ECM2419

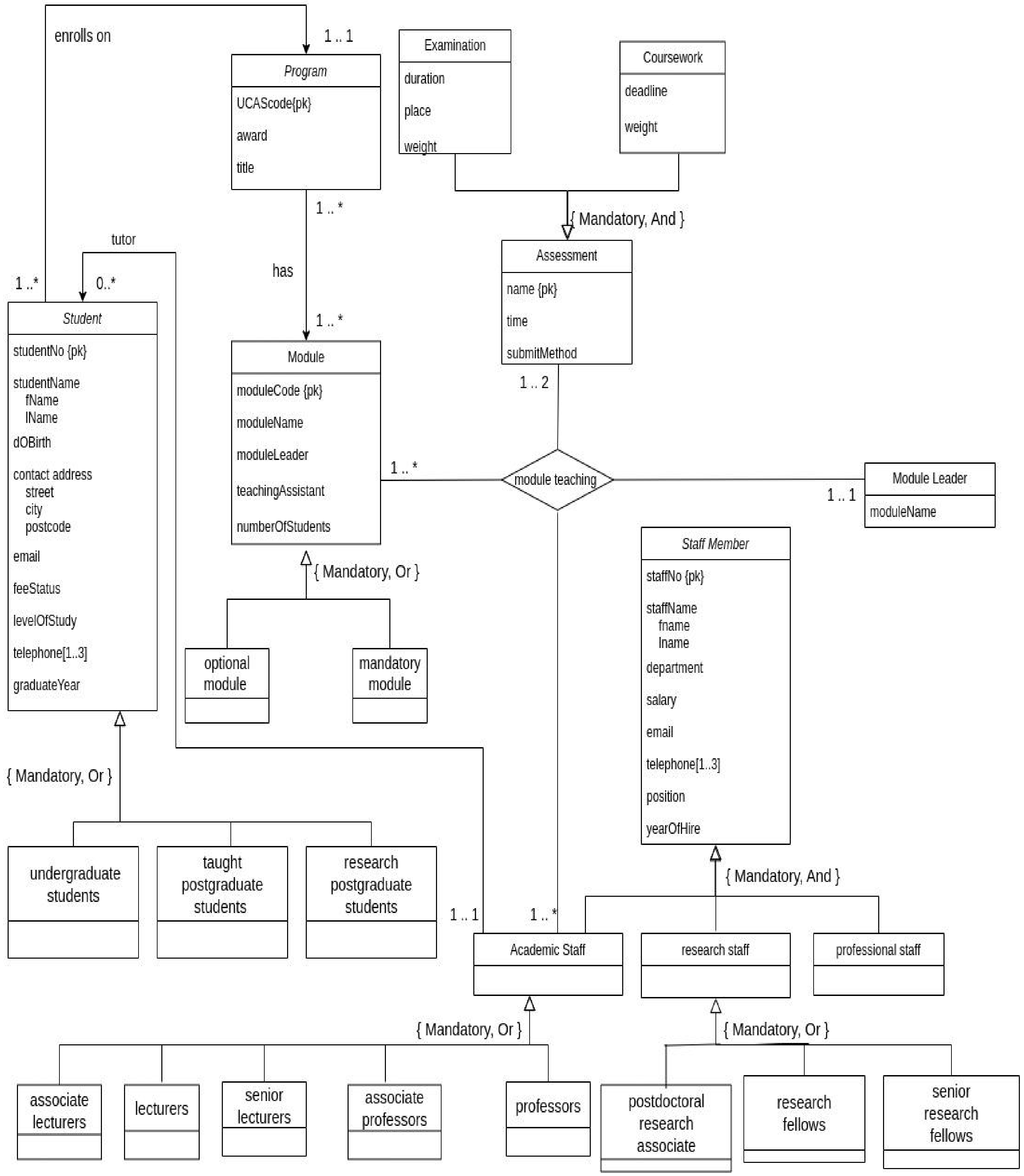
Database Theory and Design

CourseWork

Kechen Liu
StudentId 710016126

part1

UML Graph



Entity

Staff

Attributes: staffNo(primary key), staffName(composite: fname, lname), category, salary, email, telephone(min:1, max:3), position, yearOfHire.

Staff has 8 attributes, it is assumed that each staff has a unique id(staffNo) by which it can be identified, so staffNo is the prime key.

Staff need a name to distinguish who the person is, and the name attribute has two composite attributes, first name and last name. And staff also has attributes like department, salary, email, position, year of hire. Note that the phone number is a multivalued attribute which must contain at least one and at most three values.

And I use enhanced constructs to specialize the sub type of staff(academic staff, research staff, professional staff). For the participation constraints, staff must be one of the three types, so I chose mandatory. For the disjoint constrains, staffs might have several roles, an academic staff can also be a research staff, so I chose and(non-disjoint).

Using specialization to classify academic staff as associate lecturers, lecturers, senior lecturers, associate professors and professors, for the participation constraints, academic staff must be one of the three types, so I chose mandatory. For the disjoint constrains, staffs only have one role, so I chose or(disjoint).

Using specialization to classify research staff as postdoctoral research associate, research fellows and senior research fellows. For the participation constraints, academic staff must be one of the three types, so I chose mandatory. For the disjoint constrains, staffs only have one role, so I chose or(disjoint).

Staff(staffNo, fname, lname, category, salary, email, telephone, position, yearOfHire)

staffNo	fname	lname	category	salary	email	telephone	position	yearOfHire
0001	James	Walking	permanent	3,0000	jw@ex	12345678	academic staff	2021

Student

Attributes: studentNo(primary key), studentName(composite: fname, lname), dOBirth, contact address(composite: street, city, postcode), email, feestatus, levelOfStatus, telephone(min:1, max:3), graduateYear.

Student holds 9 attribute. It is assumed that each student has a unique id(studentNo) by which it can be identified, so studentNo is the prime key.

Student also need a name to distinguish who the person is, and the name attribute has two composite attributes, first name and last name. And student also has attributes like contact address(with composite attributes street, city, postcode), and postcode can serve as a foreign key to relate street and city, email address of the student, feestatus to find whether the student has paid or not, levelOfStatus to find which year is the student in, note that the phone number is a multivalued attribute which must contain at least one and at most threevalues, and the year will the student graduate.

And I use enhanced constructs to specialize the sub type of student(undergraduate student, taught postgraduate student, research postgraduate student). For the participation constraints, student must be one of the three types, so I chose mandatory. For the disjoint constrains, student only have one role at the same time, so I chose Or(disjoint).

Student(studentNo, fname, lname, dOBirth, street, city, postcode, email, feestatus, levelOfStatus, telephone, graduateYear)
Primary Key : studentNo
Foreign Key : postcode

studentNo	fname	lname	dOBirth	postcode	email	feestatus	levelOfStatus	telephone	graduateYear
001	Anna	Smith	12/03/2002	EX4 2RF	2@	done	stage2	1234567	2024

Program

Attribute: UCAScode(prime key), award, title.

Program hold three attributes, an UCAScode to serve as prime key, the award of the program, and the title of the program. For example, BSc Computer Science program has UCAS code GG41, award BSc(Hons), and title Computer Science.

Program(UCAScode, award, title)

UCAScode	award	title
RD4344	BSc	Computer Science

Module

Attribute: moduleCode(prime key), moduleName, moduleLeader, number of Students, teachAssistant

Module has 4 attributes, an moduleCode to serve as prime key to distinguish different modules. eg. Database Theory(module name), with moduleCode ECM2419, Module Leader Zeliang Wang, teaching assistant Jia Wu.

Using specialization to classify module as optional module and mandatory For the participation constraints, module must be one of the two types, so I chose mandatory. For the disjoint constrains, module must be optional module or mandatory module, so I chose or(disjoint).

Module(moduleCode, moduleName, moduleLeader, number of Students, teachAssistant)

moduleCode	moduleName	moduleLeader	number of student	teachAssistant
ECM2419	database	Zeliang Wang	200	Jia Wu

Assessment

Attribute: name(prime key), time, submitMethod

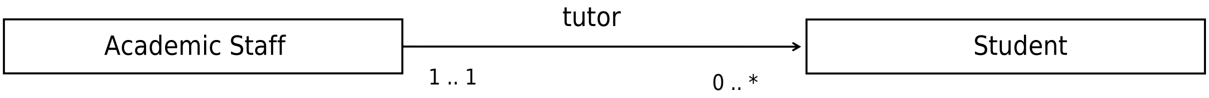
Assessment has 3 attributes, a name to serve as prime key to distinguish different assessment. A time when the assessment released, and the submitMethod.
Using specialization to classify assessment as examination and coursework.
Examination has 3 more attributes, duration, place, weight.
Coursework has 2 more attributes, deadline, weight.
Most modules are assessed through both an examination and coursework. However, some modules are entirely assessed through either a coursework or an examination, so I use Mandatory And.

Assessment(name, time submitMethod)

name	time	submitMethod
Cycling Management System	01/03/2002	bart

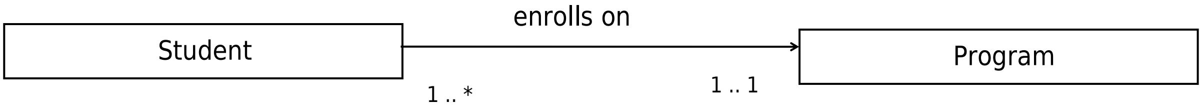
Relationship

Academic Staff tutors Student Relationship



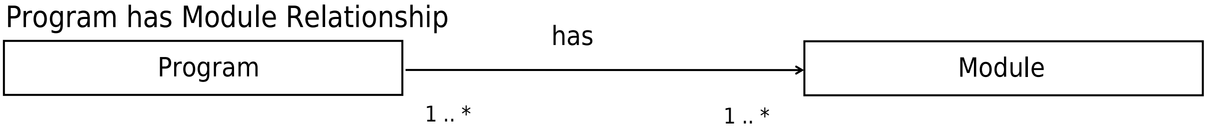
Entities involved: Academic Staff Student
Binary relationship, one-to-many cardinality, Student has mandatory participation, Academic Staff has optional participation. Each student is allocated a tutor from the academic staff on enrolling. An academic staff may tutor zero or more student.

Student enrolls on Program Relationship



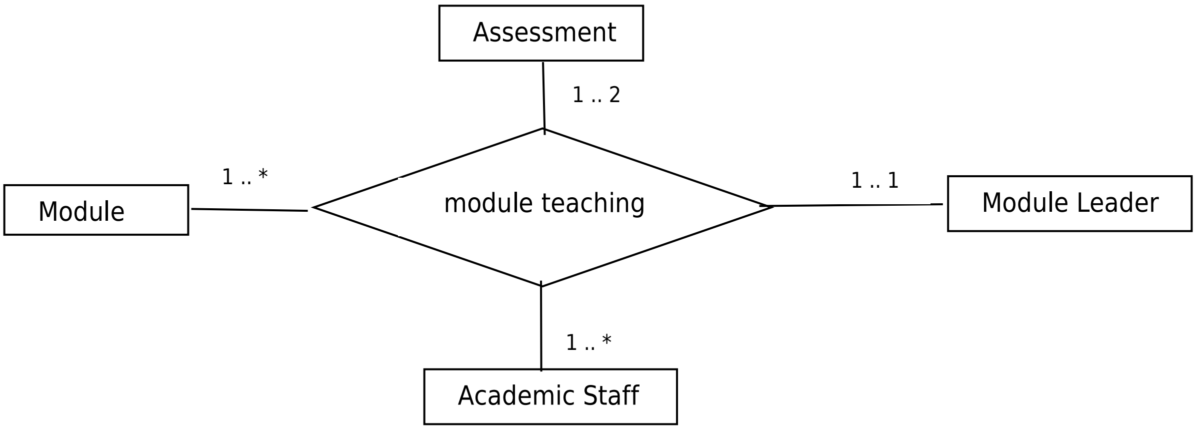
Entities involved: Student Program
Binary relationship, one-to-many cardinality, Student has mandatory participation, Program has mandatory participation. Students are enrolled on one programme at a time. And a program has 1 more students.

Program has Module Relationship



Entities involved: Program, Module
Binary relationship, many-to-many cardinality, Module has mandatory participation, Program has mandatory participation. Each degree programme has a number of mandatory modules and optional modules. A module can be shared for different programs.

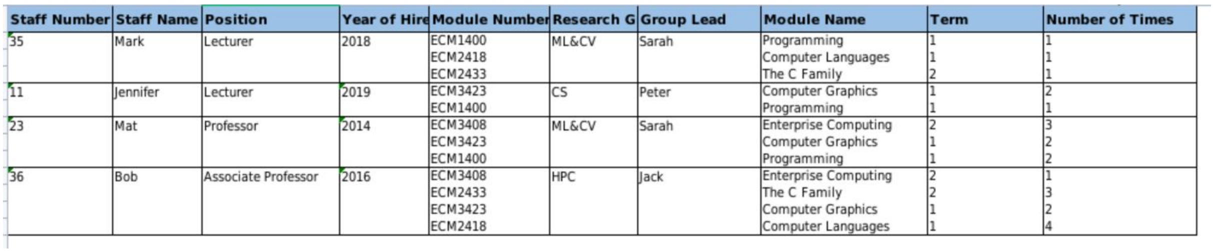
ModuleTeaching Relationship



Entities involved: Academic Staff, Module, Assessment, Module Leader
Quaternary relationship, Module has mandatory participation, Program has mandatory participation. Each degree programme has a number of mandatory modules and optional modules. A module can be shared for different programs. Each module is taught by one or more academic staff members. One of these academic staff members are designated the Module Leader. The role of the Module Leader is to organise the module’s teaching and assessment activities. Modules are assessed through both an examination and coursework. However, some modules are entirely assessed through either a coursework or an examination.

Part2

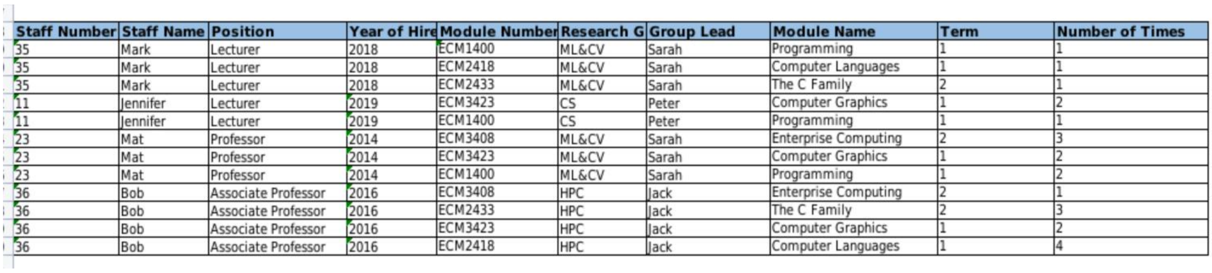
normalised table definitions



Staff Number	Staff Name	Position	Year of Hire	Module Number	Research G	Group Lead	Module Name	Term	Number of Times
35	Mark	Lecturer	2018	ECM1400 ECM2418 ECM2433	ML&CV	Sarah	Programming Computer Languages The C Family	1 1 2	1 1 1
11	Jennifer	Lecturer	2019	ECM3423 ECM1400	CS	Peter	Computer Graphics Programming	1 1	2 1
23	Mat	Professor	2014	ECM3408 ECM3423 ECM1400	ML&CV	Sarah	Enterprise Computing Computer Graphics Programming	2 1 1	3 2 2
36	Bob	Associate Professor	2016	ECM3408 ECM2433 ECM3423 ECM2418	HPC	Jack	Enterprise Computing The C Family Computer Graphics Computer Languages	2 2 1 1	1 3 2 4

- 1NF

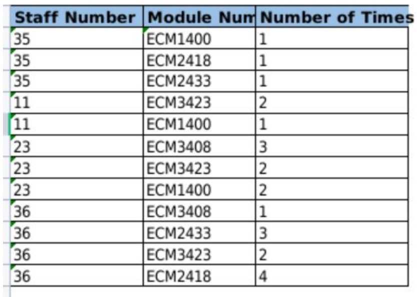
StaffTeachingRecords(Staff Number, Staff Name, Position, Year Of Hire, Module Number, Research Group, Group Lead, Module Name, Term, Number of Times)



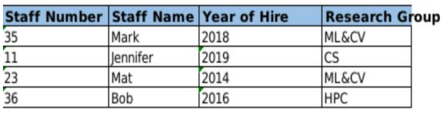
Staff Number	Staff Name	Position	Year of Hire	Module Number	Research G	Group Lead	Module Name	Term	Number of Times
35	Mark	Lecturer	2018	ECM1400	ML&CV	Sarah	Programming	1	1
35	Mark	Lecturer	2018	ECM2418	ML&CV	Sarah	Computer Languages	1	1
35	Mark	Lecturer	2018	ECM2433	ML&CV	Sarah	The C Family	2	1
11	Jennifer	Lecturer	2019	ECM3423	CS	Peter	Computer Graphics	1	2
11	Jennifer	Lecturer	2019	ECM1400	CS	Peter	Programming	1	1
23	Mat	Professor	2014	ECM3408	ML&CV	Sarah	Enterprise Computing	2	3
23	Mat	Professor	2014	ECM3423	ML&CV	Sarah	Computer Graphics	1	2
23	Mat	Professor	2014	ECM1400	ML&CV	Sarah	Programming	1	2
36	Bob	Associate Professor	2016	ECM3408	HPC	Jack	Enterprise Computing	2	1
36	Bob	Associate Professor	2016	ECM2433	HPC	Jack	The C Family	2	3
36	Bob	Associate Professor	2016	ECM3423	HPC	Jack	Computer Graphics	1	2
36	Bob	Associate Professor	2016	ECM2418	HPC	Jack	Computer Languages	1	4

- 2NF

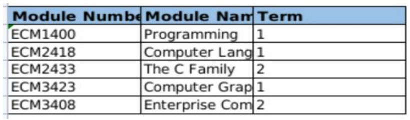
StaffTeachingRecords(Staff Number, Module Number, Number of Times)
Staff(Staff Number, Staff Name, Position, Year Of Hire, Research Group)
Module(Module Number, Module Name, Term)
Group(Research Group, Group Lead)



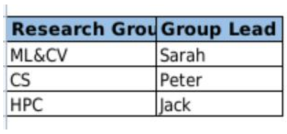
Staff Number	Module Num	Number of Times
35	ECM1400	1
35	ECM2418	1
35	ECM2433	1
11	ECM3423	2
11	ECM1400	1
23	ECM3408	3
23	ECM3423	2
23	ECM1400	2
36	ECM3408	1
36	ECM2433	3
36	ECM3423	2
36	ECM2418	4



Staff Number	Staff Name	Year of Hire	Research Group
35	Mark	2018	ML&CV
11	Jennifer	2019	CS
23	Mat	2014	ML&CV
36	Bob	2016	HPC



Module Numbe	Module Nam	Term
ECM1400	Programming	1
ECM2418	Computer Lang	1
ECM2433	The C Family	2
ECM3423	Computer Grap	1
ECM3408	Enterprise Com	2



Research Grou	Group Lead
ML&CV	Sarah
CS	Peter
HPC	Jack

the tables above are already in 3NF

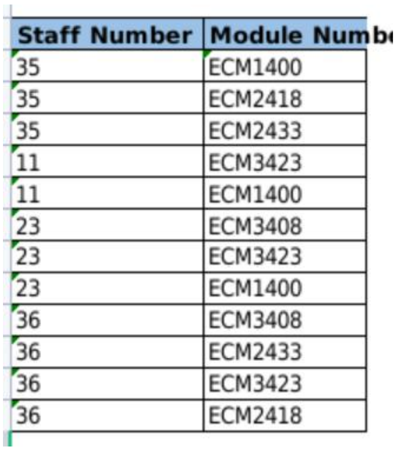
- 3NF

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

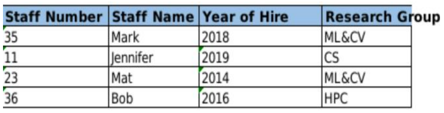
- BCNF

StaffTeachingRecords(Staff Number, Module Number)
Staff(Staff Number, Staff Name, Position, Year Of Hire, Research Group)

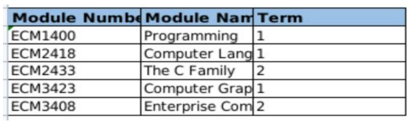
Module(Module Number, Module Name, Term)
Group(Research Group, Group Lead)
ModuleTeachingTimes(Staff Number, Module Number, Number of Times)



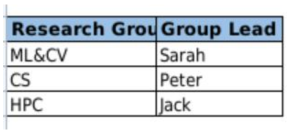
Staff Number	Module Numb
35	ECM1400
35	ECM2418
35	ECM2433
11	ECM3423
11	ECM1400
23	ECM3408
23	ECM3423
23	ECM1400
36	ECM3408
36	ECM2433
36	ECM3423
36	ECM2418



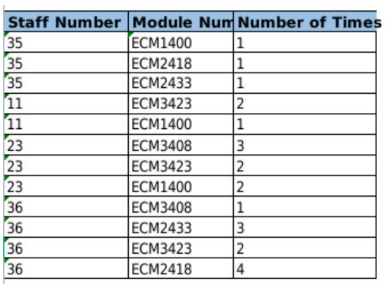
Staff Number	Staff Name	Year of Hire	Research Group
35	Mark	2018	ML&CV
11	Jennifer	2019	CS
23	Mat	2014	ML&CV
36	Bob	2016	HPC



Module Numbe	Module Nam	Term
ECM1400	Programming	1
ECM2418	Computer Lang	1
ECM2433	The C Family	2
ECM3423	Computer Grap	1
ECM3408	Enterprise Com	2



Research Grou	Group Lead
ML&CV	Sarah
CS	Peter
HPC	Jack



Staff Number	Module Num	Number of Times
35	ECM1400	1
35	ECM2418	1
35	ECM2433	1
11	ECM3423	2
11	ECM1400	1
23	ECM3408	3
23	ECM3423	2
23	ECM1400	2
36	ECM3408	1
36	ECM2433	3
36	ECM3423	2
36	ECM2418	4

Description

Functional Dependency
1. Identify FD
Fd1: Staff Number -> Staff Name, Position, Year Of Hire, Research Group
Fd2: Research Group -> Group Lead
Fd3: Module Number -> Module Name, Term
Fd4: Staff Number, Module Number -> Number of Times

2. Identify Primary Key
a. Identify all candidate key(s) by looking at the determinants.
b. Determine which candidate key is not functionally dependent on any other candidate key. This this the primary key.

e.g.
Staff Number -> Staff Name, Position, Year Of Hire, Research Group
Research Group -> Group Lead
Module Number -> Module Name, Term
Staff Number, Module Number -> Number of Times

Primary key: Staff Number, Module Number

3. Identify Partial Functional Dependencies and Full functional dependency:
a. Partial Functional Dependencies
Staff Number -> Staff Name, Position, Year Of Hire, Research Group
Research Group -> Group Lead
Module Number -> Module Name, Term

b. Full functional dependency
Staff Number, Module Number -> Number of Times

4. Identify Transitive Functional Dependencies
Staff Number -> Staff Name, Position, Year Of Hire, Research Group
Research Group -> Group Lead
Module Number -> Module Name, Term
Staff Number, Module Number -> Number of Times

Transitive Functional Dependencies
Staff Number -> Research Group -> Group Lead

Summary

- Partial Functional Dependencies

Staff Number -> Staff Name, Position, Year Of Hire, Research Group
Module Number -> Module Name, Term
Research Group -> Group Lead

- Full functional dependency

Staff Number, Module Number -> Number of Times

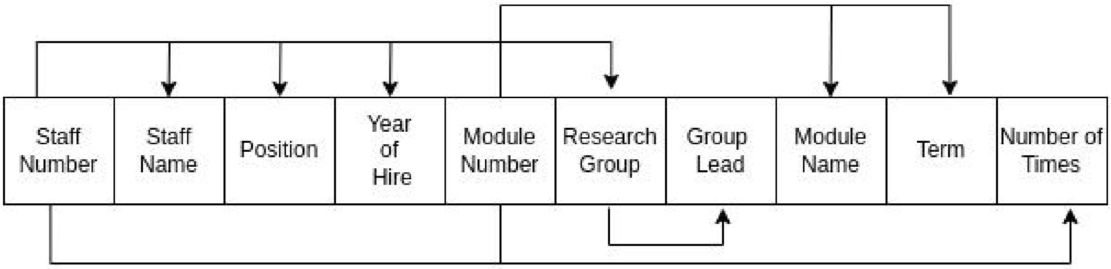
- Transitive Functional Dependencies

Staff Number -> Research Group -> Group Lead

1st Normal Form
- remove repeating groups

Repeating groups:
(Module Number, Module Name, Term, Number of Times)

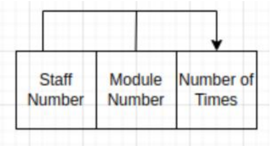
StaffTeachingRecords(Staff Number, Staff Name, Position, Year Of Hire, Module Number, Research Group, Group Lead, Module Name, Term, Number of Times)

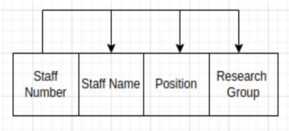


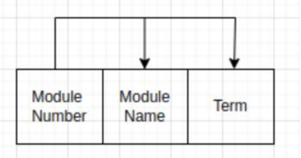
2nd Normal Form
- remove partial dependencies

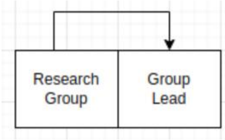
Partial Functional Dependencies
Staff Number -> Staff Name, Position, Year Of Hire, Research Group
Module Number -> Module Name, Term
Research Group -> Group Lead

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







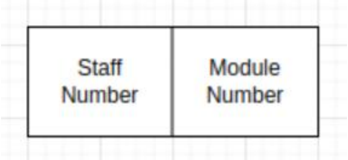


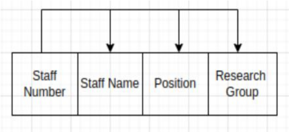
3rd Normal Form
- remove transitive dependencies

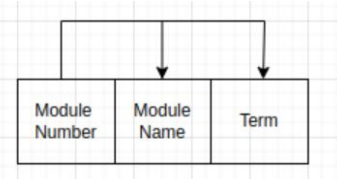
The table above already in 3NF.
StaffTeachingRecords(Staff Number, Module Number, Number of Times)
Staff(Staff Number, Staff Name, Position, Year Of Hire, Research Group)
Module(Module Number, Module Name, Term)
Group(Research Group, Group Lead)

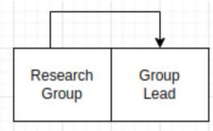
3NF to BCNF
There is one determinant is not candidate key
Staff Number, Module Number -> Number of Times

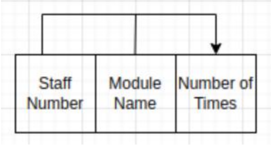
BCNF
StaffTeachingRecords(Staff Number, Module Number)
Staff(Staff Number, Staff Name, Position, Year Of Hire, Research Group)
Module(Module Number, Module Name, Term)
Group(Research Group, Group Lead)
ModuleTeachingTimes(Staff Number, Module Number, Number of Times)


Staff Number
Module Number


Staff Number
Staff Name
Position
Research Group


Module Number
Module Name
Term


Research Group
Group Lead


Staff Number
Module Name
Number of Times