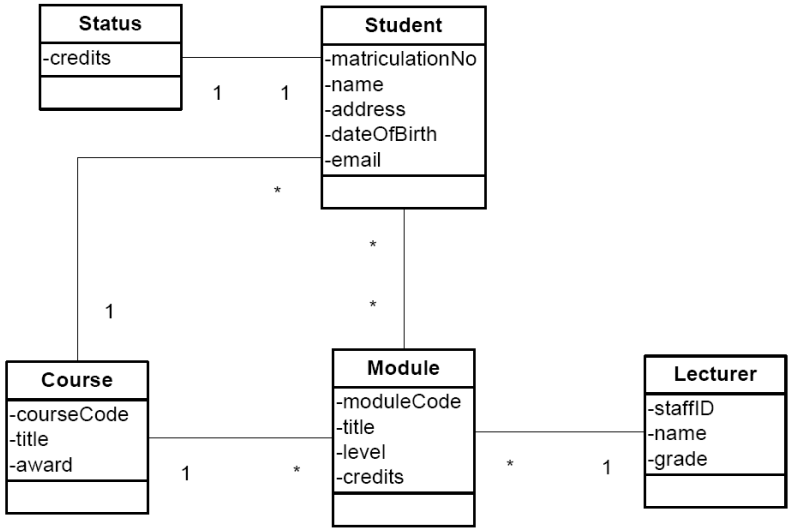
TUTORIAL 3: Tables and relationships (SOLUTIONS)

Question 1

The diagram below shows part of the domain model for a university management system.

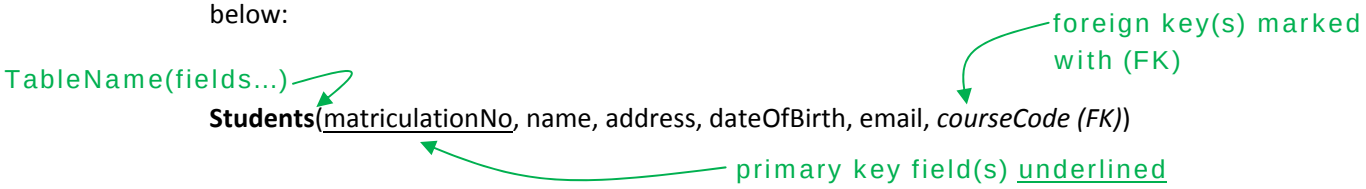


1. For each pair of related entities, identify from the diagram the type of relationship which exists between them, and suggest how the relationship would be implemented in a database.

   Write your answers in a table like this:

Entities	Type of relationship	Database implementation
Course-Student	one-to-many	foreign key in Students table
Student-Status	one-to-one	credits field in Students
Course-Module	one-to-many	foreign key in Modules table (many)
Student-Module	many-to-many	additional table (e.g. StudentModules or Enrollments) with foreign keys to both tables
Lecturer-Module	one-to-many	foreign key in Modules table (many)

2. Based on your answers, **write out the database schema**, using the notation shown below:



(Note that this is **not** a complete solution for the Students table)

Courses(courseCode, title, award)
Students(matriculationNo, name, address, dateOfBirth, email, credits, *courseCode(FK)*)
Modules(moduleCode, title, level, credits, *courseCode(FK)*, *staffID(FK)*)
StudentModules(*matriculationNo(FK)*, *moduleCode(FK)*)
Lecturers(staffID, name, grade)

The one-to-one relationship between Student and Status could also be implemented like this:

Students(matriculationNo, name, address, dateOfBirth, email, *courseCode(FK)*)
Statuses(credits, *matriculationNo (FK)*)

or even like this, as the FK can be in either table in a one-to-one relationship

Students(matriculationNo, name, address, dateOfBirth, email, *courseCode(FK)*, *statusID(FK)*) - note that statusID needs a UNIQUE constraint
Statuses(statusID, credits)

Which do you think is the best solution?

Question 2

1. The following tables are part of an IT helpdesk database:

   Tickets(ticketID, dateOpened, dateClosed, status,
   description, *administratorID(FK)*)
   Administrators(administratorID, name, grade)

   A call to the helpdesk causes a ticket to be created, and the ticket is assigned as soon as an administrator becomes available.

   Should the foreign key field in Tickets be defined as NOT NULL or NULL? Why?

NULL – Ticket needs to be able to be created without assigning it to an Administrator immediately

2. These tables are part of a library database.

   Books(bookID, title, author)
   Borrowers(borrowerID, name, address, phoneNumber)
   Loans(loanDate, *bookID(FK)*, *borrowerID(FK)*)

   Should the foreign key fields in Loans be defined as NOT NULL or NULL? Why?

NOT NULL – need to specify borrower and book when loan is created

Question 3

The following tables were used in your notes to show a one-to-one relationship. Each department has one manager, who is an employee, and each employee can only manage one department.

Departments(departmentID, *manager(FK)*)
Employees(employeeID, firstname, lastname)

foreign key *manager* references employeeID

However, each employee must also be part of a department.

How would you modify these tables to model this? What kind of relationship is this?

Employees(employeeID, firstname, lastname, *departmentID(FK)*) - one-to-many