

Databases

(6G4Z0016)

Extra Case Study: “Madcaster University”

Stephen Gordon

s.gordon@mmu.ac.uk

Madcaster University Case Study Pt.1

Madcaster University is a small, traditional university in the market town of the same name in a rural area of Wales.

The recently appointed Vice Chancellor has decided to expand outside the scope of the traditional subjects taught in the university and has set up a School of Business and Computer Studies (B&CS).

B&CS has created a modular scheme where students can choose the units they study from a range of modules. Each student must study six 20-credit modules a year and must achieve 360 credits (120 at each of levels 1, 2 and 3) to be awarded an honours degree. The scheme includes 3 degree titles and students must include the core units for their chosen course of study in their module choices for the year.

The 3 degree titles are:

- ◆ Business Information Technology (BIT)
- ◆ e-Commerce Systems (ECS)
- ◆ Software Engineering (SOE)

In addition to the taught degrees there is a Postgraduate Diploma in IT Research. This is a one-year course; there are no formal taught modules, but the students are assigned to a staff member who acts as the individual's supervisor.

The B&CS modular scheme is a first for Madcaster University and the current administrative systems do not cope with its structure. It has therefore been decided to set up a new system to register B&CS students and to record their module choices and their progress throughout their chosen course.

Madcaster University Case Study Pt.2

The requirements of the system are:

- ◆ The system should be set up with details of the: Degree Titles (name and reference code); Staff (tutor code and tutor name); Modules (module code, module title, tutor code and learning outcomes).
- ◆ Student registration: each student joining the course should be registered and associated with a degree / diploma title. The details required are name, address, gender, date of birth. Each student is allocated a unique registration number. The registration form is shown in Figure 1.
- ◆ Student amendment: at any time, student details can be amended or corrected. Students can withdraw and this is recorded as an amendment using a status indicator (the details of the withdrawn student are kept on the system). The student registration number cannot be amended.
- ◆ Module choice: at the start of each year all students (excluding research diploma students) choose six modules (including core modules for their chosen title). These choices should be registered on the system. The module choices form is shown in Figure 2. Sometimes at level 1 students immediately give their module choices for the year, and this can be done by admin staff on behalf of the student. Sometimes the student is not clear and will need to do it themselves via the web.
- ◆ Module results: at the end of the year the student's overall mark for each module is recorded together with a pass / fail indicator. To progress, students must pass all six modules taken that year; if they fail, they are out (and the status indicator is updated to show that). To obtain a degree the student must pass all three years: the student's overall degree grade will be recorded at the end of their final year. Diploma results are also recorded on the system. |

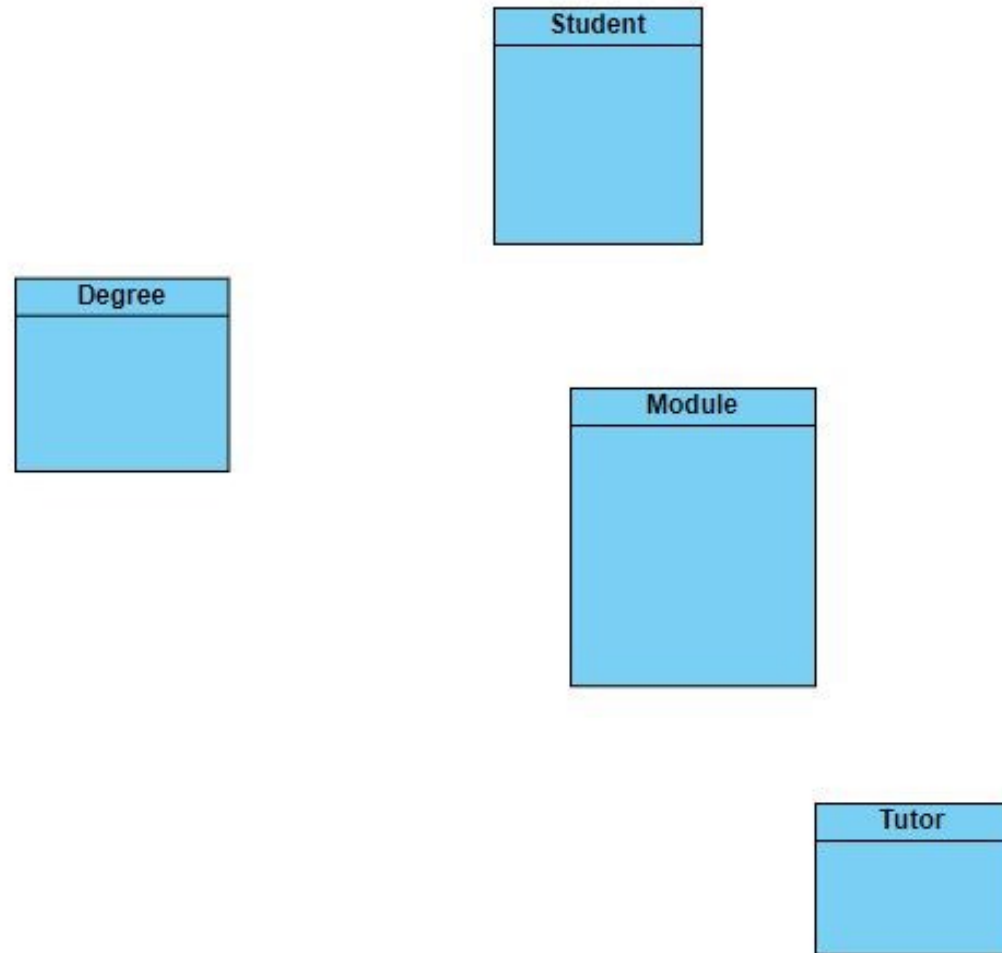
Madcaster University Exercise 1

Draw a top-down ERD for the Madcaster University B&CS system

Exercise 1 –

ERD St. 1:

Identify Entities



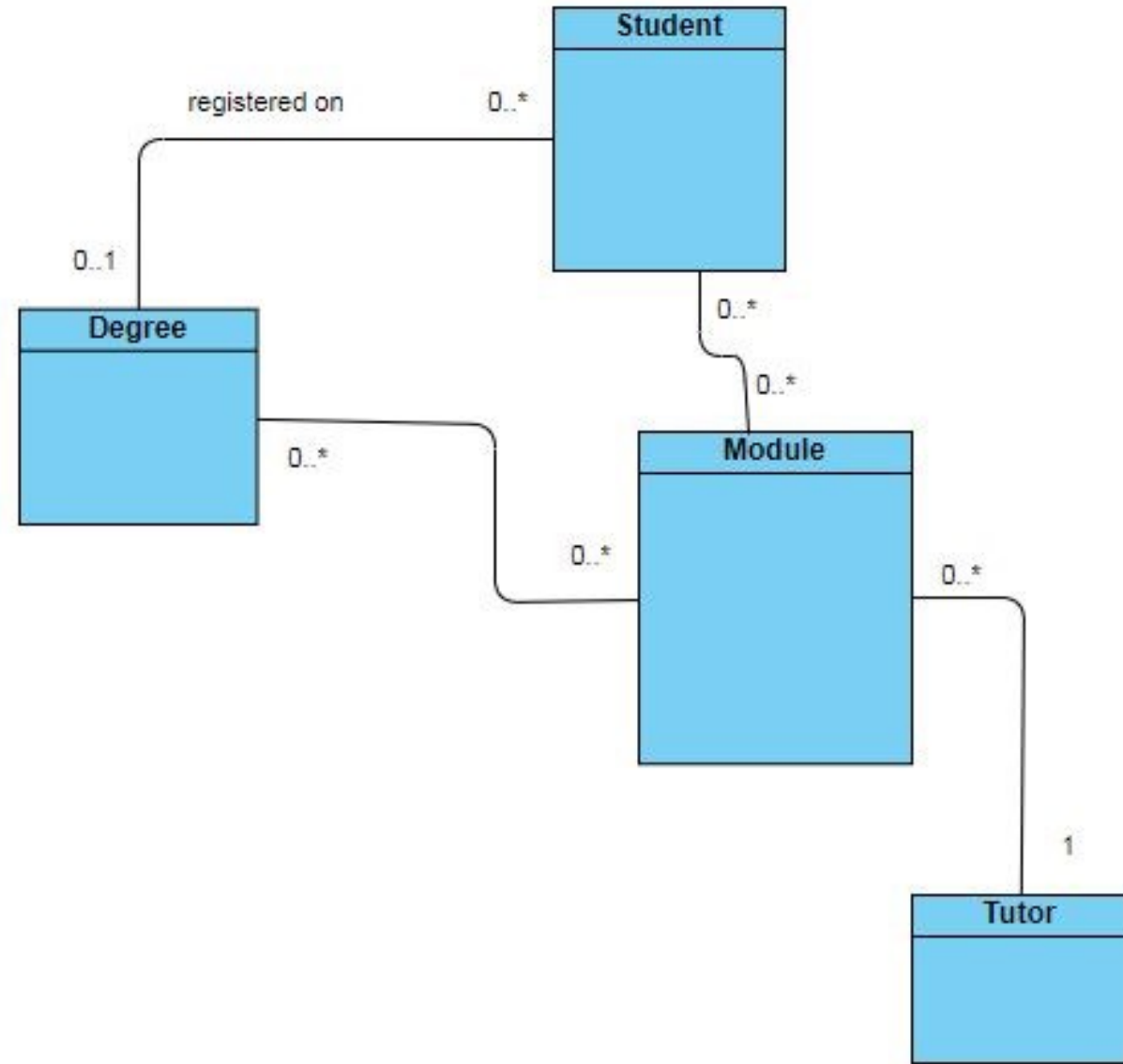
Possible entities that didn't make it: Madcaster Uni (only 1); Market Town (relevance); Wales (only 1); Vice Chancellor (only 1); B&CS (only 1); unit (same as module); credit (could be an attribute of module – but all 20 credits, so unnecessary); current admin system (only 1, relevance); core module (attribute of module); optional module (attribute of module); admin staff (not in scope); student details and student status (attributes of student); module choice and module result (attributes to be shown using relationships).

Exercise 1 –

ERD St. 2:

Add Relationships

Assumption 1: we are only going to record current module leader

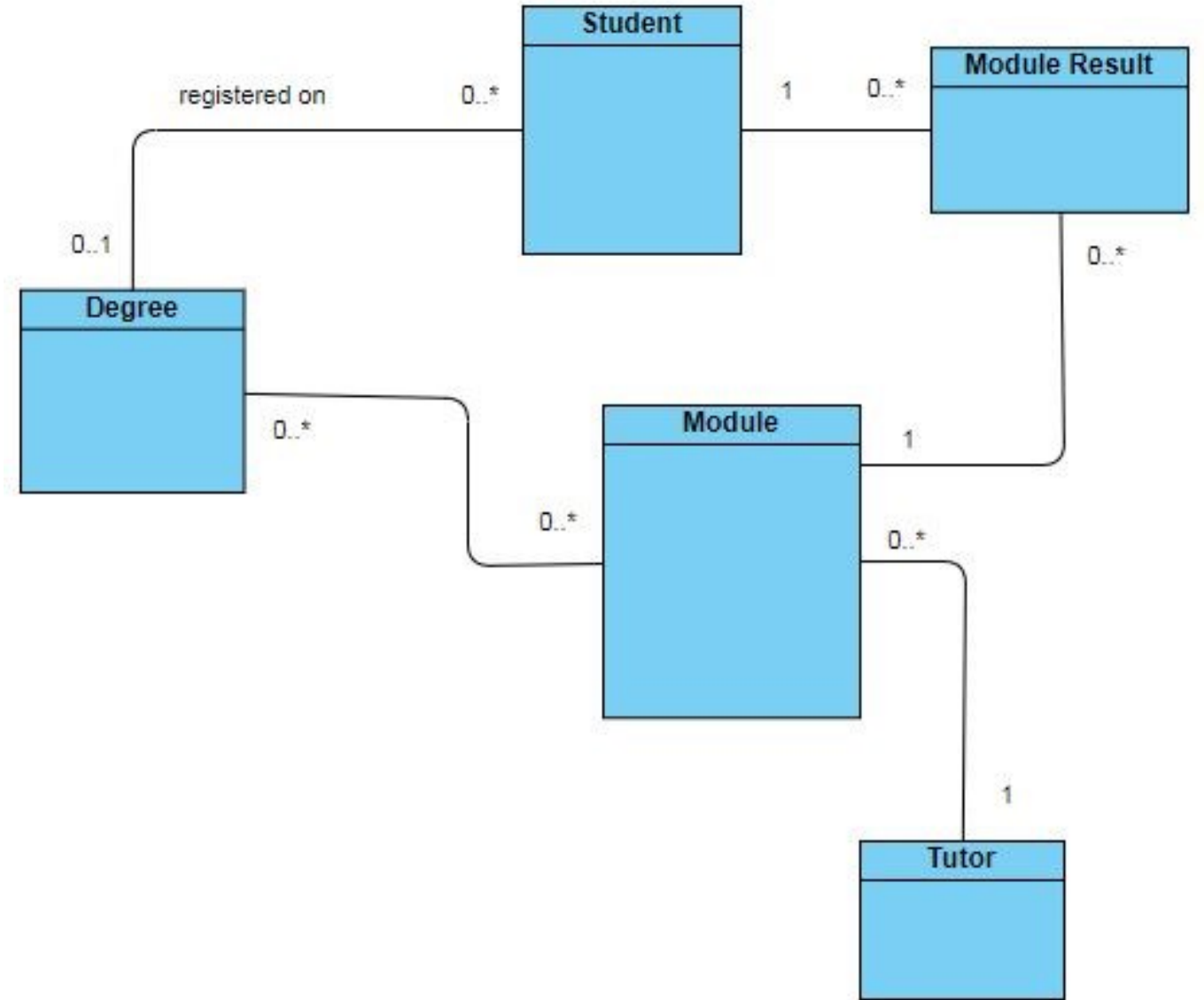


Exercise 1 –

ERD St. 3a:

Resolve M:N Relationships

Assumption 1: we are only going to record current module leader



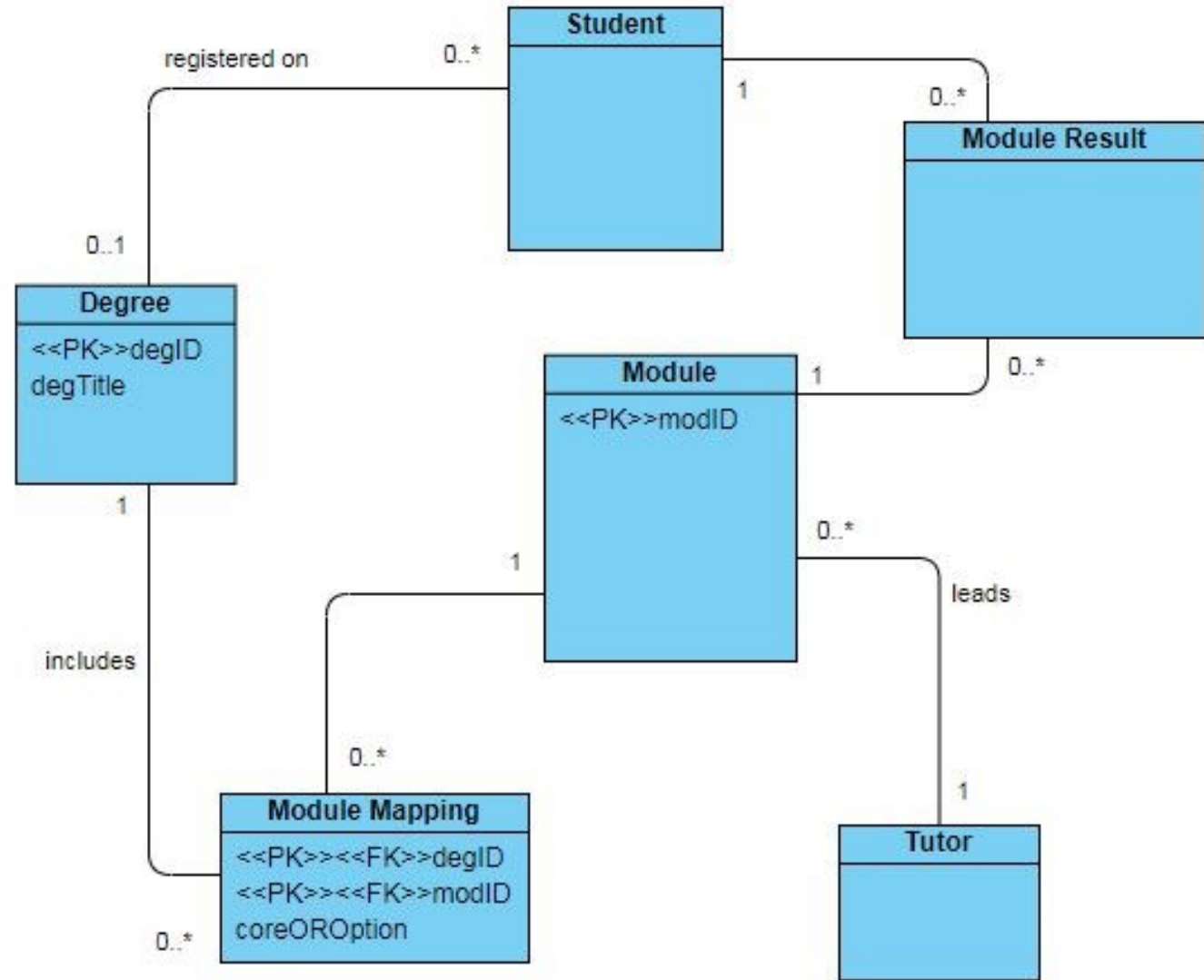
Exercise 1 –

ERD St. 3b:

Resolve M:N Relationships

Assumption 1: we are only going to record current module leader

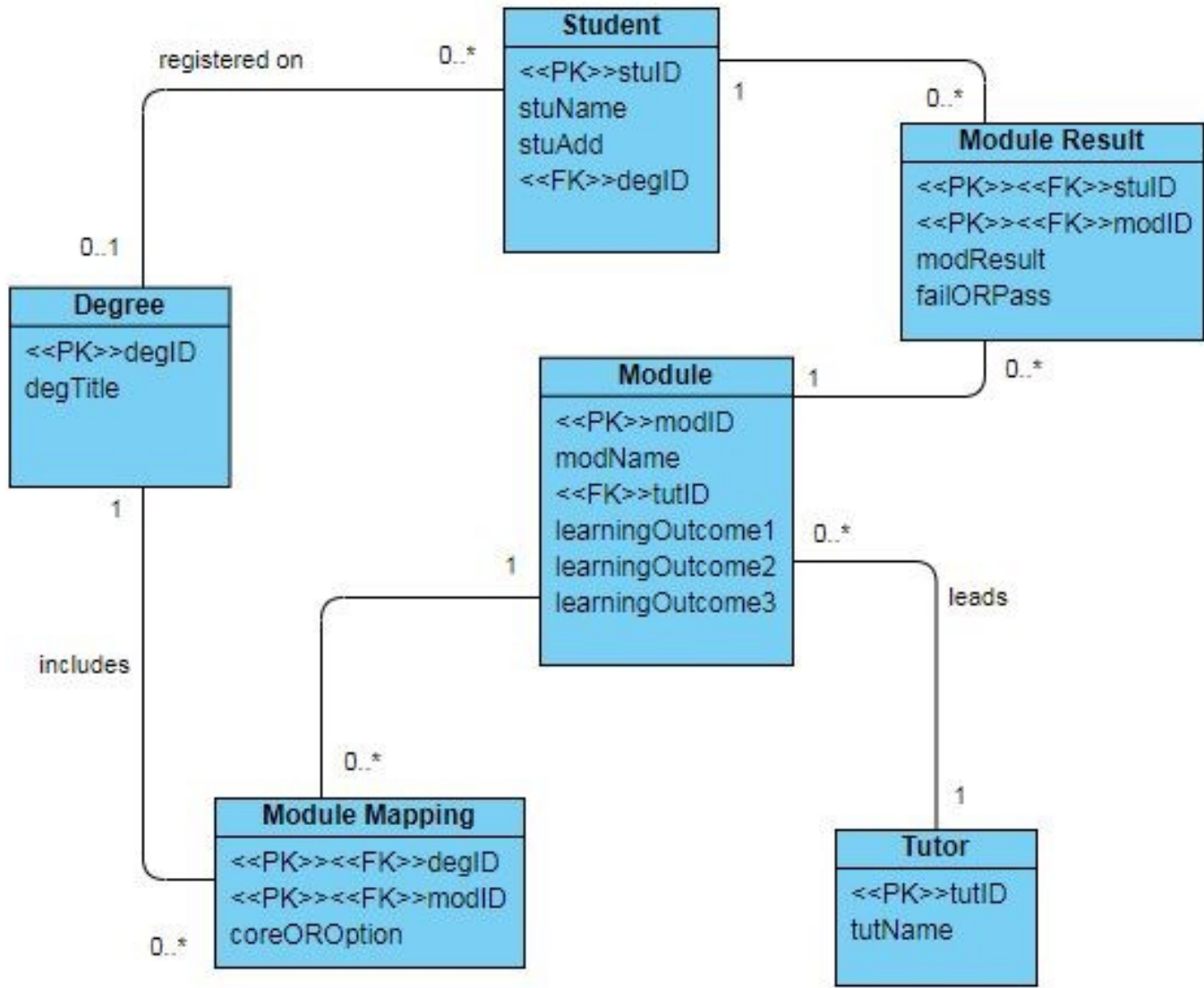
Assumption 2: a module can be optional or core on several degree paths



Exercise 1 – Suggested Solution Final

Assumption 1: we are only going to record current module leader

Assumption 2: a module can be optional or core on several degree paths



Madcaster University Exercise 2

- a) Normalise the two documents given in Madcaster University B&CS system
- b) Merge your normalisations and produce a single bottom-up data model (ERD)

Madcaster University

MU

Student Registration

Student:

Reg No

Name

Address

Gender

M / F / U

Date of birth

dd.mm.yy

Course Code

Title

Figure 1 Student Registration Form

Student Reg Normalisation

UNF

1NF

2NF

3NF

stuID

stuName

stuAddr

stuGender

stuDOB

degCode

degName

Student Reg Normalisation

UNF

stuID
stuName
stuAddr
stuGender
stuDOB
degCode
degName

1NF

stuID
stuName
stuAddr
stuGender
stuDOB
degCode
degName

2NF

3NF

1NF Rule (of thumb): Remove repeating groups

(Real rule: Do you depend upon the key?)

Student Reg Normalisation

UNF

stuID
stuName
stuAddr
stuGender
stuDOB
degCode
degName

1NF

stuID
stuName
stuAddr
stuGender
stuDOB
degCode
degName

2NF

stuID
stuName
stuAddr
stuGender
stuDOB
degCode
degName

3NF

2NF Rule: Remove part key dependencies

Or say: "Do you depend upon the whole key?"

Student Reg Normalisation

UNF

stuID
stuName
stuAddr
stuGender
stuDOB
degCode
degName

1NF

stuID
stuName
stuAddr
stuGender
stuDOB
degCode
degName

2NF

stuID
stuName
stuAddr
stuGender
stuDOB
degCode
degName

3NF

stuID
stuName
stuAddr
stuGender
stuDOB
degCode*

degCode
degName

3NF Rule: Remove transitive dependencies

Or, finally, say: “Do you depend upon the key, the whole key and nothing but the key?”

Madcaster University

MU

Module Choice

Student:

Reg No

Course:

Code

Title

Module

Code

Title

1

2

3

4

5

6

Figure 2 Module Choice Form

Module Choice Normalisation

UNF

1NF

2NF

3NF

stuID

degCode

degName

modCode

modTitle

I've chosen a relevant key. It is relevant because the student makes the module choices. I'll show you what happens if you choose mod_code shortly...

Module Choice Normalisation

UNF

stuID
degCode
degName
modCode
modTitle

1NF

stuID
degCode
degName

stuID
modCode
modTitle

2NF

3NF

1NF Rule (of thumb): Remove repeating groups

(Real rule: Do you depend upon the key?)

Module Choice Normalisation

UNF

stuID
degCode
degName
modCode
modTitle

1NF

stuID
degCode
degName

stuID
modCode
modTitle

2NF

stuID
degCode
degName

stuID
modCode

modCode
modTitle

3NF

2NF Rule: Remove part key dependencies

Or say: “Do you depend upon the whole key?”

Module Choice Normalisation

UNF

stuID
degCode
degName
modCode
modTitle

1NF

stuID
degCode
degName

stuID
modCode
modTitle

2NF

stuID
degCode
degName

stuID
modCode

modCode
modTitle

3NF

stuID
degCode*

degCode
degName

stuID
modCode

modCode
modTitle

3NF Rule: Remove transitive dependencies

Or, finally, say: “Do you depend upon the key, the whole key and nothing but the key?”

Exercise 2 – Suggested Solution

Student Reg Form

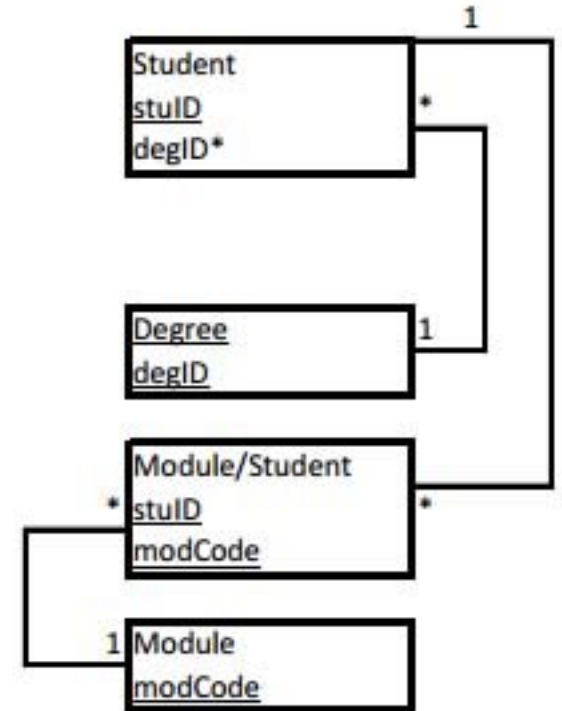
UNF	1NF	2NF	3NF
<u>stuID</u>	<u>stuID</u>	<u>stuID</u>	<u>stuID</u>
stuName	stuName	stuName	stuName
stuAddr	stuAddr	stuAddr	stuAddr
stuGender	stuGender	stuGender	stuGender
stuDOB	stuDOB	stuDOB	stuDOB
degID	degID	degID	degID*
degName	degName	degName	
			<u>degID</u>
			degName

Module Choice Form

UNF	1NF	2NF	3NF
<u>stuID</u>	<u>stuID</u>	<u>stuID</u>	<u>stuID</u>
degID	degID	degID	degID*
degName	degName	degName	
modCode			<u>degID</u>
modTitle	<u>stuID</u>	<u>stuID</u>	degName
	<u>modCode</u>	<u>modCode</u>	
	modTitle		<u>stuID</u>
		<u>modCode</u>	<u>modCode</u>
		modTitle	
			<u>modCode</u>
			modTitle

Merge 3NFs

<u>stuID</u>	
stuName	
stuAddr	>>>
stuGender	
stuDOB	
degID*	
<u>degID</u>	>>>
degName	
<u>stuID</u>	>>>
<u>modCode</u>	
<u>modCode</u>	>>>
modTitle	



To merge normalisations: just add both sets of tables together and remove anything that repeats

Exercise 2 – Suggested Solution

Extra Notes

This is for advanced students only. You can ignore it if you like

For the module form, if we use modID as the starting primary key (PK), instead of stuID, it comes out differently.

Here, with modID as starting key, we produce the extra weak entity (outlined in red) that we have on our Final Top Down ERD - “Module Mapping”.

One RDA start key ends up producing more useful info than another. Both RDAs, however, produce tables and keys that are 100% correct and useful.

This is a good example of how systems analysis and design can be quite messy sometimes, and how it is good to try lots of things out when you are deciding upon what data structure to use to solve your business problem.

Module Choice Form			
UNF	1NF	2NF	3NF
<u>modID</u>	<u>modID</u>	<u>modID</u>	<u>modID</u>
modTitle	modTitle	modTitle	modTitle
stuNo			
courseID	<u>modID</u>	<u>modID</u>	<u>modID</u>
courseTitle	<u>stuNo</u>	<u>stuNo</u>	<u>stuNo</u>
	<u>modID</u>	<u>modID</u>	<u>modID</u>
	<u>courseID</u>	<u>courseID</u>	<u>courseID</u>
	courseTitle		
		<u>courseID</u>	<u>courseID</u>
		courseTitle	courseTitle

Assumption 2: a module can be optional or core on several degree paths

Note: at 1NF we end up with 2 repeating groups: one for student (one module can have many students) and one for course (one module can appear on many courses)

Madcaster University Exercise 3

Merge your top-down and bottom-up ERDs to produce a final ERD for your database

Exercise 3 – Suggested Solution

This is the same as the top-down ERD because, in this case, everything in the bottom up ERD is already in the top-down ERD.

