Xi'an Jiaotong-Liverpool University
西交利物浦大學

Entity-Relationship Diagrams

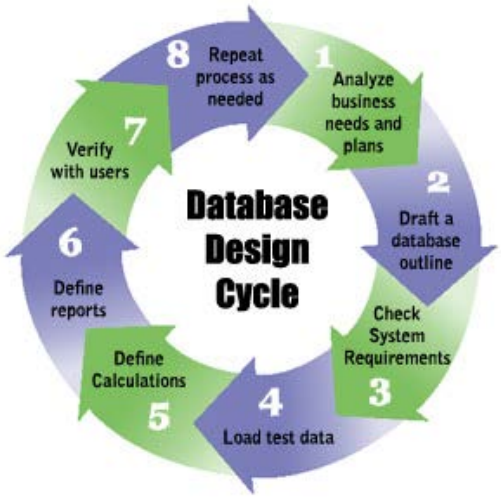
Jianjun Chen

Database Design

- This lecture introduces the technique to design a database from a piece of written requirements.

- Designing your database is important
  - Often results in a more efficient and simpler queries once the database has been created.
  - May help reduce data redundancy in the tables.

- Need to consider
  - What is the database going to be used for?
  - What tables, attributes, keys are needed?



Entity-Relationship Modelling

- ER Modelling is used for conceptual design, it consists of three types of components:
  - **Entities**: objects or items of interest.
  - **Attributes**: properties of an entity.
  - **Relationships**: links between entities.

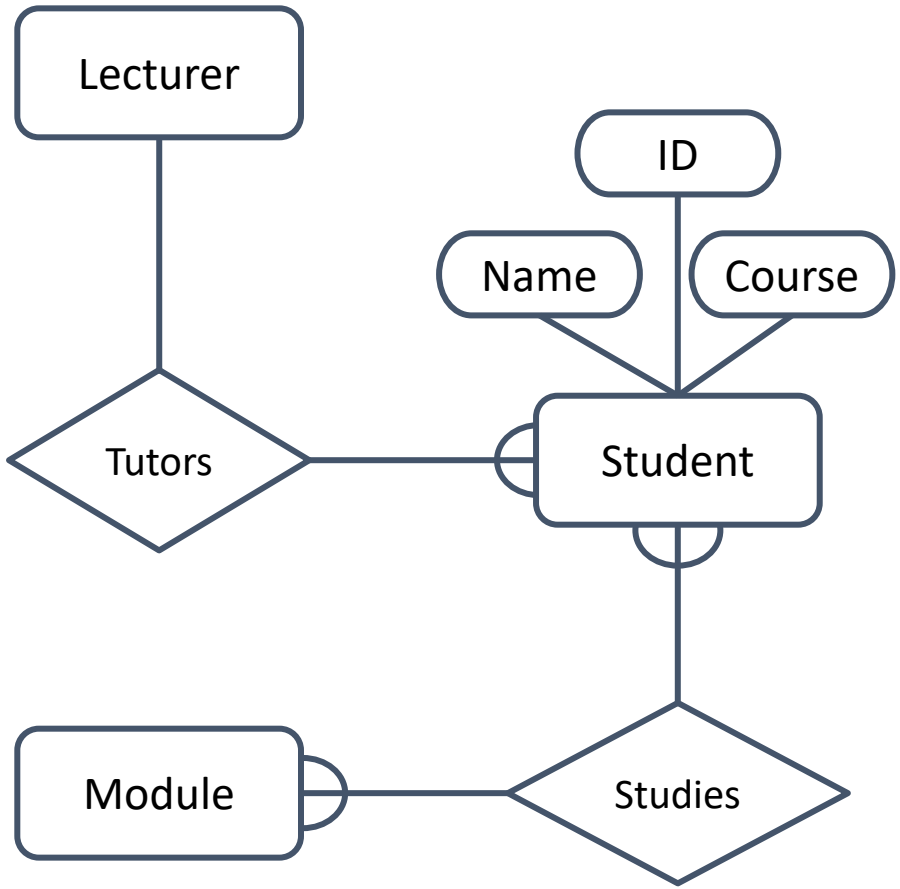
- For example, in a University database we might have entities for Students, Modules and Lecturers
  - Students might have attributes such as their ID, Name, and Course
  - Students could have relationships with Modules (enrolment) and Lecturers (tutor/tutee)

- What are the corresponding elements in a real database? ☺

Entity-Relationship Diagrams

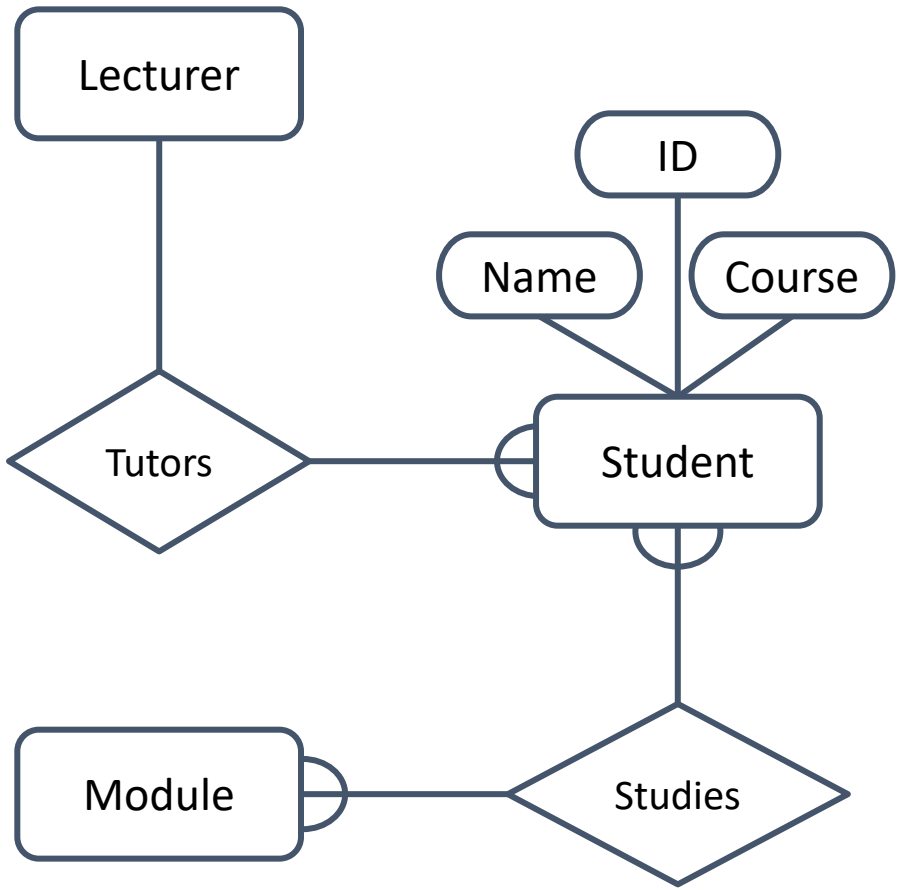
ER Models are often represented as ER diagrams that

- Give a conceptual view of the database
- Are independent of the choice of DBMS
- Can identify some problems in a design



ER: Diagram Conventions

- There are various notations for representing ER diagrams
- These specify the shape of the various components, and the notation used to represent relationships
- For this introductory module, we will use simplified notation



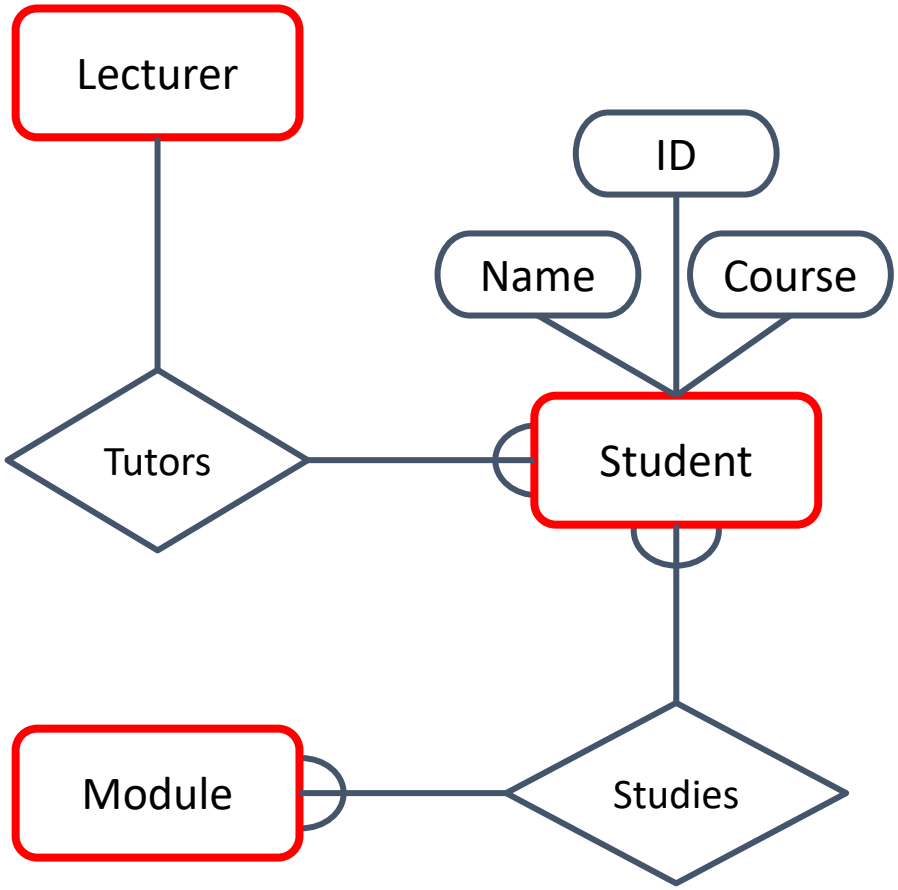
Entities

- **Entities** represent objects or things of interest
  - Physical things like students, lecturers, employees, products
  - More abstract things like modules, orders, courses, projects

- Each entity:
  - Is a general type or class, such as Lecturer or Module
  - Has instances of that particular type. E.g. DBI and IAI are instances of Module
  - Has attributes (such as name, email address)

Entities

- In ER Diagrams, we will represent Entities as boxes with rounded corners
- The box is labelled with the name of the class of objects represented by that entity



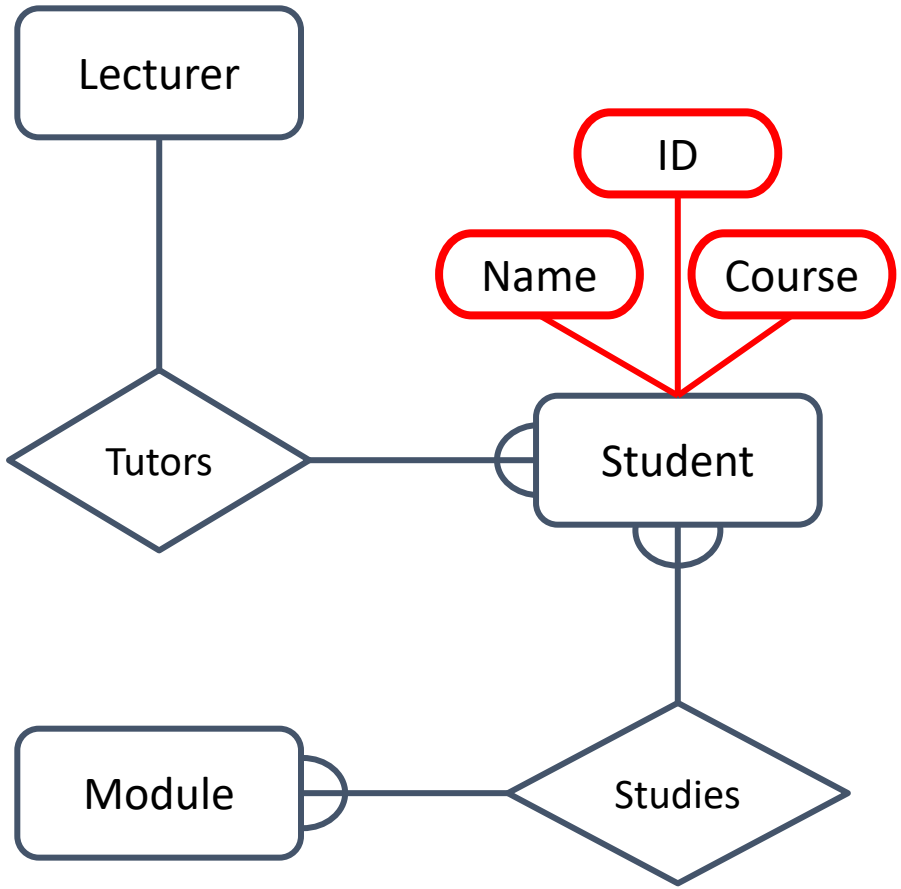
Attributes

- Attributes are facts, aspects, properties, or details about an entity
  - Students have IDs, names, courses, addresses, …
  - Modules have codes, titles, credit weights, levels, …

- Each attribute have:
  - A name
  - An associated entity
  - Domains of possible values
  - For each instance of the associated entity, a value from the attributes domain

Attributes

- In an ER Diagram attributes are drawn as ovals
- Each attribute is linked to its entity by a line
- The name of the attribute is written in the oval



Relationships

- A relationship is an association between two or more entities
  - Each Student takes several Modules.
  - Each Module is taught by a Lecturer.
  - Each Employee works for a single Department.

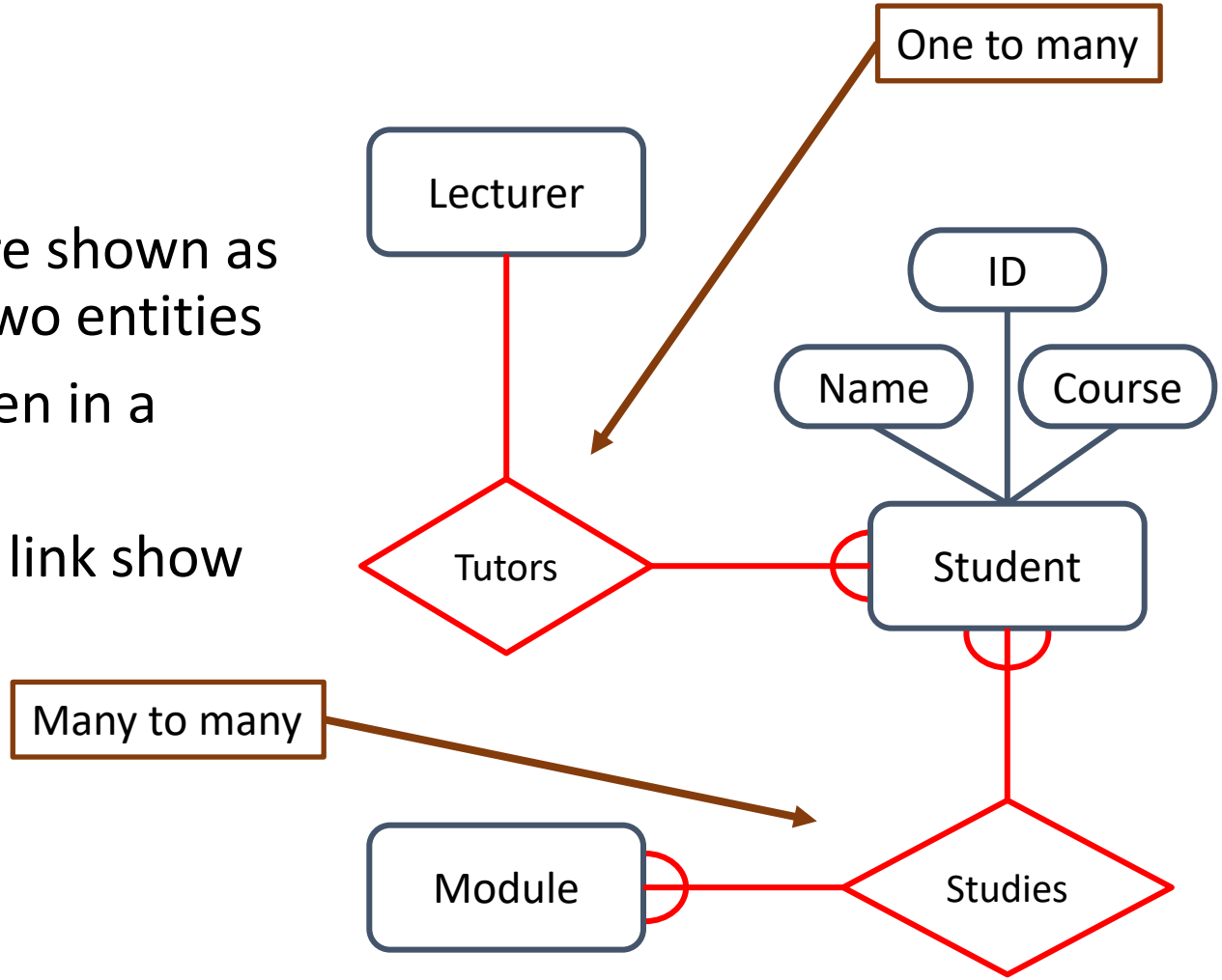
- Relationships have
  - A name.
  - A set of entities that participate in them.
  - A degree: the number of entities that participate (usually 2).
  - A cardinality ratio.

Cardinality Ratios

- One to one
  - Written as (1:1).
  - Each lecturer has a unique office & offices are single occupancy
- One to many
  - Written as (1:M) or (1:*)
  - A lecturer may tutor many students, but each student has just one tutor
- Many to many
  - Written as (M:M) or (M:N) or (*:*)
  - Each student takes several modules, and each module is taken by several students

Relationships

- Relationships are shown as links between two entities
- The name is given in a diamond box
- The ends of the link show cardinality



More about Cardinality Ratios

- One to Many
- **staff ---- branch**: Many staff members can be assigned to a single branch.
- How is this relationship reflected in database tables?

Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

More about Cardinality Ratios

- Many rows of staff can map to a single branch record.
- Each row of staff corresponds to the information of a staff in real life.
  - Same for branch.
- In the staff table, B003 appears several times with SG37, SG14 and SG5.

Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

Designing ER Models

Create ER diagram from simple real word requirements

Designing ER Models

- To make an ER model you need to identify:
  - Entities
  - Attributes
  - Relationships and Cardinality ratios
- We obtain these from a problem description
- General guidelines
  - Since entities are things or objects they are often nouns in the description
  - Attributes are facts or properties, and so are often nouns also
  - Verbs often describe relationships between entities

Example 1

A university consists of a number of departments. Each department offers several courses. A number of modules make up each course. Students enroll in a particular course and take modules towards the completion of that course. Each module is taught by a lecturer from the appropriate department (several lecturers work in the same department), and each lecturer tutors a group of students. A lecturer can teach more than one module but can work only in one department.

Entity, Attributes, Relationships: What shall be identified first? Followed by what?....

Example 1: Entities

A university consists of a number of departments. Each department offers several courses. A number of modules make up each course. Students enroll in a particular course and take modules towards the completion of that course. Each module is taught by a lecturer from the appropriate department (several lecturers work in the same department), and each lecturer tutors a group of students. A lecturer can teach more than one module but can work only in one department.

Entities – Department, Course, Module, Student, Lecturer

Example 1: Relationships

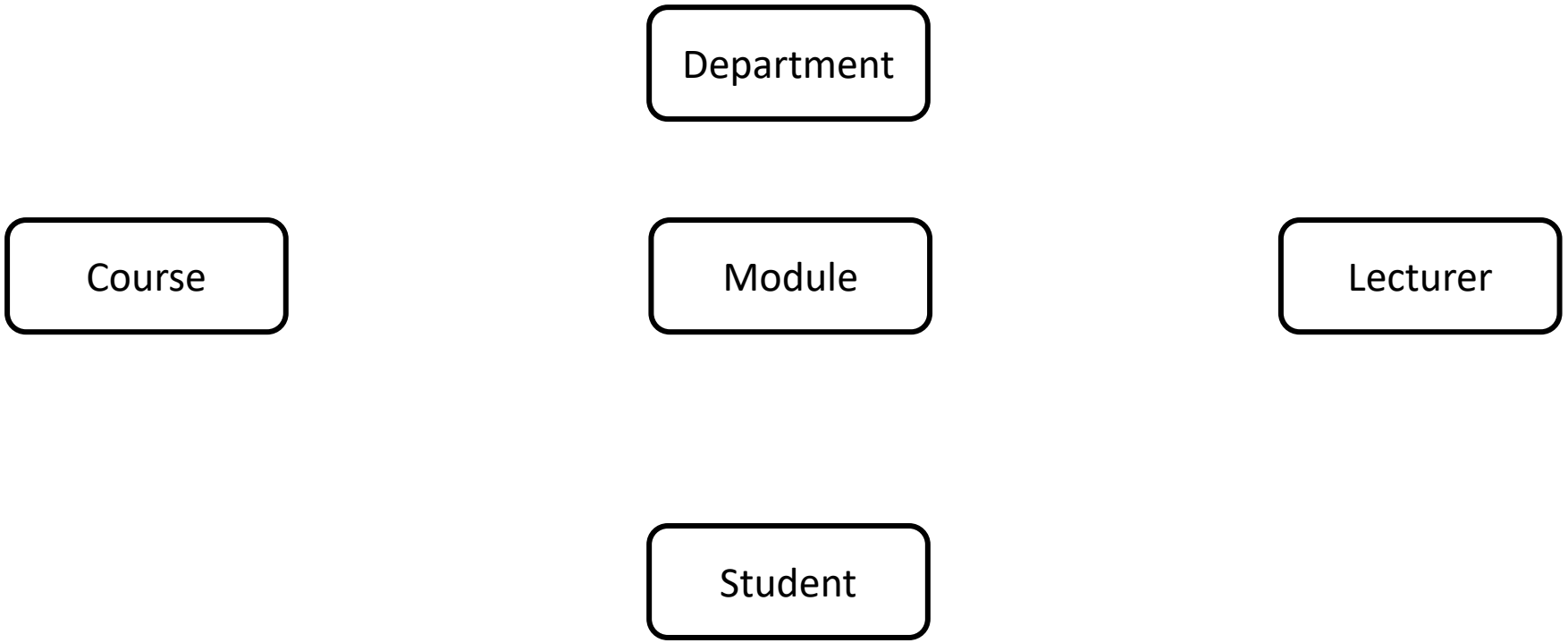
A university consists of a number of departments. Each department offers several courses. A number of modules make up each course. Students enroll in a particular course and take modules towards the completion of that course. Each module is taught by a lecturer from the appropriate department (several lecturers work in the same department), and each lecturer tutors a group of students. A lecturer can teach more than one module but can work only in one department.

Entities – Department, Course, Module, Student, Lecturer

Relationships – Offers, Make Up, Enroll, Take, Taught By, Work in, Tutors

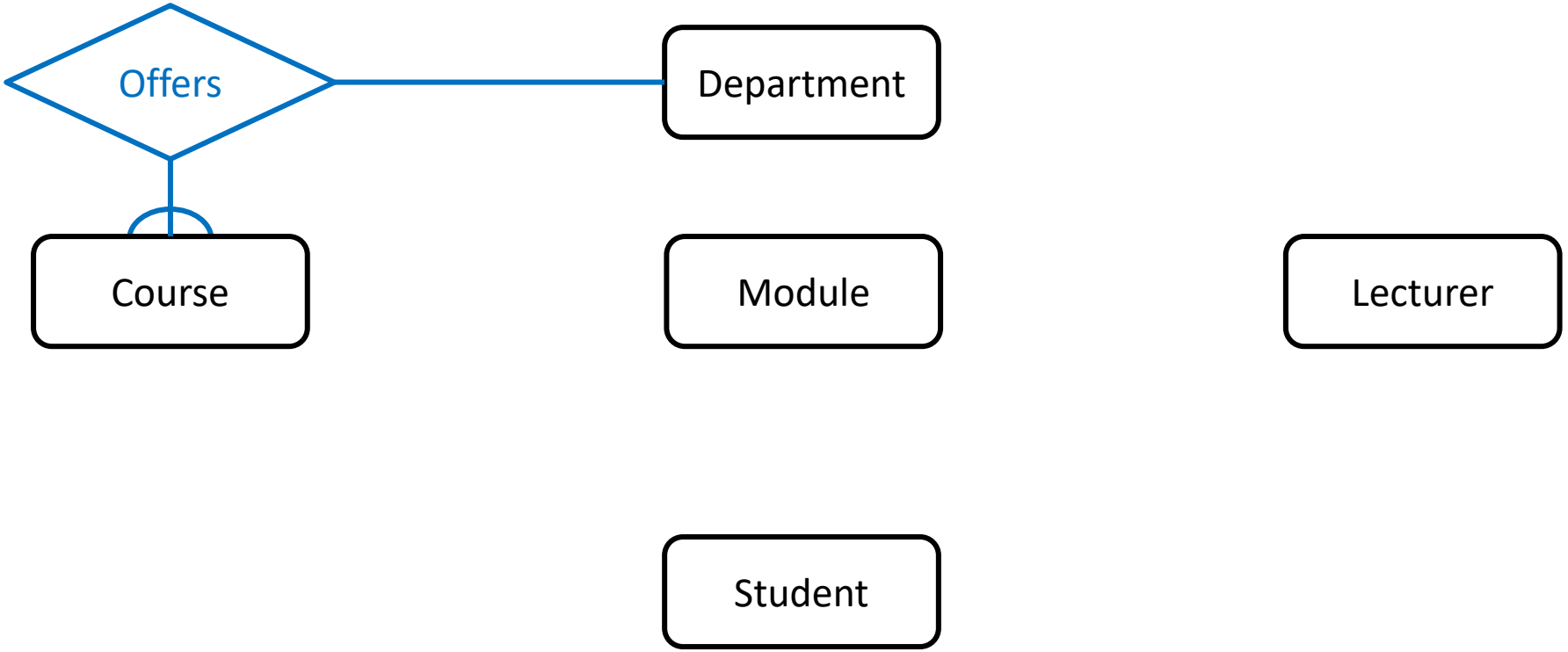
Entities in ER Diagram

Entities – Department, Course, Module, Student, Lecturer



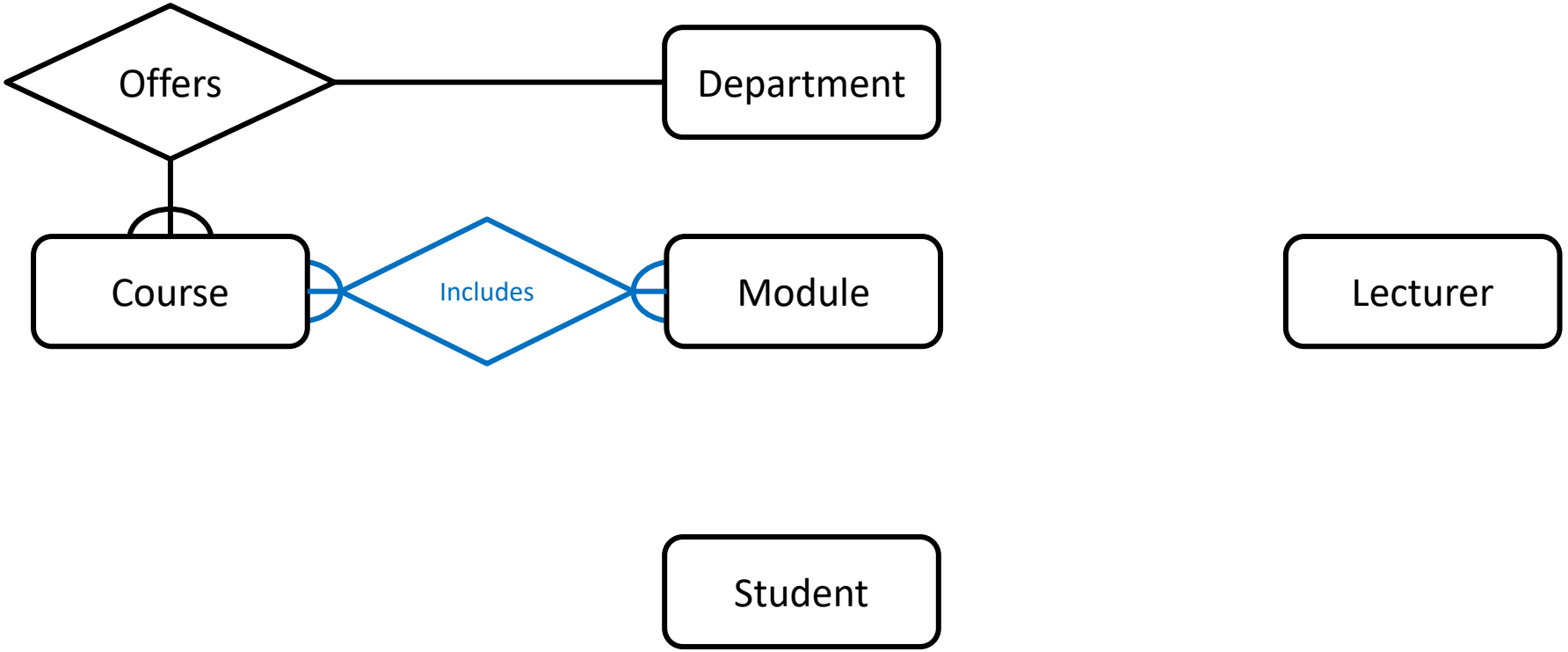
Relationships in ER Diagram

Each Department offers several Courses



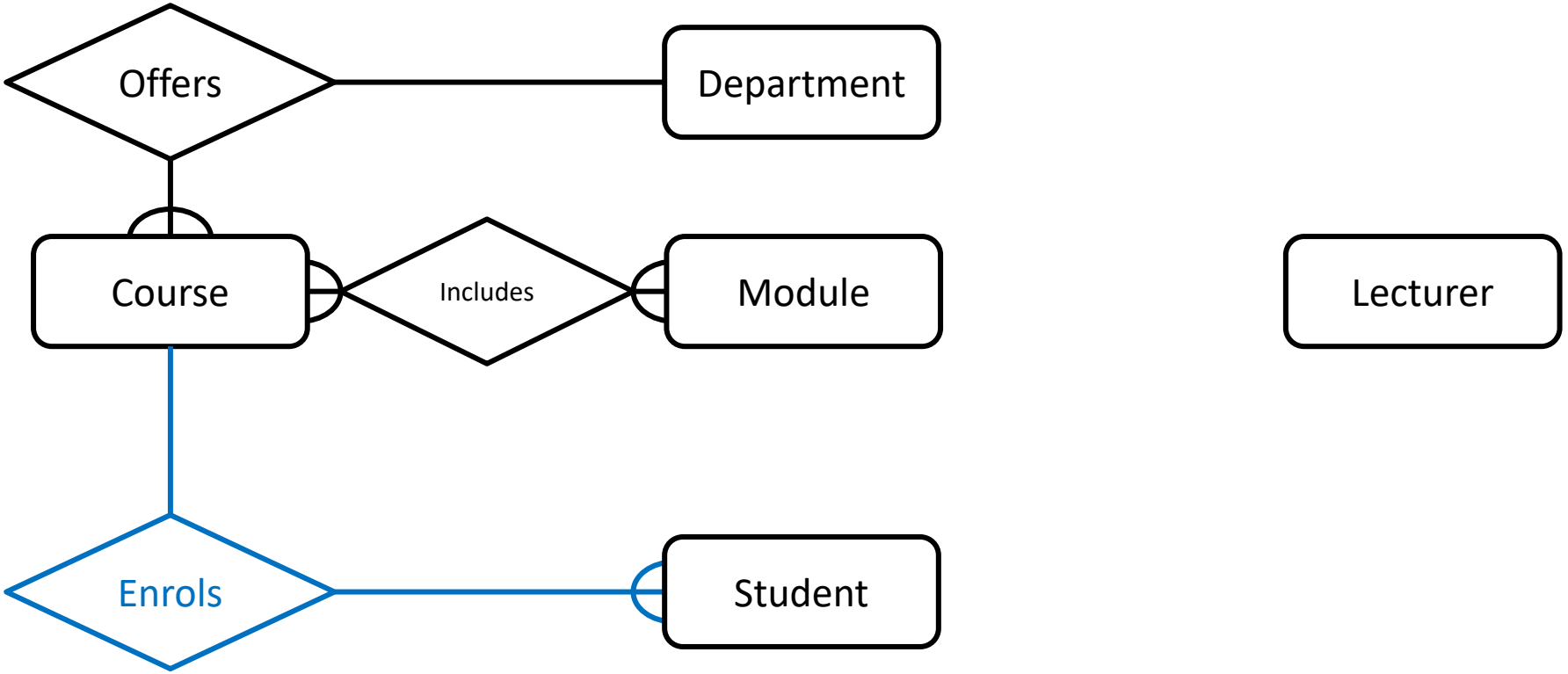
Relationships in ER Diagram

A number of modules make up each Course



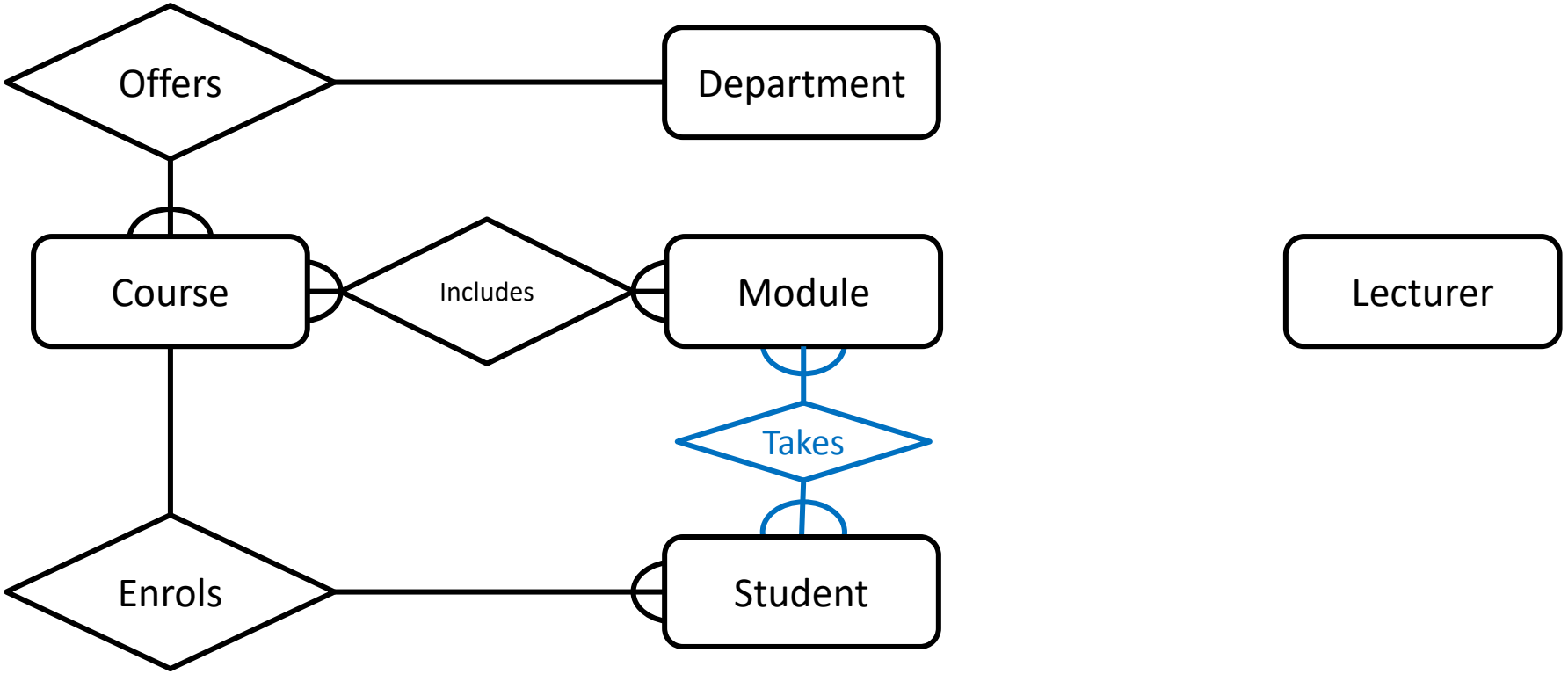
Relationships in ER Diagram

Students enroll in a particular course



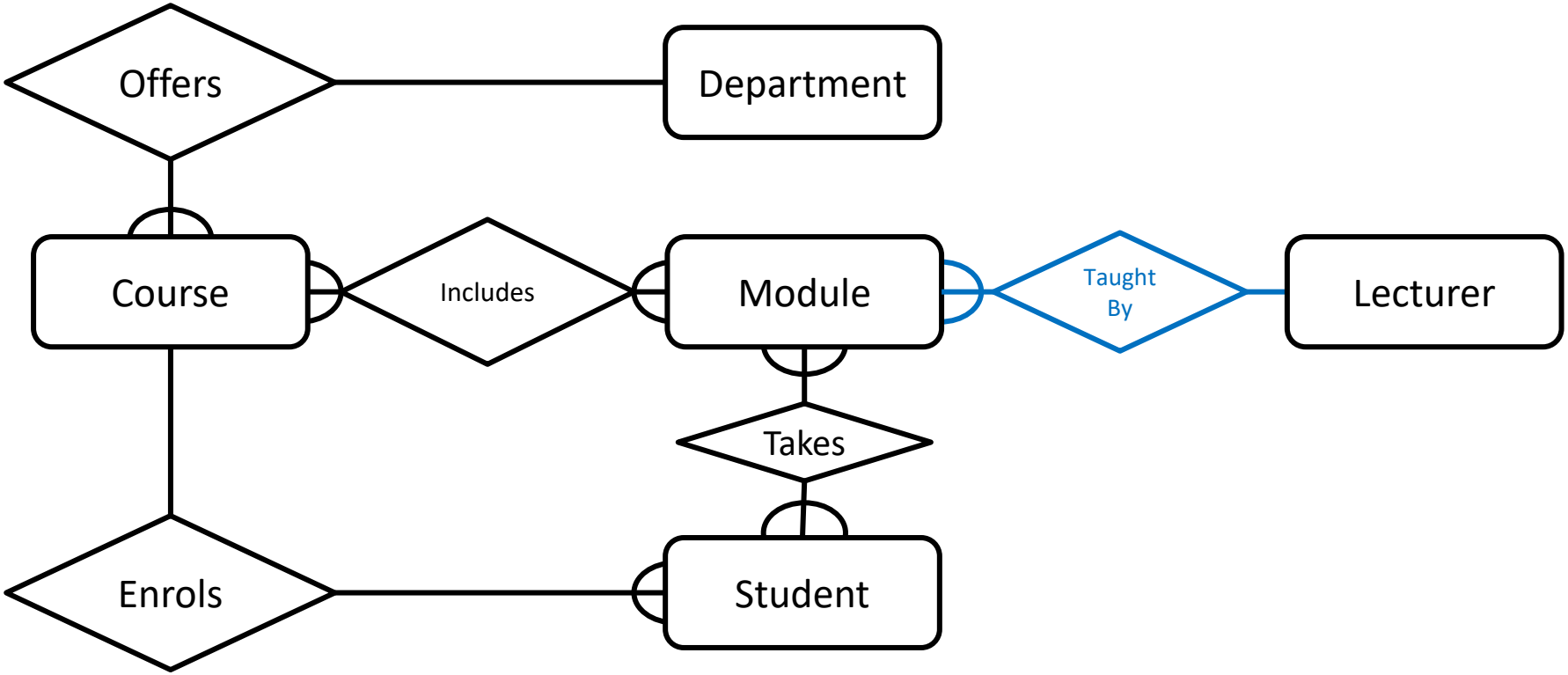
Relationships in ER Diagram

Students take several modules



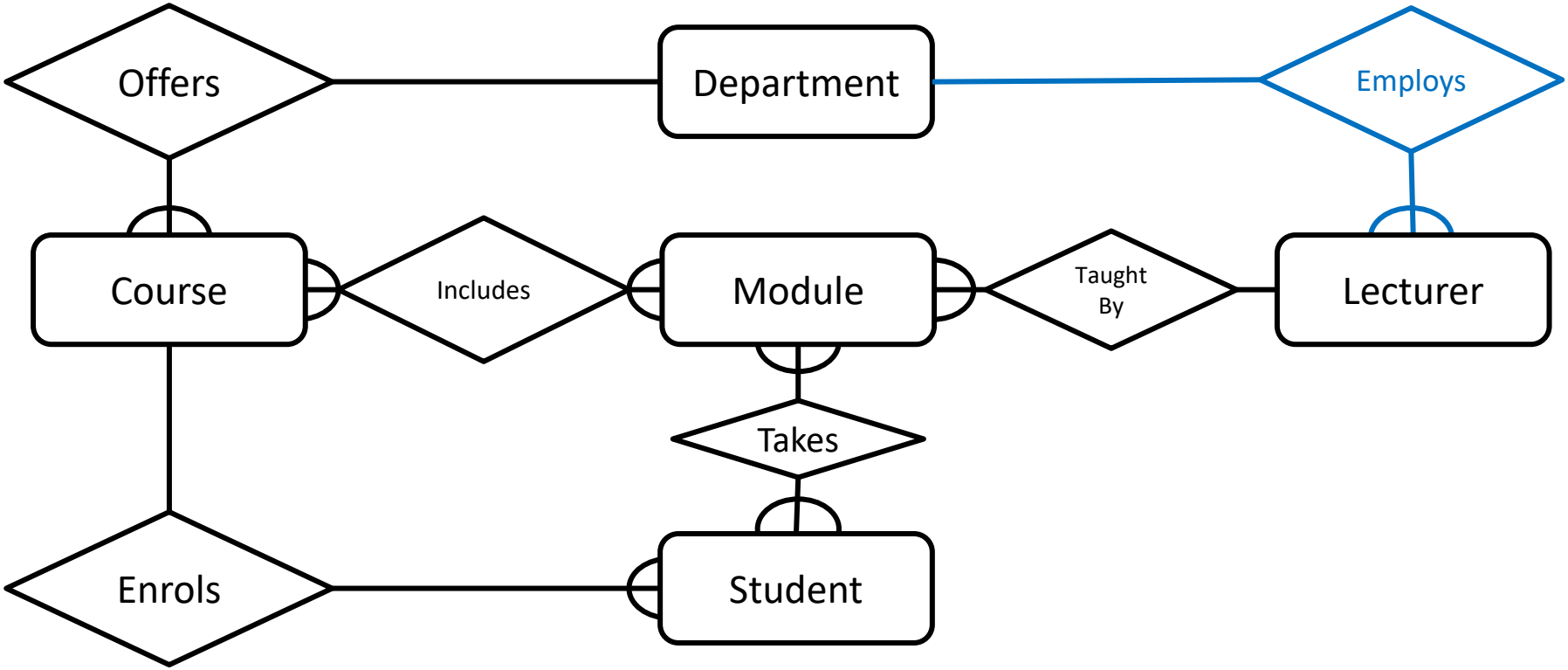
Relationships in ER Diagram

A lecturer can teach more than one module



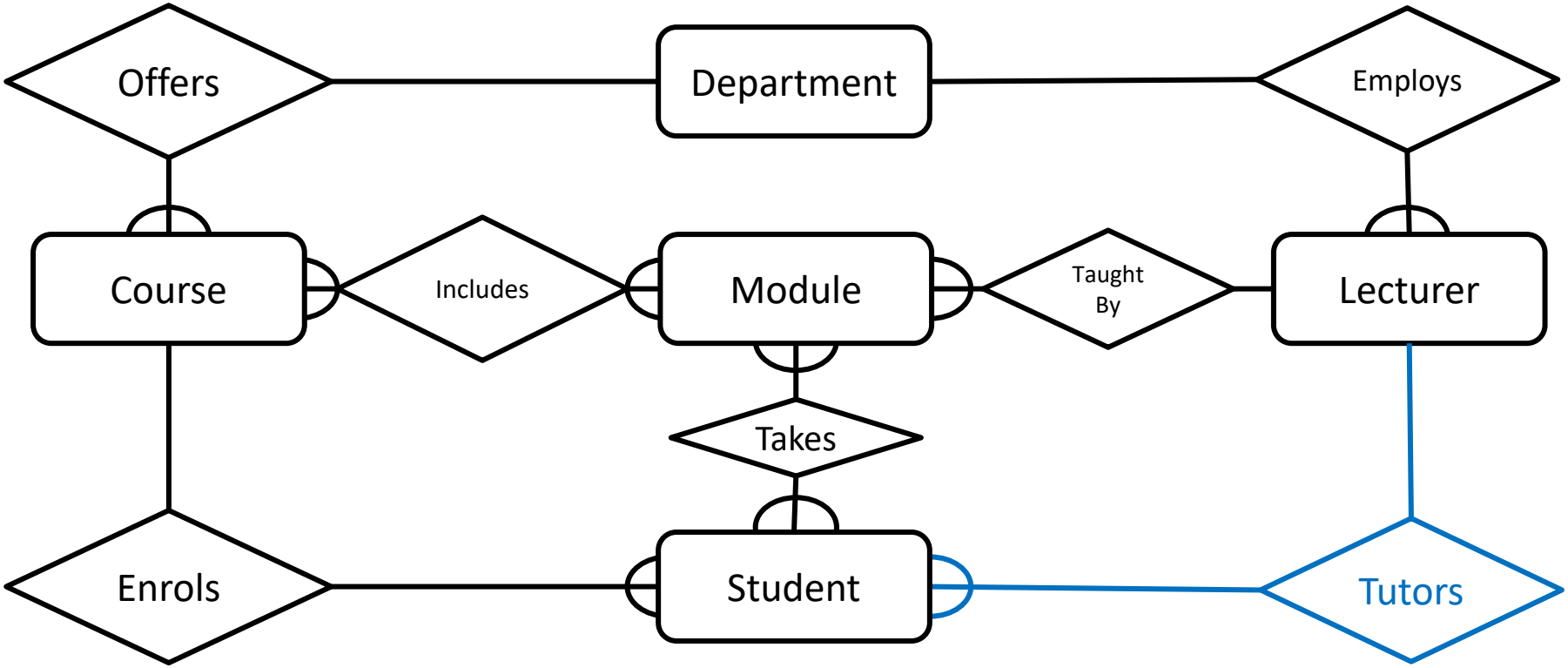
Relationships in ER Diagram

Each department employs a number of lecturers



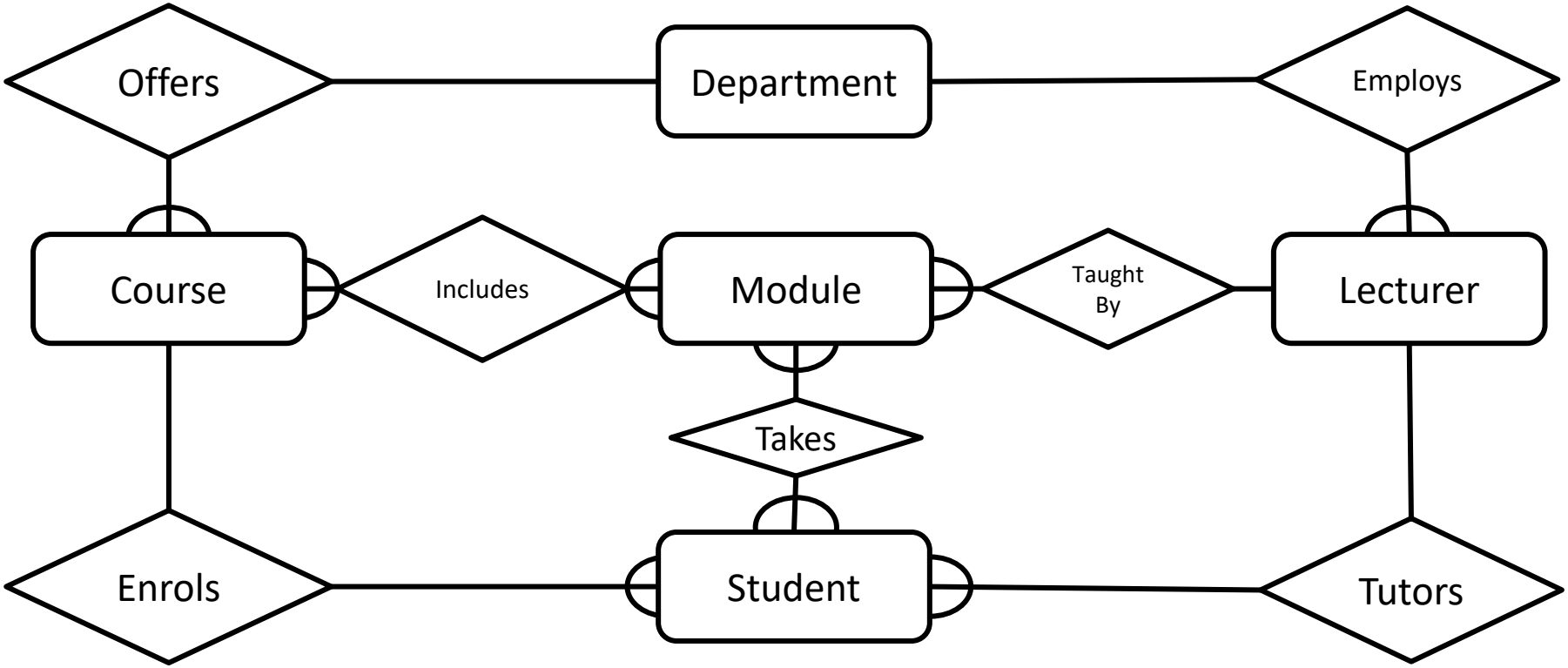
Relationships in ER Diagram

Each Lecturer tutors a number of Students



The Complete ER Diagram

- The completed diagram. All that remains is to remove M:M relationships



Issue: M:M Relationships

- M:M relationships are difficult to represent in a database:

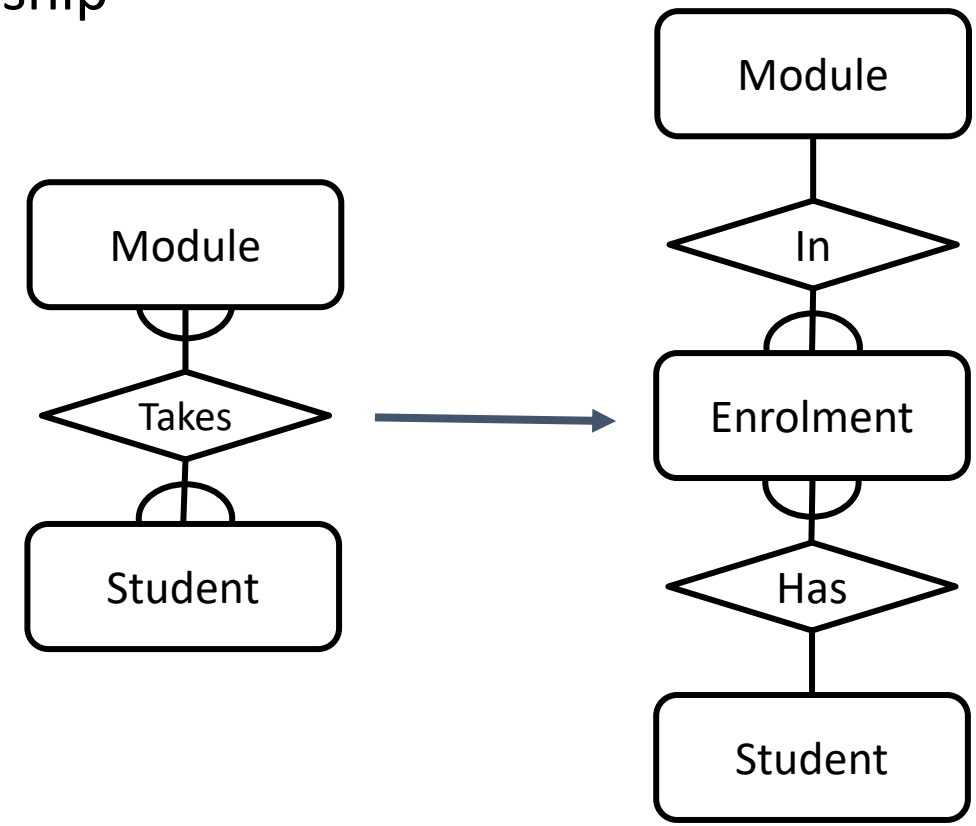
Student (design 1)		
SID	**sName**	**sMod**
1001	Jack Smith	DBI
1001	Jack Smith	PRG
1001	Jack Smith	IAI
1002	Anne Jones	PRG
1002	Anne Jones	IAI
1002	Anne Jones	Vis

Module	
MID	**mName**
DBI	Databases and Interfaces
PRG	Programming
IAI	AI
VIS	Computer Vision

Student (design 2)		
SID	**sName**	**sMod**
1001	Jack Smith	DBI, PRG, IAI
1002	Anne Jones	VIS, IAI, PRG

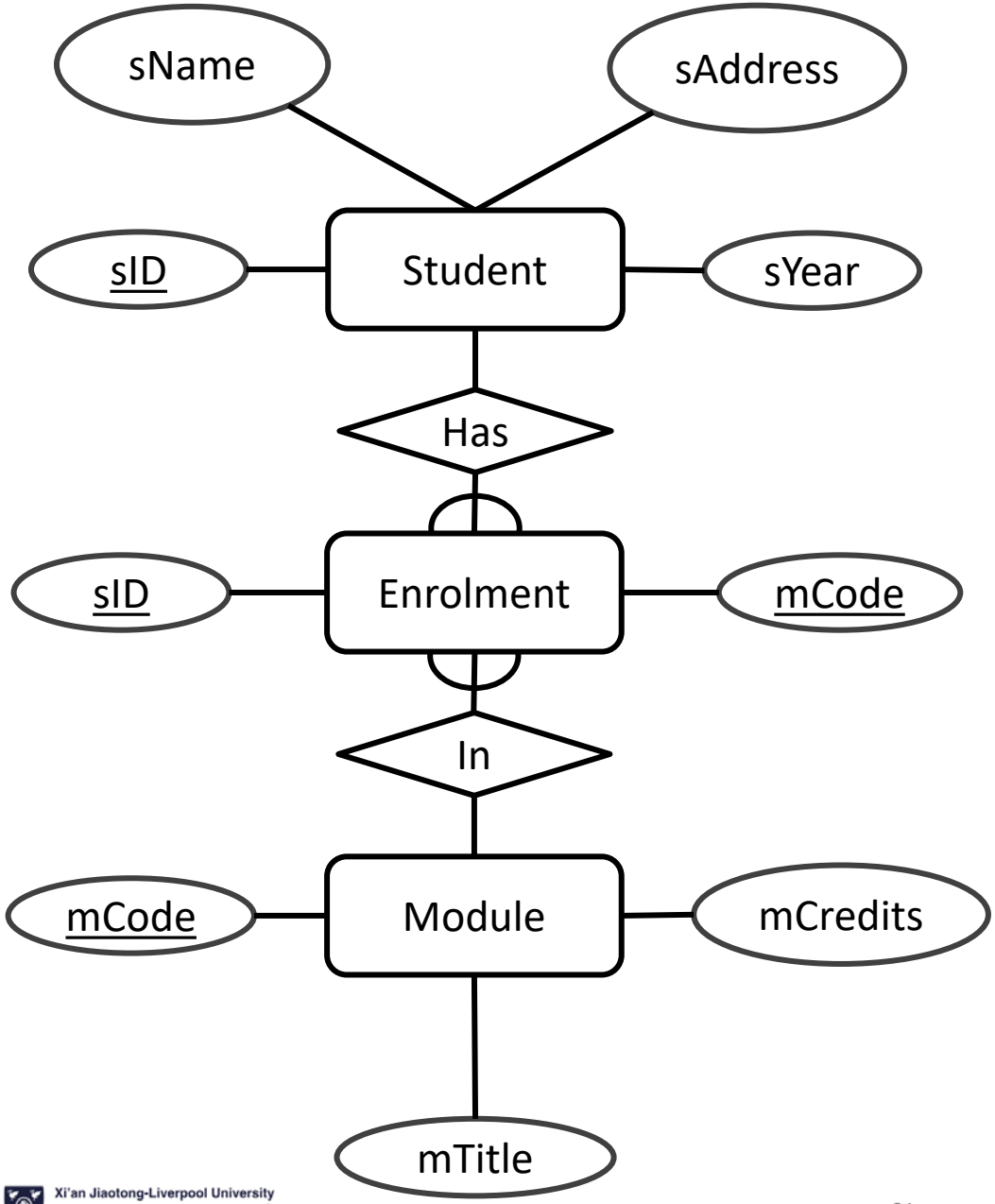
Removing M:M Relationships

- We can split a M:M relationship into two 1:M relationships.
- An additional entity is created to represent the M:M relationship



Removing M:M …

- The Enrolment table
  - Will have columns for the student ID and module code attributes
  - Will have a foreign key to Student for the ‘has’ relationship
  - Will have a foreign key to Module for the ‘in’ relationship



Entities and Attributes

- Sometimes it is hard to tell if something should be an entity or an attribute
  - They both represent objects or facts about the world
  - They are both often represented by nouns in descriptions

- General guidelines
  - Entities can have attributes but attributes have no smaller parts
  - Entities can have relationships between them, but an attribute belongs to a single entity

Example 2

We want to represent information about products in a database. Each product has a description, a price and a supplier. Suppliers have addresses, phone numbers, and names. Each address is made up of a street address, a city name, and a postcode.

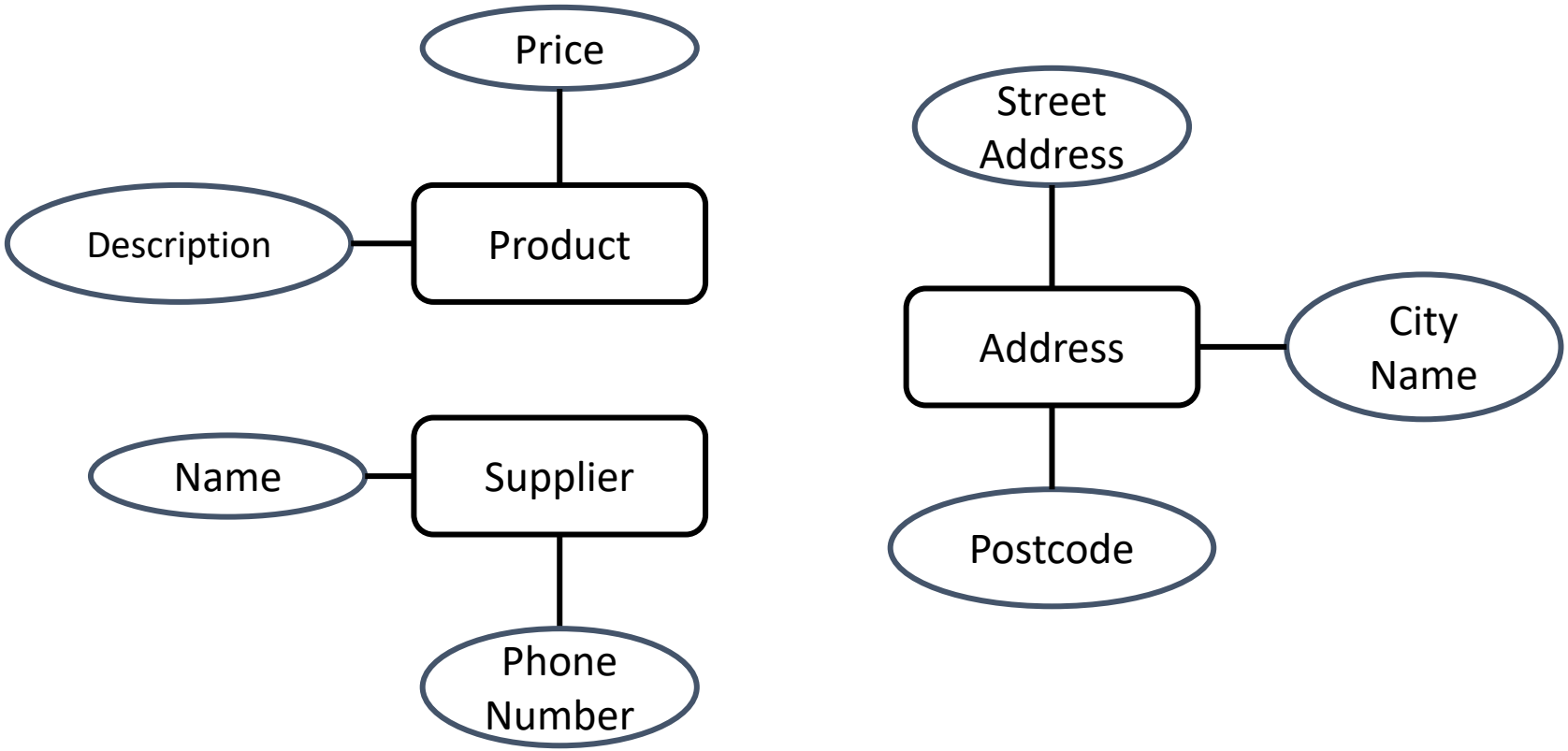
Example 2: Entities/Attributes

- Entities or attributes:
  - product
  - description
  - price
  - supplier
  - address
  - phone number
  - name
  - street address
  - city name
  - postcode

- Products, suppliers, and addresses all have smaller parts so we make them entities

- The others have no smaller parts and belong to a single entity

Example 2: ER Diagram

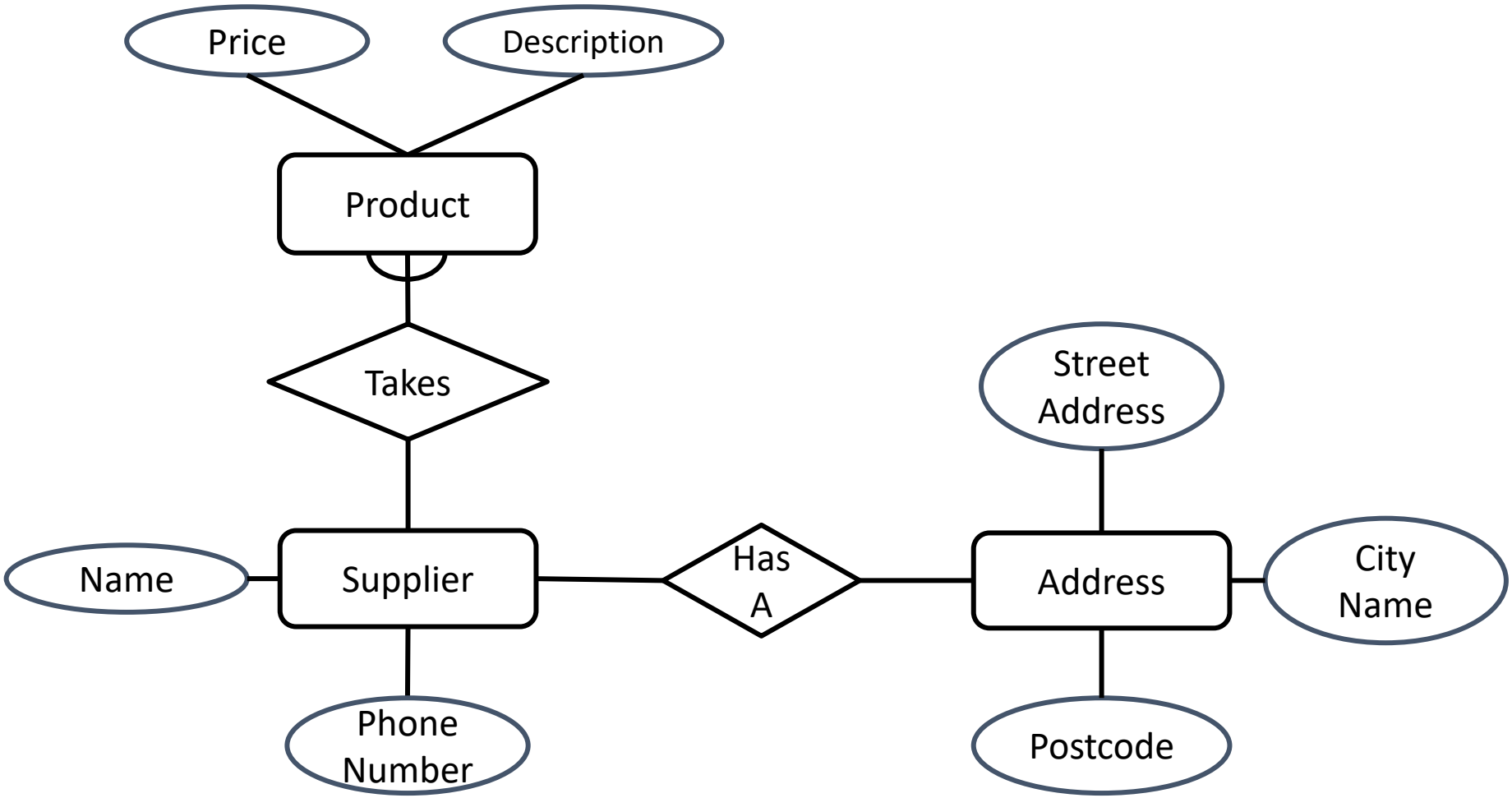


Example 2: Relationships

- Each product has a supplier
  - Each product has a single supplier but there is nothing to stop a supplier supplying many products
  - A many to one relationship

- Each supplier has an address
  - A supplier has a single address
  - It does not seem sensible for two different suppliers to have the same address
  - A one to one relationship

Example 2: Initial ER Diagram



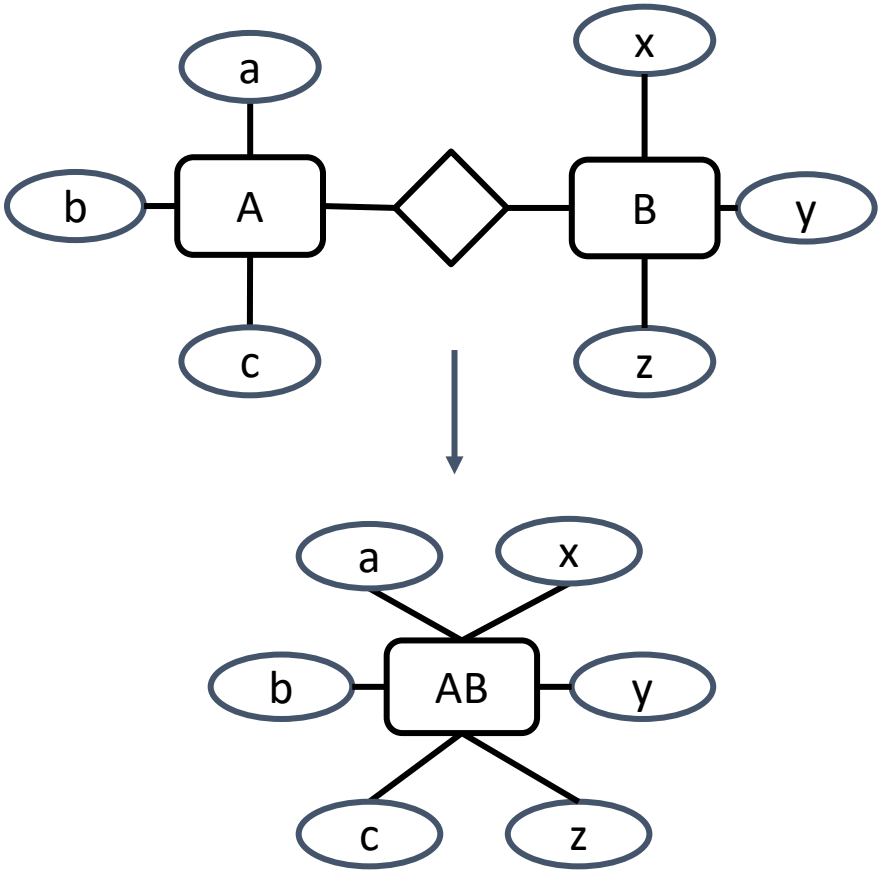
Issue: One-to-One Relationships

- Some relationships between entities, A and B, might be redundant if:
  - It is a 1:1 relationship between A and B
  - Every A is related to a B and every B is related to an A.

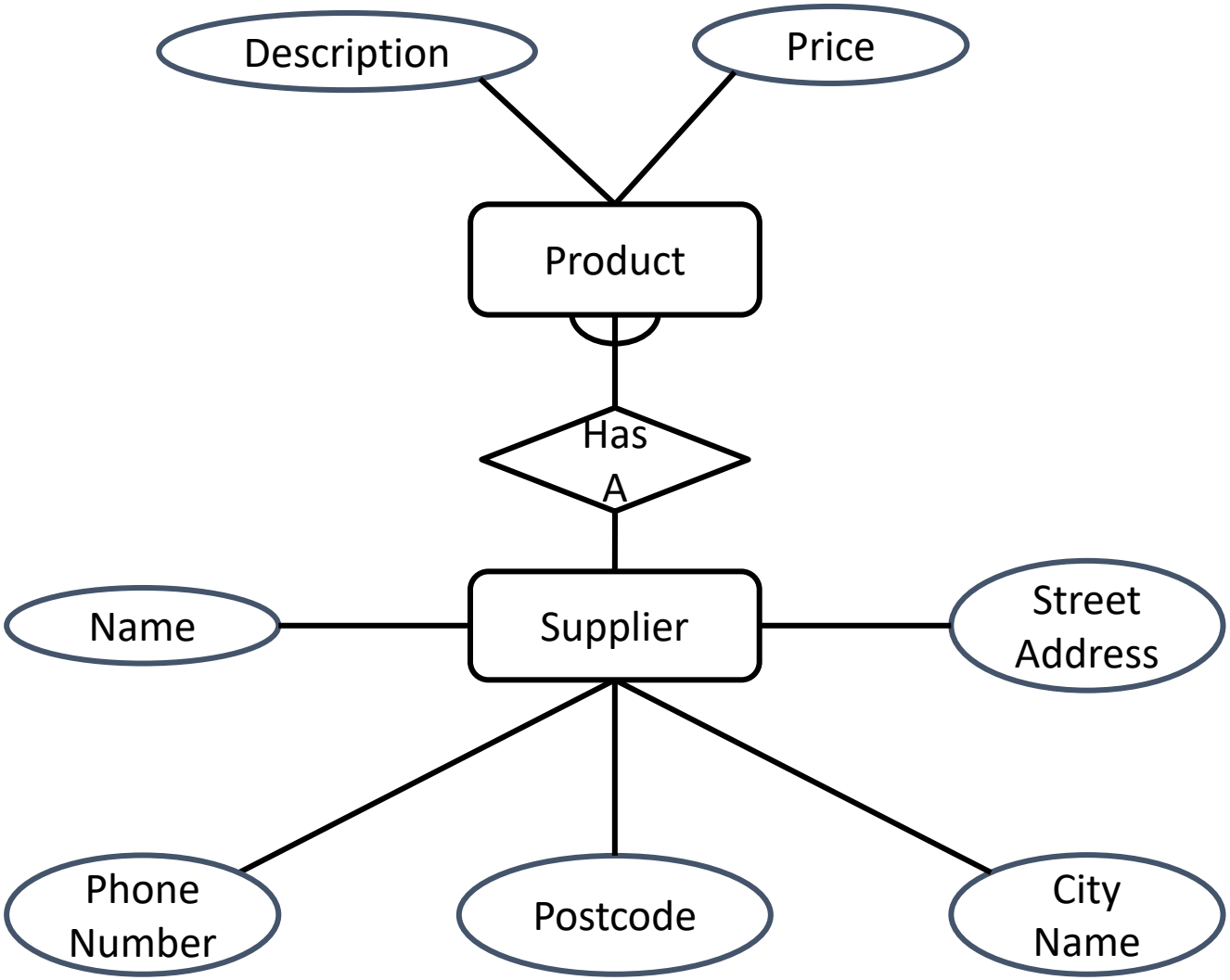
- Example: The supplier-address relationship is one-to-one
  - Every supplier has an address
  - We don’t need addresses that are not related to a supplier

One-to-One Relationships

- We can merge the two entities that take part in a redundant relationship together
  - They become a single entity
  - The new entity has all the attributes of the old ones



Example 2: The Final ER Diagram



ER Diagram: Summary of Steps

1) From a description of the requirements, identify:
   - Entities
   - Attributes
   - Relationships
   - Cardinality ratios of the relationships

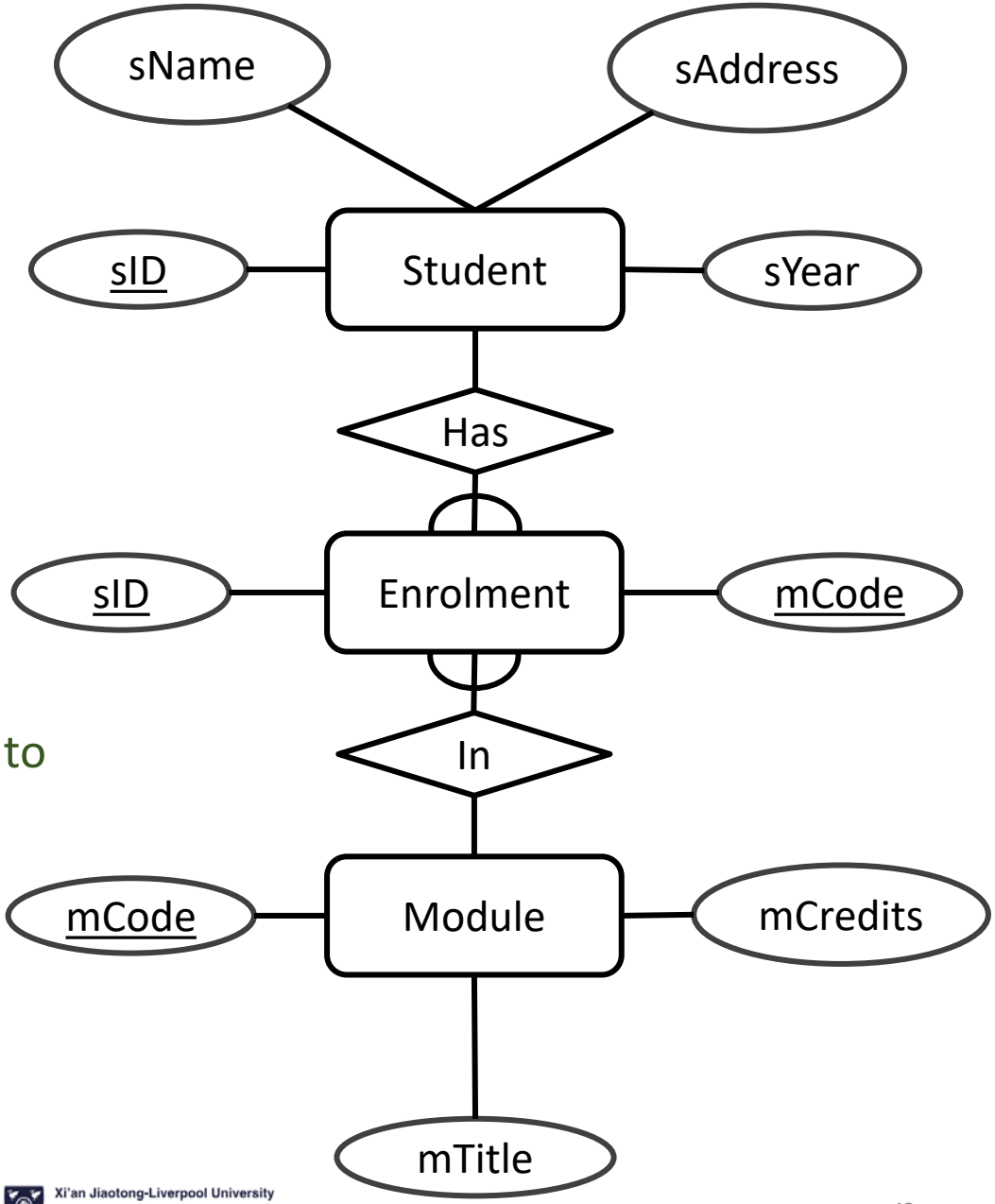
2) Draw the ER diagram and then
   - Look at one to one relationships as they might be redundant
   - Look at many to many relationships as they will often need to be split into two one to many links, using an intermediate entity

From ER Diagram to SQL Tables.

- Entities Become table names.
- Attributes of an entity becomes the columns.
- Relationships become foreign keys.

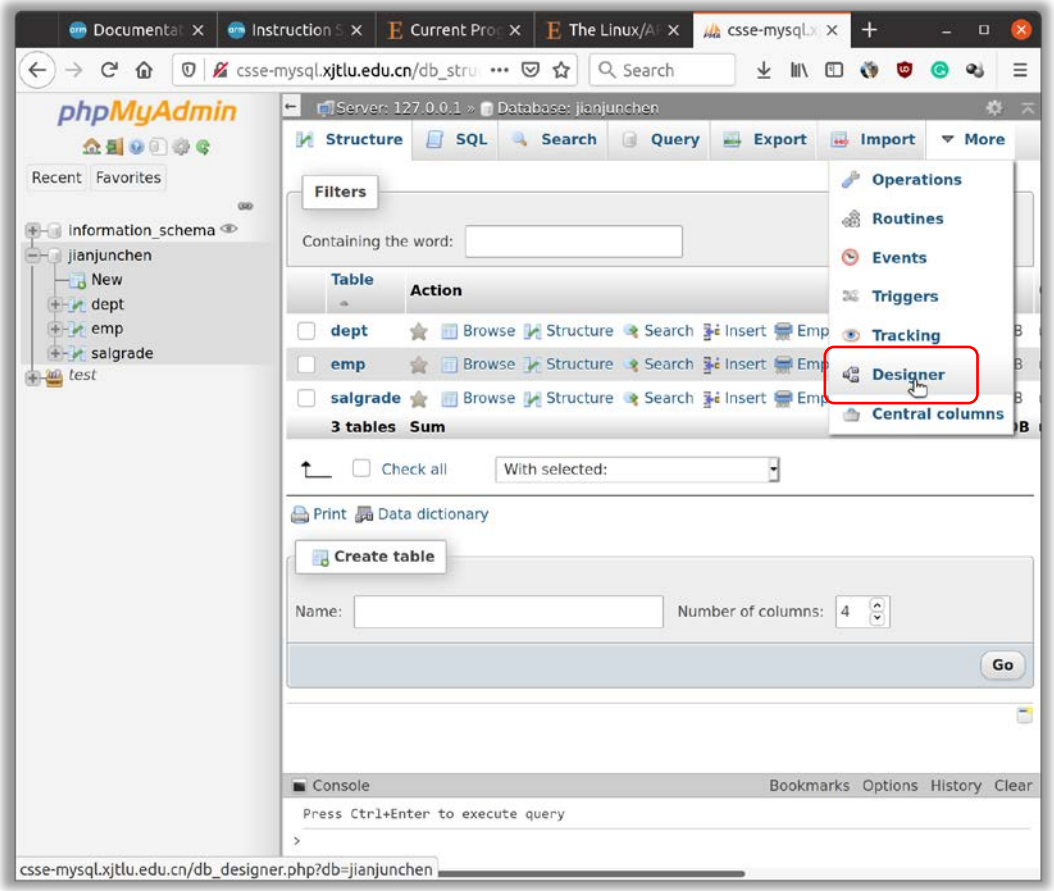
Relationships

- Relationships as Foreign Keys:
  - 1:1 are usually not used, or can be treated as a special case of M:1
  - M:1 are represented as a foreign key from the M-side to the 1.
  - M:M are split into two M:1 relationships.



ER Diagram in SQL

- If you already have an existing database, you can use your favourite database tool to generate the ER diagram.



The Sunny Isle Hotel

Database design example

*and the excellent unnc 2019 spring semester class

Step by Step: Entities and Attributes

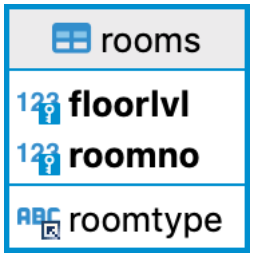
- "The application must allow a guest to book rooms or cancel booked rooms."
  - Tables: Guest, Room
  - Relationship: Book

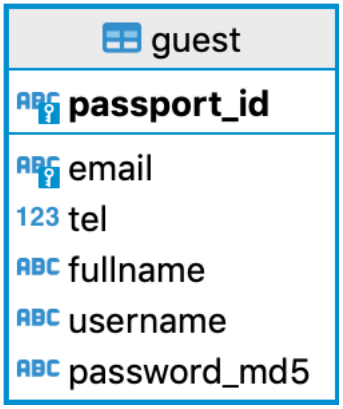
- Attributes of Guest?
  - "... guest will have a username, guest's real name, passport ID, telephone number and email address"
  - "To log in, a guest is required to input his username and password correctly."

- Attributes of Room?
  - "Rooms in the Sunny Isle Hotel have four different types"
  - "The room number..."

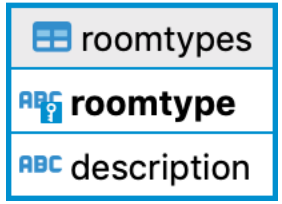
Step by Step: Entities and Attributes

- The design here looks okay.





- The values for roomtype belong to a set.
  - Large room with double beds.
  - Large room with a large single bed.
  - Small room with a single bed.
  - VIP room.
- You can either use domain to limit its values,
- Or add a new table `roomtypes` and reference to it.



Step by Step: Relationship

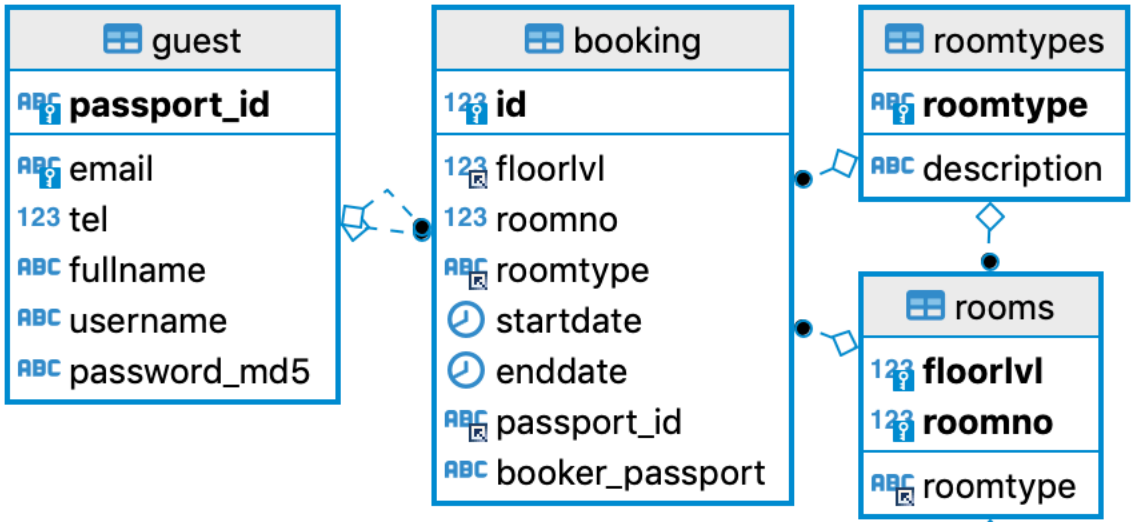
- Guests book rooms:
  - One guest can book multiple rooms?
  - A room can be booked by multiple guests?

- It is an ____ relationship.
  - M:M
  - 1:1
  - 1:M

- What should be the final design of “guest-book-room”?

Step by Step: Relationship

- For an M:M relationship, it is good idea to use an additional table.
  - (floorlvl, roomno) references room.
  - Passport_id references guest.
  - Booker_passport also references guest.



Step by Step: Relationship, a Closer Look

```
create table booking (
    id int primary key,
    floorlvl int,
    roomno int,
    constraint fk_booking_room foreign key
    (floorlvl, roomno) references rooms (floorlvl, roomno),

    roomtype varchar(255) not null
        references roomtypes (roomtype),
    startdate date not null,
    enddate date not null,
    passport_id varchar(255) not null
        references guest (passport_id),
    booker_passport varchar(255)
        references guest (passport_id)
);
```

In real life, guests can check in when they arrive at the hotel, room numbers are decided at that moment. So floorlvl and roomno can be null.

A room type must be selected when booking

A guest can book multiple rooms for himself and his friends.

Design Issues

- How about the design of the “rooms” below? Any issues?

rooms

floorlvl	roomno	booked
10	02	true

- Surrogate key VS Natural Key:
  - https://www.mssqltips.com/sqlservertip/5431/surrogate-key-vs-natural-key-differences-and-when-to-use-in-sql-server/
  - The examples in this module prefers natural keys. But feel free to make your own decisions.

Questions?