Department to keep a record of the students and staff.

*Entity Relationship Diagram*

This section of the report will cover the analysis of ER Diagram based on the Department of Computer Science which has different entities, relationships, attributes and constraints.

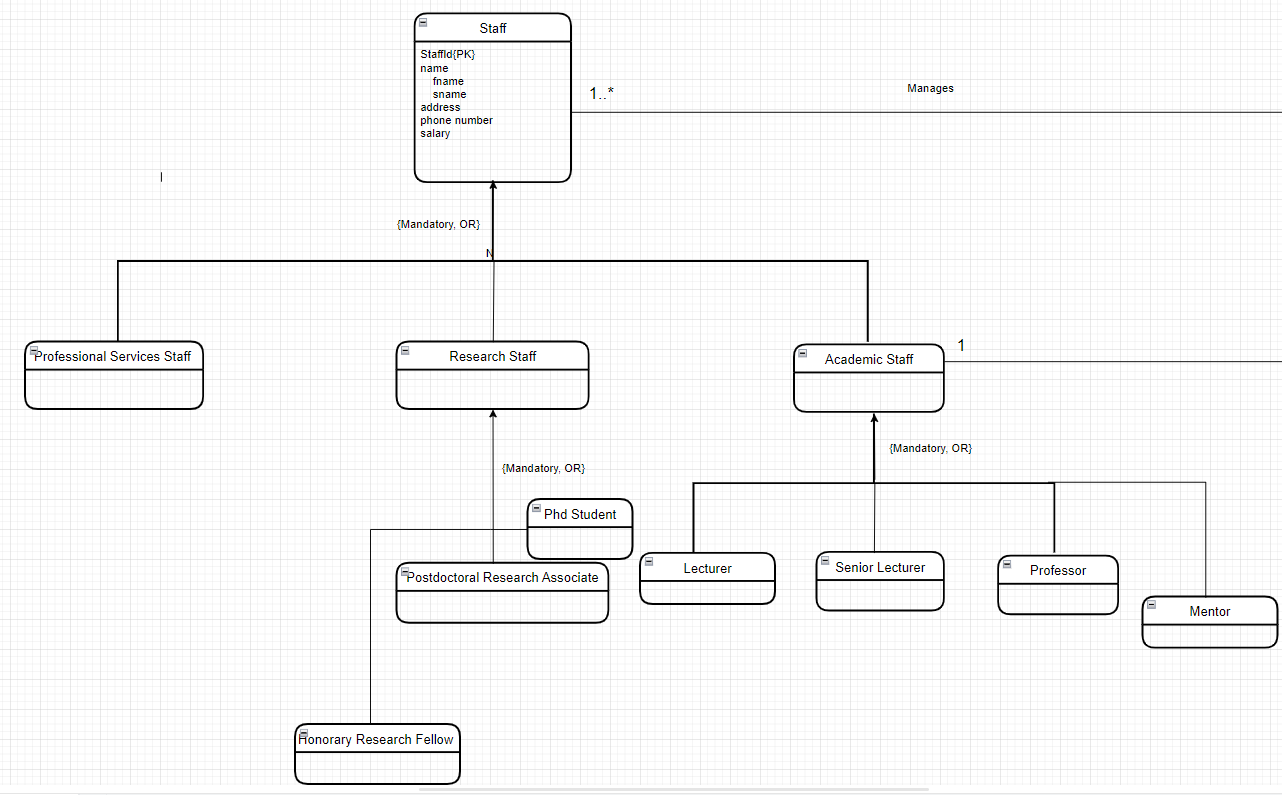
The entities of the ER Diagram in this report covers every single detail that is needed for the students and staff. There will be a section which covers each entity with their relationships why these have same or different attributes and constraints.

Staff

The staff entity covers all the members that are involved in working for the Department of Computer Science. This entity is linked with entities which has all the types of staff which are academic staff, research staff and professional services staff.

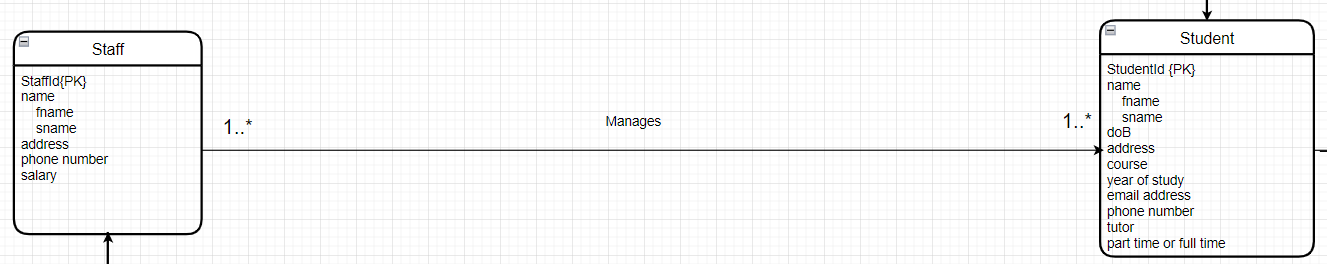
The attributes that covers the details of the staff are their ID, name, address, phone number, email address and salary. The primary key for this entity is the staff ID and having an ID as a primary key can give a huge benefit for the database system. The reason for this is that it is easier and faster to find the details of a particular staff we need if two or three people have the same first name and second name.

On Figure 1.0, it shows the constraint that has been used to link the staff entity with its entity types is ‘{Mandatory , Or}’. It is ‘Mandatory’ because a staff member has to be either one of these as there cannot be additional types of staff. It is a disjoint constraint(‘OR’) as there is no overlap between the attributes of the staff type entities e.g. a PhD student cannot become an academic staff and research staff at the same time.



**Figure 1.0**- Staff Entity

Student and Staff Relation



**Figure 1.1-** Student and Staff Relation

On Figure 1.1, the relationship between the staff and students are shown as many to many(\*:\*). This shows that different staff members can manage various number of students e.g. in a computer science tutorial session, there might be a lecturer and one or more PhD students to assist helping students in their lab work.

Student

In the student entity we have included attributes such as student ID(primary key), name, date of birth, contact details, course they study, part time or full time student, year of study and their tutor. The attribute for tutor was include in this entity as all the students will be allocated to one of the staff members teaching the course as their tutors throughout their years of study.

On Figure 1.2, it has a constraint of ‘{Mandatory, OR}’ for the link between entities of the student type and student. It is ‘Mandatory’ as there are only two types of students that study a course. This is a disjoint constraint(‘OR’) as an undergraduate or postgraduate student cannot do each other’s course at the same time.

Diagram

Description automatically generated

**Figure 1.2**- Student Entity

Student and Degree Relation

Diagram

Description automatically generated

**Figure 1.3**- Student and Degree Relation

On Figure 1.3 shown above, it shows the two entities have many to one relationship. This implies that many students can be study one degree as they cannot study two degrees at the same time.

Degree and Module

For the degree entity, the degree title has been included as the primary key since there is no type of code or ID associated with the degree.

As shown on Figure 1.4, degree entity has one to many relationship with the module as one degree can have many modules.

The entities of the modules and its types have a constraint of {Mandatory, OR} as there is no overlap between the module types due to it being a disjoint constraint. Also, there no other types of modules proving it being ‘Mandatory’.

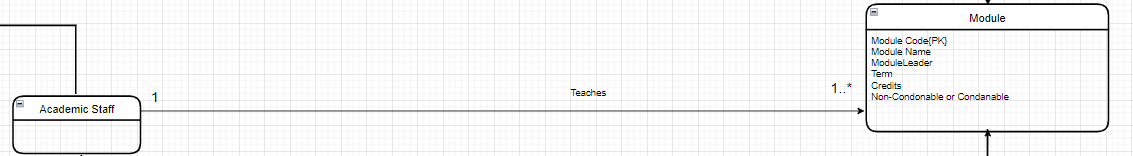
The module and its types have a many to one relationship as various range of modules has to be one of the types as a module cannot be both mandatory and optional at the same time.

Diagram, schematic

Description automatically generated

**Figure 1.4**- Degree and Module Entities and Relation

Academic Staff and Module Relation



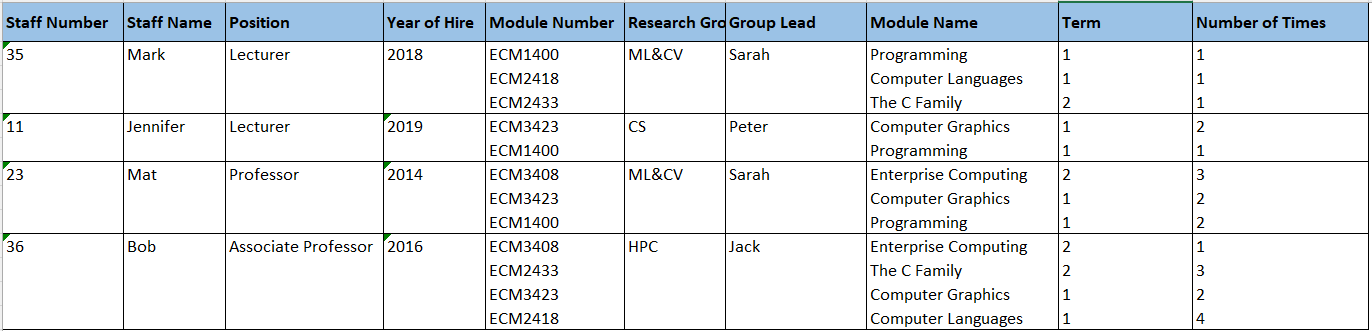
**Figure 1.5**- Academic Staff and Module Relation

As you can see on Figure 1.5, the relation between the academic staff and module is one to many(1:\*) as one module can be taught by 1 academic staff and one academic staff can teach one or more modules in the course.

***Part 2***

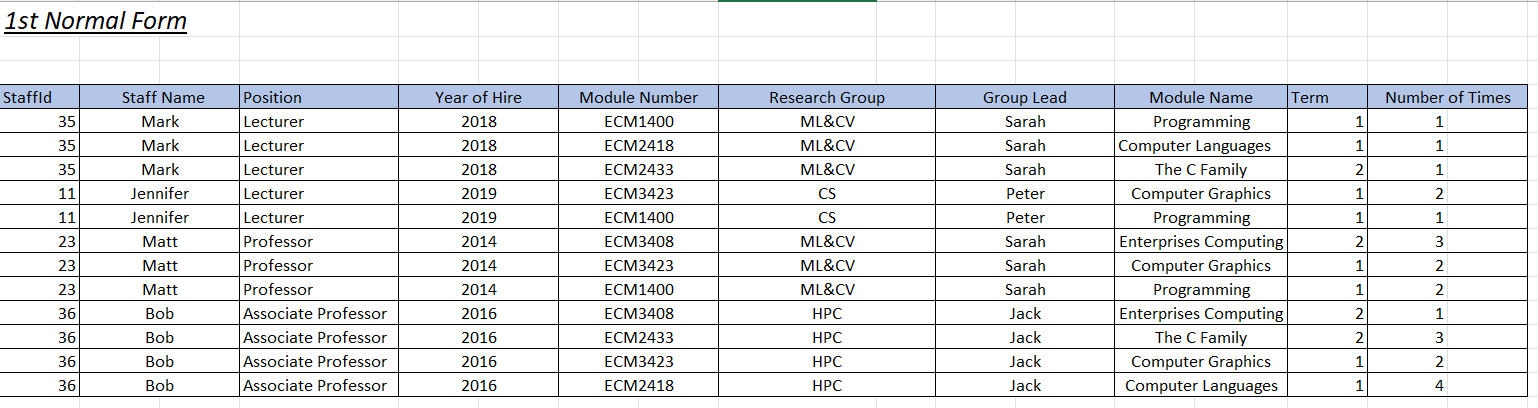
In this section, the report will describe the process of normalisation by converting a relational data into 3rd normal form. The main objective of normalisation is to remove redundancy in the relations so there might be some anomalies when you trying to find a specific data in a table so to resolve those anomalies we divide the big table into separate smaller tables. This process of normalisation can be useful for finding the specific data that you want in an easier and faster manner.

Before converting into 3rd normal form, the relational data has to be in unnormalized form then we convert it into 1st and 2nd normal form and finally it reaches into 3rd normal form.

1st Normal Form

**Figure 1.6-** Unnormalized Form Data

In order for converting an unnormalized form data into 1st form normalized data we need to remove the repeating groups in the table. As you can see on Figure 1.6, there are repeating groups from columns of ‘Module Number’ to ‘Number of Times’. In a relation we cannot have a multi valued attributes, so each tuple needs to have an atomic value.



**Figure 1.7**- 1st Normal Form Data

As you can see on Figure 1.6, there are no repeating groups in the table and each attribute in this relation contains only one single value.

As this table does not consist of multi valued attributes we can convert this into a **relation schema**:

Staff(StaffId, Staff Name, Position, Year of Hire, Module Number, Research Group, Group Lead, Module Name, Term, Number of Times)

***Staff*** is the relation name

StaffId is the primary key and Module Number is the foreign key

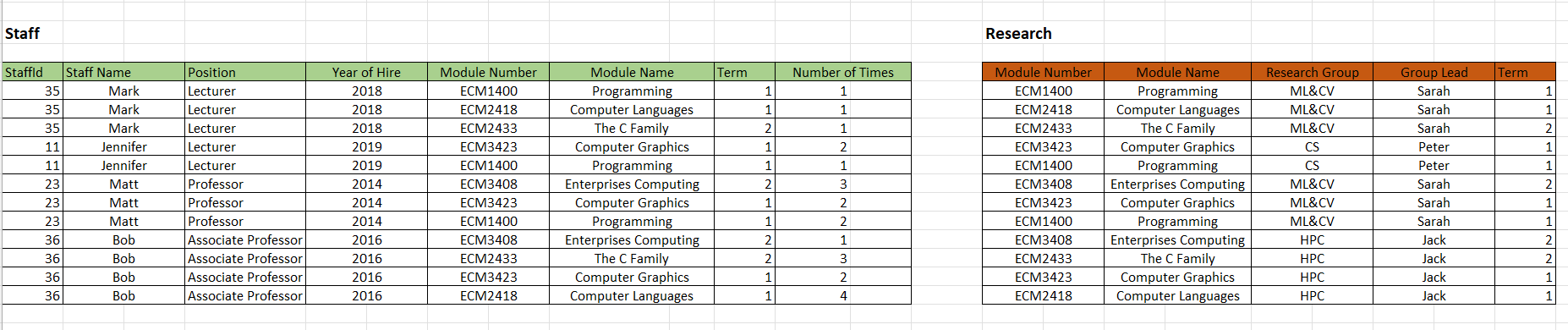
2nd Normal Form

The reason for transforming from 1st to 2nd normal form is to remove the partial dependencies.

To remove the partial dependency of the 2nd Normal Form table shown on Figure 1.6, we create an additional relation table.

The two relation tables for the 2nd Normal Form which are associated with both academic staff and research staff.

The only three attributes that are included in both tables are ‘Module Number’, ‘Module Name’ and ‘Term’. ‘Module Number’ was included as the primary key in the research relation table and the other two attributes were included to know in which modules the research staff were included in and when they were involved in those. This also goes with having the same reason of including these three attributes in the staff relation as well.



**Figure 1.8**- 2nd Normal Form Data

Relation Schema:

Staff(StaffId, Staff Name, Year of Hire, Module Number, Module Name, Term, Number of Times)

Research(Module Number, Module Name, Research Group, Group Lead, Term)

3rd Normal Form

For transforming from 2nd Normal Form to 3rd Normal Form we have to remove transitive dependencies. We can determine the module number and name by the staff ID and name then, by using the module number and name we can determine the term the module was taught in. This represents transitive dependency.

So this enables us to create another table which represents the module relations which has attributes of module number(primary key), module name and term.

The 3rd Normal Form data has staff relation table with attributes such as staff Id(primary key), staff name, position and year of hire.

A picture containing application

Description automatically generatedWe also have the research relation table which consists of the attributes research group(primary key) and group lead.

**Figure 1.9**- 3rd Normal Form Data

Relation Schema:

Staff(StaffId, Staff Name, Position, Year of Hire)

Module(Module Number, Module Name, Term)

Research(Research Group, Group Lead)