# Data Base Group Assignment 1

Jessica Harris Tiffany Woodley

April 2019

# 1 Introduction

Following a faculty guidebook can be quite intimidating and requires a lot of faculty-specific knowledge. The purpose of this assignment is to create a data base which will aid an application to take a student through this process during registration.

# 2 Assumptions

The assumptions we made when designing the data base were:

- The only people stored in the database are students, lecturers or lecturers who are still studying.
- Co-requisites and pre-requisites are considered to be the same thing.
- Lectures, practicals and tutorials are compulsory for all subjects
- Lectures and practicals can have different lengths.
- The timetable follows the form shown in Figure 1.
- Planned courses depends on a given student registering in a particular year - this is why we decided to make it a weak entity. When that student leaves, this entity is deleted. However, their academic record remains in the database so that we can access the results obtained for courses actually taken (not just planned). This allows for retrieval of transcripts and for analysis of previous students' performances.
- Each course is run by one department, even if the course is multi-disciplinary.
- Multivalued attributes in the ER diagram will be split into multiple rows (one value per row), in order to maintain atomicity and hence keep the data base in first normal form.

| | Mon | Tue | Wed | Thurs | Fri |
|---|---|---|---|---|---|
| 8:00 | Period 1 | Period 11 | Period 21 | Period 31 | Period 41 |
| 9:00 | Period 2 | Period 12 | Period 22 | Period 32 | Period 42 |
| 10:00 | Period 3 | Period 13 | Period 23 | Period 33 | Period 43 |
| 11:00 | Period 4 | Period 14 | Period 24 | Period 34 | Period 44 |
| 12:00 | Period 5 | Period 15 | Period 25 | Period 35 | Period 45 |
| 13:00 | Period 6 | Period 16 | Period 26 | Period 36 | Period 46 |
| 14:00 | Period 7 | Period 17 | Period 27 | Period 37 | Period 47 |
| 15:00 | Period 8 | Period 18 | Period 28 | Period 38 | Period 48 |
| 16:00 | Period 9 | Period 19 | Period 29 | Period 39 | Period 49 |
| 17:00 | Period 10 | Period 20 | Period 30 | Period 40 | Period 50 |

Figure 1: Outline of Period allocation throughout the week.

# 3 Entity-Relationship Model

## 3.1 Overview of diagram

The ER diagram for the data base designed to support the Science Faculty undergraduate enrollment app is shown in Figure 2.

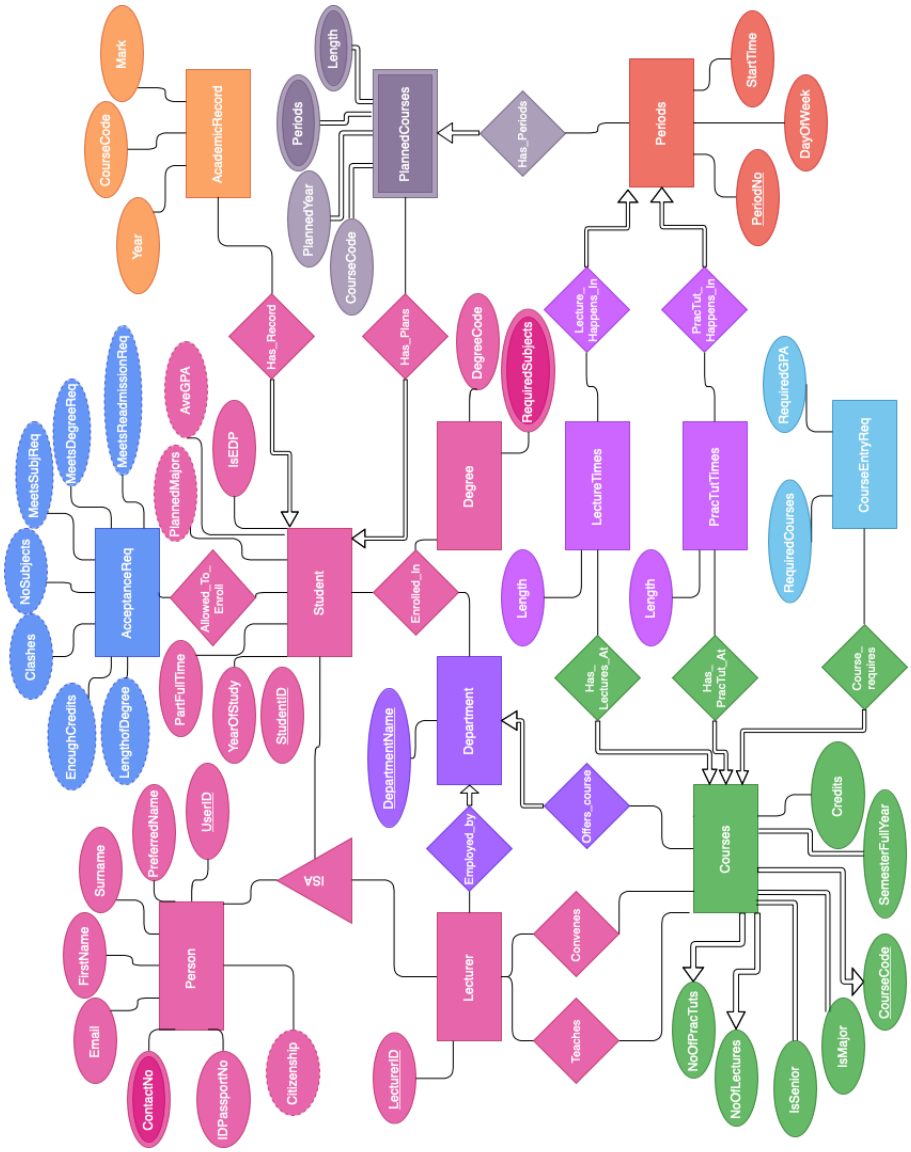



Figure 2: ER diagram for the Science Faculty undergraduate enrollment app

### 3.2 Explanation of calculated attributes

The list below explains the calculated attributes, including those based on academic rules given in the UCT Science Faculty Handbook. For each rule, an explanation is given as to how the data base was set up to accommodate the rule.

- **LengthOfDegree**
  - This column checks that the BSc degree extends over no less than three academic years of study.
  - The database stores the year of study for each planned course, this allows a formula to be applied which checks that there is atleast courses registered for third year
- **EnoughCredits**
  - The course code links to the number of credits each course carries (via the Courses table). From this it can be worked out if the student has registered for enough credits.
- **Clashes**
  - This checks if there will be clashes within the timetable based on the chosen courses. This determines whether a given combination of courses is possible within an academic year.
  - A period ID (PeriodNo), defined in Figure 1, was created to easily identify clashes. Every planned course will have a row for each lecture period. These can then be checked to see if they are unique. If not, then there will be a subject clash.
- **NoSubjects**
  - This checks a student has not registered for more courses than are allowed.
  - The database has a planned courses entity which holds all the courses the student will take with course code. The course code links to whether it is a semester/full year course. From these it can be calculated if the student is taking too many subjects
  - The course code also links to whether it is a senior course and which department it falls under. From these it can be checked if a student has the right amount of senior courses and that their are enough from each department
- **MeetsSubjReq**
  - This checks that the courses a student plans to take in earlier years of their degree meet the requirements for entry into their specified major subjects. It does this by considering the course entry requirements (CourseEntryReq) for each of the planned majors (PlannedMajors).

- **MeetsDegreeReq**
  - RequiredSubjects is a multivalued attibute in the Degree table that indicates which degrees (given by DegreeCode) require which courses as compulsory subjects. This formula checks the PlannedCourses against the RequiredSubjects to see that the student has selected the compulsory courses for their chosen degree path.
- **MeetsReadmissionReq**
  - This checks if the student is enrolled in enough courses to be able to meet readmission requirements by the end of the years of courses - this will be checked using different formulas for standard and extended development programme (EDP) students.
  - The database has a PlannedCourses entity which holds all the courses the student intends to take, together with its course code. The course code links to the Courses table, where it can be determined whether it is a semester/full year course. Using this information, it can be calculated if the student is going to make the subject requirements by the end of their consecutive years of study. The StudentID also has a linked attribute which indicates whether the student is doing the EDP or not (IsEDP), so this can be worked out on this basis.
- **PlannedMajors**
  - The data base stores whether each course is a major subject using IsMajor, which allows a formula to be applied to check which of the student's planned courses are third-year-level, major subjects that will be taken when the student is registered for third year (or later).
- **AveGPA**
  - A student's average GPA is calculated based on previous years' academic performance. While this is less of a concern for first year students, it becomes more relevant for second- and third-year students.
- **Citizenship**
  - A person's citizenship is worked out from their ID number.

# 4 Relational data base design

A relational database design for the data shown in the ER Diagram (Figure 2) is given below. The name of each relation is followed by the names of its attributes in brackets; and the attribute(s) in each primary key are underlined.

**Entity tables:**

- Person(UserID, IDPassportNo, FirstName, Surname, PreferredName, Citizenship, ContactNo, Email)
- Lecturer(LecturerID, UserID, DepartmentName)
- Student(StudentID, UserID, YearOfStudy, PlannedMajors, AveGPA , IsEDP, DegreeCode)
- PlannedCourses(StudentID,CourseCode, PlannedYear, Periods, Length)
- AcademicRecord(StudentNo, CourseCode, Year, Mark)
- Department(DepartmentName, UserID)
- Degree(DegreeCode, RequiredSubjects)
- Courses(CourseCode, IsSenior, IsMajor, SemesterFullYear, NoOfLectures, NoOfPracTuts, LecturerID, DepartmentName, )
- CourseEntryReq(CourseCode, RequiredCourses, RequiredGPA)
- AcceptanceReq(StudentID, EnoughCredits, Clashes, NoSubjects, MeetsSubjReq, MeetsDegreeReq, MeetsReadmissionReq, LengthOfDegree )
- LectureTimes(CourseCode, Length, PeriodNo)
- PracTutTimes(CourseCode, Length, PeriodNo)
- Periods(PeriodNo, DayOfWeek, StartTime)

**Relationship tables:**

- Enrolled_In(StudentNo, DegreeCode)

# 5 Questions about the database

## 5.1 Limitations in Capturing Needs/Knowledge

Prerequisites for a course are given by RequiredCourses and RequiredGPA. A current limitation of the database design is that it is not clear whether only *one* of the prerequisite courses is required, or if a *combination* of the prerequisite courses is needed. A potential workaround for this was to add an attribute to the relationship, as shown in Figure 3, that specifies whether there is an OR or AND relationship between the prerequisite courses of a chosen Course. The idea is to convey, given an array of prerequisite courses, whether *all* the listed courses are required, or any one of them. Unfortunately with this workaround it is still not possible to portray if there are multiple OR relationships for the course.

Another limitation of the current design is the assumption that tuts and lectures are compulsory. This could also potentially be fixed using an AND/OR attribute to the relationship between Courses and (Lecture and Tutorial Times), as described above.

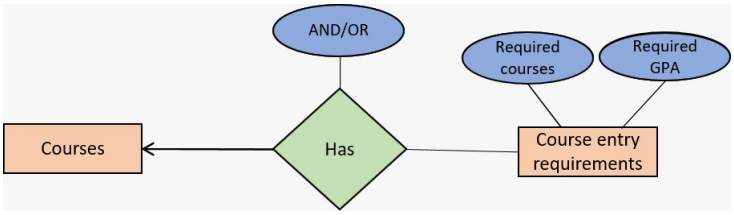



Figure 3: Suggestion for defining an AND/OR relationship in an array of pre-requisite courses.

## 5.2 Alteration from 2nd normal form

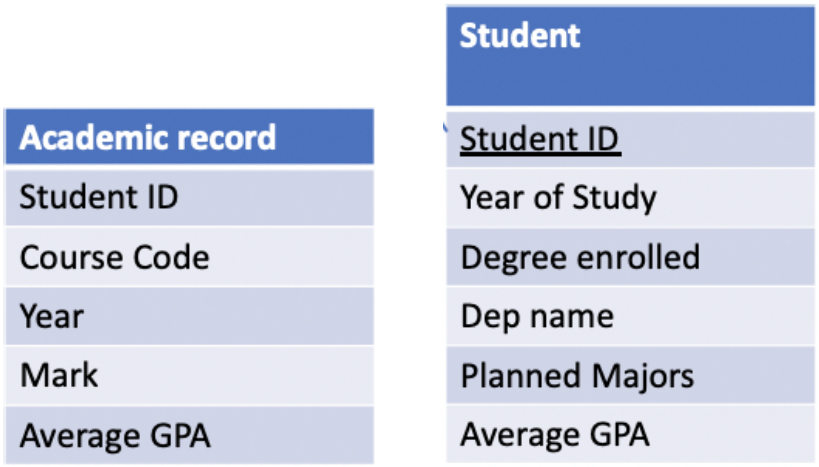
The requirements for a database to be in second normal form are:

- Be in First Normal Form
- There should be no partial dependencies

Alterations which would take the database out of first normal form are as follow:

- Having multivalued attributes within a table - should rather be stored on multiple lines
- Having a column contain different domains
- Having a column name that is not unique
- Having the database rely on the order entries are put into it

The specific alteration we consider is adding in a partial dependency (Figure 4). In the AcademicRecord table, the candidate key is made up of both *Student ID* and *Course Code*. If we added in a column for AverageGPA, this would only be dependent on Student ID and result in partial dependency.




(a) Alteration (b) Fix

Figure 4: Alteration to take data base out of 2NF.

This is why in our design *AverageGPA* is in the **Student** table so that it is directly related to the primary key.

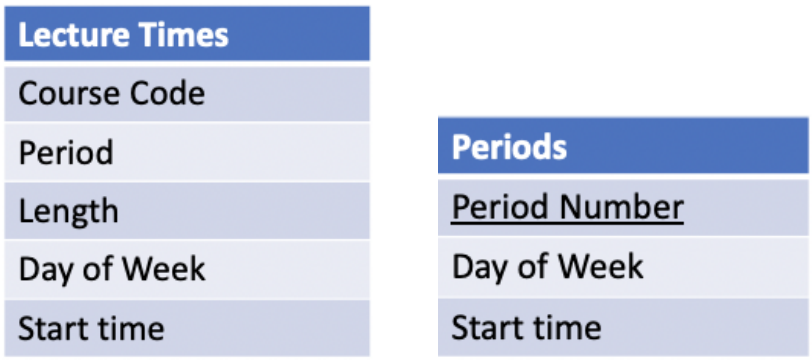
### 5.3 Alteration into 3rd normal form

The requirements for a database to be in third normal form are:

- Must already be in second normal form
- Must not include transitive dependencies

To take it out of third normal form any changes to the requirements listed in the previous question can be done.

The specific alteration we looked into was removing it from 3NF by adding in a transitive dependency. This could be achieved by adding in *Day of the week* and *Start time* into the **Lecture Times** table (Figure 5). This leads to a non-prime attribute being dependent on a non-prime attribute and in turn a transitive dependency.




(a) Alteration (b) Fix

Figure 5: Alteration to take data base out of 3NF.

This is remedied by moving the information *Day of Week* and *Start time* into a separate table with its primary key being *Period Number*. The primary key *Period Number* is then used to link the *Day of Week* and *Start time* back to the **Lecture Times** table

### 5.4 Aspects to monitor

We've set up the database in a way that can detect clashes between courses. This is done using attributes period and length. This may cause issues when checking clashes as if a lecture starts in a different period but lies within the same time length of another lecture the system could miss it if not implemented properly.

If this does turn out to be an issue an adjustment we could make would be to change it to a start and end time rather than a period number.

### 5.5 Suggested View

Our suggested view (Figure 6) is one intended for each student, once they have chosen their planned courses. It gives them a list of their chosen courses with information about each course in a descending order of average marks, based on historical data from the academic records. This will give a student an indication of the difficulty of their chosen courses. From this view, a user could do a query to find the average of all the chosen courses to see how difficult they may find their chosen course load.

It is intended that access to historic information will help the student make an informed decision about their chosen subject combinations.

| Course | Avg Mark | Min Mark | Max Mark | Median Mark | No. of fails | No. of students enrolled |
|---|---|---|---|---|---|---|
| MATHS201 | 73 | 40 | 95 | 65 | 43 | 2056 |
| STATS102 | 62 | 33 | 84 | 60 | 56 | 1031 |
| PHYSICS311 | 53 | 36 | 79 | 55 | 68 | 877 |

Figure 6: Suggested view for a student, showing historical information about the marks achieved for each course they intend to take.