Entity Relationship Modelling

Mattew Pike & Yuan Yao

Univeristy of Nottingham Ningbo China (UNNC)

Overview

Content

- Database Design
- Entity Relationship Model
    - Entity
    - Attributes
    - Relationship
    - E/R Diagrams

Learning Outcomes

- Understand what is database design.
- Understand what is the Entity Relationship Model.
- Familiar with each components in the ER Model.
- Understand what is an E/R Diagrams.

Database Design

Recap

What have we learnt so far?

- Some database concepts ...
- Relational Model
- Relational Algebra

What can we do?

- Represent data in the form of relations.
- Write relational algebra to operate on relations.
- Generate relations that satisfy given relational algebra.

Database Design

We yet haven't leanrt how to design relations/tables...

- **Logical Design**
    - Create the database in a given DBMS.
- **Conceptual Design**
    - Build a model independent of the choice of DBMS.
- **Physical Design**
    - How to organise the storage in the hardware.

We will focus on **Conceptual Design** in this lecture.

Entite Relationship Model

Entity Relationship Model

In a University database, we might have data for students, modules and lecturers.

- Students might have their IDs, Names and Age.
- Students may have relationship with modules and lecturers.

Questions:

- How many relation/tables do we need?
- How many attributes in each relation?
- What are the keys?

Example: University Database

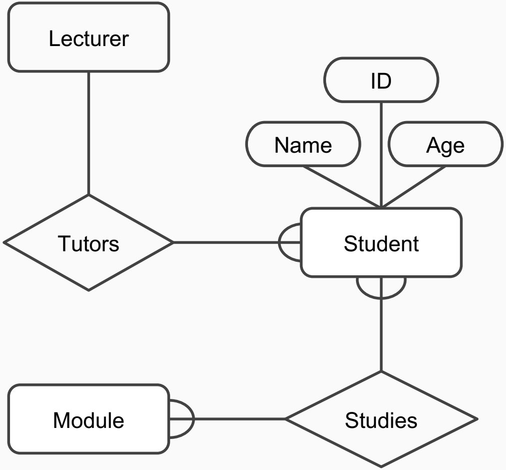
University							
sID	sName	sAge	mId	mName	lId	lName	lAge
...	...	...	...	...	...	...	...

- What are the potential primary keys?
- Is it a good database design? Why?

Entity Relationship Model

Entity Relationship Model are often represented as ER Diagrams.

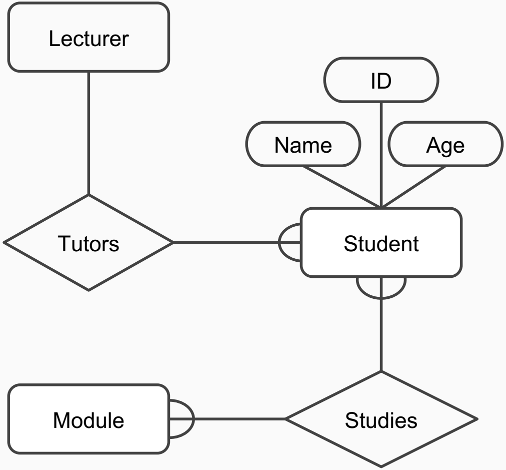
- A conceptual view of the database.
- Independent of the choice of DBMS.
- Can identify problems in a design



Entity Relationship Diagram

Basic Components in an ER Diagram:

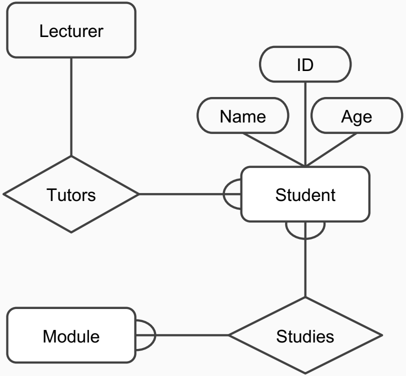
- Entities: objects or things of interest.
- Attributes: properties of an entity.
- Relationships: links between entities.



Entities

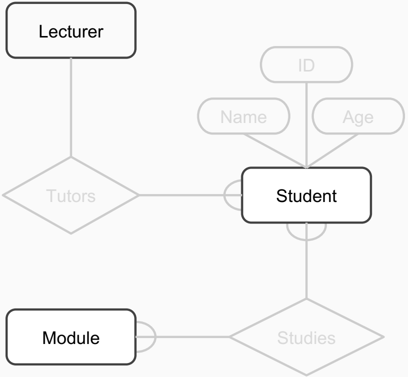
- **Entities** represent objects or things of interest.
  - Physical things like *students*, *lecturers*, *employee* and *products*.
  - More abstract things like *modules*, *orders* and *projects*.
- **Entity types:**
  - A group of objects with same properties, e.g., Lecturer.
- **Entity instance (occurrence):**
  - A uniquely identifiable of that particular type, e.g., Yuan Yao is an instance of **Lecturer**.

Example: Entities



What are entities in this ER Diagram?

Example: Entities

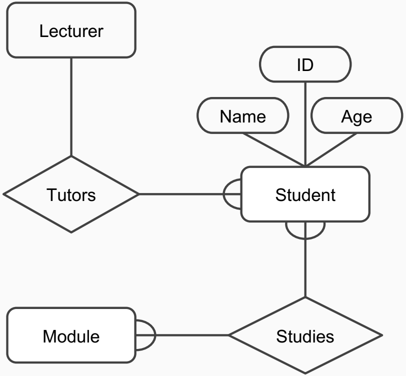


- Entities are represented as boxes with rounded corners.
- Each box is labelled with the name of the entity type.

Attributes

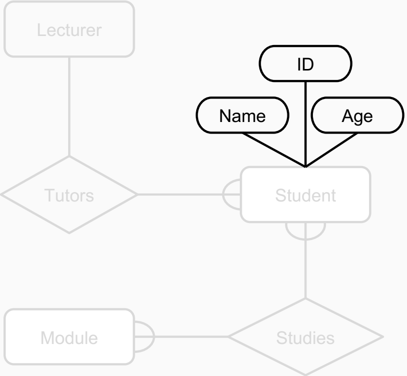
- **Attributes** are facts, aspects, properties or other details about an **entity**.
    - Students have *IDs*, *names* and so on.
    - Projects have *titles*, *codes* ...
- An **Attribute** has
    - A name.
    - An associated entity (type? or instance?).
    - Domains of possible values.
    - Each instance of the associated entity has a value from the domain.

Example: Attributes



What are the attributes in this ER Diagram?

Example: Attributes



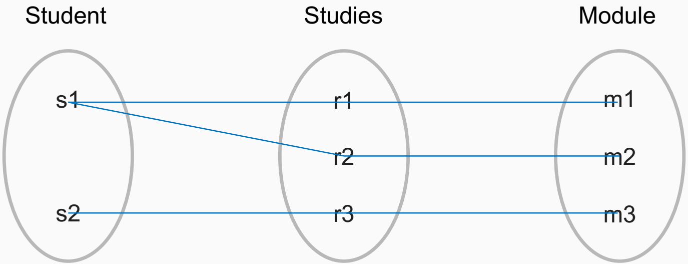
- Attributes are drawn as ovals.
- Each attribute is linked to its entity by a line.
- The name of the attribute is written in the oval.

Relationships

- **Relationships** are associations between two or more **entities**.
    - Each student *takes* several modules.
    - Each module is *taught* by a lecturer.
    - Each employee *works* for a company.
- A **Relationship** has
    - A name.
    - A set of entities that participate in them.
    - A degree.
    - A cardinality ratio.

Relationships

- **Relationship Type:** an association between two or more entity types.
- **Relationship Instance:** a uniquely identifiable association that includes one instance from each participating entity type.



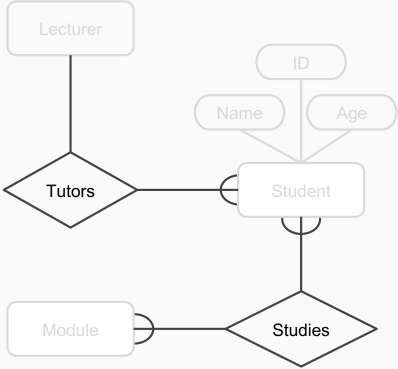
Degree of a Relationship

- **Degree of a Relationship:** the number of participating entity (type? or instance?)
- Examples:
  - A teacher teaches a particular course with a particular class.
  - A module may have other modules as its prerequisites.

Cardinality Ratio

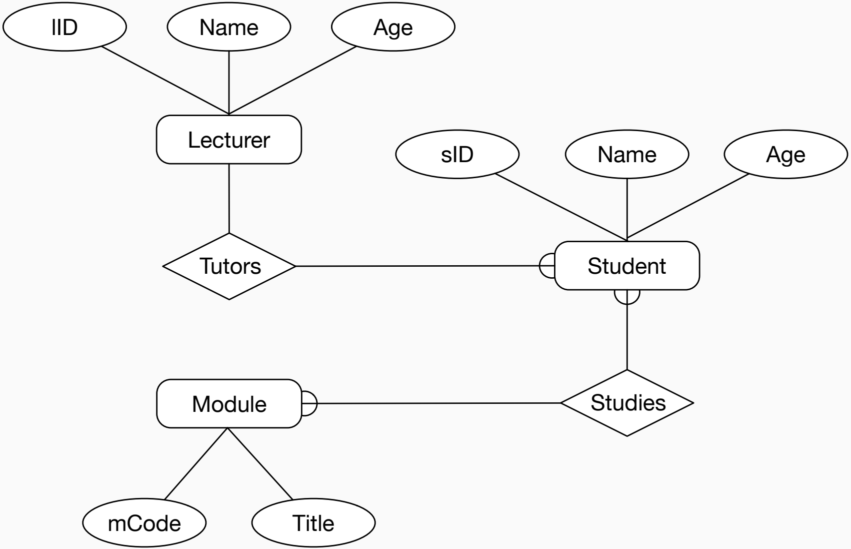
- Each entity instance in a relationship can participate in zero, one or more than one instance of that relationship.
- We are not considering zero instance here.
- Three types of relationships with different cardinality ratio:
  - One to One (1:1)
    - E.g., each lecturer has a unique office, and offices are all single occupancy.
  - One to Many (1:M)
    - E.g., a lecturer may tutor many students, but each student has exactly one tutor.
  - Many to Many (M:M)
    - E.g., Each student takes several modules and each module is taken by several students.

Example: Relationship



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

ER Diagram


lID
Name
Age
Lecturer
sID
Name
Age
Tutors
Student
Module
Studies
mCode
Title

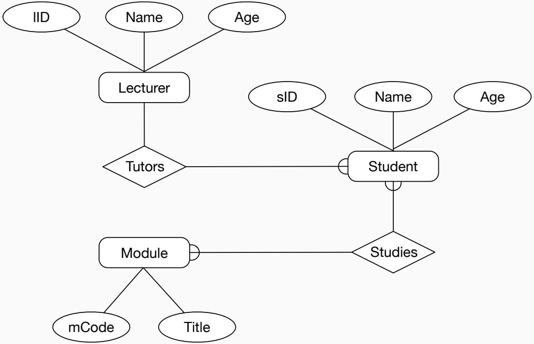
Relational Model vs ER Diagram

Relational Model	ER Diagram
Relation	
Tuple	
Attribute	
Primary Key	
Foreign Key	

Relational Model vs ER Diagram

Relational Model	ER Diagram
Relation	Entity Type
Tuple	Entity Instance
Attribute	Attribute
Primary Key	Attribute
Foreign Key	1:M relationship

ER Diagram to Relations



Lecturer		
lID	Name	Age

Student			
sID	Name	Age	lID

How about M:M relationship between **Student** and **Module**?