Members: Manish, Nahid, Saleem, Tarik, Javed

DATABASE

Coursework 1: Design

Explanation of Diagram

The first Diagram is Conceptual; the purpose of it is to show the objects that exist in a system and the relationships between them. In the diagram, it's split up into different sections from students to exams and coursework.

Students –

The students section must have their first and last names in the database as well as their Student ID, Date of Birth and the year that they have enrolled in. The reason to have year enrolled is that it can know the subjects that you are taking that year as you don't want errors as some students might get enrolled in the wrong year.

Student to Subjects – Once the student is enrolled to the subjects; its split throughout the academic year and each semester there will be different subjects taking place. The subject needs to have a name and subject ID.

Subjects to Assessments –

Assessments are a weak entity because if an exam or coursework did not exist, then the entity would break. Even though it mentions that the subject contains exams, we decided to go for under assessment as coursework and exam are part of being assessed. So, many students will take on many subjects.

Coursework –

Coursework is a major factor in this as it can have a different set of marks for each coursework. As a result, the attributes we have under coursework are Coursework ID, Max Mark and we have a multi-value attribute called Coursework Number. This is because it mentions that there can be more than one coursework and that the marks would be different from another coursework. So, under the attribute coursework number, we included Gain marks as students can get the most possible parks; Mark out of as the max a student can get; and Module Percentage as how much does each coursework weigh and how much it counts to the final grade.

Exam to Attempt –

Exam is another key factor as a student can pass an exam or fail it and do a re-sit of the exam. Under the exam I have Max mark and Exam ID as attributes. So what we have done is leave attempt as an attribute than being an entity because a student can pass an exam but they can be able to sit it by choice however, if you fail you would have to take a re-sit. Instead of doing a pass/fail attribute, we included a result and then re-sit as an attribute. There was a lot of consideration on this section but that's what we decided to go for and the data would then come back to the database.

Student to Coursework –

Students must complete coursework and their marks will be recorded onto the database.

Students to Exam – Students must take the exam and it will be recorded from the result if you passed or failed and depending on the scenario you would have to take a resist of the exam.

Coursework & Exam to Lecturer –

The lecture would set and create the questions for the coursework and for the exam.

Questions to Staff Members & Lecturers -

When students have completed their exams, the staff members or lectures mark their work. Diff staff members can mark different questions

Coursework Marked by Staff Member –

Lecturers set coursework that students complete and submit their coursework and are then marked by staff members

Staff Members to Lecturer –

A lecturer is also a staff member and it means they do not have to mark the exams or coursework as it mentions that the author does not have to mark it,

Relationships between each entity

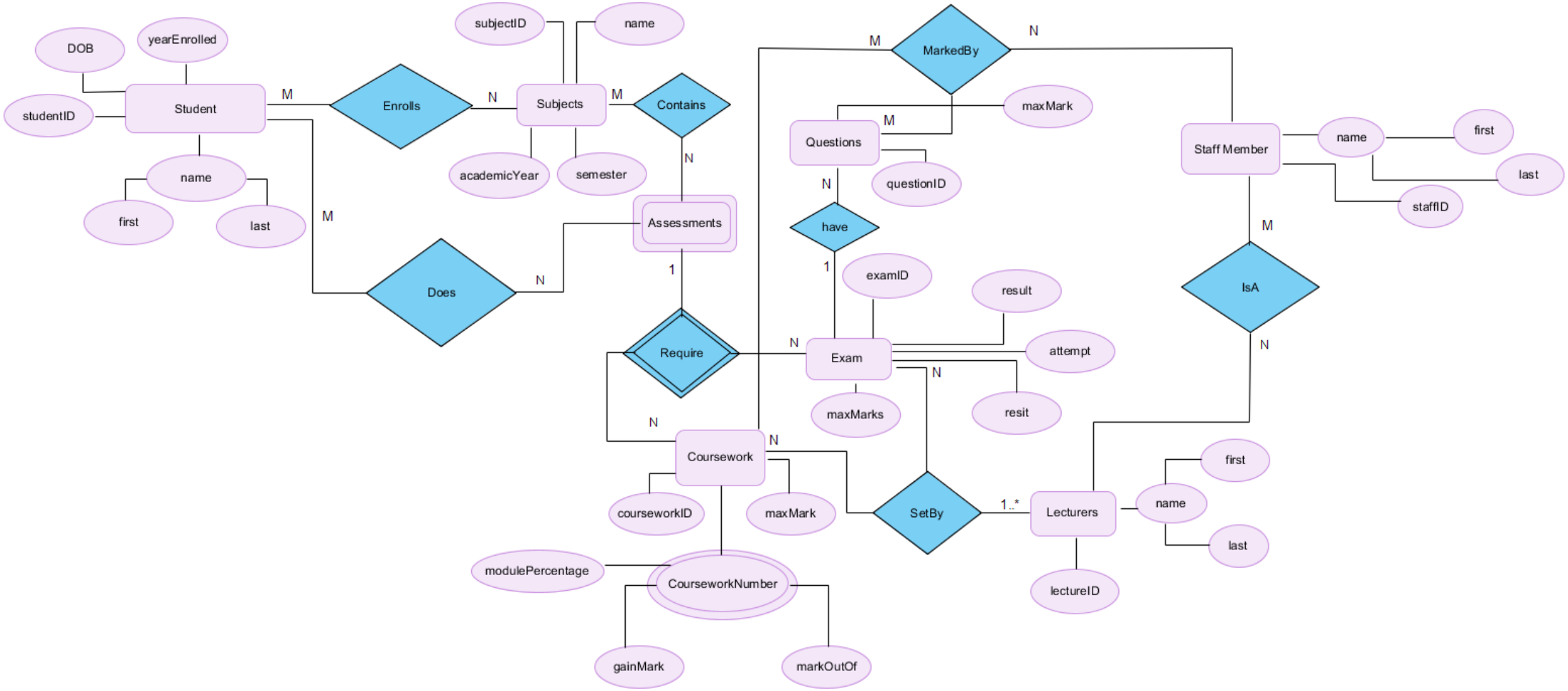
- Many Students Enrol to Many Subjects
- Many Students Do Many Assessments
- Many Subjects Contain Many Assessments
- One Assessment Requires Many Exams
- One Assessment Requires Many Coursework
- Several Coursework is set by One to Many Lecturers
- Several Exams are set by One or Many Lecturers
- Several Coursework is Marked By many Staff Members
- One Exam has Many Questions
- Many Questions are Marked By Many Staff Members
- Staff Members have many Lecturers

Entity Relationship Diagram

The primary keys with the key symbol are foreign keys (green ones) in other tables which link all of them together. The different Types are set for each entity column. We translated the conceptual diagram to a physical data model and the relationships between each entity are shown by one to one; one to many and many to many

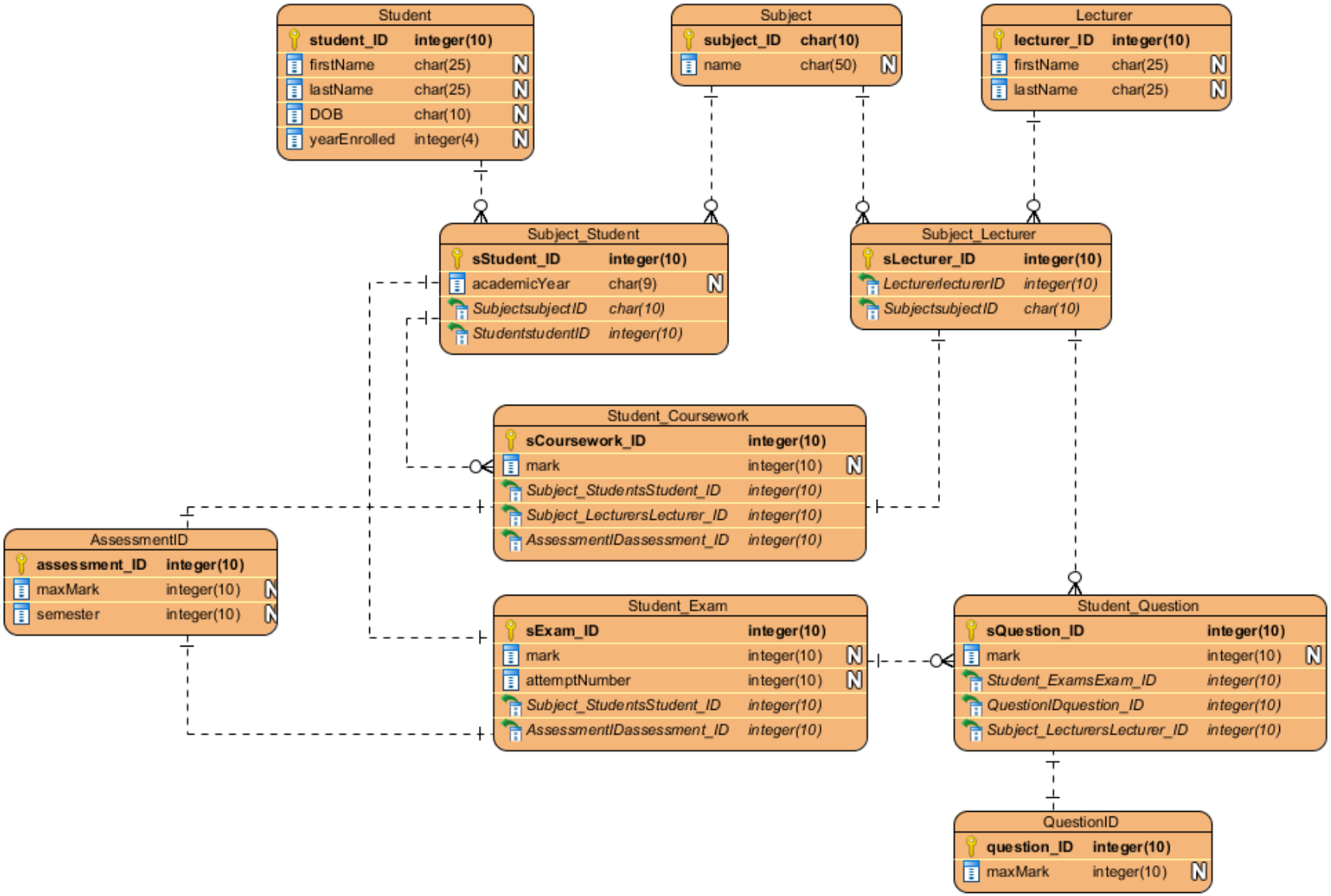
Members: Manish, Nahid, Saleem, Tarik, Javed

CONCEPTUAL DIAGRAM



Members: Manish, Nahid, Saleem, Tarik, Javed

PHYSICAL DATA MODEL



Members: Manish, Nahid, Saleem, Tarik, Javed

Relational Schema

Student

student_ID	firstName	lastName	DOB	yearEnrolled

Subject

subjectID	name	academicYear	studentID	lecturerID	semester

Lecturer

lecturerID	firstName	lastName

Assessment

asseementID	lecturerID	subjectID	assessmentCentre

Coursework

courseworkID	maxMark	assessmentID	courseworknumber	modulePercentage	gainMark	markOutOF

Exam

examID	maxMark	assessmentID	attempt	resit

Questions

questionID	maxMark	examID	lecturerID

Staff

Firstname	Lastname	**staffID**	lecturerID

Assumptions

The database will store data about newly enrolled and already enrolled students, it will keep track of the subjects that each student is enrolled to, and their performance in academic year and all form of assessments e.g. exams and coursework.

The lecturers will be able to set coursework and exams for students, be able to mark coursework and exams that the student sit.

Keep track of the staff that will be marking the coursework and exams.

The database will store all information about students, exams, courseworks and their marks for each semester and academic year.

Members: Manish, Nahid, Saleem, Tarik, Javed

Normalisation Forms

Why our data is in 4th form

First Normal Form (1NF)
All the values are atomic. There are no multi-valued attributes.

Class: Student

studentID	firstName	lastName	DOB	yearEnrolled
160	Malcolm	Reiner	11/3/1998	2016
161	Jennifer	Black	14/5/1997	2015
162	Niyaz	Gayy	21/07/1993	2016

Class: Lecturer

lecturerID	firstName	lastName
1001	Bob	Marley
1002	Peter	Jones
1003	Kamrul	Islam

Class: Subject

subjectID	name
ECS5028U	Object-Orientated Programming
ECS5038U	Algorithms and Data Structures
ECS5043P	Graphical User Interfaces

Class: Subject_Student

subjectStudentID	academicYear	studentID	subjectID
1	2017/2018	160	ECS5028U
2	2016/2017	161	ECS5028U
3	2017/2018	162	ECS5043P

Class: Subject_Lecturer

subjectLecturerID	lecturerID	subjectID
5	1001	ECS5028U
2	1002	ECS5043P
6	1003	ECS5028U

Class: Coursework

courseworkID	subjectStudentID	mark	maxMark	semester	subjectLecurerID
1	1	28	30	A	6
1	2	24	30	A	5
2	1	58	75	B	6

Class: Exam

examID	subjectStudentID	mark	attemptNumber	maxMark	semester
F23	1	50	1	100	B
F23	2	78	1	100	B
F23	4	82	1	100	B

Class Question

questionID	subjectStudentID	mark	maxMark	subjectLecturerID	examID
1	1	5	10	6	F23
1	2	9	10	5	F23
2	1	12	25	6	F23

Members: Manish, Nahid, Saleem, Tarik, Javed

Second Normal Form (2NF)
These were the classes that were changed to switch to 2NF from 1NF. There were some changes to get rid of the composite primary keys to make fully dependent primary keys. Now all the classes only contain 1 primary key and the non-primary attributes are reliant on the primary key therefore no partial dependency.

Class: AssessmentID

assessment_ID	maxMark	semester	sLecurer_ID
1	30	A	6
1	30	A	5
2	75	B	6

Class: Student_Coursework

sCoursework_ID	mark	sStudent_ID	coursework_ID
1	20	160	1
2	28	161	1
3	68	160	2

Class: Student_Exam

sExam_ID	mark	attemptNumber	sStudent_ID	exam_ID
10001	25	1	160	F23
10002	73	1	161	F23
10003	93	2	160	F25

Class QuestionID

question_ID	maxMark
1	10
1	10
2	25

Class: Student_Question

sQuestion_ID	mark	sExam_ID	question_ID	sLecturer_ID
201	6	10001	1	6
202	15	10001	2	5
203	9	10002	1	6

Third normal form (3NF)

There were no changes made as the table was already in 3NF. This is because all the attributes relied on the primary key therefore there were no non-primary attributes that relied on other non-primary attributes therefore the table could not be split further. However, if the resit was dependant on the mark achieved by the student you could potentially have a different table as follows:

Class: ExamID

exam_ID	**threshold_mark**	**maxMark**	**semester**
1	30	50	A
2	50	100	B
3	60	100	B

Class: Resit

threshold_mark	**sExam_ID**
40	10001
60	10002
50	10003

This would mean that if the threshold_mark was achieved there is no need to resit, but otherwise there would be a resit. However, it isn't specified that resits are only available to those who do not achieve the threshold.

Boyce and Codd Normal Form (BCNF)

There were no changes made as the tables were already in BCNF. All the other values other than the primary key could be repeated making them non-unique and hence not a candidate key. The only candidate keys in the tables are the primary keys.

Fourth Normal Form (4NF)

There were no changes made to the tables since there the primary key values could not be duplicated therefore the other values could not be separated into different tables. Also, there were no multi-valued dependencies as the values all relied on the primary key.