



ERD ENTITY RELATIONSHIP DIAGRAM

M. Rasti-Barzoki

Website: rasti.iut.ac.ir

91-2



Entity Relationship Diagrams for Data Modelling

- An Entity-Relationship Diagram (ERD) shows how the data that flows in the system is organised and used.



Entity Relationship Diagrams for Data Modelling

- The elements of an **ERD** are the following:
 - Entity –
 - Attributes –
 - Relationships – associations between entities.



Entity Relationship Diagrams for Data Modelling

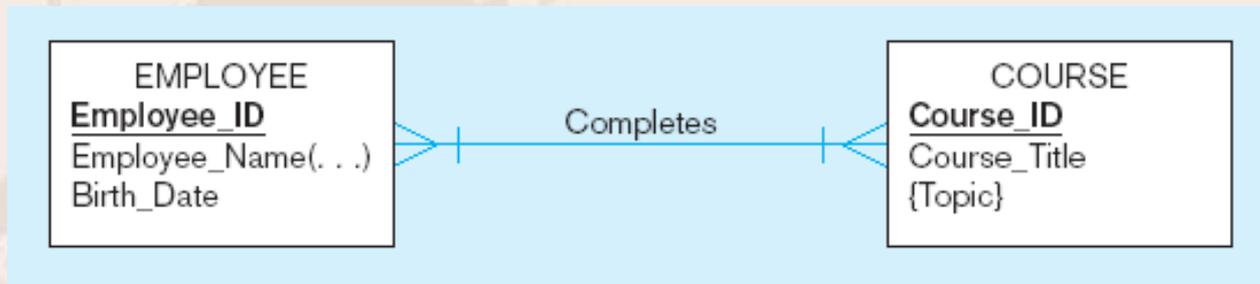
Relationships

- **A one-to-one relationship (1:1).**
- **A one-to-many (1:N).**
- **A many-to-many (M:N).**



Relationship types and instances

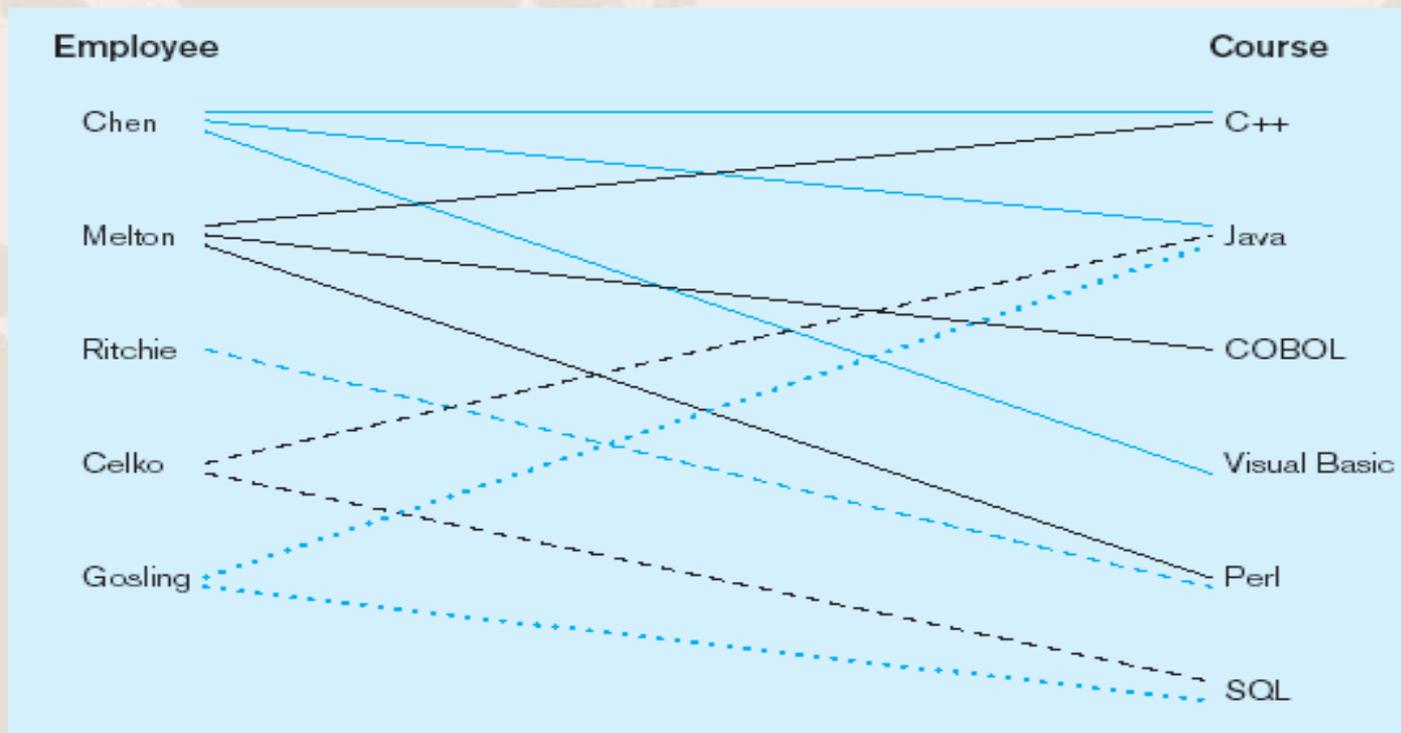
- a) Relationship type





Relationship types and instances

- b) Relationship instances



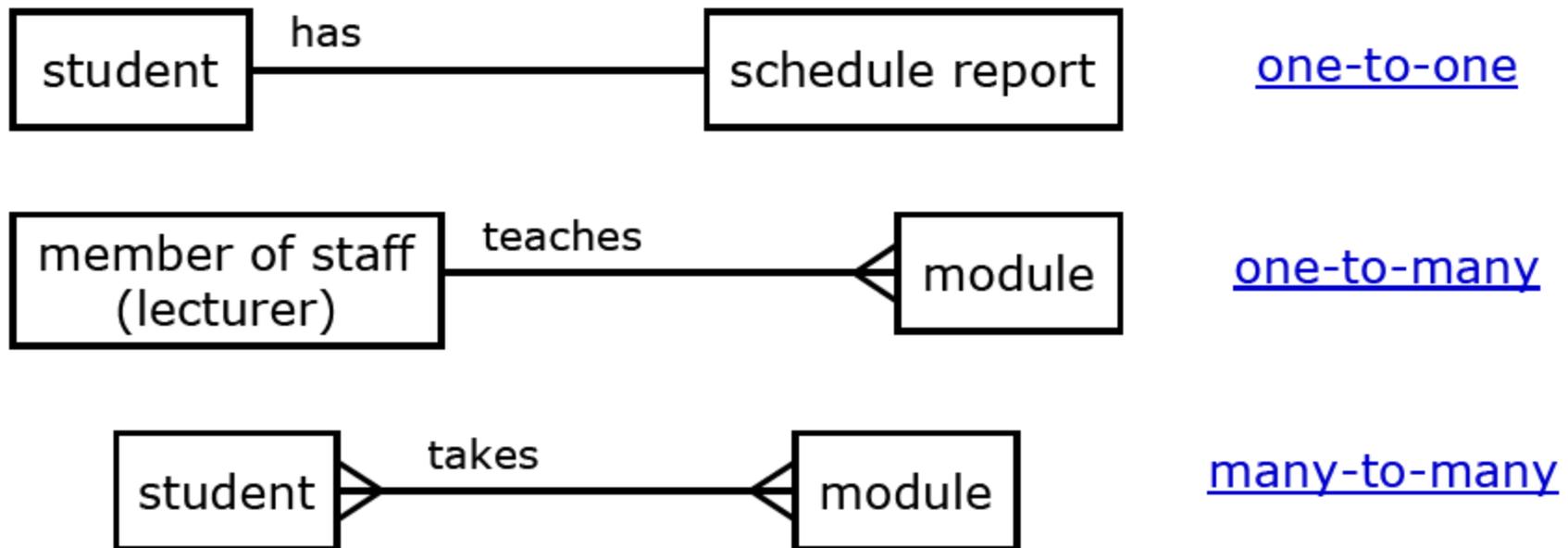


Management Information Systems

ERD

ERD

- Types of entity relationships and example of their representation:



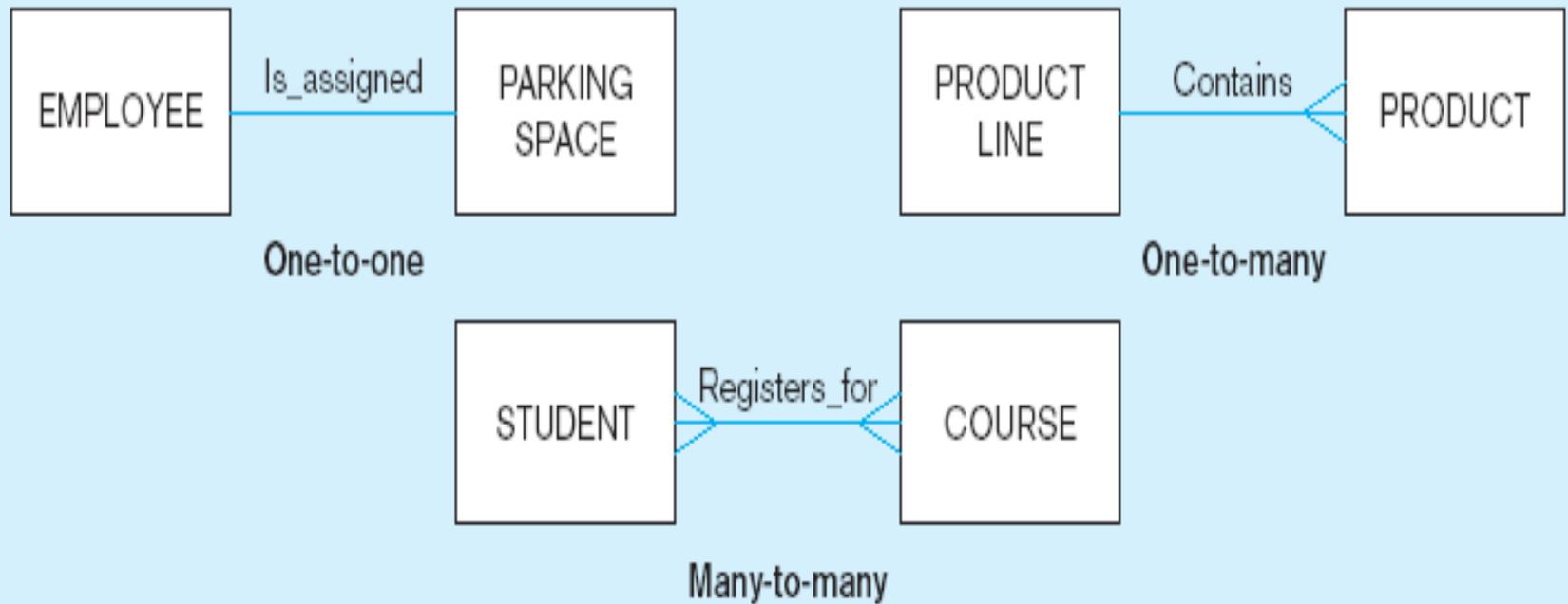


Management Information Systems

ERD

ERD

- **Examples**





Management Information Systems

ERD

ERD

- **Cardinalities**

Cardinalities



Mandatory One



Optional One



Mandatory Many



Optional Many

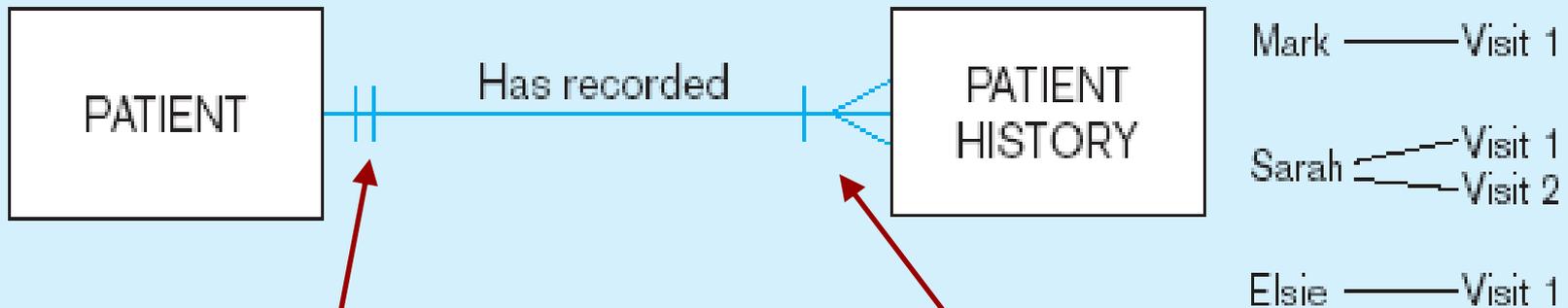


Management Information Systems

ERD

ERD

- Examples



A patient history is recorded for one and only one patient

A patient must have recorded at least one history, and can have many

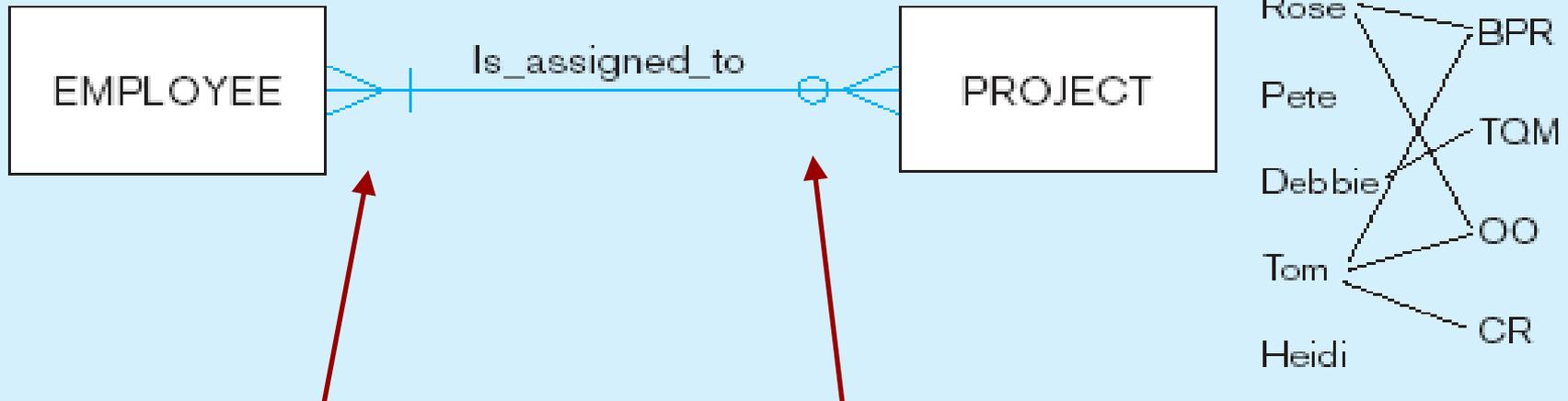


Management Information Systems

ERD

ERD

- Examples



A project must be assigned to at least one employee, and may be assigned to many

An employee can be assigned to any number of projects, or may not be assigned to any at all



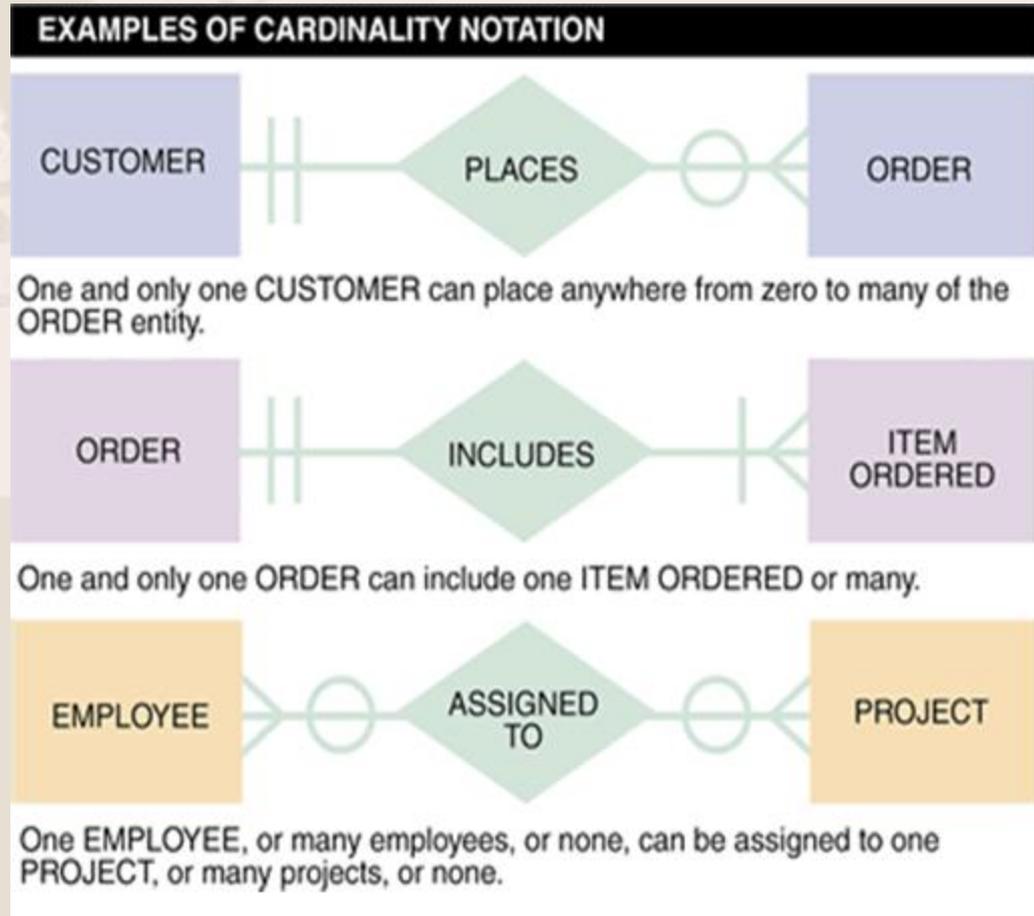
Management Information Systems

ERD

ERD

- Examples

SYMBOL	MEANING	UML REPRESENTATION
	One and only one	1
	One or many	1..*
	Zero, or one, or many	0..*
	Zero, or one	0..1



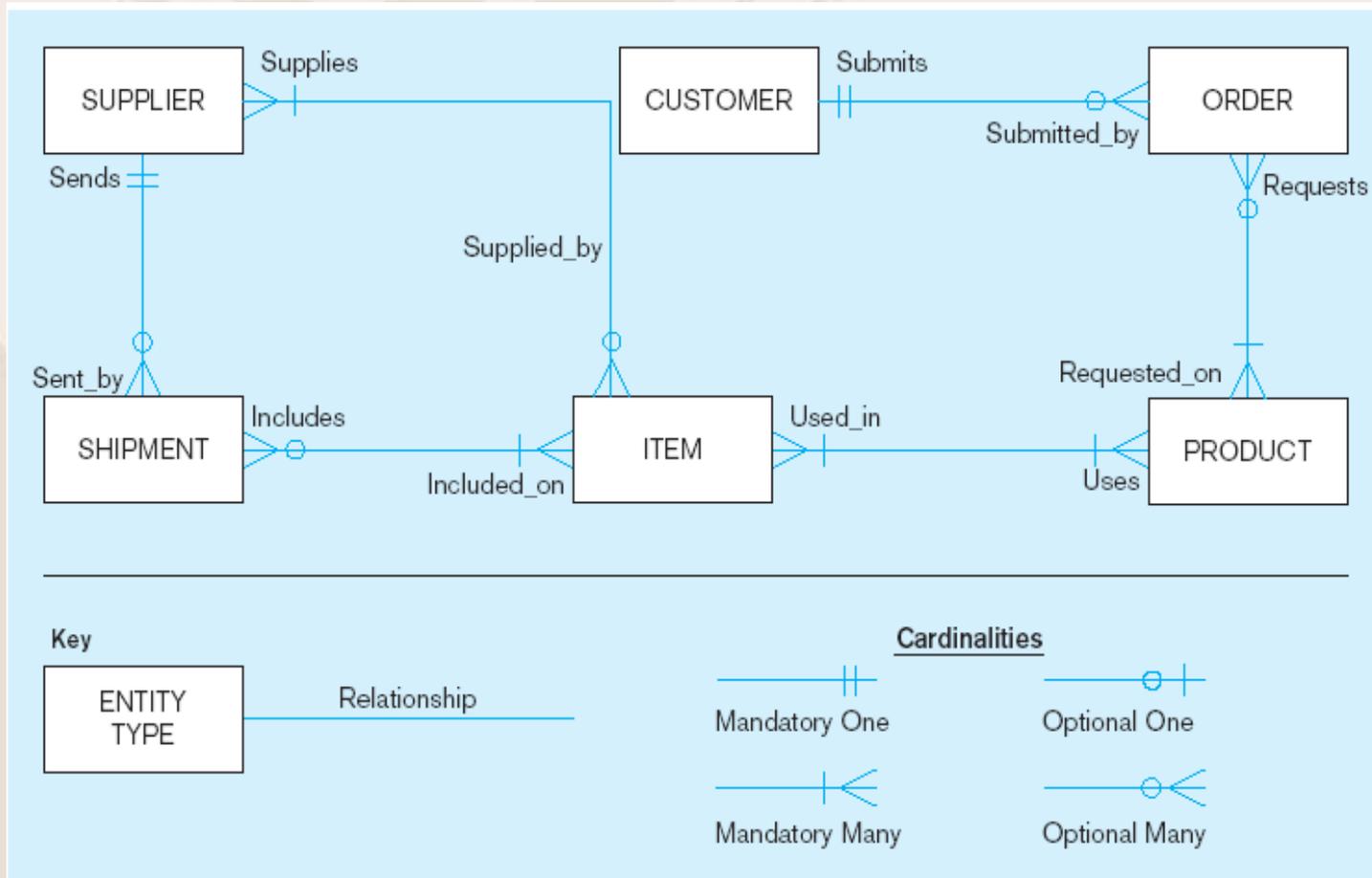


Management Information Systems

ERD

ERD

- Examples





Database Models

- A database model is a collection of logical constructs used to represent the data structure and the data relationships found within the database.
- Two Categories of Database Models
 - Conceptual models focus on the logical nature of the data representation. They are concerned with *what* is represented rather than *how* it is represented.
 - Implementation models place the emphasis on *how* the data are represented in the database or on *how* the data structures are implemented.



Steps to Construct ERD

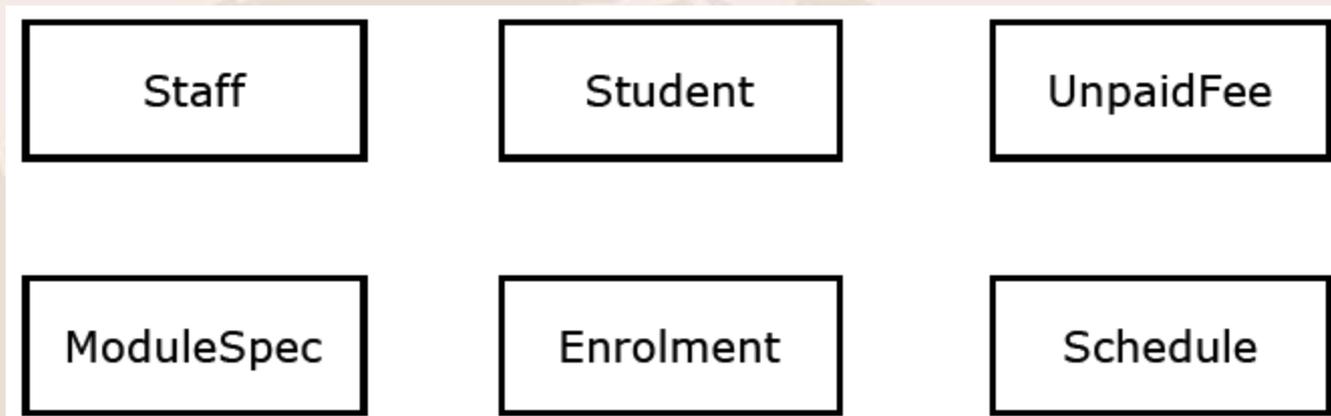
- Like constructing *DFDs* and *data dictionaries*, the construction of ERDs is an iterative and evolutionary process that requires refinement as a better understanding of the system to be developed is achieved.
 1. **Identify entities** – use DFDs to identify multiple instance entities.
 2. **Add attributes** – requires detailed analysis and perhaps use of requirements gathering techniques.
 3. **Assign identifiers or Primary keys** – Each entity type must have an attribute or set of attributes that distinguishes one instance from other instances of the same type.
 4. **Foreign Keys** - The attributes of a foreign key in one entity are the attributes of a primary key in another entity.
 5. **Identify Relationships** –



Steps to Construct ERD

Example: Create a logical data model (based on an ERD) for the University Registration System.

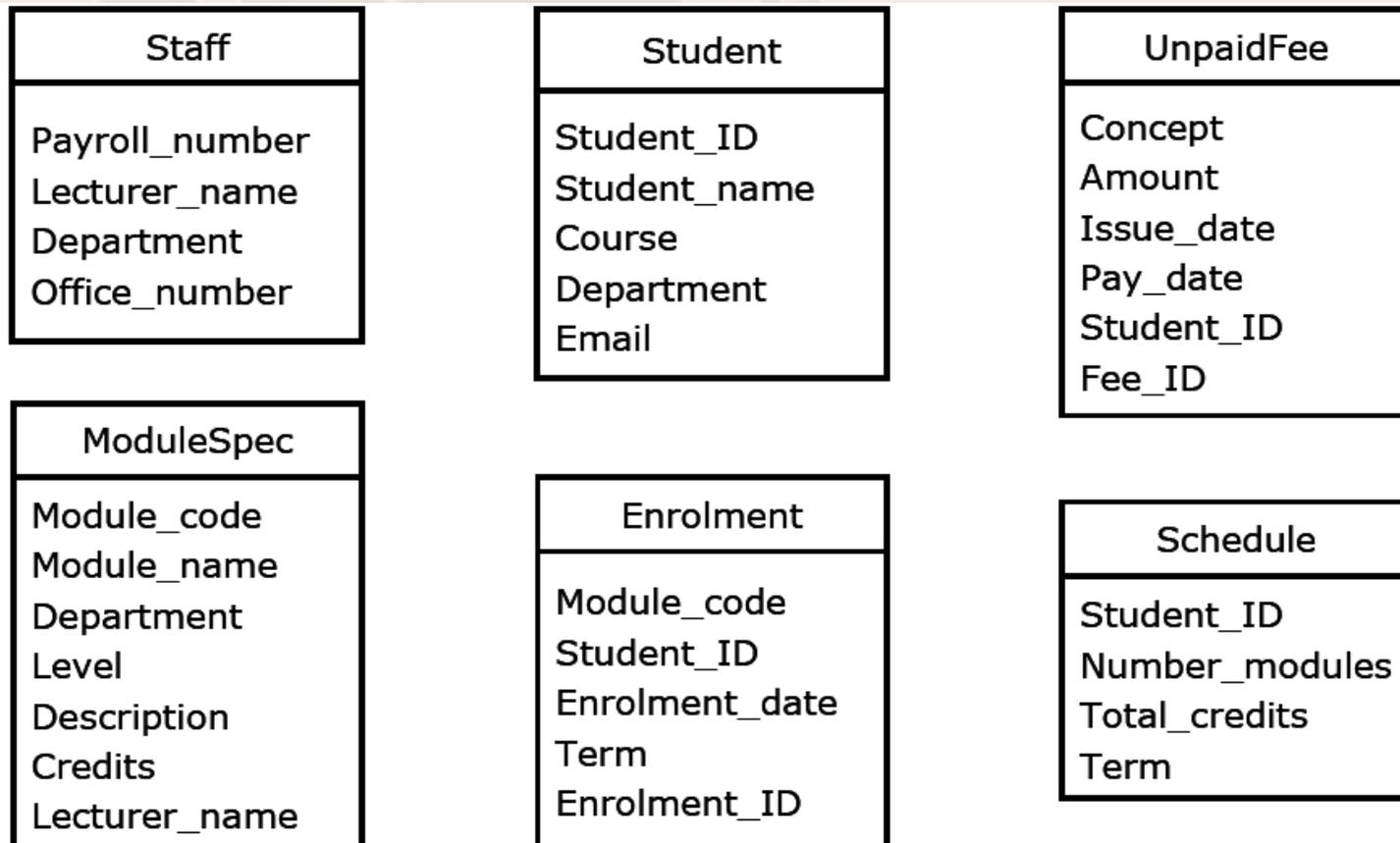
1. Identify Entities





Steps to Construct ERD

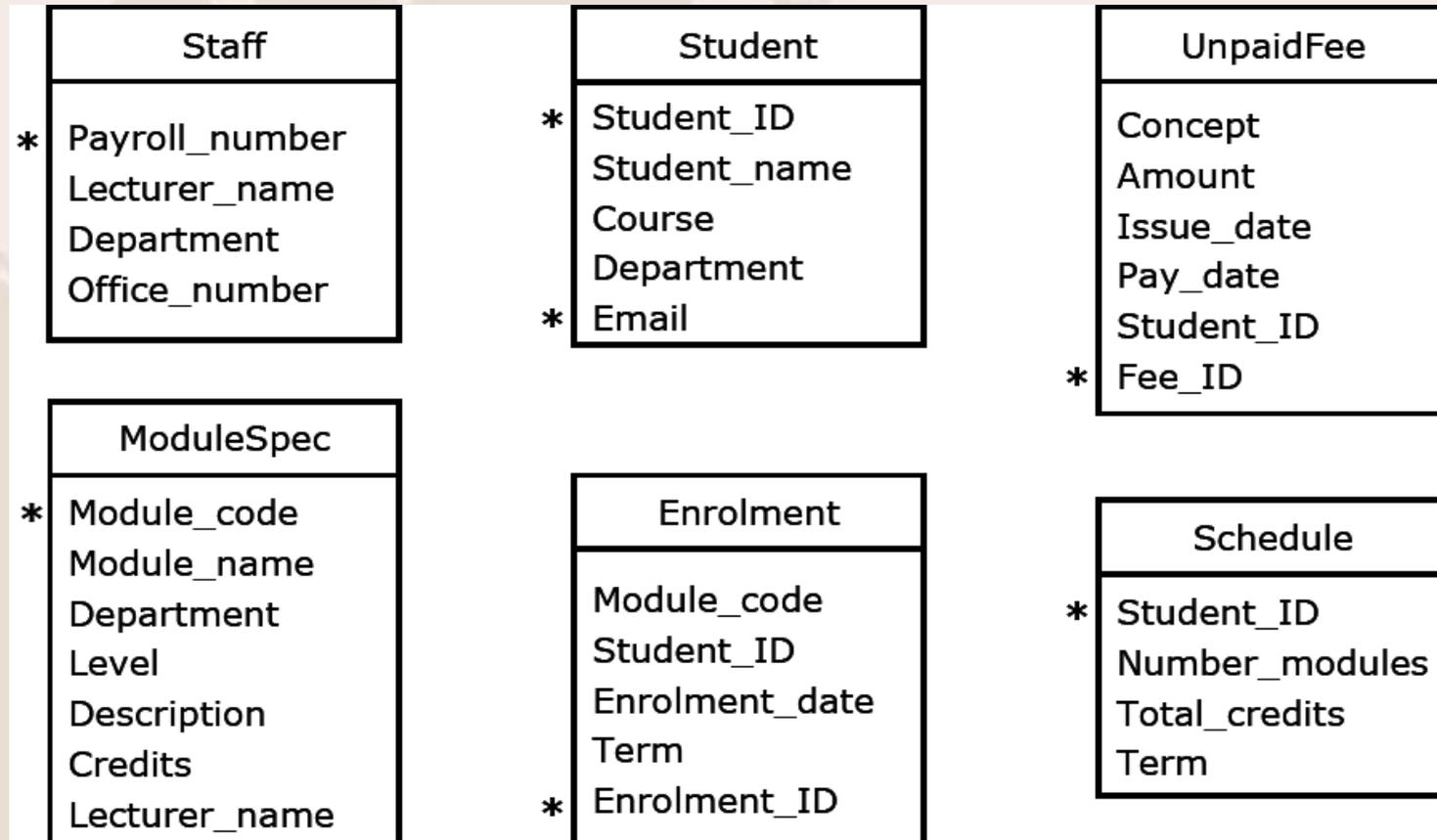
2. Add Attributes





Steps to Construct ERD

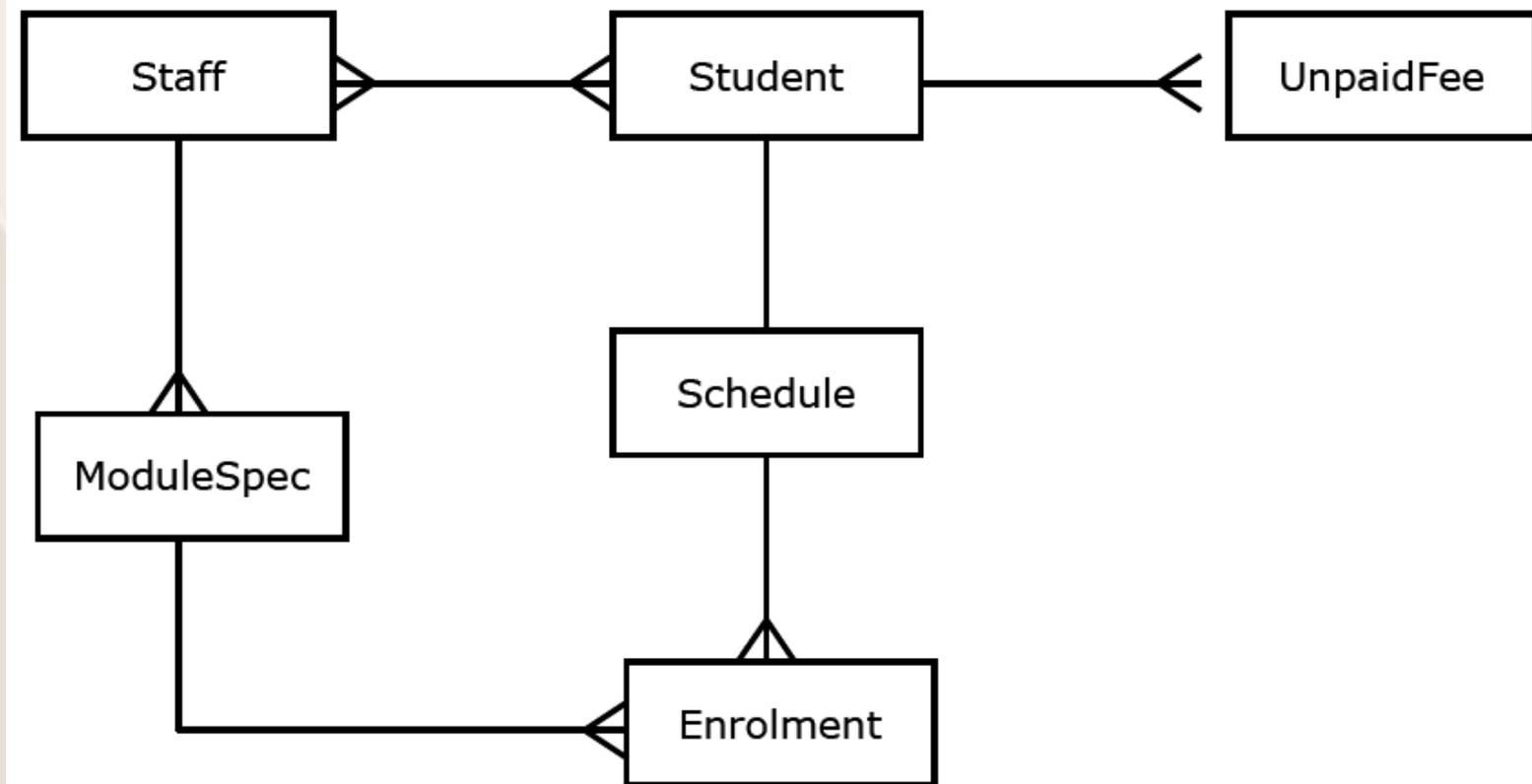
3. Assign Identifiers





Steps to Construct ERD

4. Identify Relationships





ERD

- **Three Types of Relationships**

- **One-to-one relationships (1:1)**

- Each store is managed by a single employee and each store manager (employee) only manages a single store.

- EMPLOYEE (1) manages STORE (1)

- **One-to-many relationships (1:M)**

- A painter paints many different paintings, but each one of them is painted by only that painter.

- PAINTER (1) paints PAINTING (M)

- **Many-to-many relationships (M:N)**

- An employee might learn many job skills, and each job skill might be learned by many employees.

- EMPLOYEE (M) learns SKILL (N)



Entity Relationship Diagrams for Data Modelling

- Implement One-to-One Relationships
 - In a relational database, *one-to-one relationships* between entities can be implemented by merging the two tables/entities together.
- Implement One-to-Many Relationships
 - Place the *primary key* of the entity that can occur only once as a *foreign key* in the entity that can occur many times.
- Implement Many-to-Many Relationships
 - In a relational database, *many-to-many relationships* between entities can be implemented by creating a new entity that links the other two entities together. In most cases, the primary key of the new entity consists of the primary keys of the entities that it is relating together.
 - Spit it into two one-to-many relationships!



Transforming ERD to Tables

- Applying these rules ensures all relationships simplify to 1:N
- 1:1 - Relationships merge into one table. One entity becomes an attribute of the other
- 1:N - Add primary key from the “one” entity as a foreign key in the “many” entity
- N:M - Add intersection table which includes primary keys from both tables

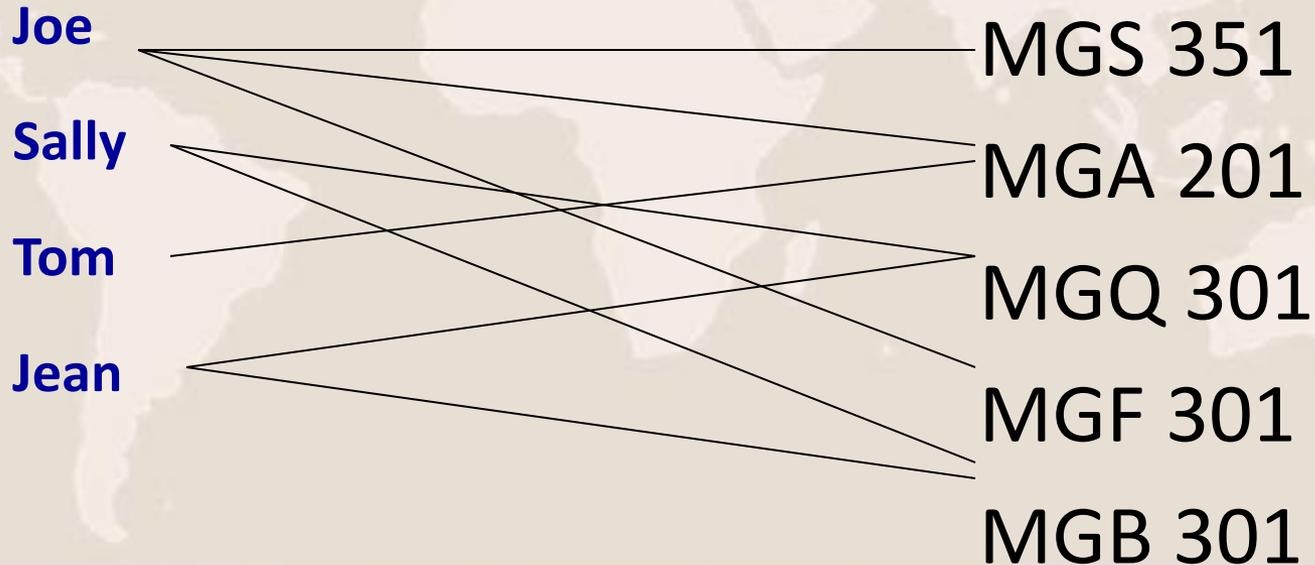


Multiple Relationships





Many-to-Many Relationship (N:M)





Transforming ERD to Tables – N:M Detailed Steps

- 1. Add intersection table**
- 2. Flip relationship lines around so “many” side faces the intersection table**
- 3. Add primary keys from original tables as fields in the intersection table**
- 4. Assign a name the intersection table**
- 5. Add other relevant fields to intersection table if necessary**



Transforming ERD to Tables – N:M (Before)

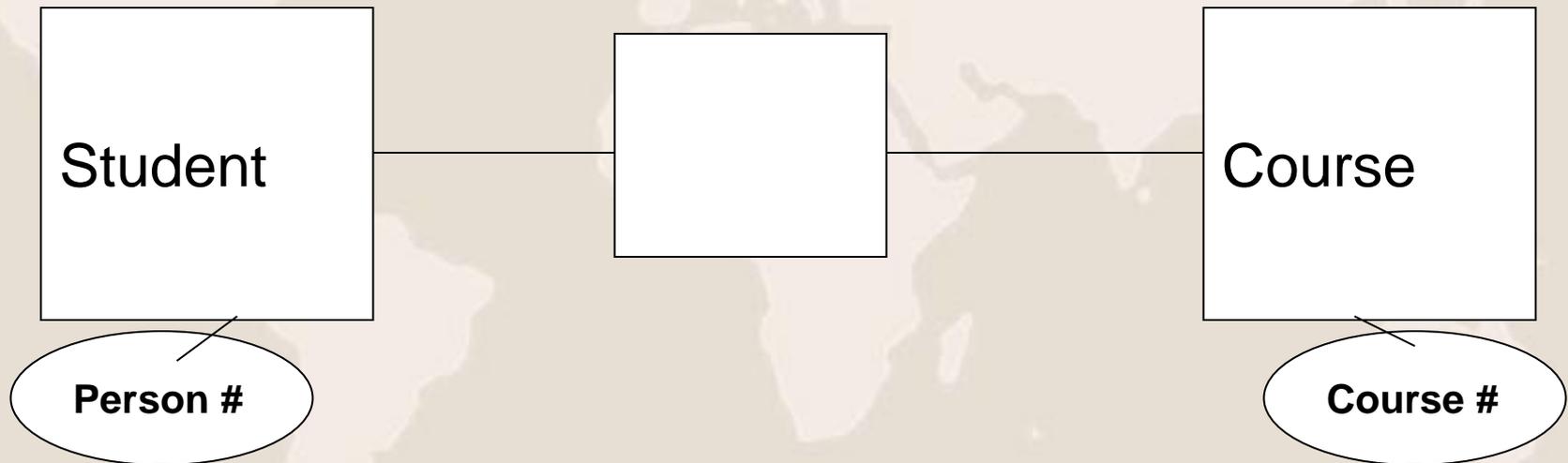
- **M:N - Add intersection table which includes primary keys from both tables**





Transforming ERD to Tables – N:M (After)

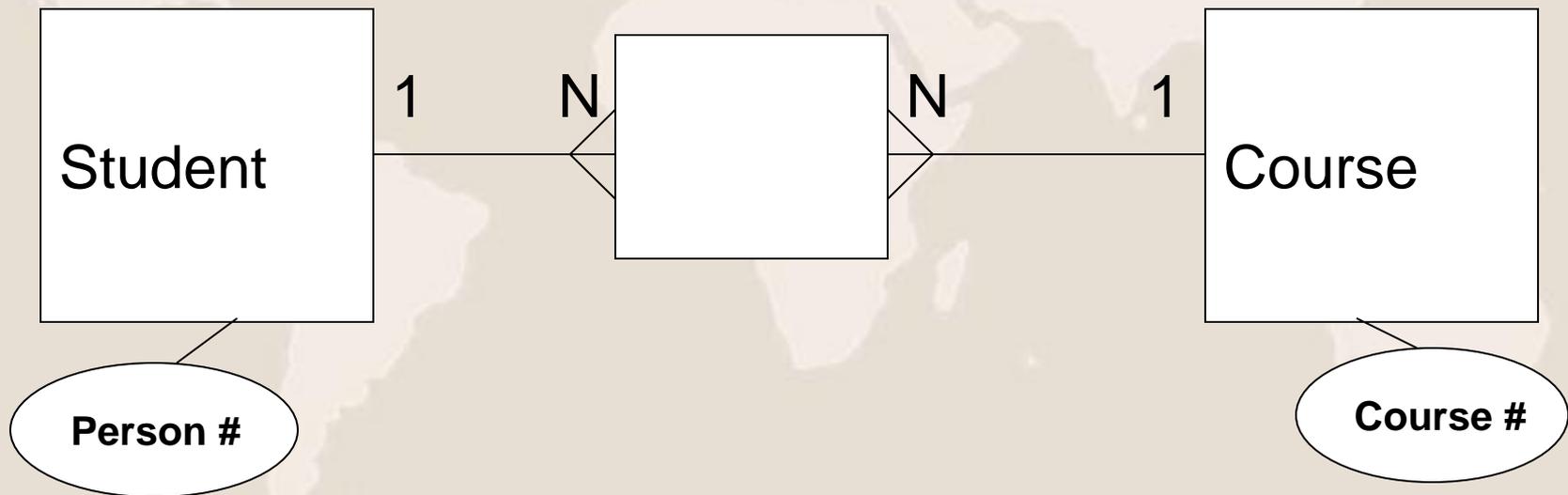
- Add intersection table





Transforming ERD to Tables – N:M (After)

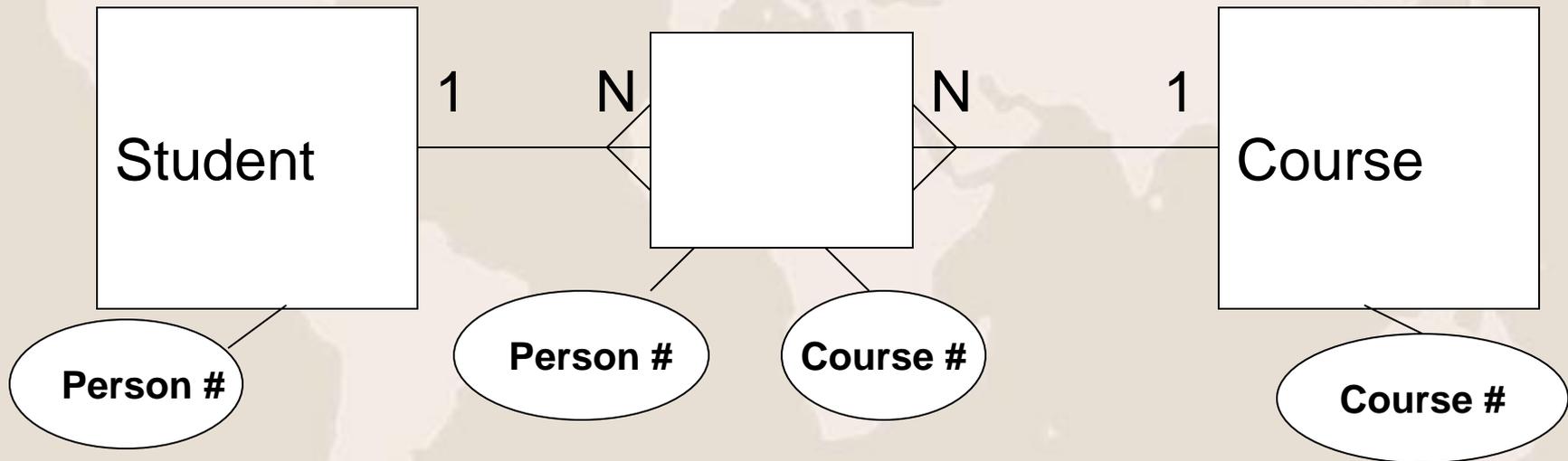
- Flip relationship lines around so “many” side faces the intersection table





Transforming ERD to Tables – N:M (After)

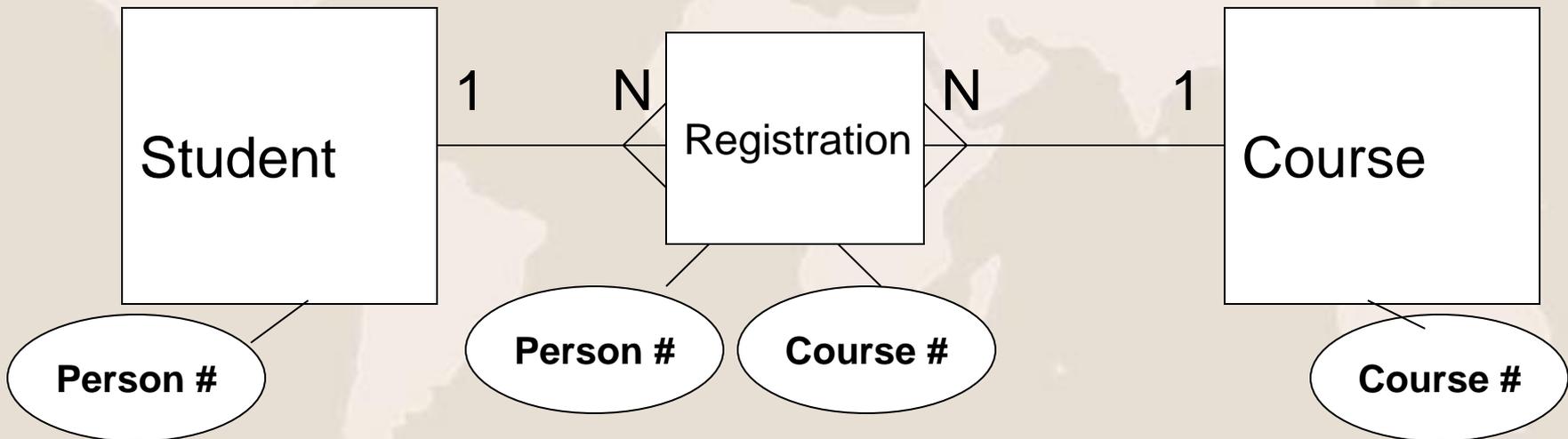
- Add primary keys from original tables as foreign key fields in the intersection table





Transforming ERD to Tables – N:M (After)

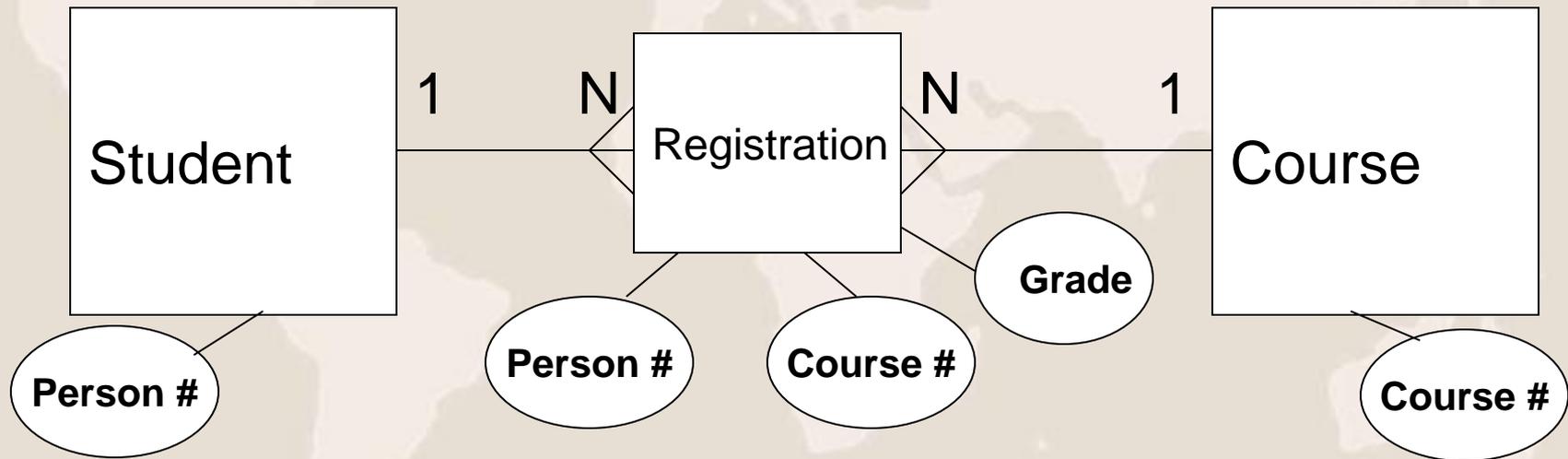
- Assign a name the intersection table





Transforming ERD to Tables – N:M (After)

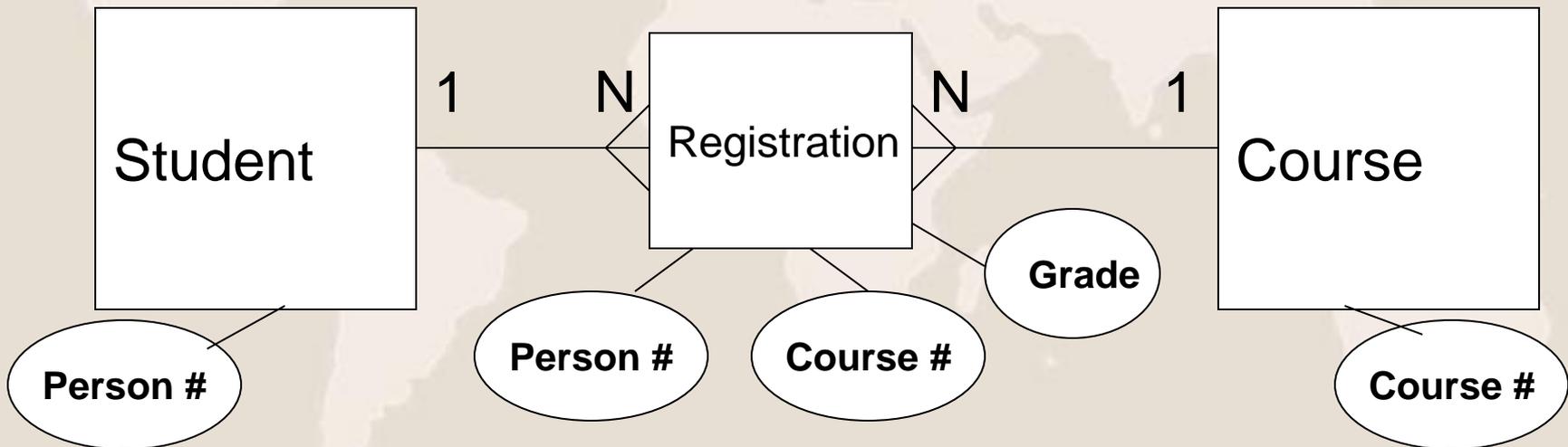
- Add other relevant fields to intersection table if necessary





What should be used as Primary Key in the intersection table?

Composite Primary Key or Auto-number





Steps to Construct ERD

- **Important guidelines when creating ERDs:**
 - Use consistent labels and names
 - Avoid unnecessary attributes
 - Break-down the model if needed
 - Determine cardinality correctly



Validation of Entity Relationship Diagrams

- The **DFD** and **ERD** are balanced when:
 - There is a **one-to-one** correspondence between data stores in the DFD and entities in the ERD.
 - Data flows in the DFD are made of data elements corresponding to attributes in the ERD.



پایان

