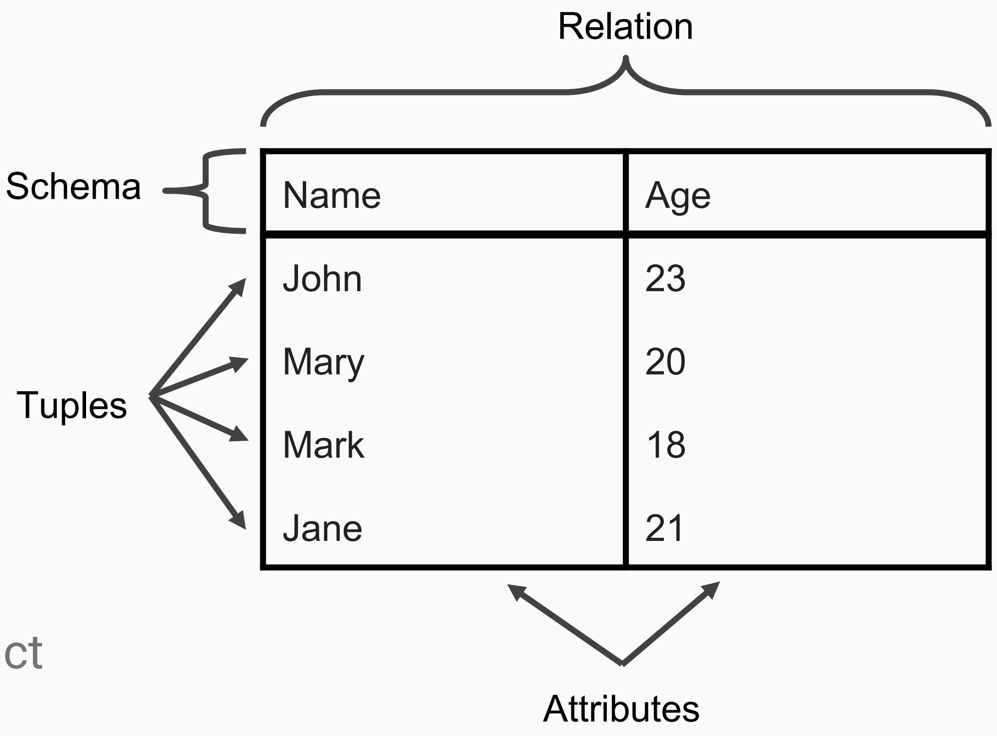
Entity Relationship Modelling

DBI - Databases and Interfaces
Dr Matthew Pike & Dr Yuan Yao

Recap

- Relational Model
  - Relations, Attributes, Schema, Tuples, Domains
  - Degree and Cardinality
  - Candidate Key, Primary Key, Foreign Key
- Relational Algebra
  - Selection, Projection
  - Union, Set Difference, Intersection, Cartesian Product
  - Natural Joint, Theta Joint, Rename



Name	Age
John	23
Mary	20
Mark	18
Jane	21

Cartesian Product vs Natural Joint

- Extended Cartesian product:
- A = $\{(c_1^1, \dots, c_1^n), \dots (c_r^1, \dots, c_r^n)\}$
- B = $\{(d_1^1, \dots, d_1^m), \dots (d_s^1, \dots, d_s^m)\}$
- $A \times B = \{(c_i^1, \dots, c_i^n, d_j^1, \dots, d_j^m) \mid 1 \leq i \leq r, 1 \leq j \leq s\}$
- Natural Joint
- A = $\{(c_1^1, \dots, c_1^n), \dots (c_r^1, \dots, c_r^n)\}$
- B = $\{(d_1^1, \dots, d_1^m), \dots (d_s^1, \dots, d_s^m)\}$
- Attr^a, … , Attr^b in A and B have the same name
- $A \bowtie B = \{(c_i^1, \dots, c_i^n, d_j^1, \dots, d_j^{a-1}, d_j^{b+1}{}_{, \dots,}\, d_j^m) \mid 1 \leq i \leq r, 1 \leq j \leq s, c_i^x = d_j^x, a \leq x \leq b\}$

Cartesian Product vs Natural Joint

R	
A	**C**
3	3
6	4
7	1

S		
B	**C**	**D**
5	1	6
4	3	9

What is the result of R x S?

What are the result of R ⋈ S?

Cartesian Product vs Natural Joint

Cartesian product:

R X S				
A	**R.C**	**B**	**S.C**	**D**
3	3	5	1	6
3	3	4	3	9
6	4	5	1	6
6	4	4	3	9
7	1	5	1	6
7	1	4	3	9

Natural Joint:

R ⋈ S			
A	**C**	**B**	**D**
3	3	4	9
7	1	5	6

R	
A	**C**
3	3
6	4
7	1

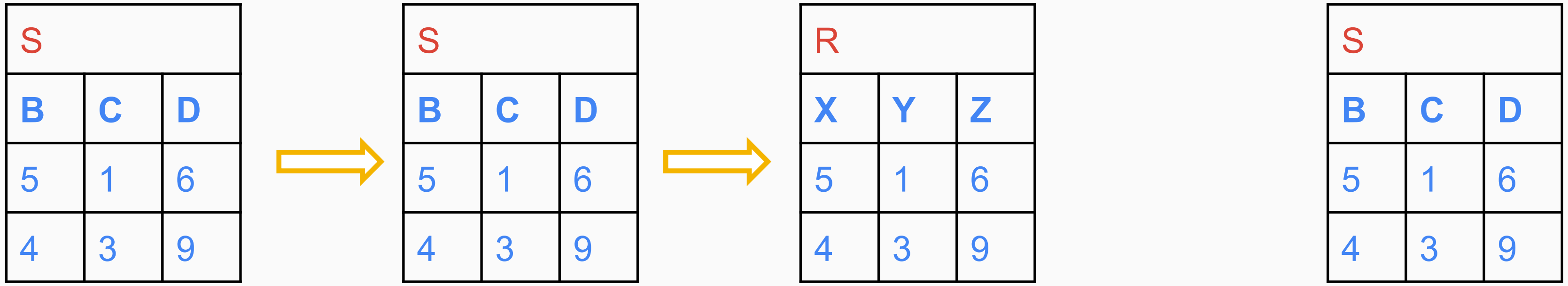
S		
B	**C**	**D**
5	1	6
4	3	9

Questions:

1) What happens if no tuple from R and S have the same C value?
2) What happens if R and S do not have common attribute?

Rename

ρR(X, Y, Z)(S)



S		
B	**C**	**D**
5	1	6
4	3	9

S		
B	**C**	**D**
5	1	6
4	3	9

R		
X	**Y**	**Z**
5	1	6
4	3	9

S		
B	**C**	**D**
5	1	6
4	3	9

Create a copy

Rename in the new copy

Why we need Relational Model and Algebra?

- Relational Model and Algebra
  - Purely mathematical model based on set theory and first-order logic.
  - Operations are purely mathematical calculations.

- SQL
  - Based on the relational model
  - Tables, queries implements relational model and algebra

What we have learnt so far

- Some Database Concepts …
- Some Mathematical Models …
- Some Mathematical Operations …

- Before creating and learning Database we need to know how to design it
  - What tables, keys and constraints are needed?
  - What is the database going to be used for?

Database Design

- Physical Design
  - How the database is stored in hardware
- Logical Design
  - Create the Database in a given DBMS
- Conceptual Design
  - Build a model independent of the choice of DBMS

- We will learn conceptual design in this lecture

This Lecture

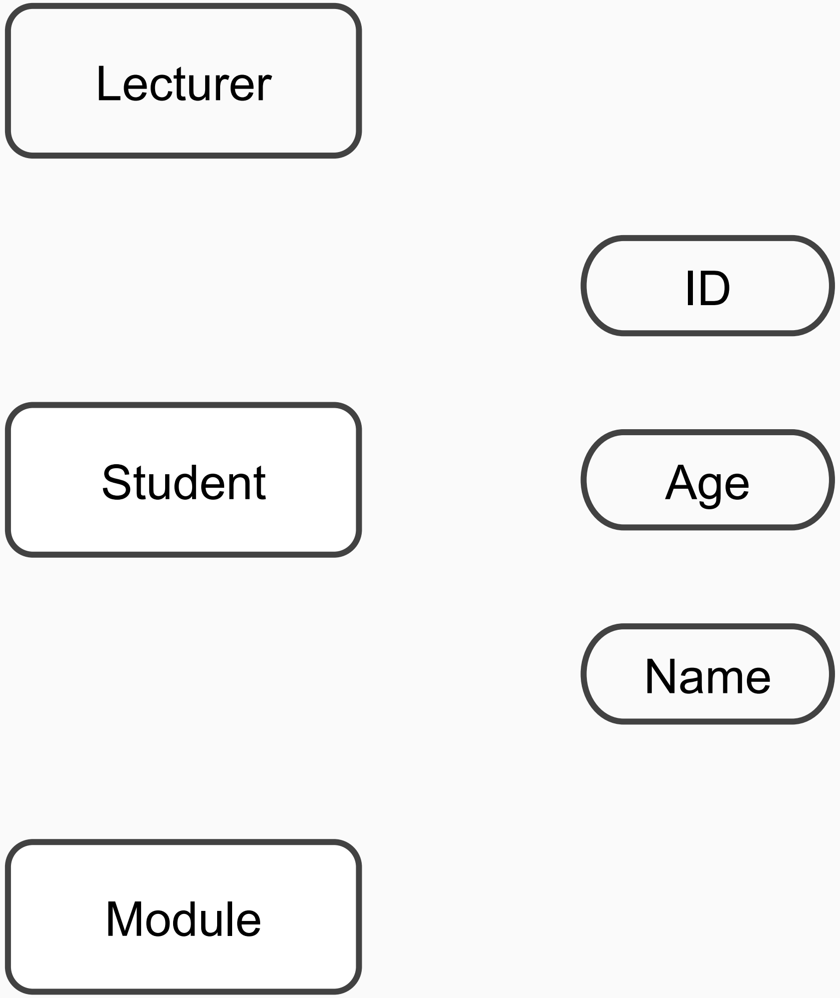
- Entity/Relationship models
  - Entities and Attributes
  - Relationships
  - E/R Diagrams

- Further Reading
  - Database Systems, Connolly & Begg, Chapter 12

Entity/Relationship Model

In a University database we might have Students, Modules and Lecturers

- Students might have their ID, Name, and Age
- Students could have relationships with Modules and Lecturers



Lecturer

ID

Student

Age

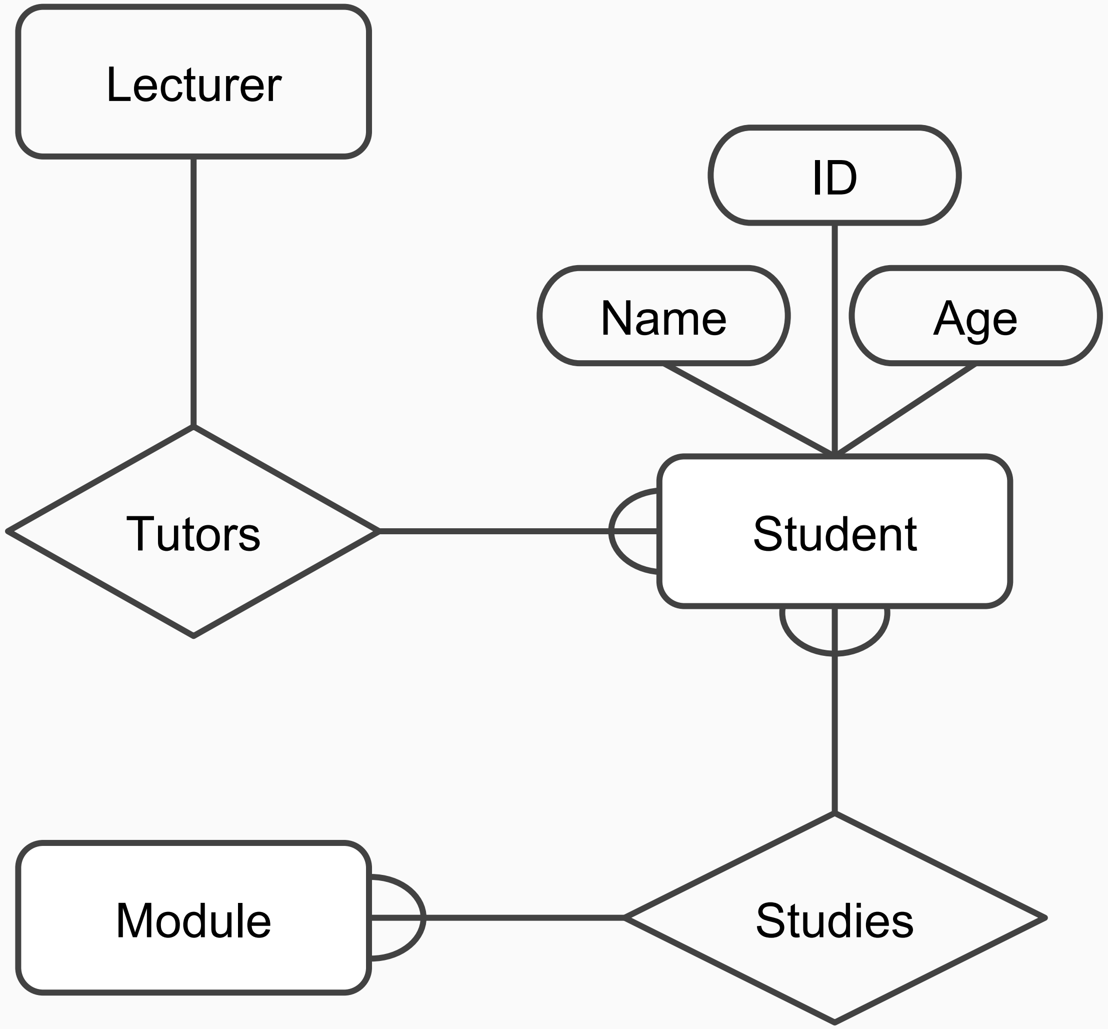
Name

Module

Entity/Relationship Diagrams

E/R Models are often represented as E/R diagrams

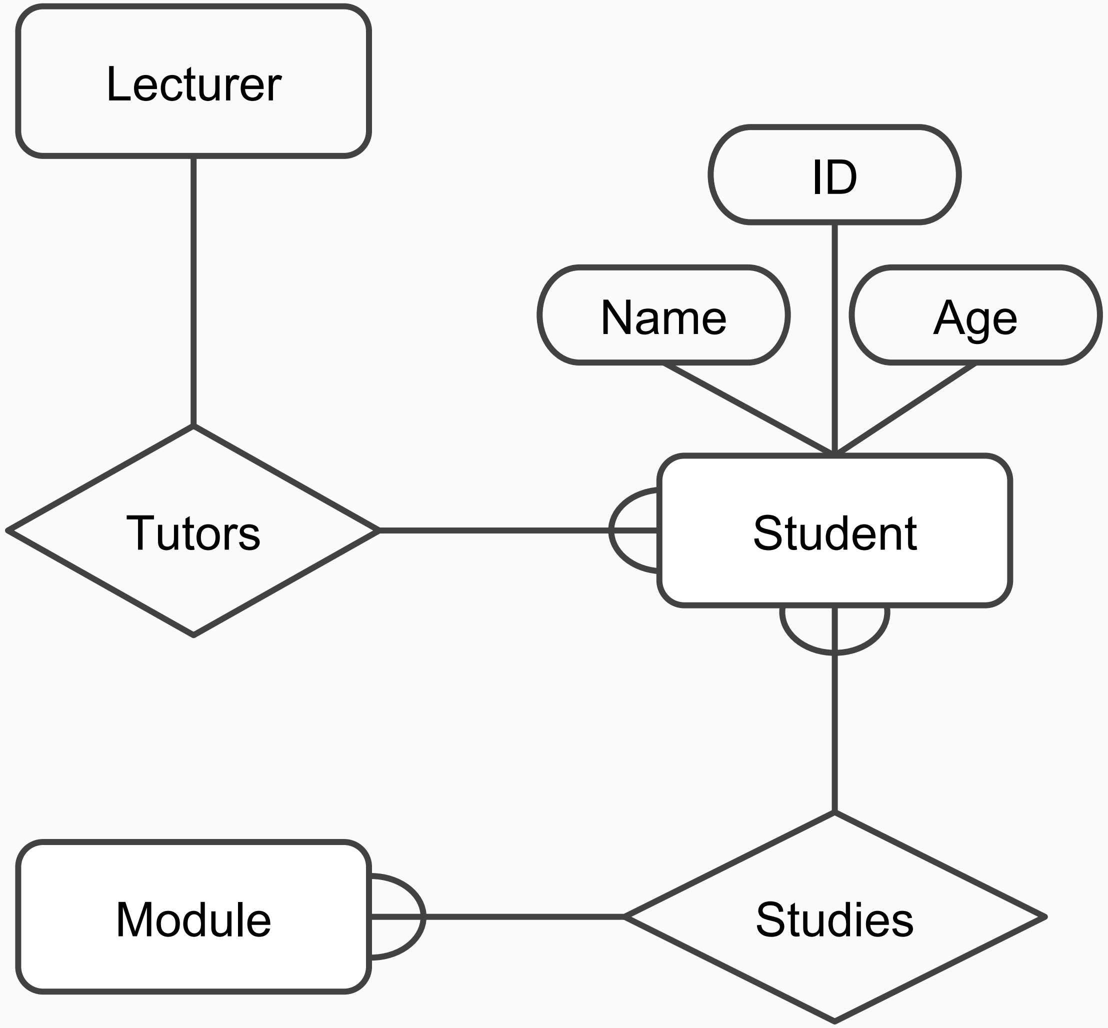
- A conceptual view of database
- Independent of the choice of DBMS
- Can identify some problems in a design



Entity/Relationship Diagrams

Basic components:

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



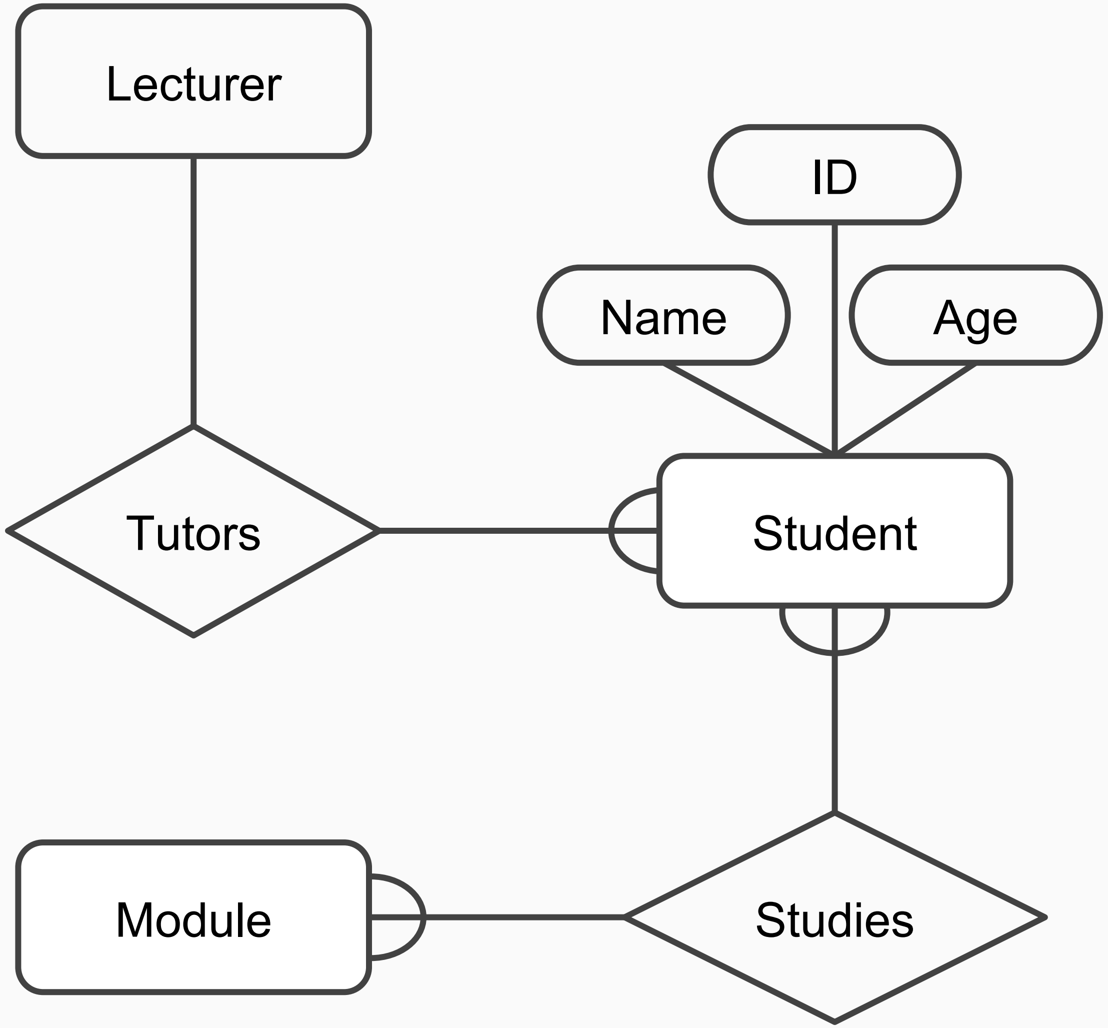
Entities

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

- Entities type:
    - A group of objects with same properties in the real world, e.g., Lecturer

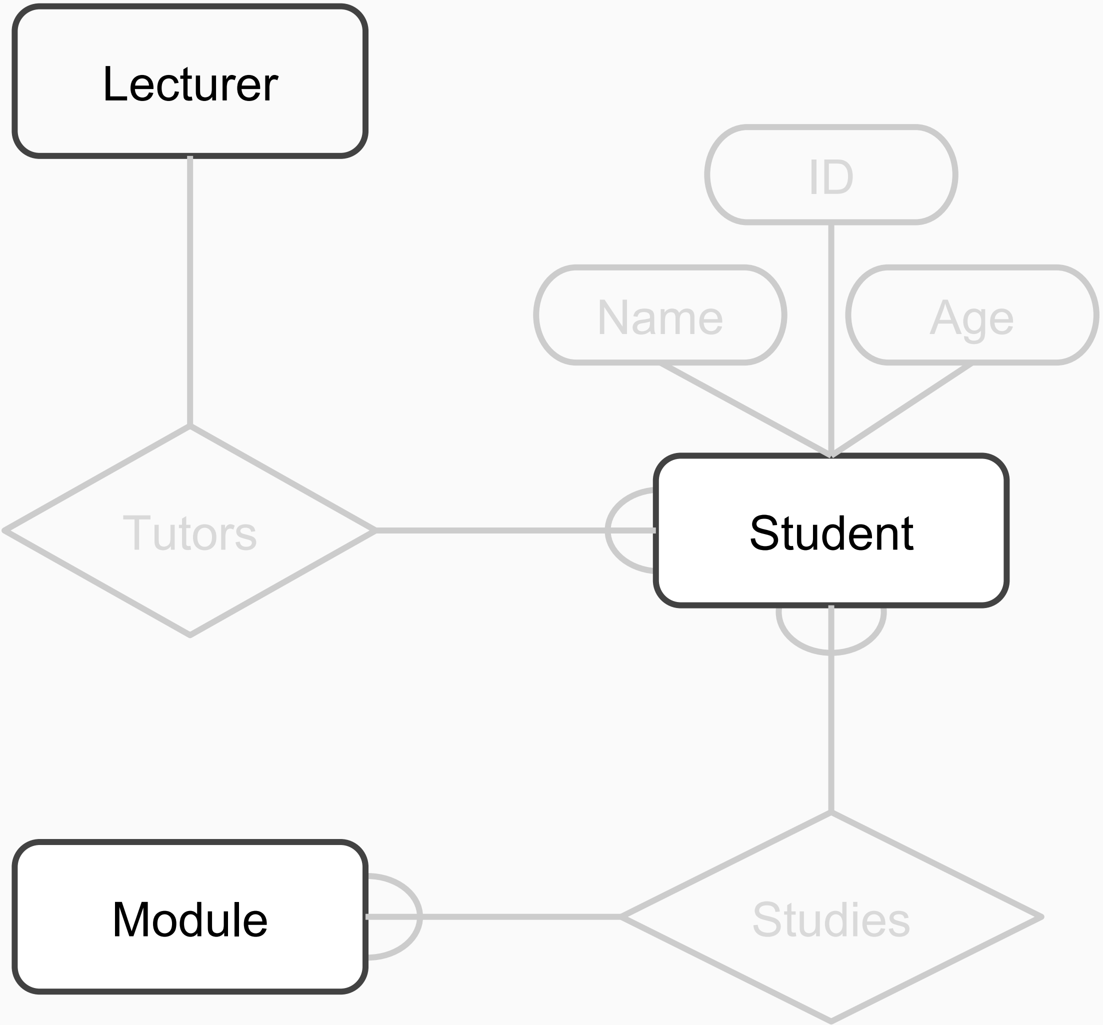
- Entities instance (occurrence):
    - A uniquely identifiable of that particular type. E.g. Yuan Yao is an instances of Lecturer

Which are entities in this E/R Diagram?



E/R: 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

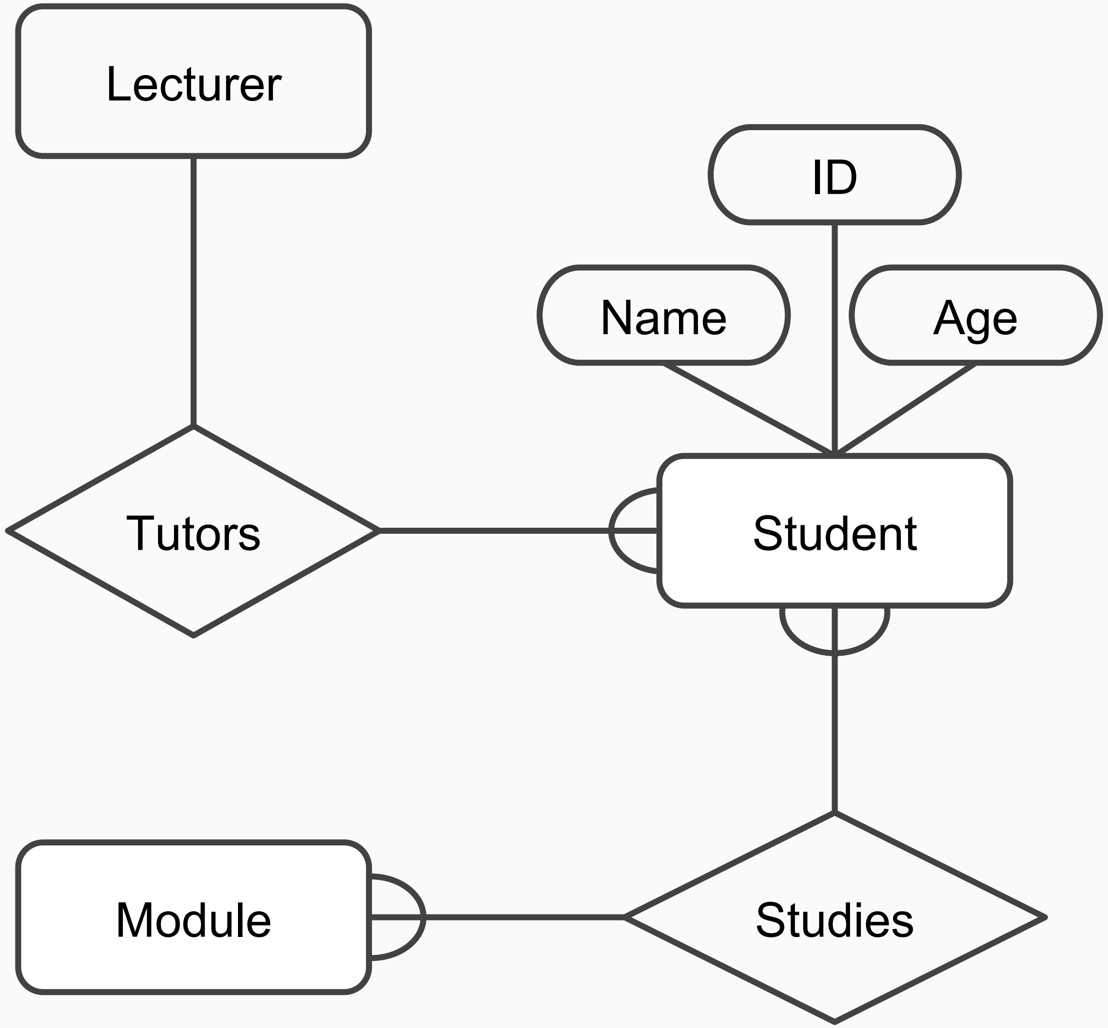


Attributes

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

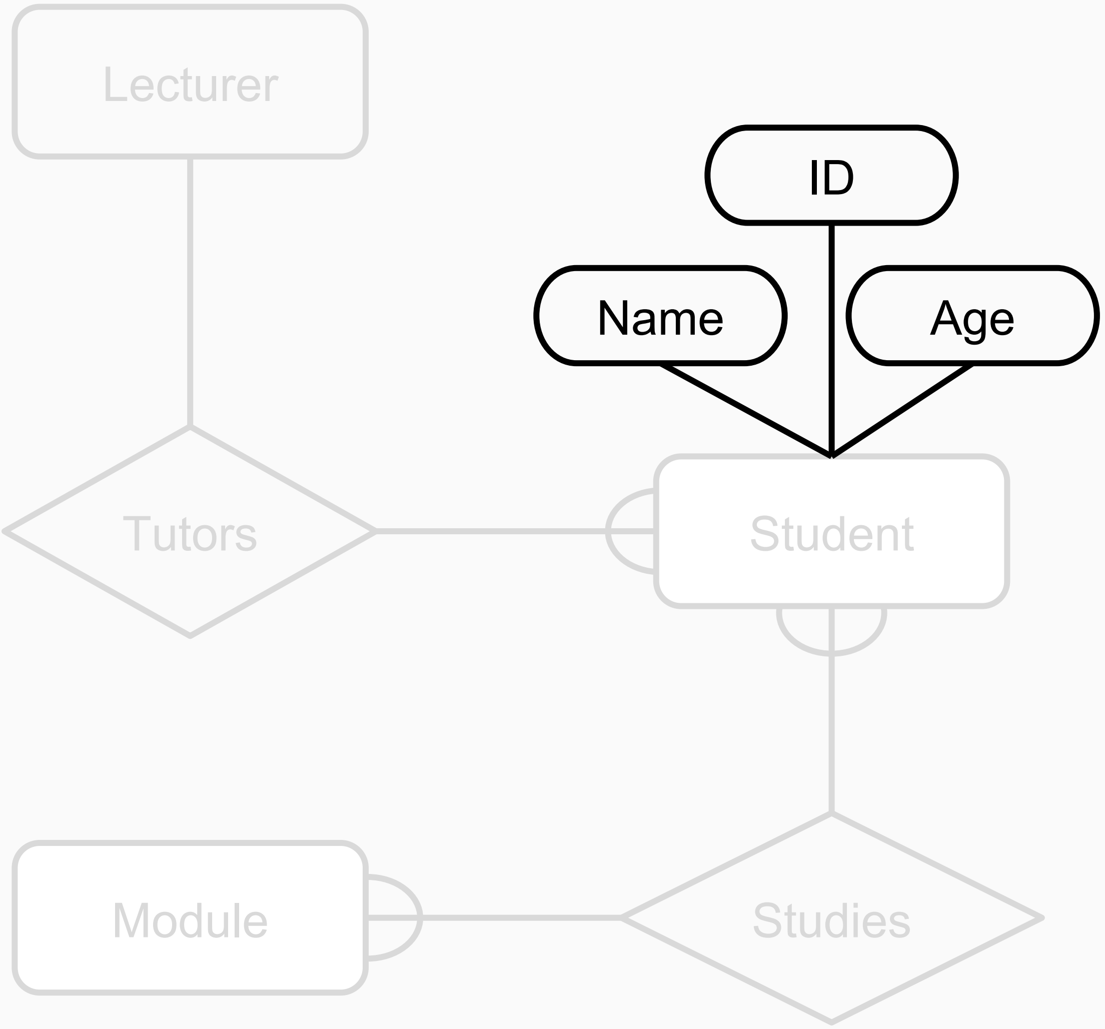
- Attributes have
  - A name
  - An associated entity (type or instance?)
  - Domains of possible values
  - For each instance of the associated entity, a value from the attributes domain

Which are attributes in this E/R Diagram?



E/R: Diagram 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

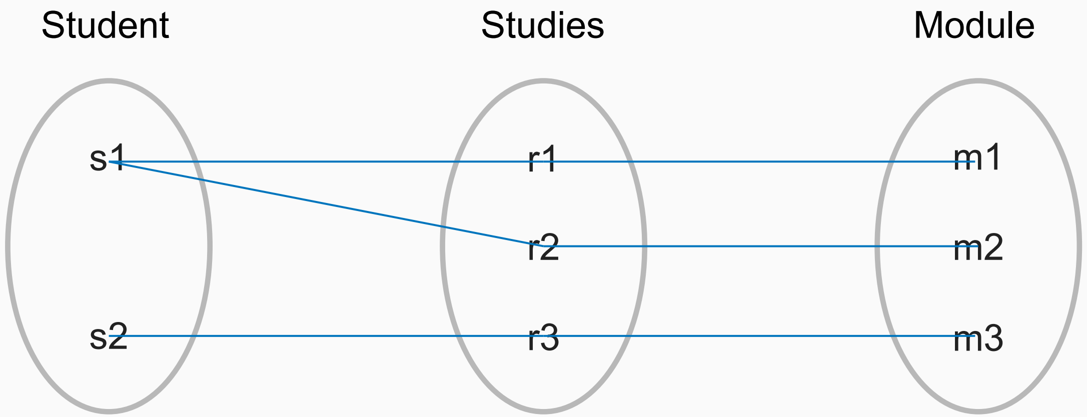
- Relationships are an association between two or more entities
  - Each Student takes several Modules
  - Each Module is taught by a Lecturer
  - Each Employee works for a single Department

- 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

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.

- Example:
    - Each student has ID number.
    - A teacher teaches a particular course and a particular class.
    - A module may have other modules as its prerequisites.

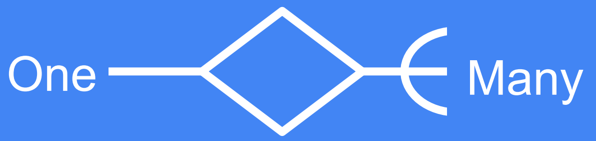
Cardinality Ratios

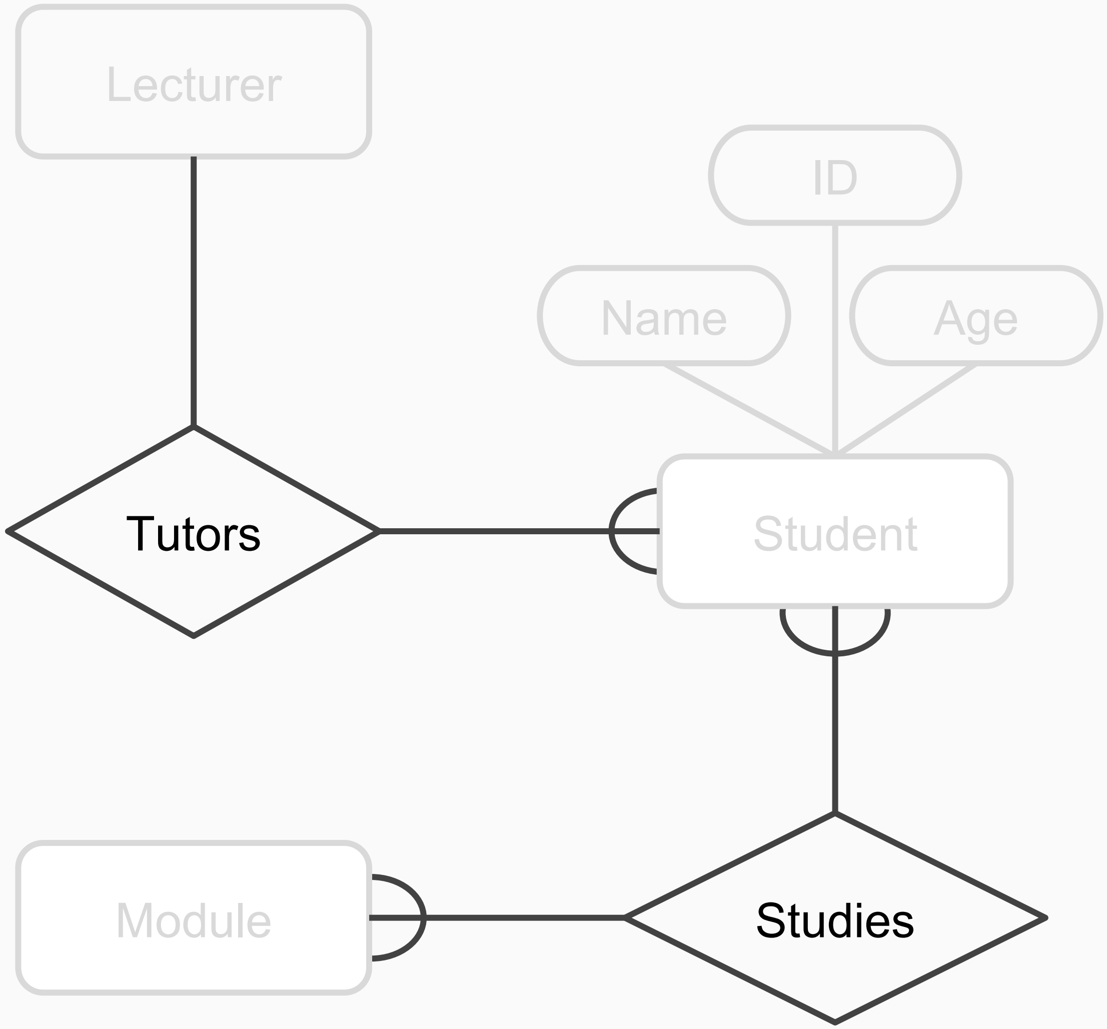
- Each entity in a relationship can participate in zero, one, or more than one instances of that relationship
- We won’t be dealing with optional (zero instances) of relationships
- This leads to 3 types of relationship...

- 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

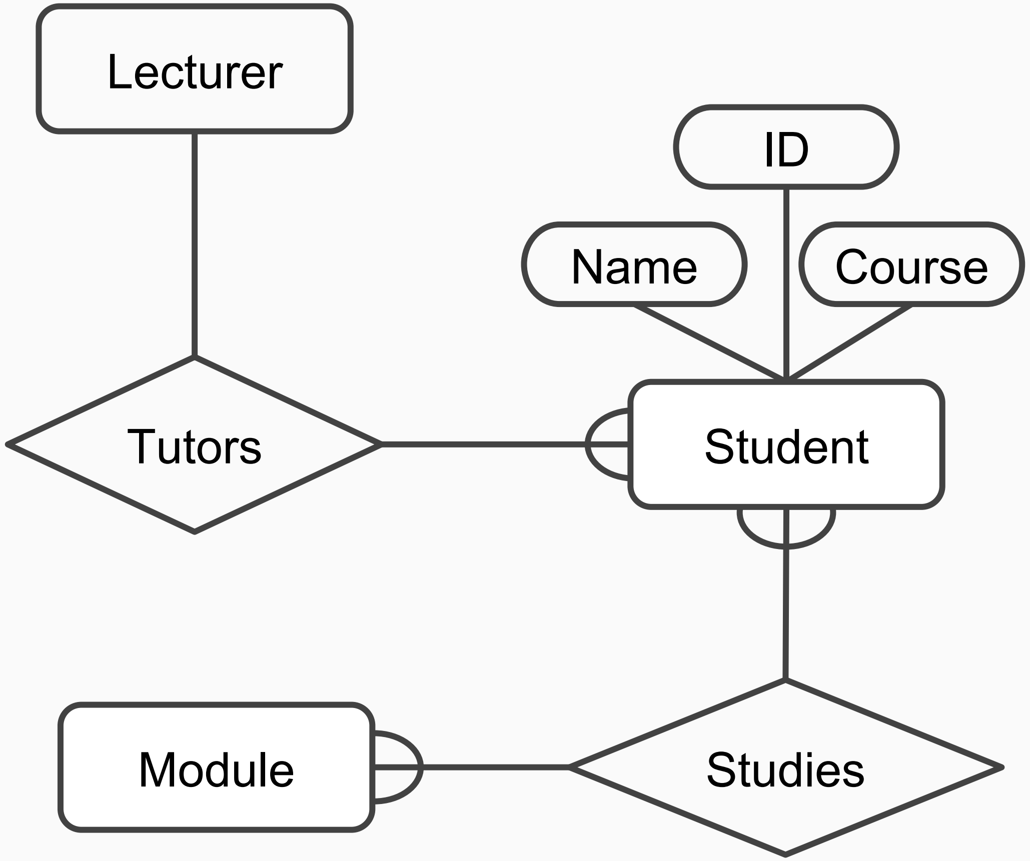
E/R: Diagram Relationships

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





Final Diagram



E/R Model vs Relational Model

Relational Model	E/R Model
Relation	
Tuple	
Attribute	
Primary Key	
Foreign Key	

E/R Model vs Relational Model

Relational Model	E/R Model
Relation	
Tuple	
Attribute	Attribute
Primary Key	
Foreign Key	

E/R Model vs Relational Model

Relational Model	E/R Model
Relation	Entity Type
Tuple	Entity Instance
Attribute	Attribute
Primary Key	
Foreign Key	

E/R Model vs Relational Model

Relational Model	E/R Model
Relation	Entity Type
Tuple	Entity Instance
Attribute	Attribute
Primary Key	Attribute
Foreign Key	

E/R Model vs Relational Model

Relational Model	E/R Model
Relation	Entity Type
Tuple	Entity Instance
Attribute	Attribute
Primary Key	Attribute
Foreign Key	1:M relationship

What’s next?

- We have learnt E/R Model and Diagram
- We will be given the problem description
- Task: design database for given problem

Problem Description → E/R Model