Entity Relationship Modelling

G51DBI – Databases and Interfaces

Yorgos Tzimiropoulos
yorgos.tzimiropoulos@nottingham.ac.uk

This Lecture

A university consists of a number of departments. Each department offers several courses. A number of modules make up each course. Students enrol 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 lecturer 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, Enrol, Take, Taught By, From The, Tutors

This Lecture

- Entity/Relationship models
  - Entities and Attributes
  - Relationships
  - E/R Diagrams
- Further Reading
  - Database Systems, Connolly & Begg, Chapter 12
  - The Manga Guide to Databases, Chapter 3

Database Design

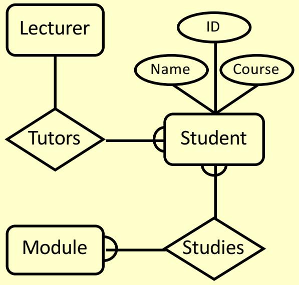
- Designing your database is important
  - We can create a database design that is independent of DBMS
  - Often results in a more efficient and simpler queries once the database has been created
- Before we look at how to create and use a database we'll look at how to design one
- Need to consider
  - What tables, keys, and constraints are needed?
  - What is the database going to be used for?

Entity/Relationship Modelling

- E/R Modelling is used for conceptual design
  - 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)

Entity/Relationship Diagrams

- E/R Models are often represented as E/R diagrams that
  - Give a conceptual view of the database
  - Are independent of the choice of DBMS
  - Can identify some problems in a design

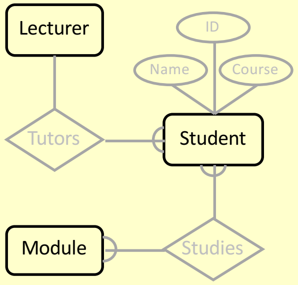


Entities

- Entities represent objects or things of interest
  - Physical things like students, lecturers, employees, products
  - More abstract things like modules, orders, courses, projects
- Entities have
  - A general type or class, such as Lecturer or Module
  - Instances of that particular type. E.g. Boriana Koleva, Steve Bagley are instances of Lecturer
  - Attributes (such as name, email address)

Diagramming Entities

- In E/R 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

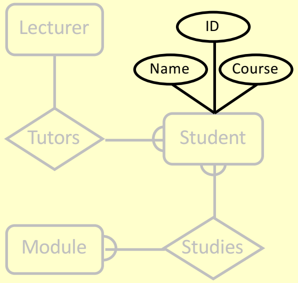


Attributes

- Attributes are facts, aspects, properties, or details about an entity
  - Students have IDs, names, courses, addresses, …
  - Modules have codes, titles, credit weights, levels, …
- Attributes have
  - A name
  - An associated entity
  - Domains of possible values
  - For each instance of the associated entity, a value from the attributes domain

Diagramming Attributes

- In an E/R 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

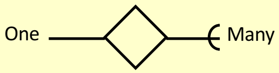
- Relationships are an association between two or more entities
  - Each Student takes several Modules
  - Each Module is taught by a Lecturer
  - Each Employee works for a single Department
- Relationships have
  - A name
  - A set of entities that participate in them
  - A degree - the number of entities that participate (most have degree 2)
  - A cardinality ratio

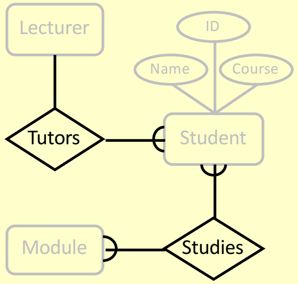
Cardinality Ratios

- Each entity in a relationship can participate in zero, one, or more than one instances of that relationship
- We won’t be dealing with optional (zero instances) of relationships
- This leads to 3 types of relationship...
- One to one (1:1)
  - Each lecturer has a unique office & offices are single occupancy
- One to many (1:M)
  - A lecturer may tutor many students, but each student has just one tutor
- Many to many (M:M)
  - Each student takes several modules, and each module is taken by several students

Entity/Relationship Diagrams

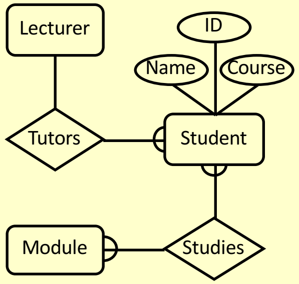
- Relationships are shown as links between two entities
- The name is given in a diamond box
- The ends of the link show cardinality





Entity/Relationship Diagrams

- Final E/R diagram looks like this:



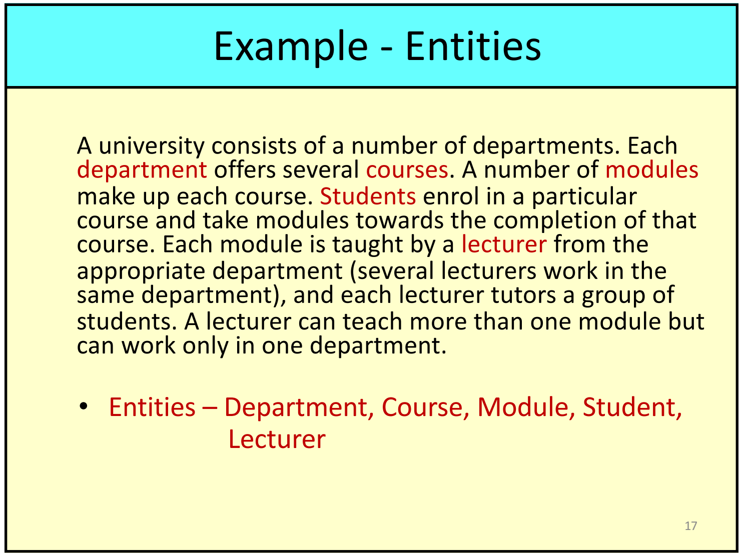
Making E/R Models

- To make an E/R model you need to identify
  - Entities
  - Attributes
  - Relationships
  - 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

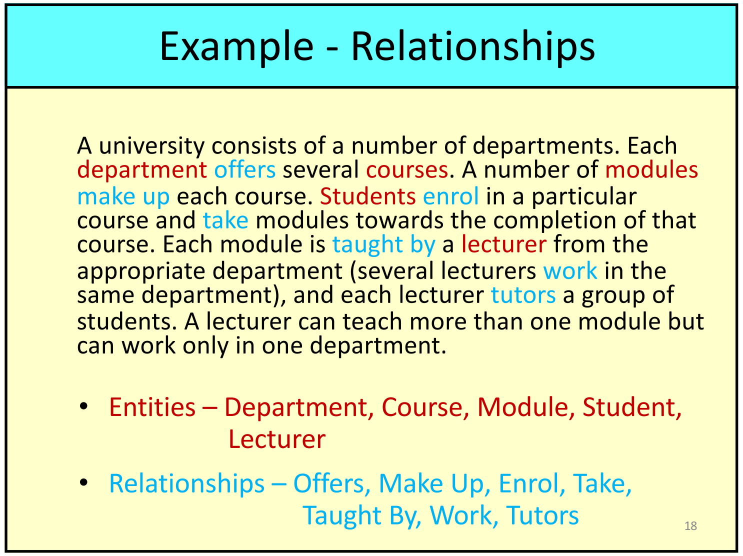
A university consists of a number of departments. Each department offers several courses. A number of modules make up each course. Students enrol in a particular course and take modules towards the completion of that course. Each module is taught by a lecturer from the appropriate department (several 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.



Example - Entities

A university consists of a number of departments. Each department offers several courses. A number of modules make up each course. Students enrol 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.

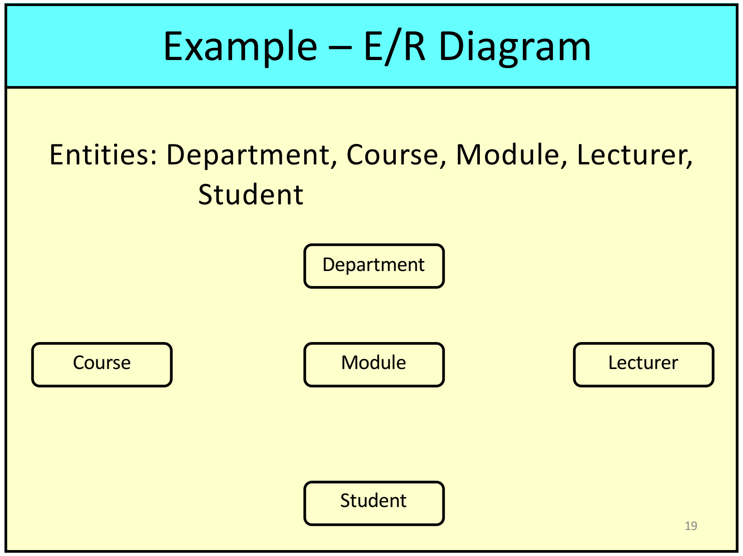
- Entities – Department, Course, Module, Student, Lecturer



Example - Relationships

A university consists of a number of departments. Each department offers several courses. A number of modules make up each course. Students enrol 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, Enrol, Take, Taught By, Work, Tutors



Example – E/R Diagram

Entities: Department, Course, Module, Lecturer, Student

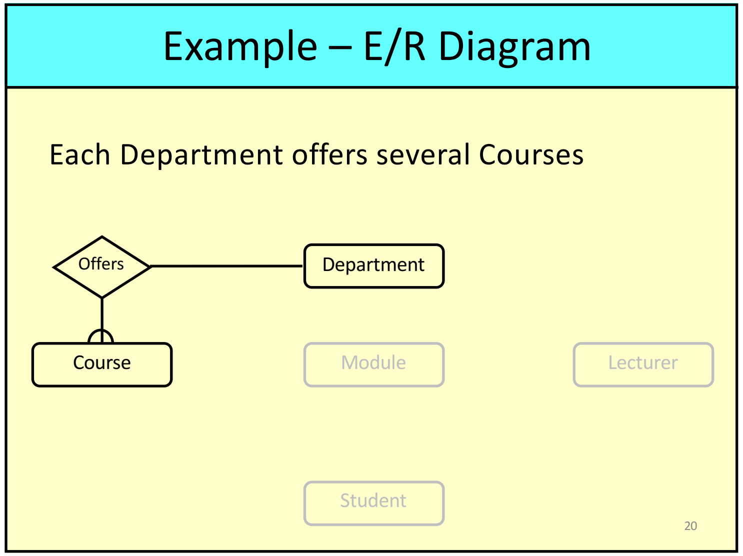
Department

Course

Module

Lecturer

Student



Example – E/R Diagram

Each Department offers several Courses

Offers

Department

Course

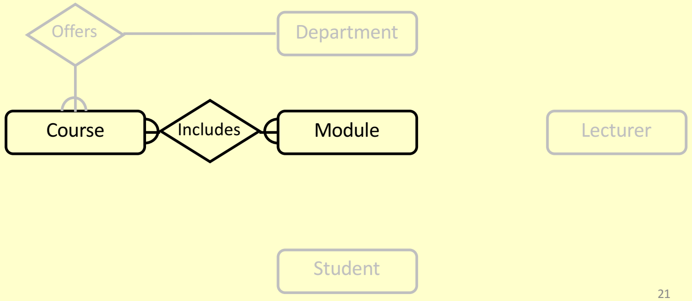
Module

Lecturer

Student

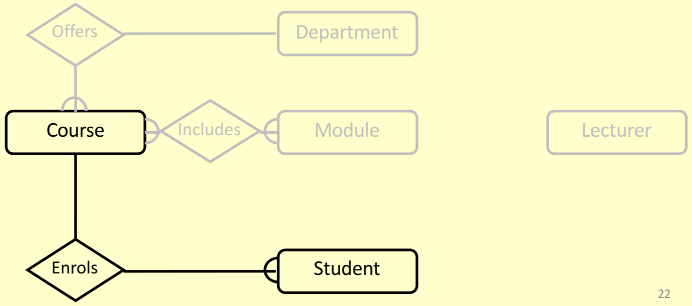
Example – E/R Diagram

A number of modules make up each Course



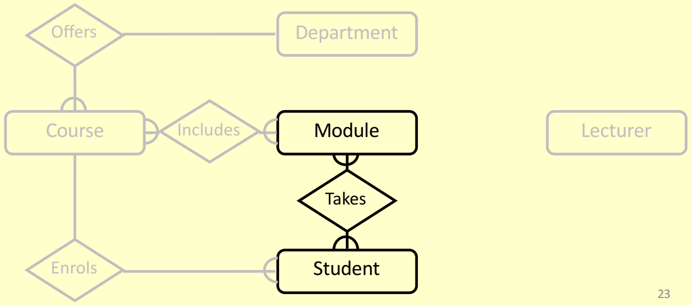
Example – E/R Diagram

Students enrol in a particular course



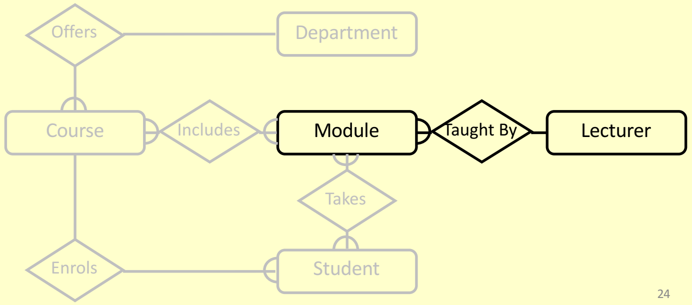
Example – E/R Diagram

Students take several modules



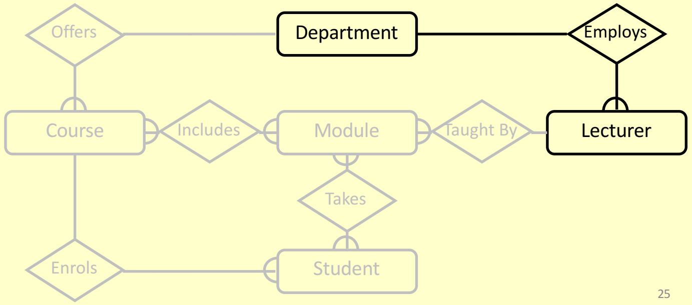
Example – E/R Diagram

Each Module is taught by a Lecturer



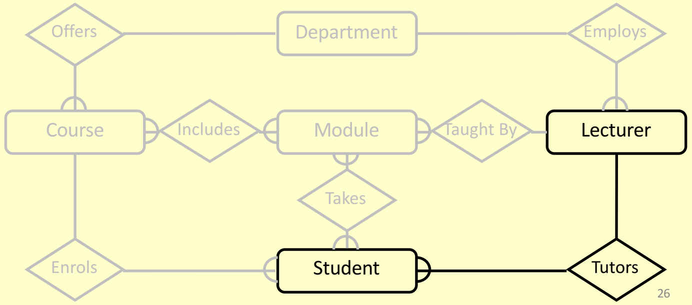
Example – E/R Diagram

Each department employs a number of lecturers



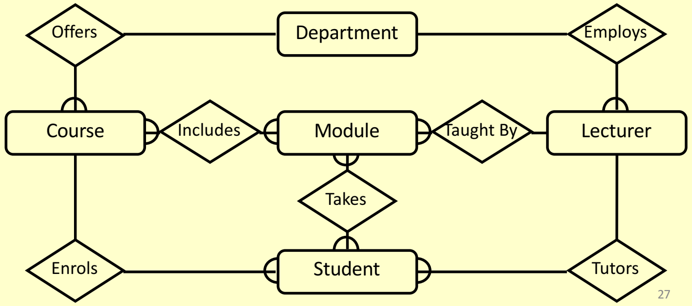
Example – E/R Diagram

Each Lecturer tutors a number of Students



Example – E/R Diagram

The completed diagram.



1 to Many: Department offers several Courses

dID	dName
1	CS
2	ENG
3	B
4	PH

cID	cName
1	CS
2	CS + AI
3	E/E
4	Market

1 to Many: Department offers several Courses

dID	dName
1	CS
2	ENG
3	B
4	PH

dID	cID	cName
1	1	CS
1	2	CS + AI
2	3	E/E
3	4	Market

Many to Many: Students take several Modules

sID	sName	mID
1	J	1, 2
2	M	
3	A	
4	M	

mID	mName
1	AI
2	DBI
3	Circuits
4	Stats

Many to Many: Students take several Modules

sID	sName
1	J
2	M
3	A
4	M

sID	mID
1	1
1	2
3	1
3	2

mID	mName
1	AI
2	DBI
3	Circuits
4	Stats

Removing M:M Relationships

- Many to many relationships are difficult to represent in a database:

Student

SID	SName	SMod
1001	Jack Smith	DBS
1001	Jack Smith	PRG
1001	Jack Smith	IAI
1002	Anne Jones	PRG
1002	Anne Jones	IAI
1002	Anne Jones	VIS

Module

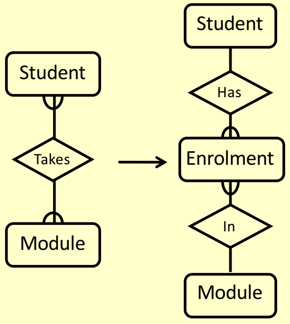
MID	MName
DBS	Database Systems
PRG	Programming
IAI	AI
VIS	Computer Vision

Student

SID	SName	SMods
1001	Jack Smith	DBS, PRG, IAI
1002	Anne Jones	PRG, IAI, VIS

Removing M:M Relationships

- Many to many relationships are difficult to represent in a database
- We can split a many to many relationship into two one to many relationships
- An additional entity is created to represent the M:M 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

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.

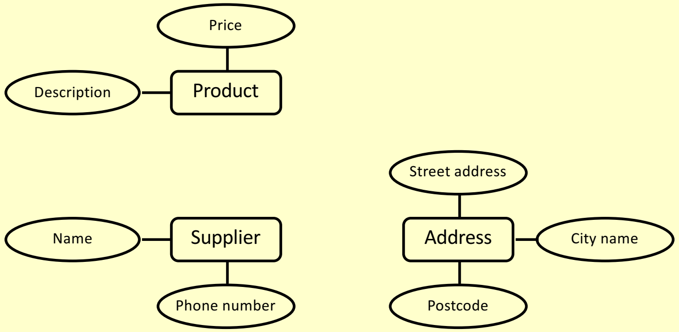
Example - Entities/Attributes

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

Example - 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

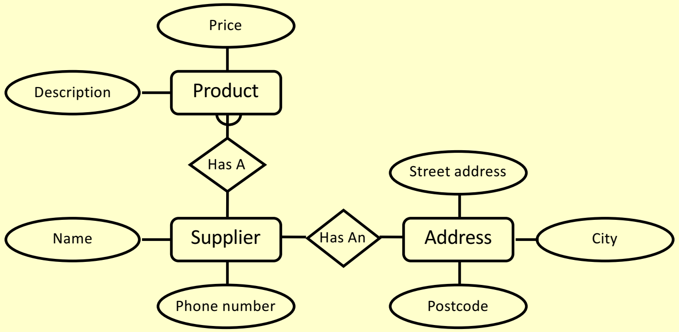
Example - E/R Diagram



Example - 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 - E/R Diagram

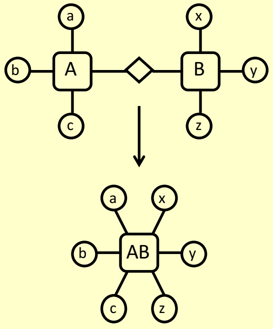


One to One Relationships

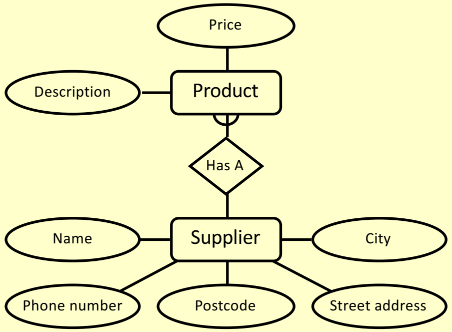
- *Some* relationships between entities, A and B, *might* be redundant if
  - It is a 1:1 relationship between A and B
  - Every A is related to a B and every B is related to an A
- Example - the supplier-address relationship
  - Is one to one
  - Every supplier has an address
  - We don't need addresses that are not related to a supplier

Redundant 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 - E/R Diagram

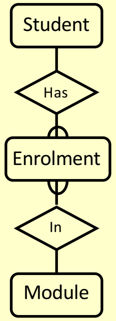


Making E/R Diagrams

- From a description of the requirements identify the
  - Entities
  - Attributes
  - Relationships
  - Cardinality ratios of the relationships
- Draw the E/R 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

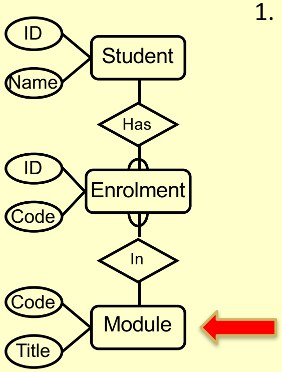
Debugging Designs

- With a bit of practice E/R diagrams can be used to plan queries
  - You can look at the diagram and figure out how to find useful information
  - If you can't find the information you need, you may need to change the design



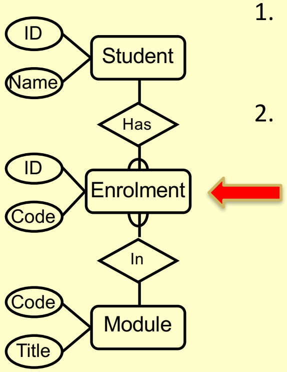
How can you find a list of students who are enrolled in Database systems?

Debugging Designs



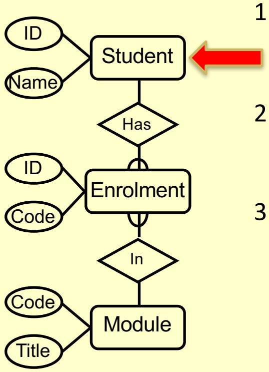
1. Find the instance of Module with the title 'Database Systems'

Debugging Designs

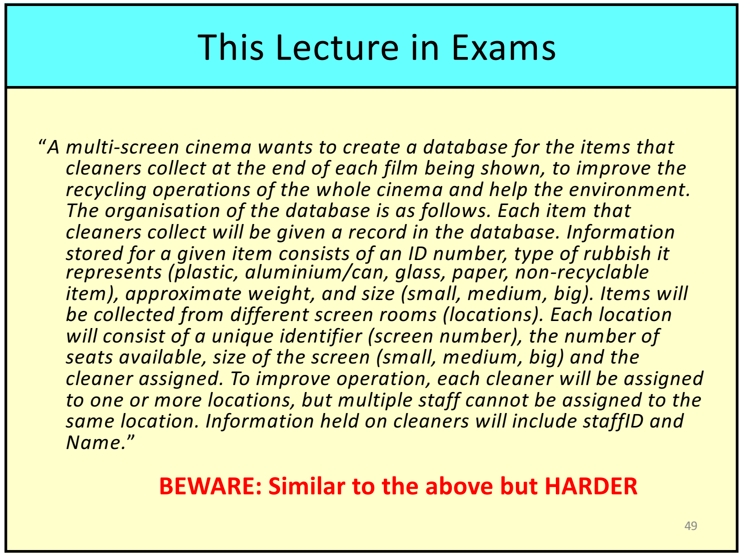


1. Find the instance of Module with the title 'Database Systems'
2. Find instances of the Enrolment entity with the same Code as the result of (1)

Debugging Designs



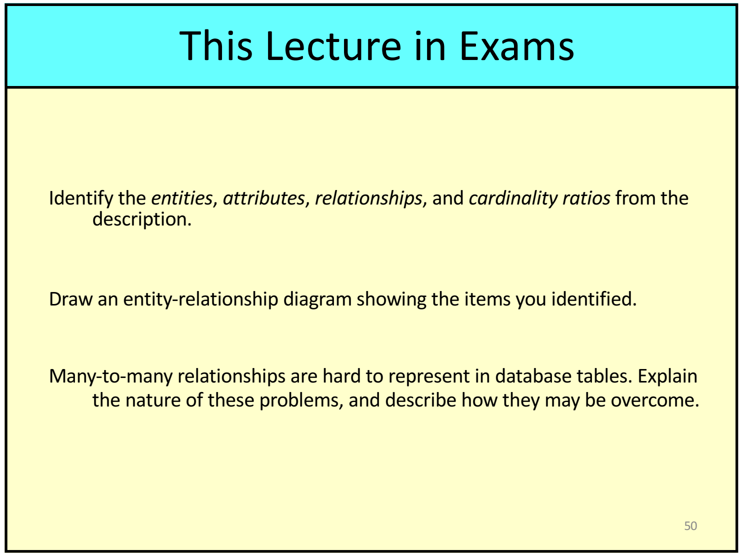
1. Find the instance of Module with the title 'Database Systems'
2. Find instances of the Enrolment entity with the same Code as the result of (1)
3. For each instance of Enrolment in the result of (2) find the corresponding student



This Lecture in Exams

"A multi-screen cinema wants to create a database for the items that cleaners collect at the end of each film being shown, to improve the recycling operations of the whole cinema and help the environment. The organisation of the database is as follows. Each item that cleaners collect will be given a record in the database. Information stored for a given item consists of an ID number, type of rubbish it represents (plastic, aluminium/can, glass, paper, non-recyclable item), approximate weight, and size (small, medium, big). Items will be collected from different screen rooms (locations). Each location will consist of a unique identifier (screen number), the number of seats available, size of the screen (small, medium, big) and the cleaner assigned. To improve operation, each cleaner will be assigned to one or more locations, but multiple staff cannot be assigned to the same location. Information held on cleaners will include staffID and Name."

BEWARE: Similar to the above but HARDER

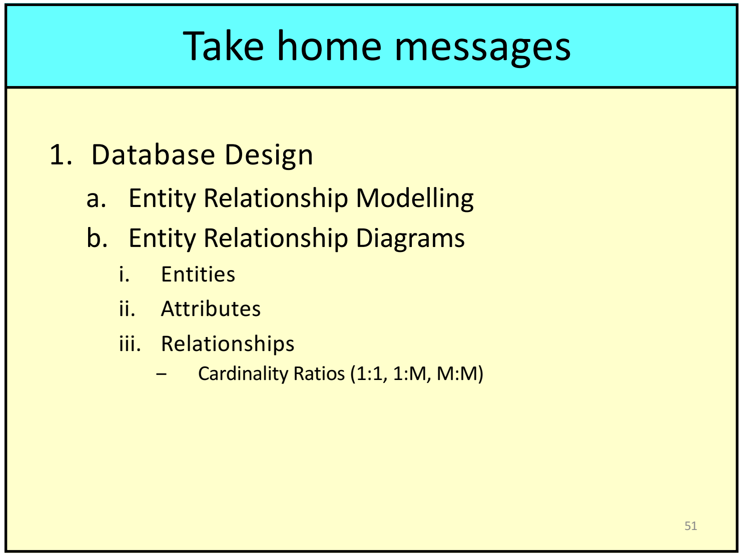


This Lecture in Exams

Identify the *entities*, *attributes*, *relationships*, and *cardinality ratios* from the description.

Draw an entity-relationship diagram showing the items you identified.

Many-to-many relationships are hard to represent in database tables. Explain the nature of these problems, and describe how they may be overcome.



Take home messages

1. Database Design
   a. Entity Relationship Modelling
   b. Entity Relationship Diagrams
      i. Entities
      ii. Attributes
      iii. Relationships
         - Cardinality Ratios (1:1, 1:M, M:M)