



Information systems

Unit 5 Database design

Database design

- Entity-Relationship model
- Conceptual design
- Logic design
- Normalization



Database design

Entity-Relationship Model

Entity-Relationship model

- Life cycle of an information system
- Databases design
- Entities and relationships
- Attributes
- Identifiers
- Generalization
- E-R schema documentation
- UML and E-R



Entity-Relationship model

**Life cycle of an information
system**

Databases design

- The design of a database is one of the activities in the development of an information system
 - must be considered in the broader context of the life cycle of an information system

Life cycle of an information system

Feasibility study

Life cycle of an information system

➤ Feasibility study

- determination of the costs of the various alternatives and of the implementation priorities of the system components

Life cycle of an information system

Feasibility study



Collection and analysis
of the requirements

Life cycle of an information system

- Collection and analysis of the requirements
 - definition of properties and functions of the information system
 - requires interaction with the user
 - produces a complete but informal description of the system to be implemented

Life cycle of an information system

Feasibility study



Collection and analysis
of the requirements



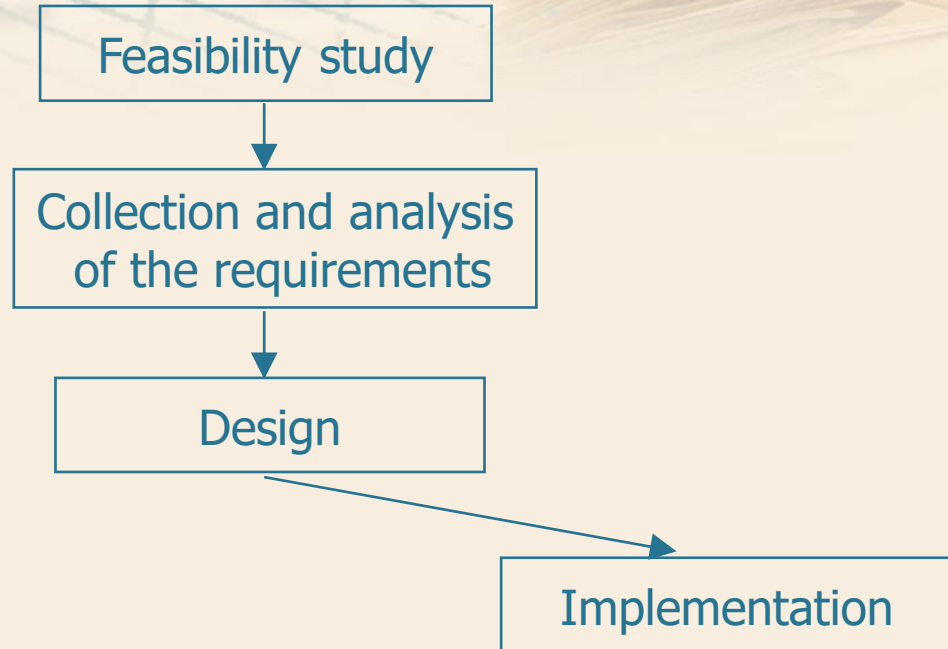
Design

Life cycle of an information system

➤ Design

- divided into data and application design
- produces formal descriptions

Life cycle of an information system

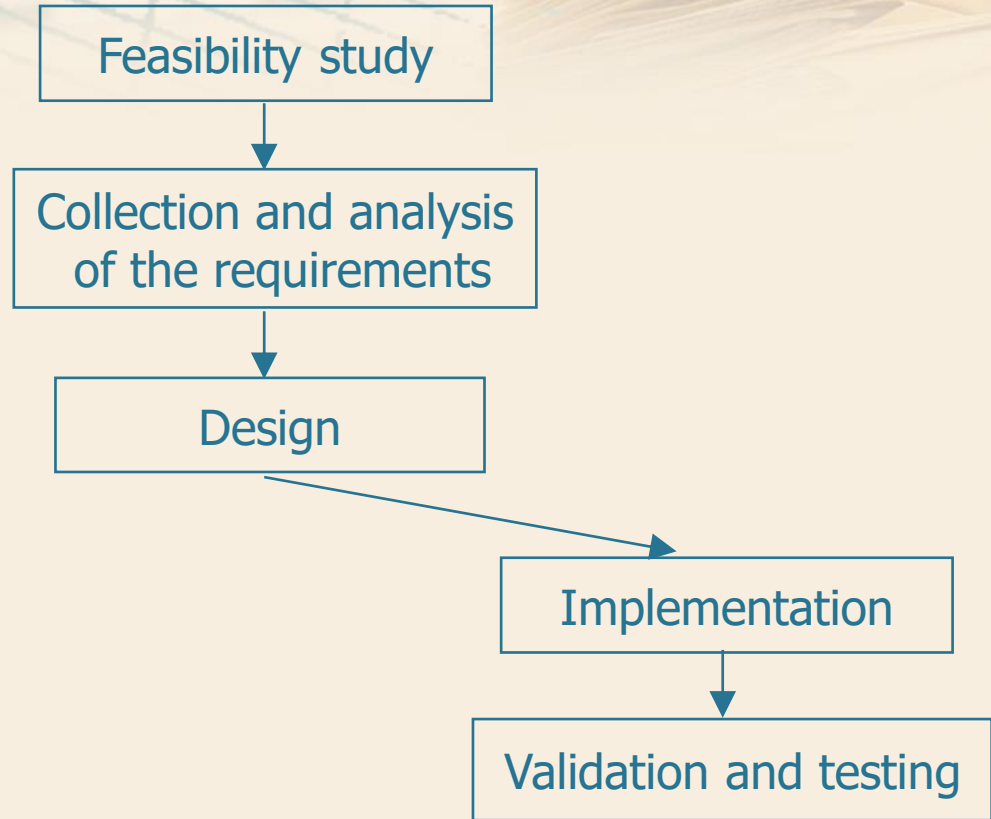


Life cycle of an information system

➤ Implementation

- creation of the information system according to the characteristics defined in the design phase

Life cycle of an information system

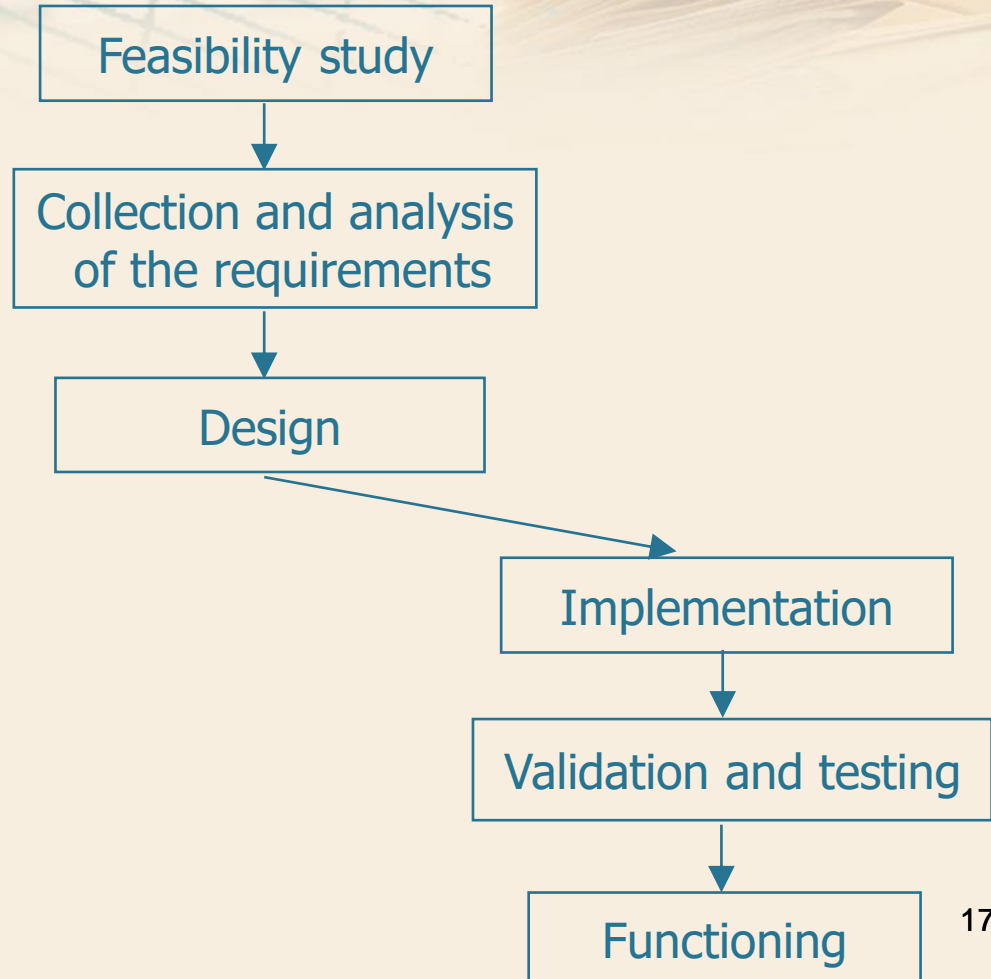


Life cycle of an information system

➤ Validation and testing

- verification of the correct functioning and quality of the information system
- can lead to changes in requirements or a revision of the design

Life cycle of an information system

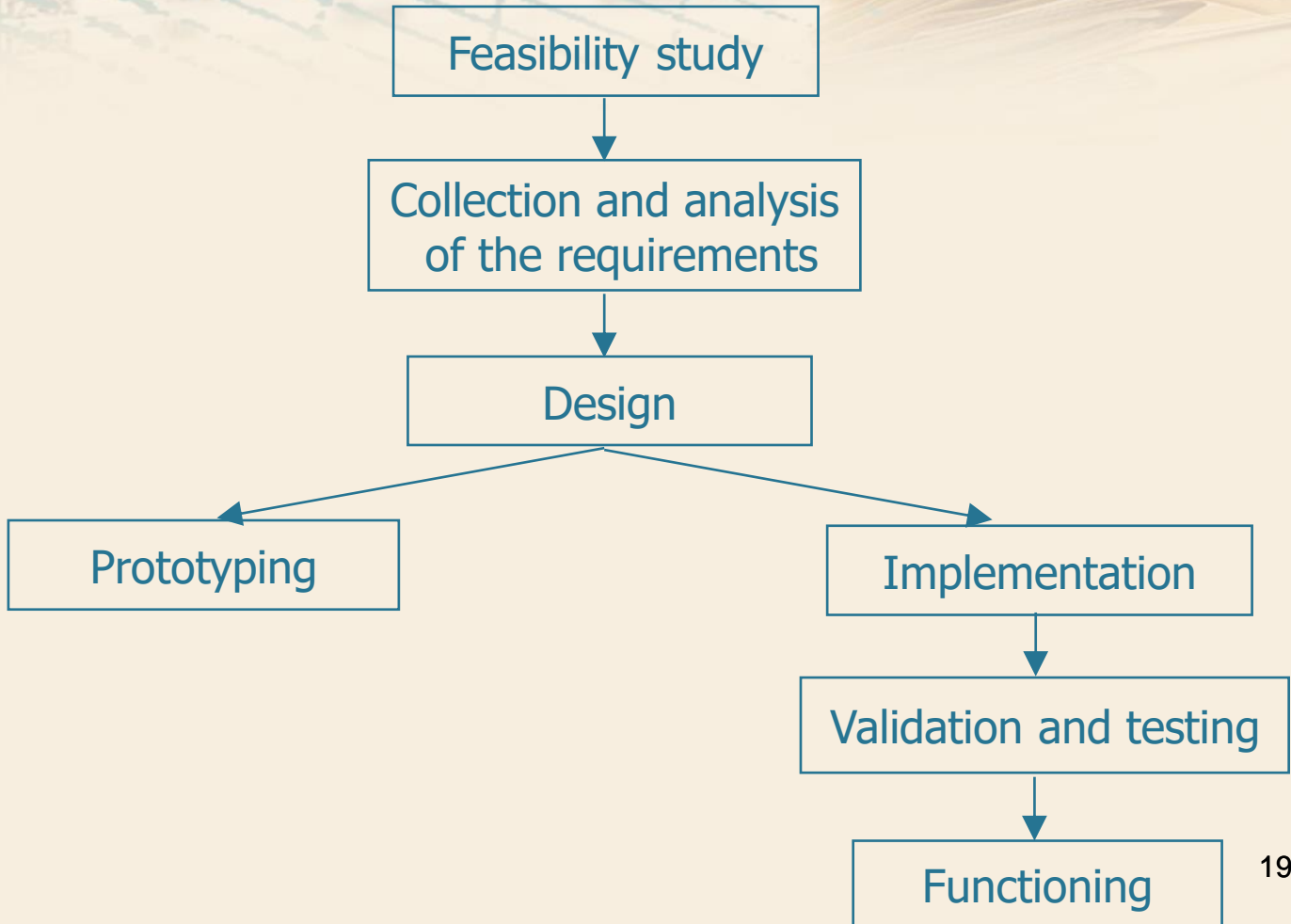


Life cycle of an information system

➤ Functioning

- system operation
- requires management and maintenance operations

Life cycle of an information system

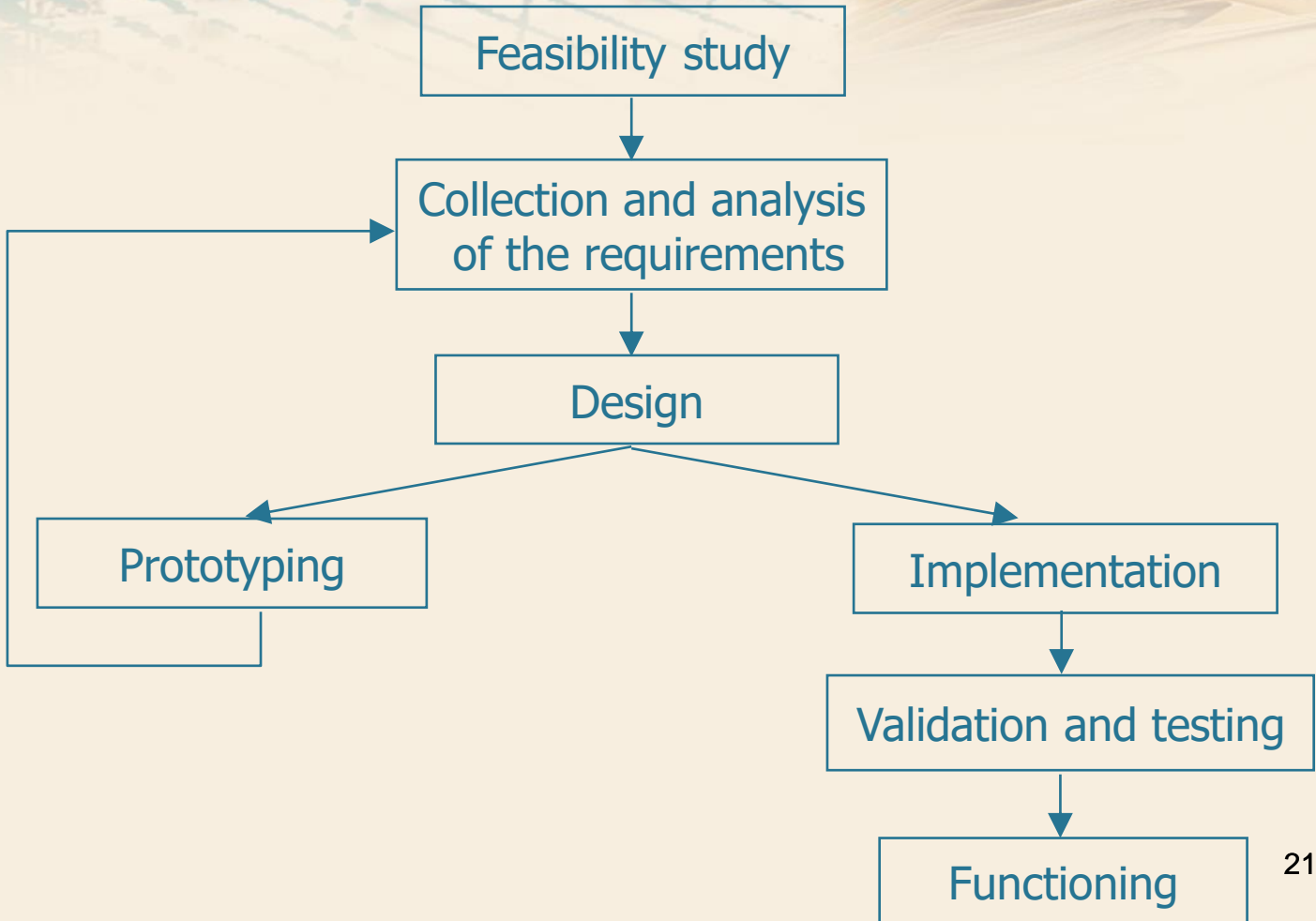


Life cycle of an information system

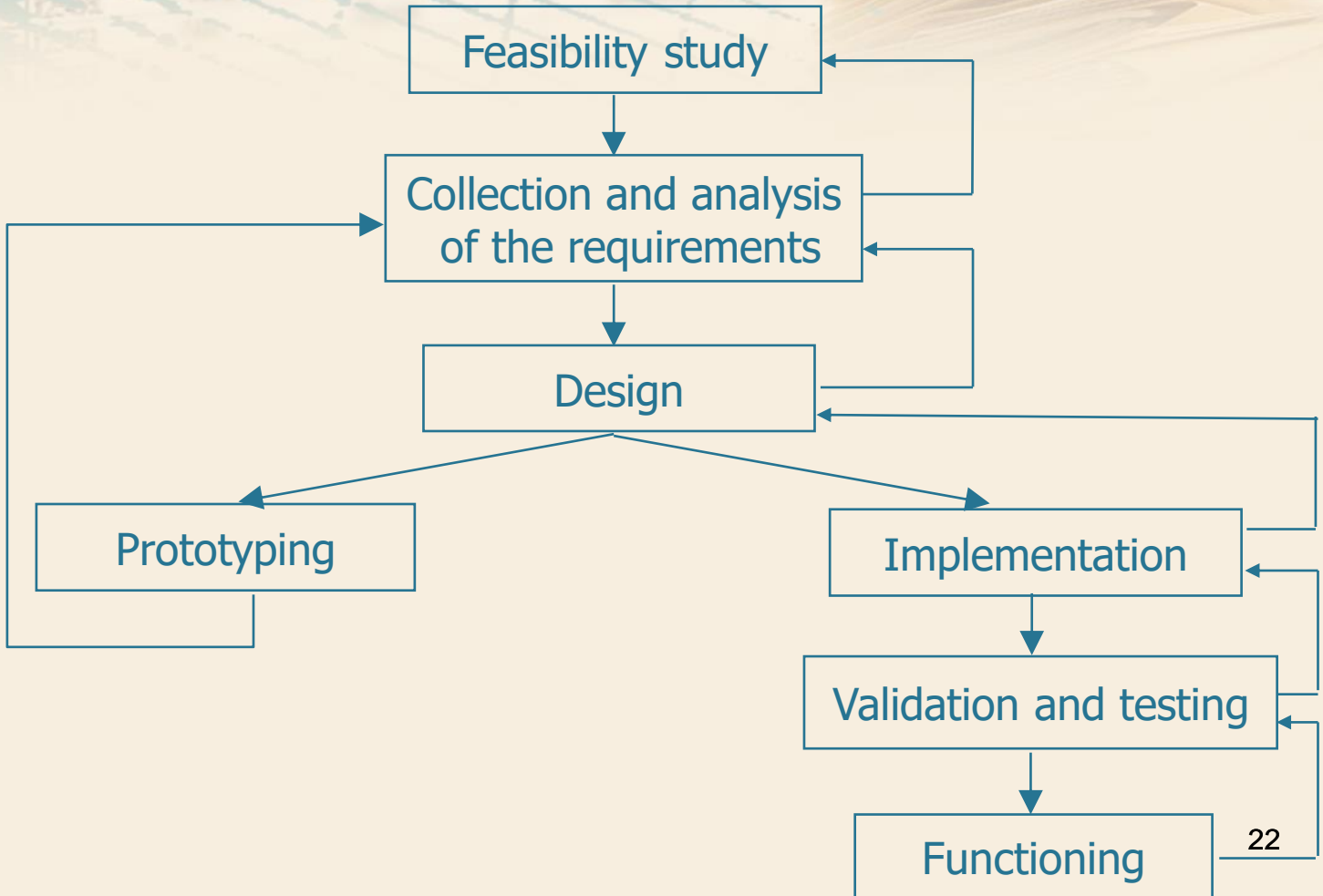
➤ Prototyping

- rapid creation of a simplified version of the system in order to evaluate its characteristics
- can lead to changes in requirements or a revision of the design

Life cycle of an information system



Life cycle of an information system





Entity-Relationship model

Database design

Database design

- The database is an important component of the overall system
- Data-driven design methodology
 - database design precedes the design of the applications that use it
 - greater attention to the design phase than the other phases

Design methodology

- A design methodology consists of
- decomposition of the project activity into successive independent steps
 - strategies to be followed in the various steps and criteria for choosing strategies
 - reference models to describe the input and output data of the various phases

Design methodology: Example

- Athletic training
 - Activity decomposition
 1. physical condition
 - 2a. enhancement
 - 2b. velocity

Design methodology: Example

➤ Athletic training

- activity decomposition
- strategies to follow in the various steps
 1. A) food diet
B) exercises in order to reduce the percentage of body fat
 - 2a. A) strength exercises
B) resistance exercises

Design methodology: Example

➤ Athletic training

- activity decomposition
- strategies to follow in the various steps
- reference models to describe the input and output data of the various phases
 1. input data: actual weight, % of body fat
output data: model of the body structure of the fit person
 - 2a. input data: fit person model
output data: body structure model of the average athlete

Properties of the methodology

➤ Generality

- possibility of use regardless of the problem and the tools available

➤ Quality of result

- in terms of correctness, completeness and efficiency with respect to the resources used

➤ Ease of use

- of both strategies and reference models

Data-driven design

- For databases, methodology based on the separation of decisions
 - *what* to represent in the database
 - conceptual design
 - *how* to represent it
 - logical and physical design

Stages of database design

Application
requirements



Application requirements

- Informal specifications of the reality of interest
 - application properties
 - application functionalities

Stages of database design

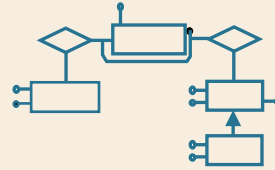
Application requirements



Conceptual design



Conceptual schema



Conceptual design

- Representation of informal specifications in the form of a *conceptual schema*
- formal and complete description, which refers to a conceptual model
 - independence from implementation aspects (data model)
 - the target is the representation of the *information content* of the database

Stages of database design

Application requirements



Conceptual design



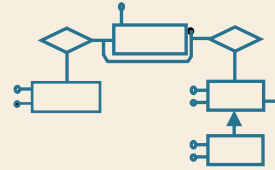
Conceptual schema



Logical design



Logical schema



- Translation of the conceptual schema into the logical schema
- refers to the chosen logical data model
 - criteria are used to optimize the operations which must be performed on the data
 - quality of the schema verified by formal techniques (normalization)

Stages of database design

Application requirements



Conceptual design



Conceptual schema



Logical design



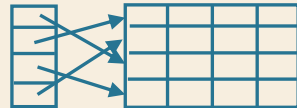
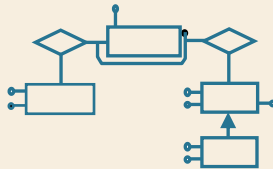
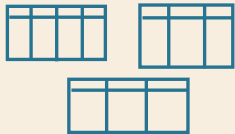
Logical schema



Physical design



Physical schema



Physical design

- Specification of physical data storage parameters (organization of files and indexes)
 - produces a physical model, which depends on the chosen DBMS

Stages of database design

Application requirements



Conceptual design



Conceptual schema



Logical design



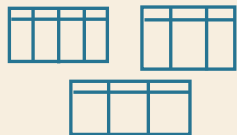
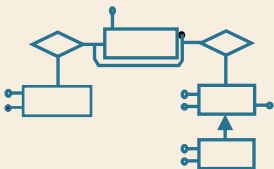
Logical schema



Physical design



Physical schema





Entity-Relationship model

Entities and relationships

E-R model (Entity-Relationship)

- It is the most widespread conceptual model
- Provides constructs to describe data structure specifications
 - in a simple and understandable way
 - with graphic formalism
 - regardless of the data model, which can be chosen later
- There are numerous variations

Main constructs of the E-R model

- Entities
- Relationships
- Attributes
- Identifiers
- Generalizations and subsets

Entity name

- Represents classes of real-world objects (people, things, events, ...), which they have
 - common properties
 - autonomous existence
- Examples: employee, student, item
- An occurrence of an entity is an object of the class that the entity represents



Relationship name

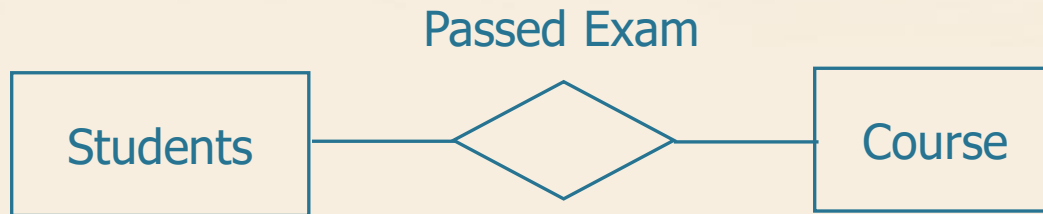
- Represents a logical link between two or more entities
- Examples: exam between student and course, residence between person and municipality
- Not to be confused with the relationship of the relational model
 - sometimes it is named association

Relationships examples

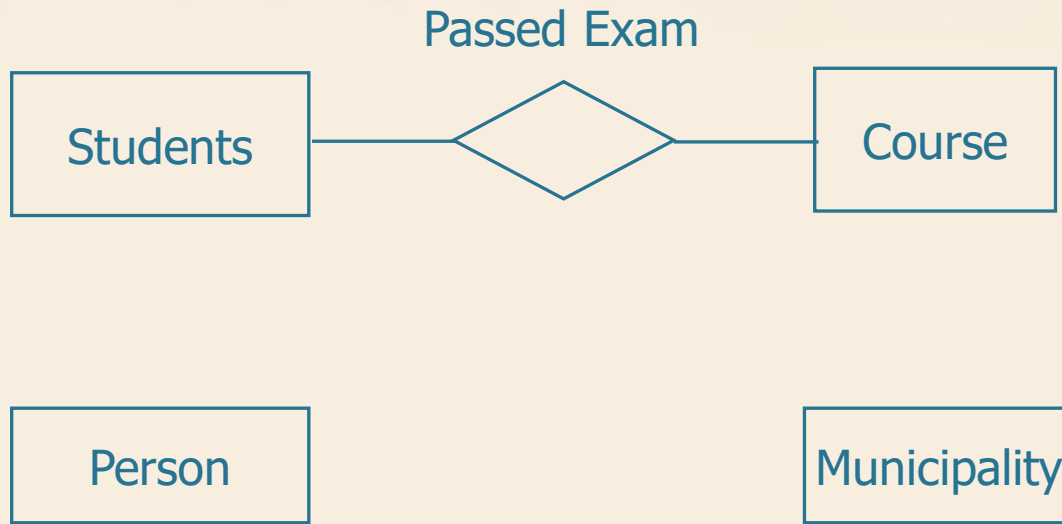
Student

Course

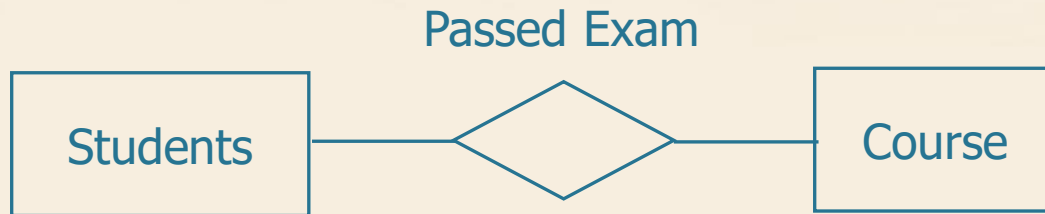
Relationships examples



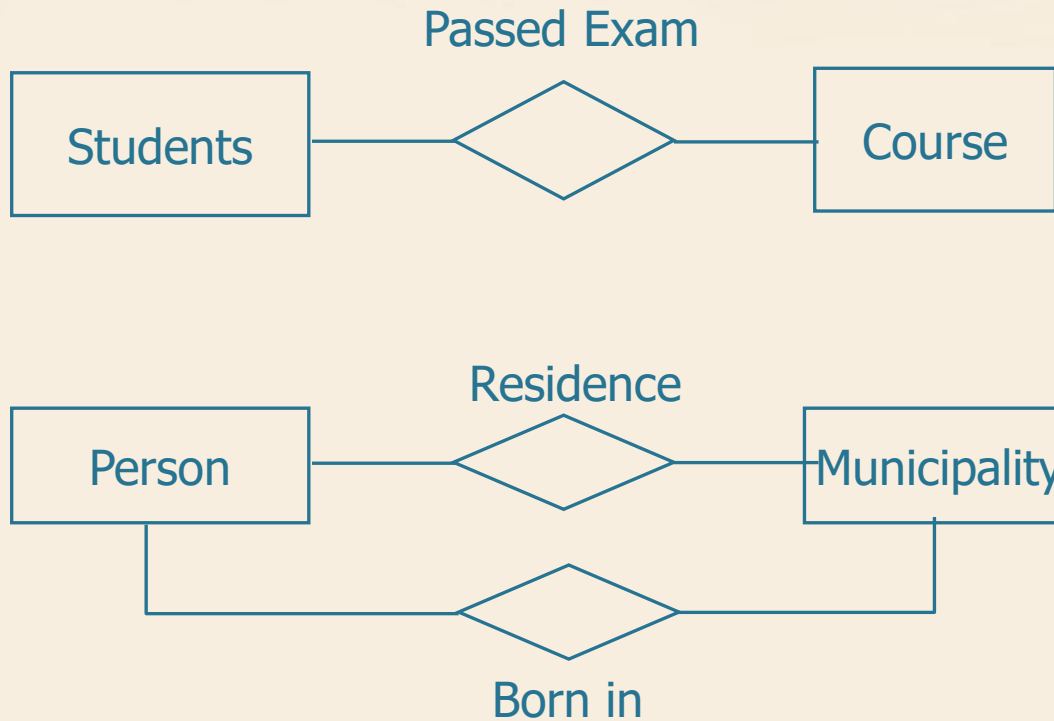
Relationships examples



Relationships examples

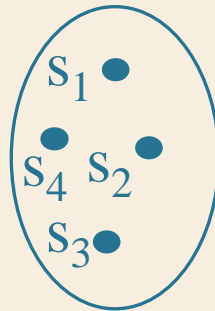


Relationships examples

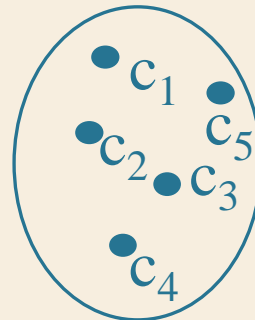


Occurrences of a relationship

Student

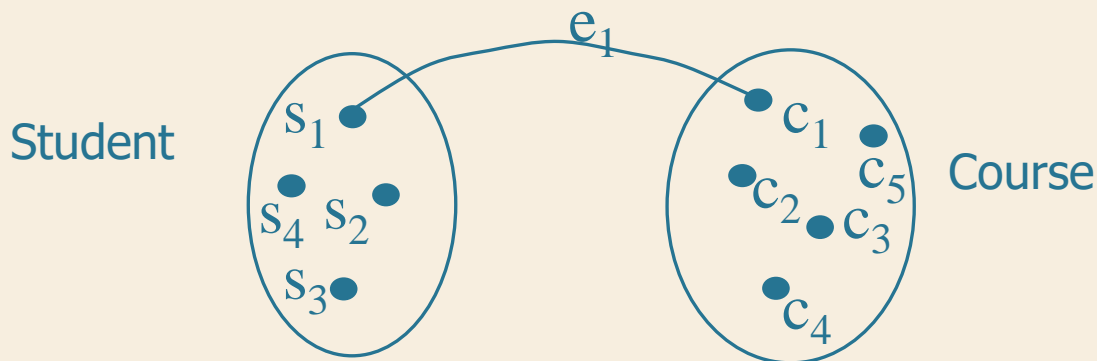


Course



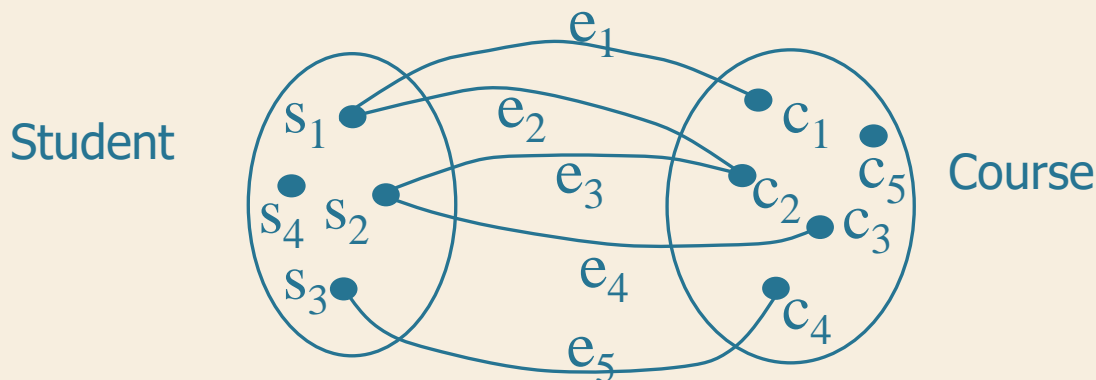
Occurrences of a relationship

- ⇒ An occurrence of a relationship is an n-tuple (pair in the case of a binary relationship) consisting of occurrences of entities, one for each of the entities involved



Occurrences of a relationship

- An occurrence of a relationship is an n-tuple (pair in the case of a binary relationship) consisting of occurrences of entities, one for each of the entities involved
- No identical n-tuples are allowed

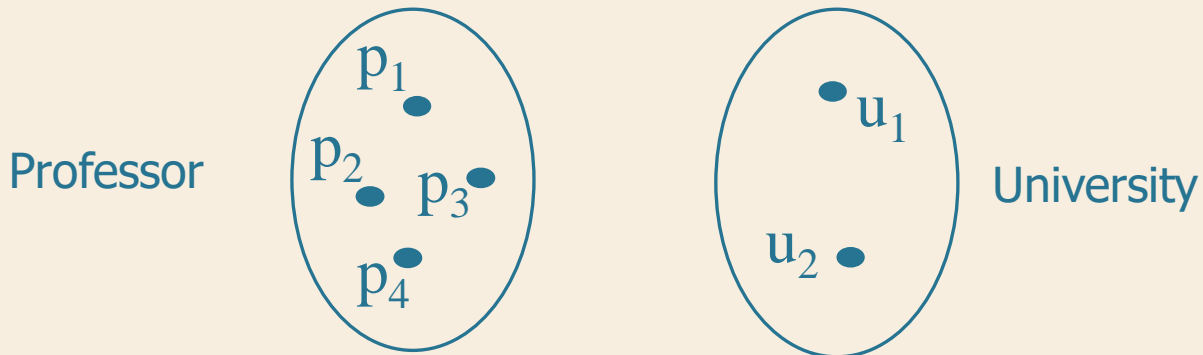


Cardinality of binary relationships

- They are specified for each entity participating in a relationship
- They describe the maximum and minimum number of relationship occurrences in which an entity occurrence can participate
 - **minimum cardinality**
 - 0 (optional participation)
 - 1 (mandatory participation)
 - **Maximum cardinality**
 - 1 (at most one occurrence)
 - N (arbitrary number of occurrences)

Cardinality of binary relationships

➤ 1 to 1 correspondence

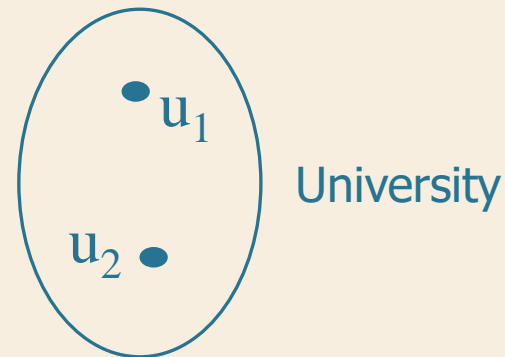
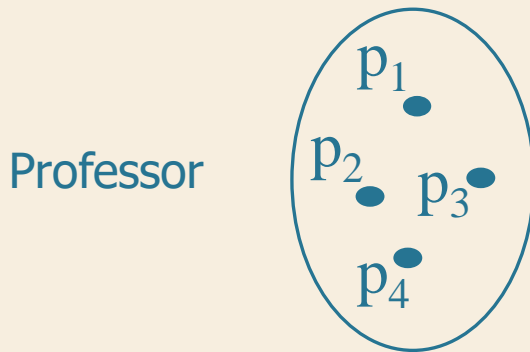


Cardinality of binary relationships

➤ 1 to 1 correspondence

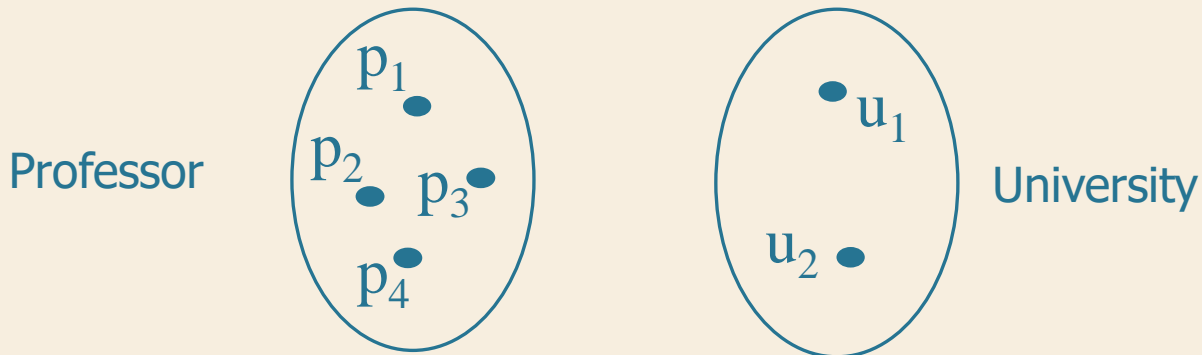
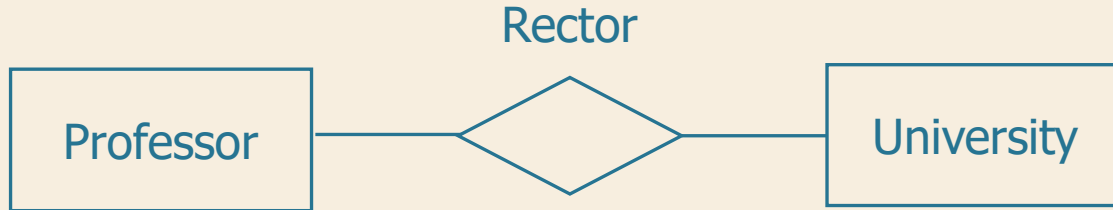
Professor

University



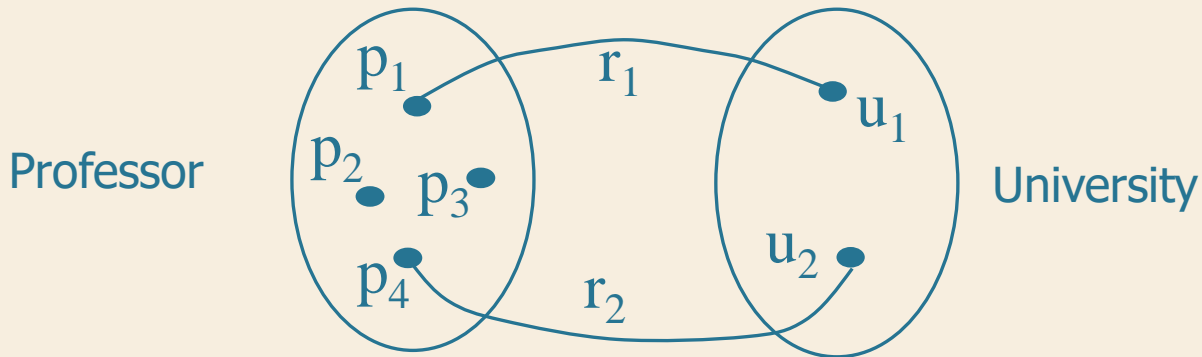
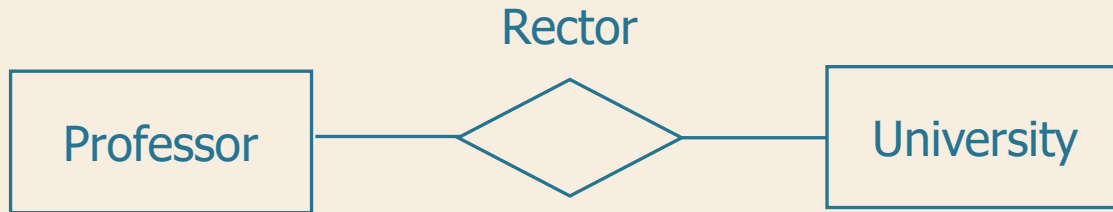
Cardinality of binary relationships

➤ 1 to 1 correspondence



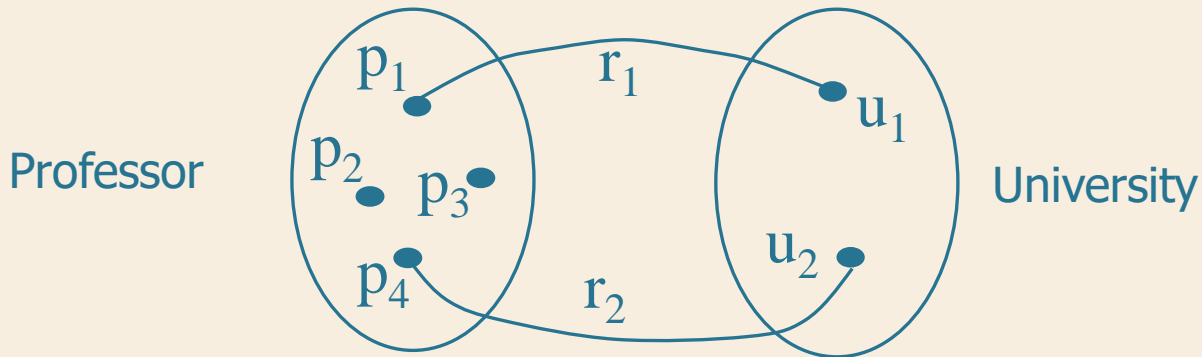
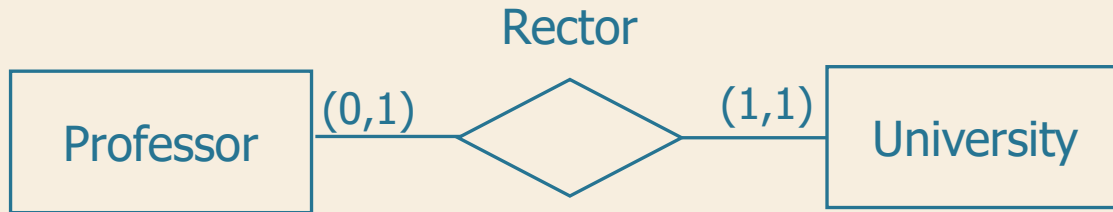
Cardinality of binary relationships

➤ 1 to 1 correspondence



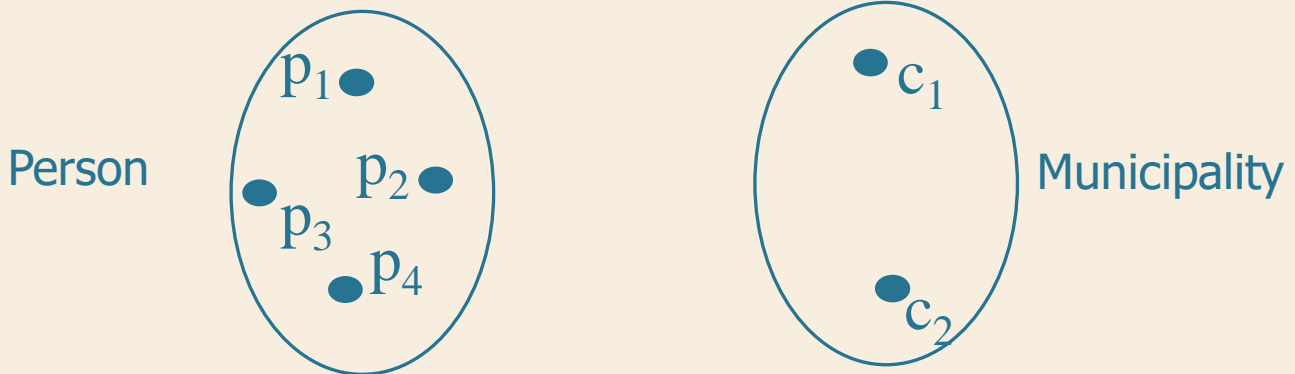
Cardinality of binary relationships

➤ 1 to 1 correspondence



Cardinality of binary relationships

➤ 1 to N correspondence



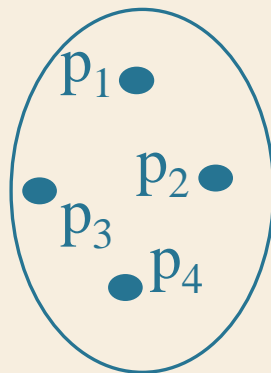
Cardinality of binary relationships

➤ 1 to N correspondence

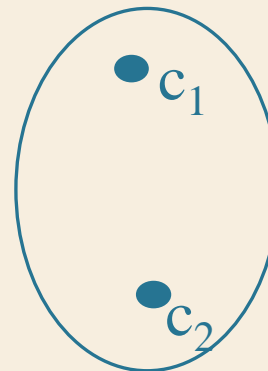
Person

Municipality

Person

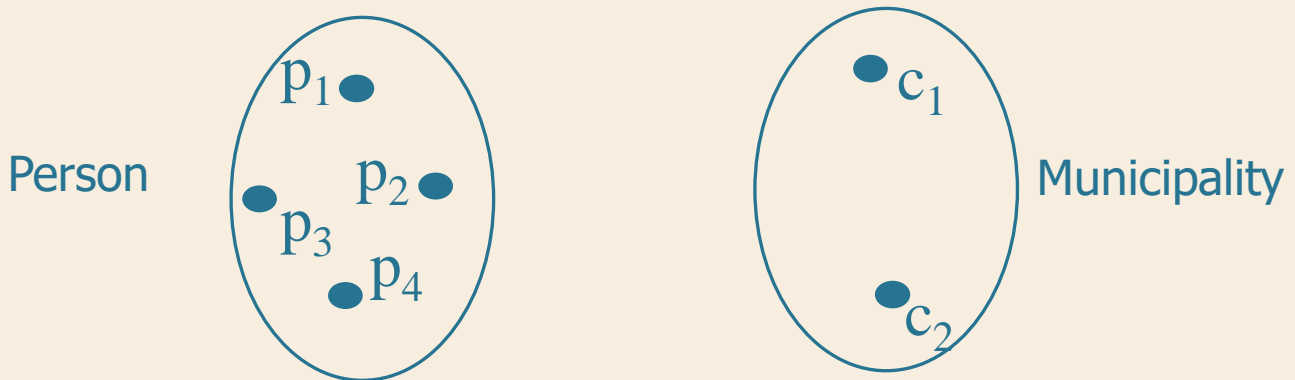
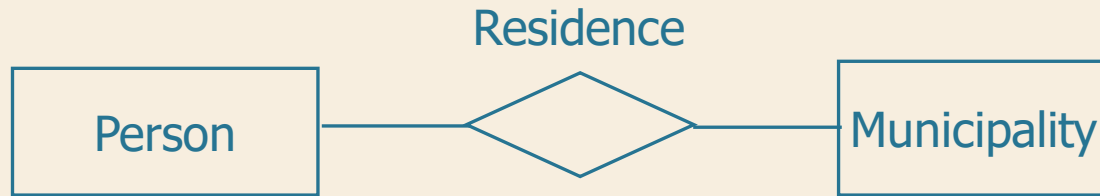


Municipality



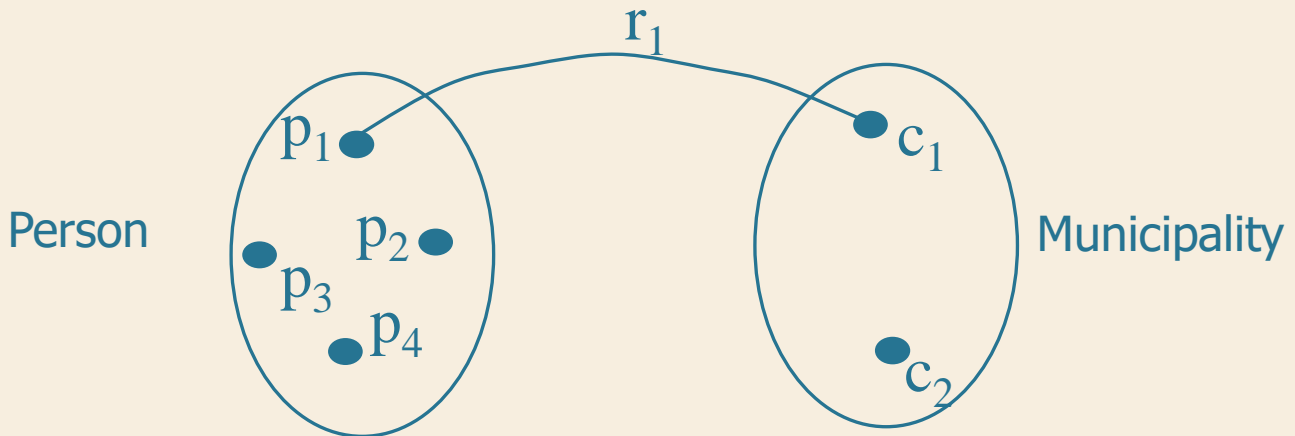
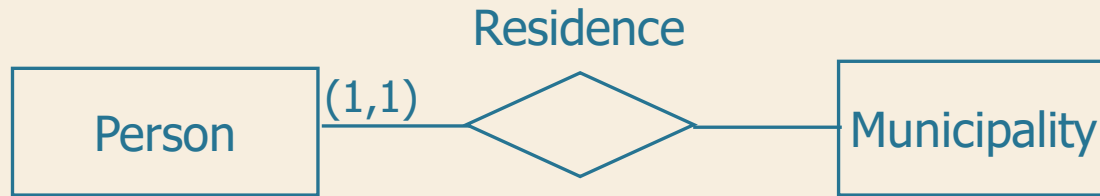
Cardinality of binary relationships

➤ 1 to N correspondence



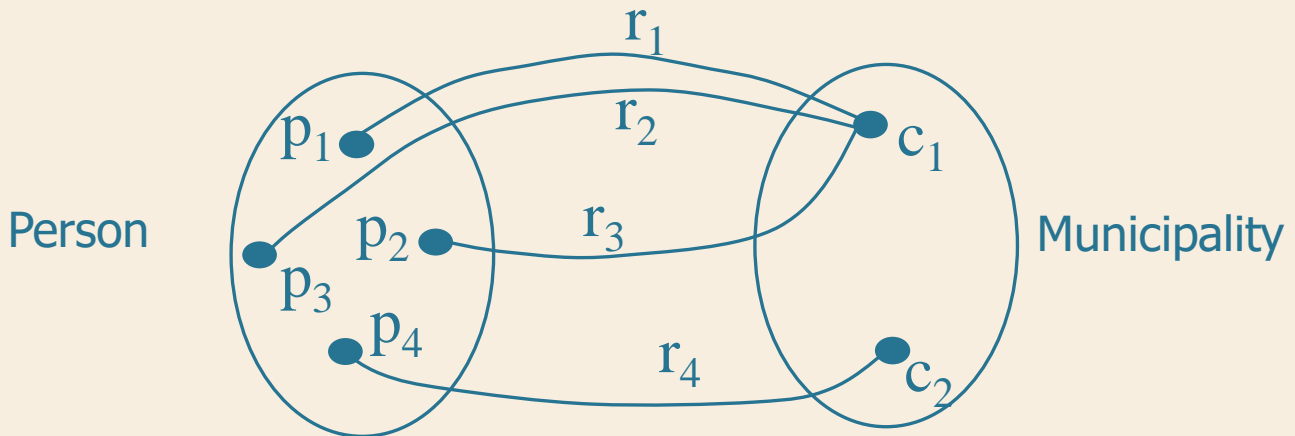
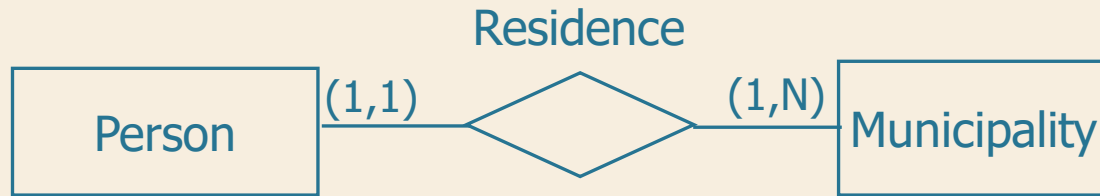
Cardinality of binary relationships

➤ 1 to N correspondence



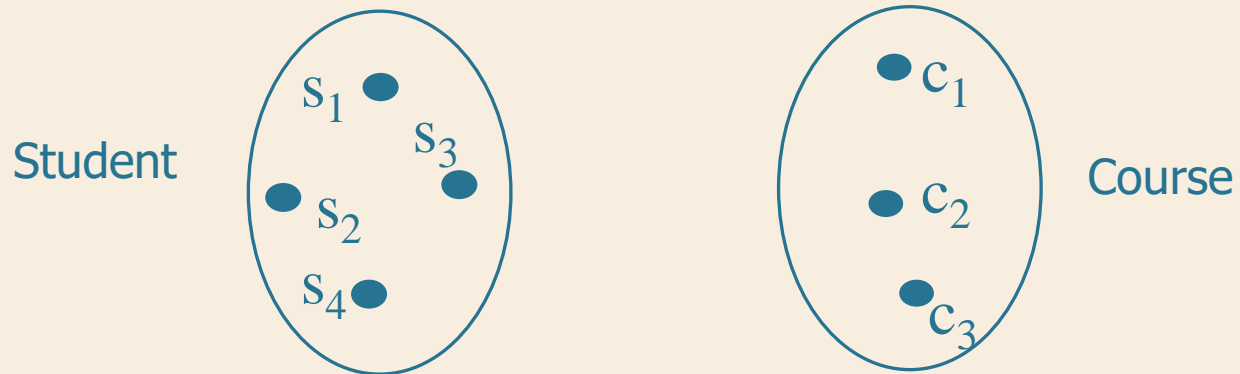
Cardinality of binary relationships

➤ 1 to N correspondence



Cardinality of binary relationships

➤ N to N correspondence



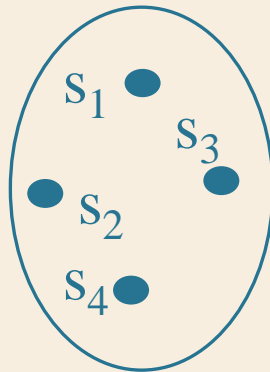
Cardinality of binary relationships

➤ N to N correspondence

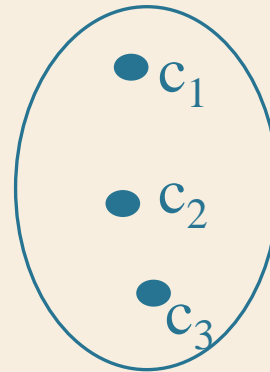
Student

Course

Student

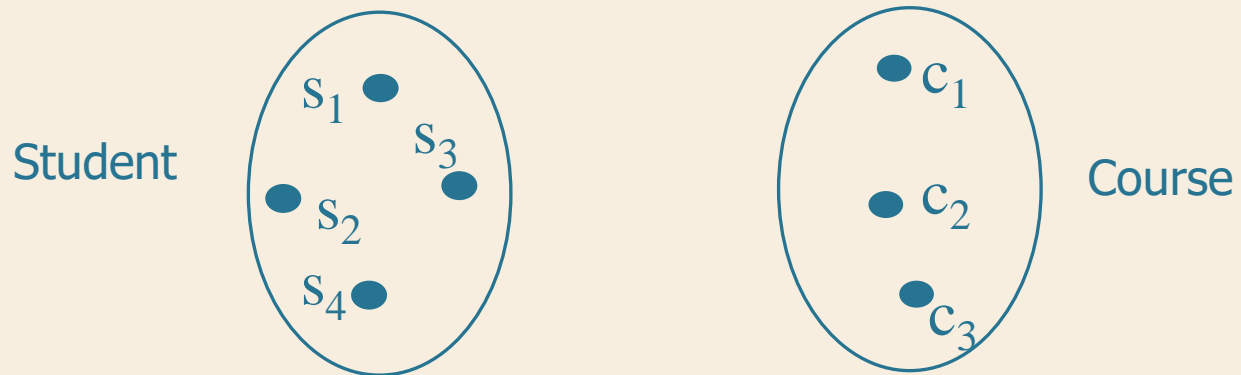
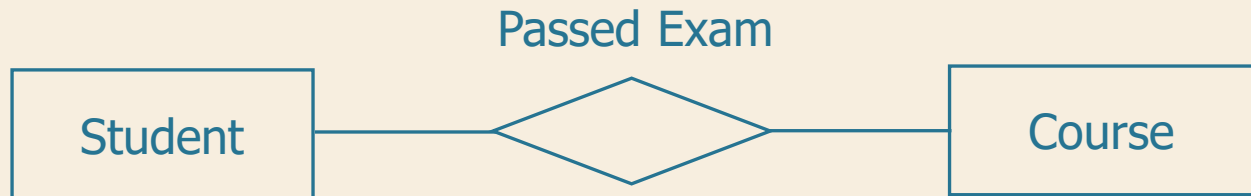


Course



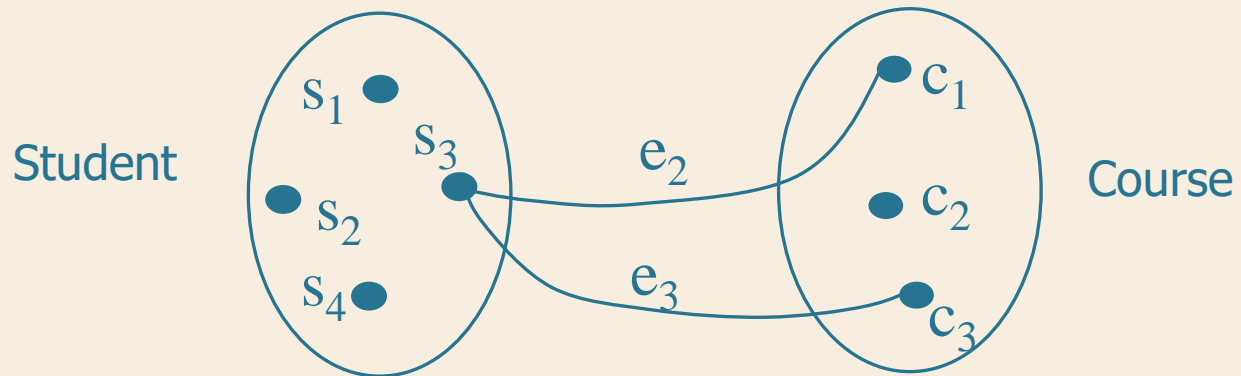
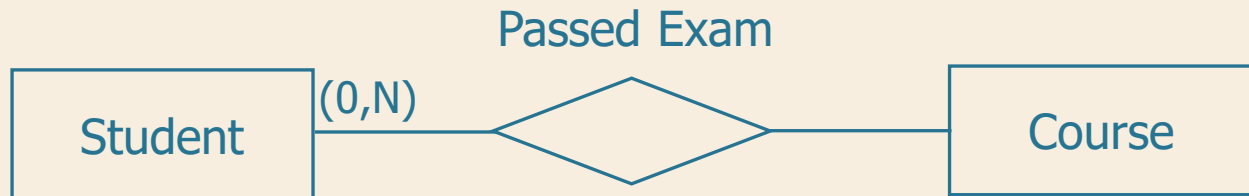
Cardinality of binary relationships

➤ N to N correspondence



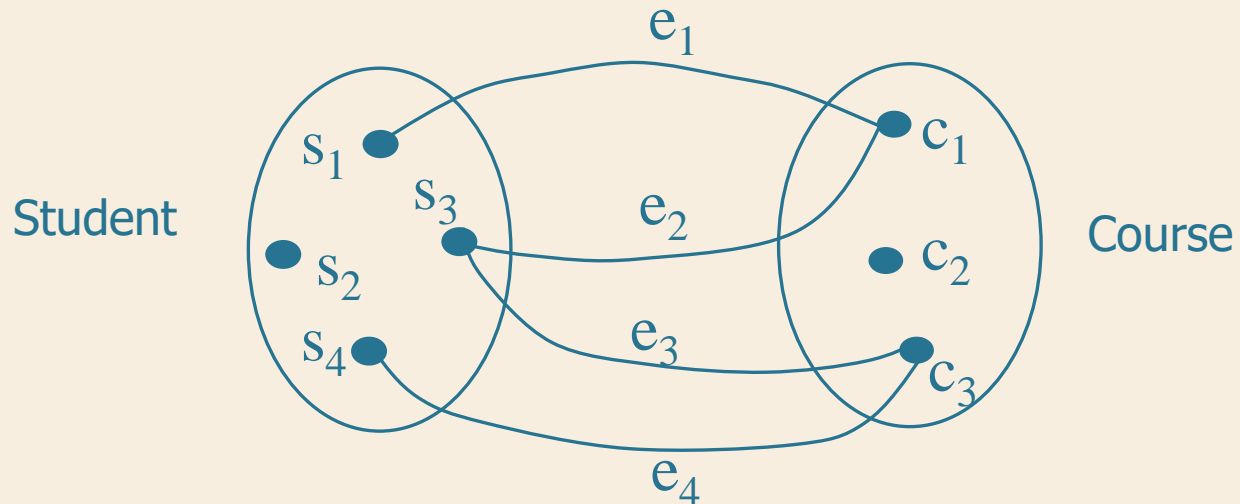
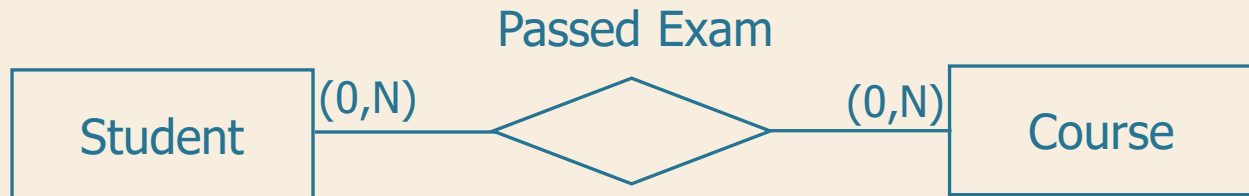
Cardinality of binary relationships

➤ N to N correspondence

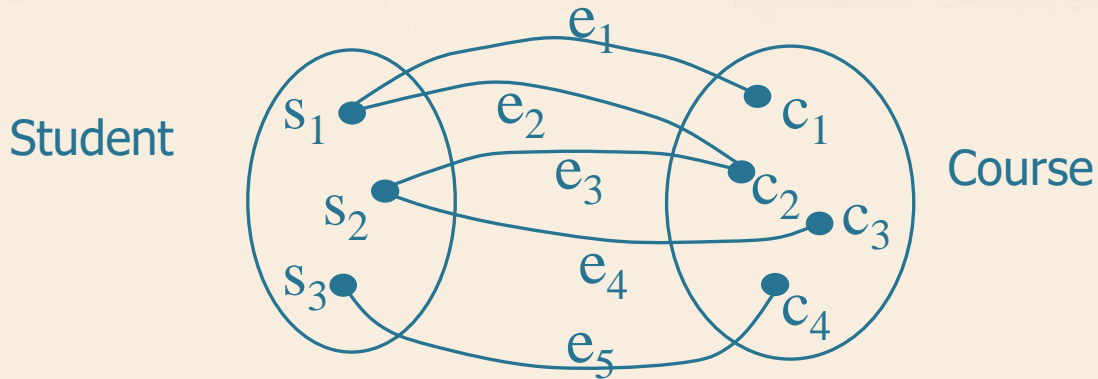


Cardinality of binary relationships

➤ N to N correspondence



Limitations of a binary relationship



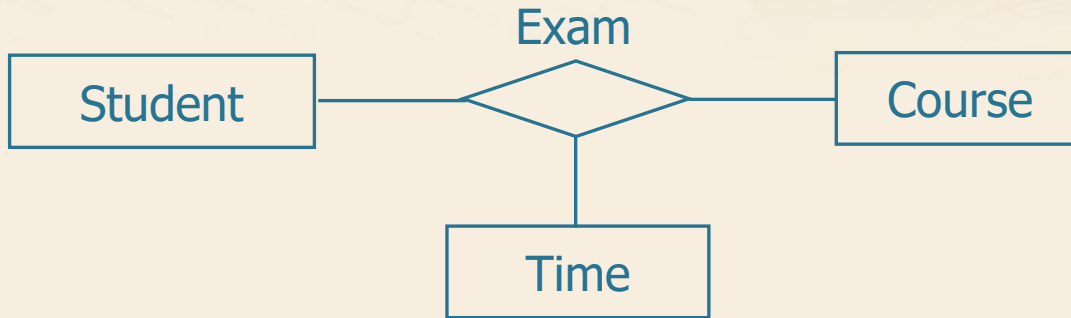
- It is not possible that a student takes the same exam more than once

Ternary relationship

- A student can take the same exam more than once at different times
- Example of an exam instance

s_1	c_1	t_1
s_1	c_1	t_2
...		

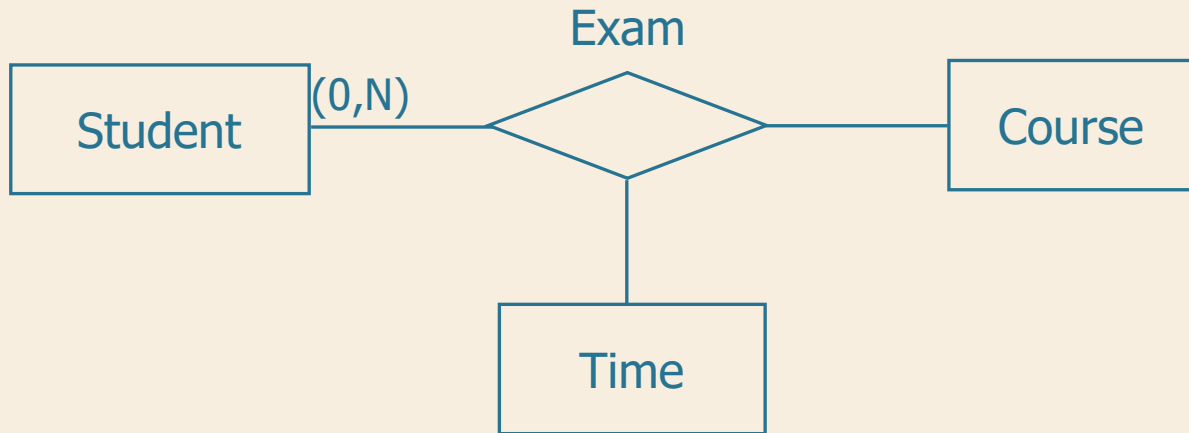
Ternary relationship



- A student can take the same exam more than once at different times
- Example of an exam instance

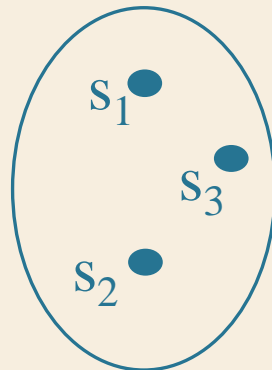
s_1	c_1	t_1
s_1	c_1	t_2
...		

Cardinality of ternary relationships

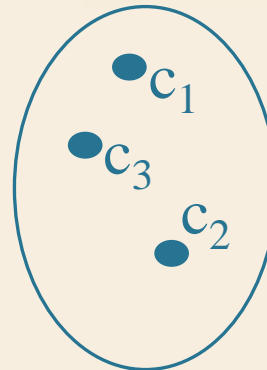


Occurrences of a ternary relationship

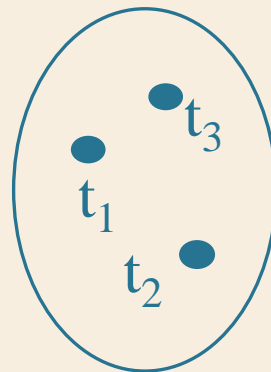
Student



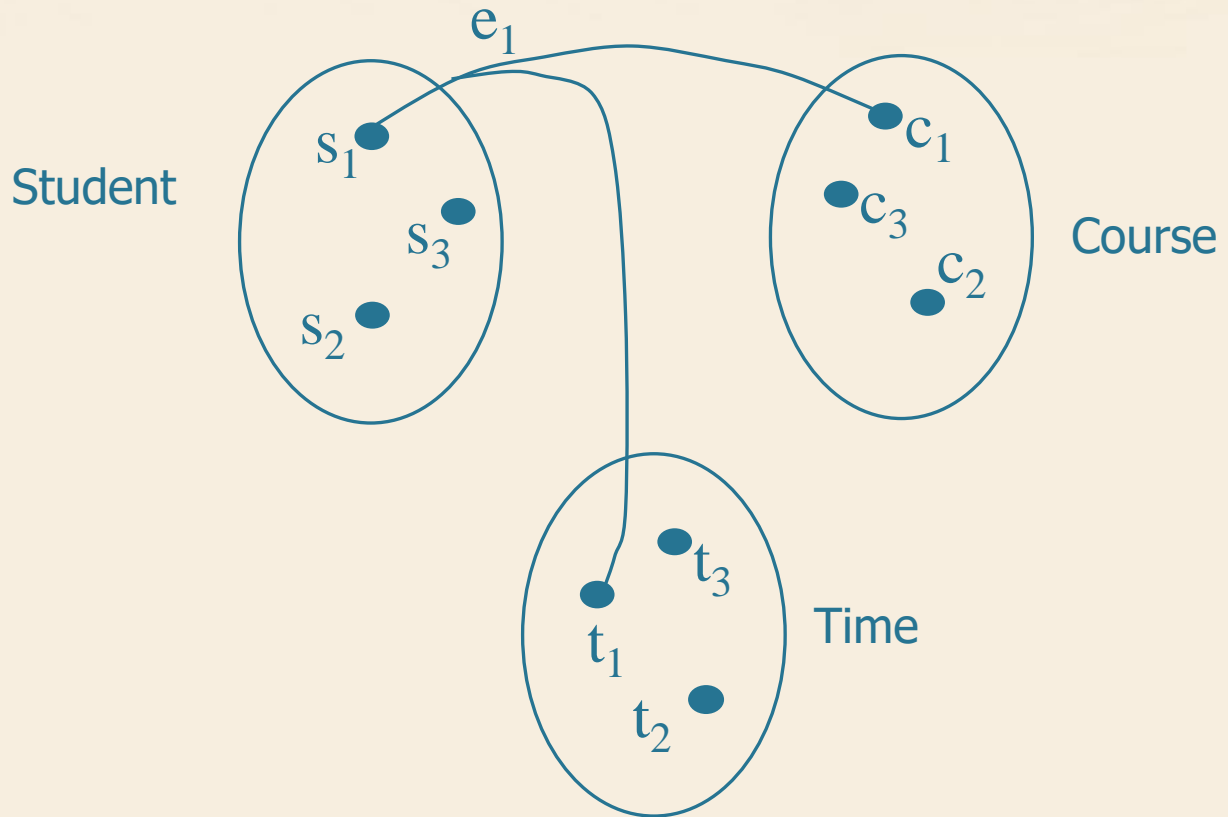
Course



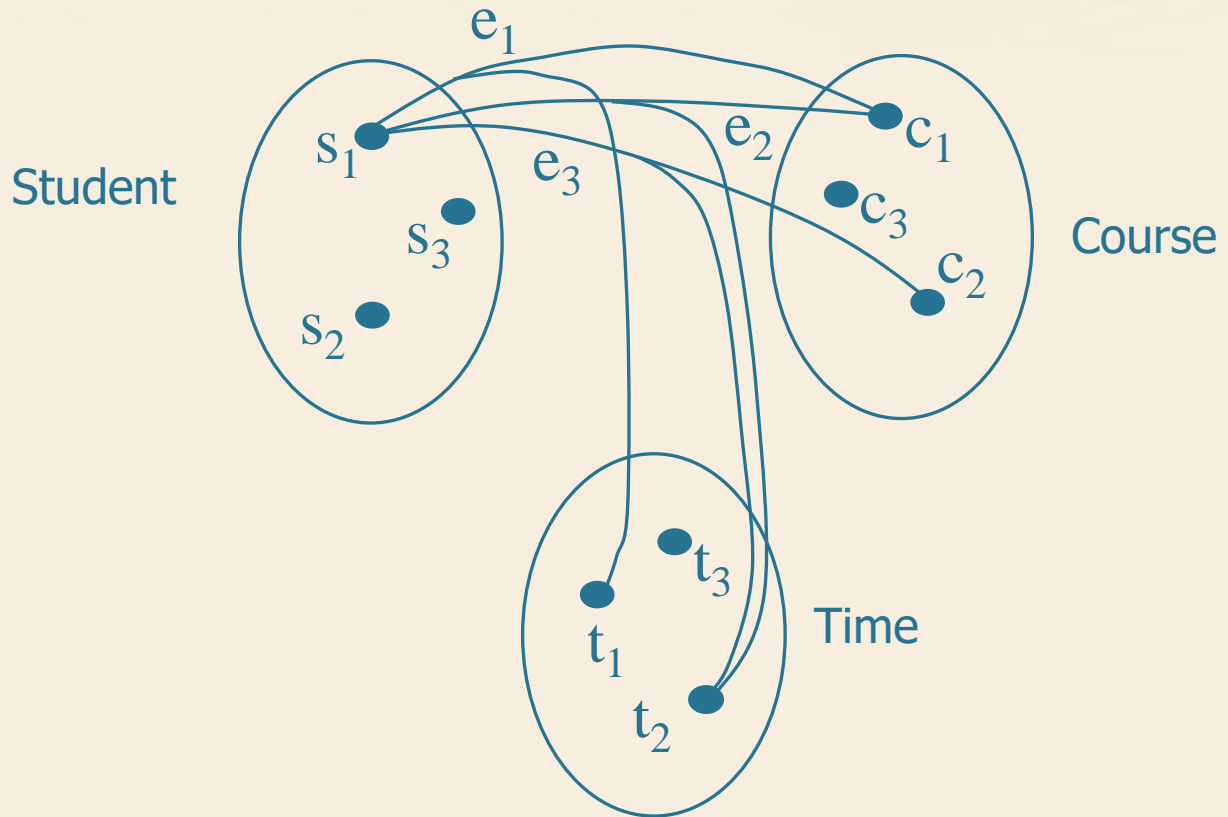
Time



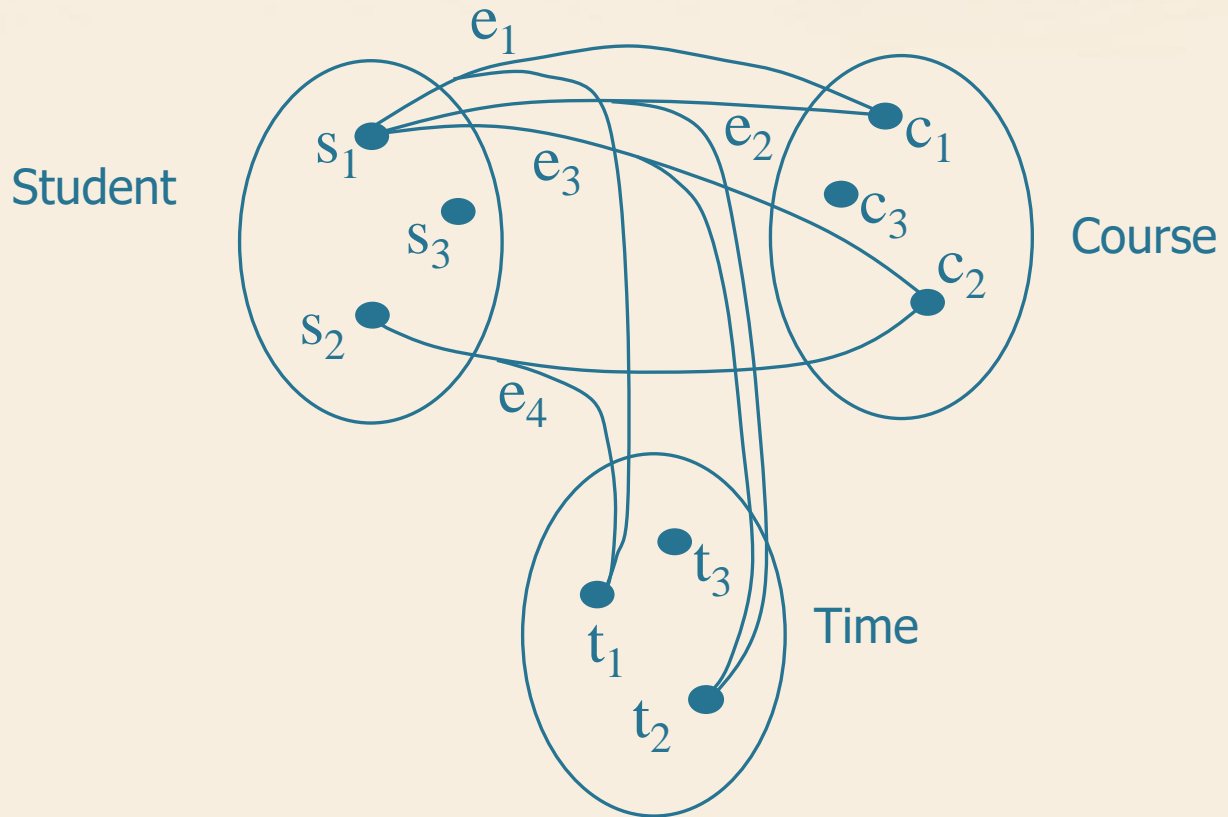
Occurrences of a ternary relationship



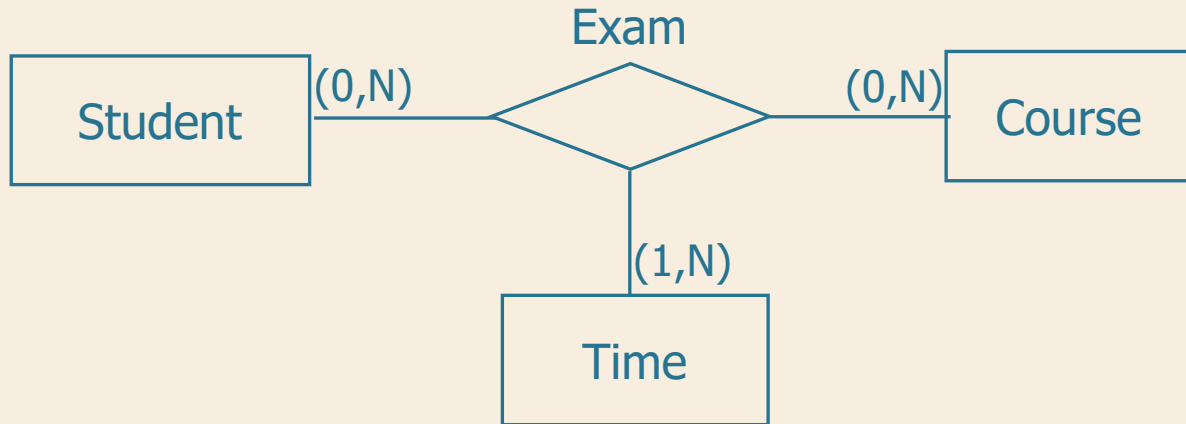
Occurrences of a ternary relationship



Occurrences of a ternary relationship



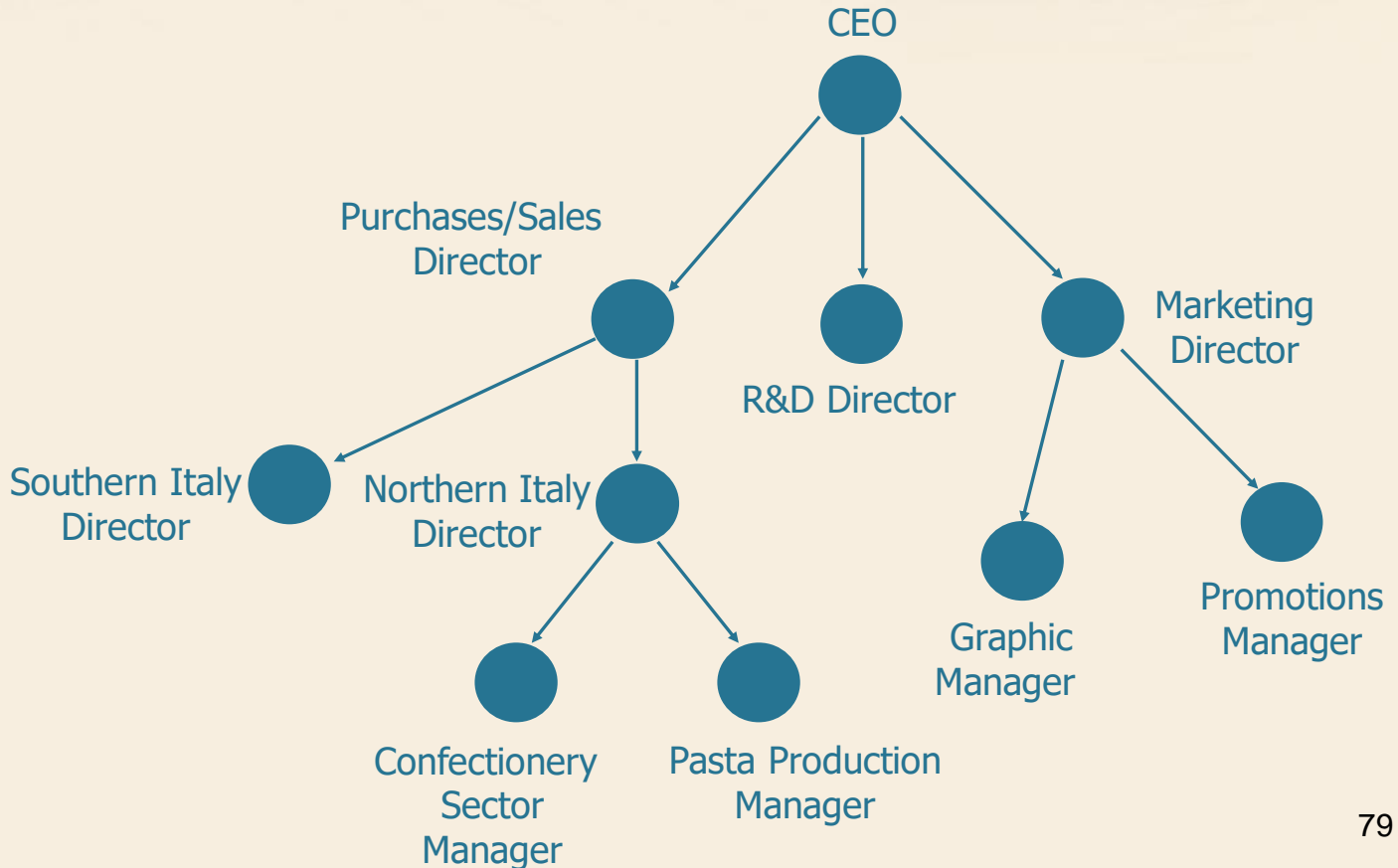
Cardinality of ternary relationships



Observations

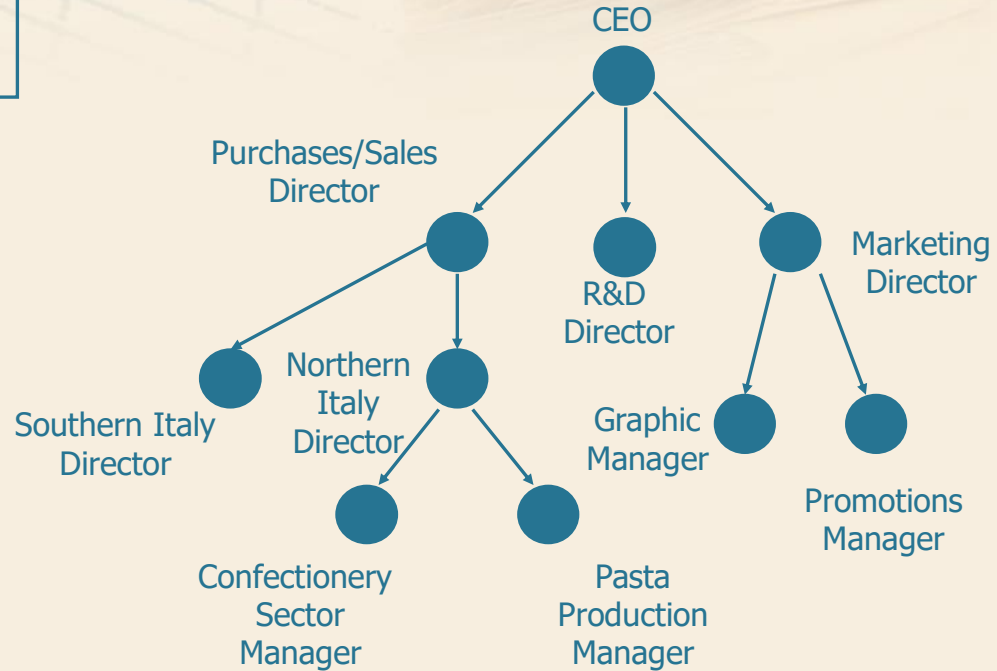
- Minimum cardinalities are rarely 1 for all entities involved in a relationship
- The maximum cardinalities of an n-ary relationship are (practically) always N
 - if the participation of an entity E has a maximum cardinality of 1, it is possible to eliminate the n-ary relationship and link entity E with the others through binary relationships

Recursive relationship

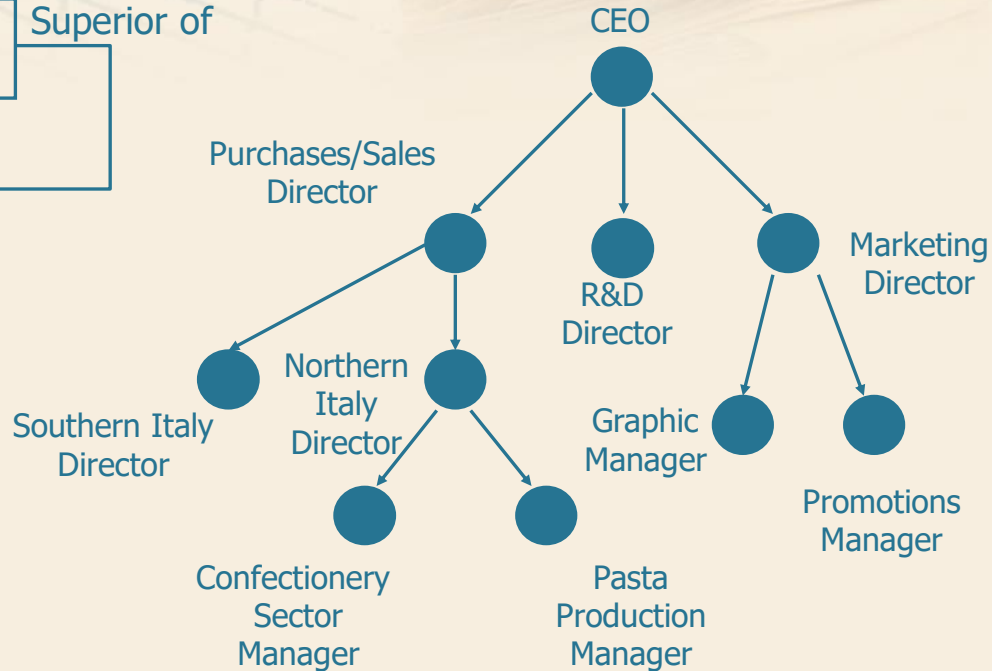
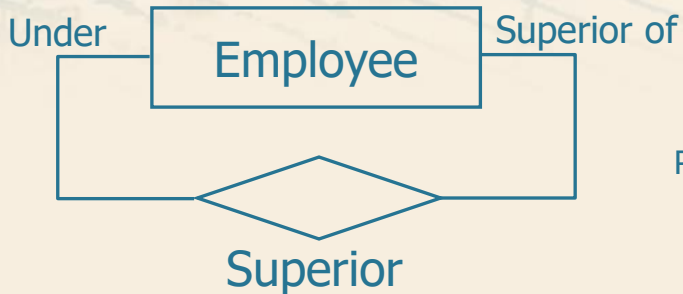


Recursive relationship

Employee

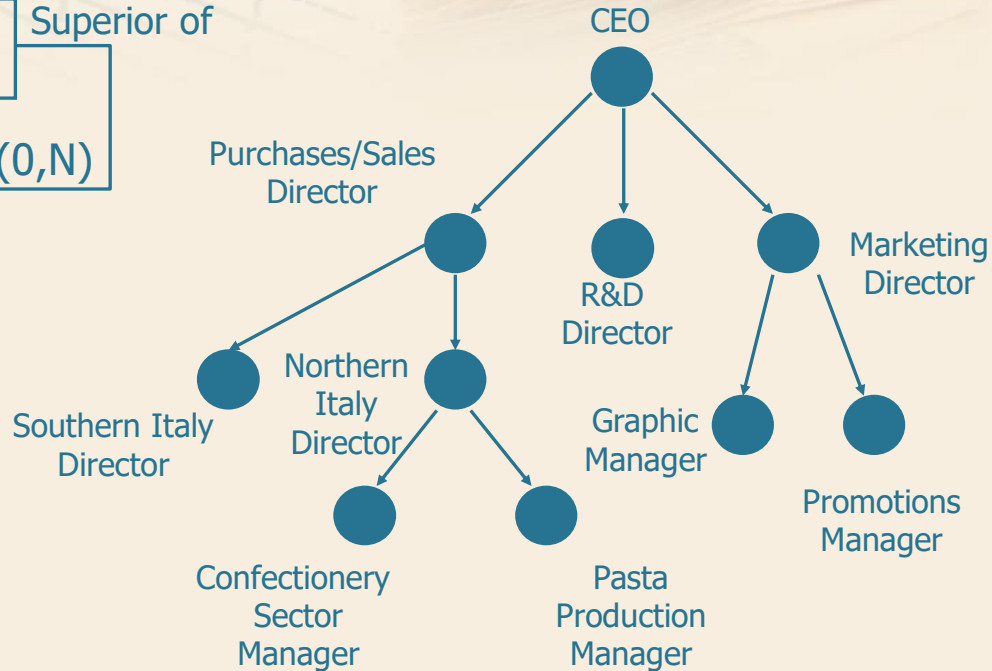
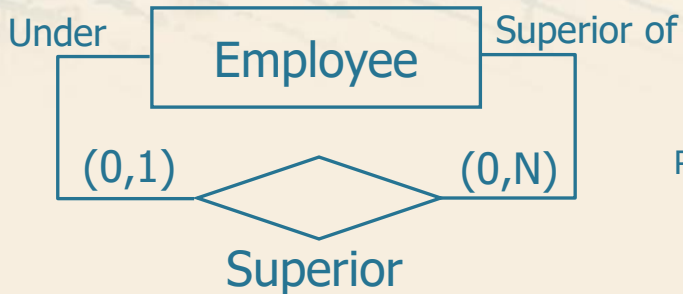


Recursive relationship



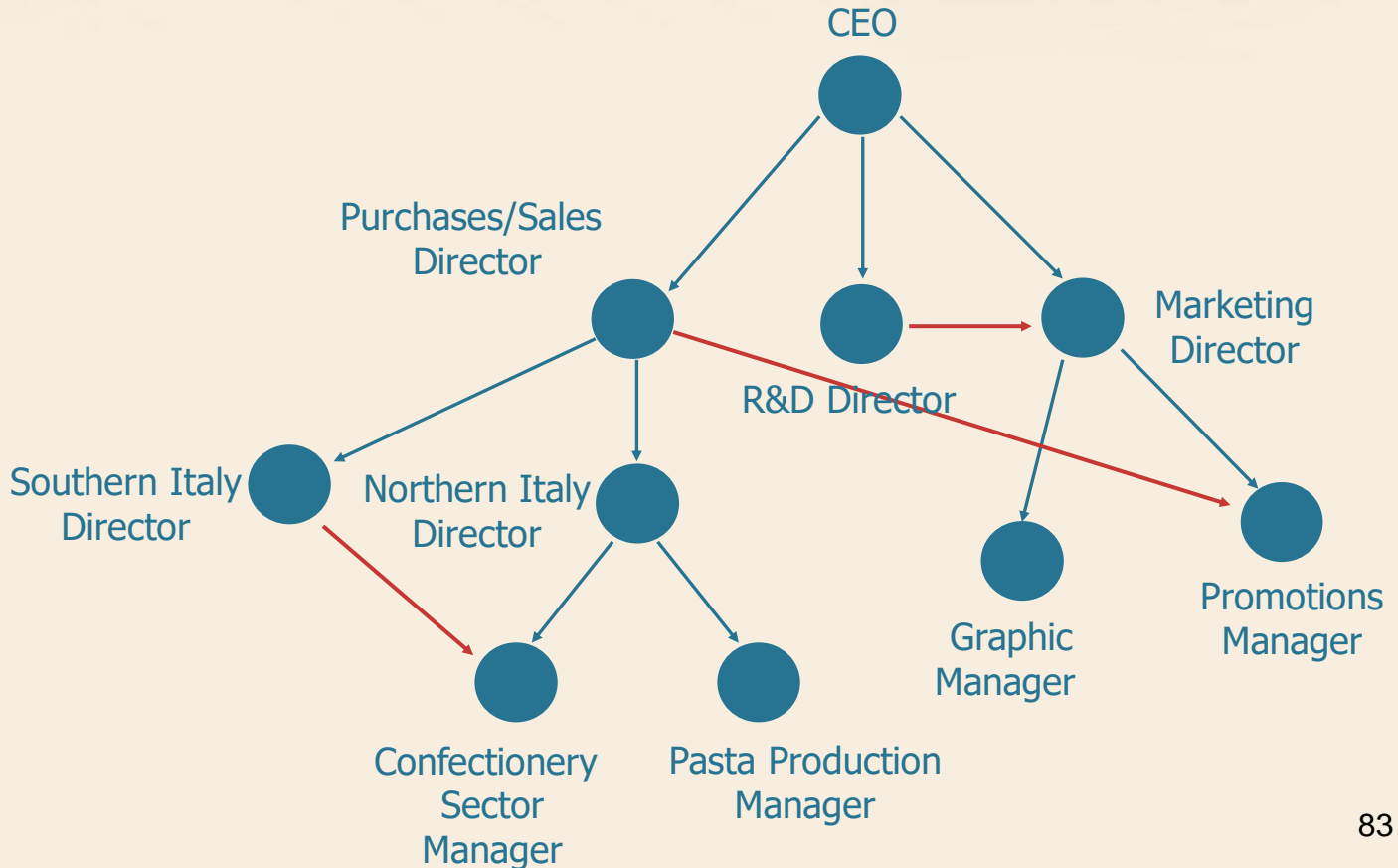
- Relationship between an entity and itself
- If the relationship is not symmetrical, the two roles of the entity must be defined

Recursive relationship



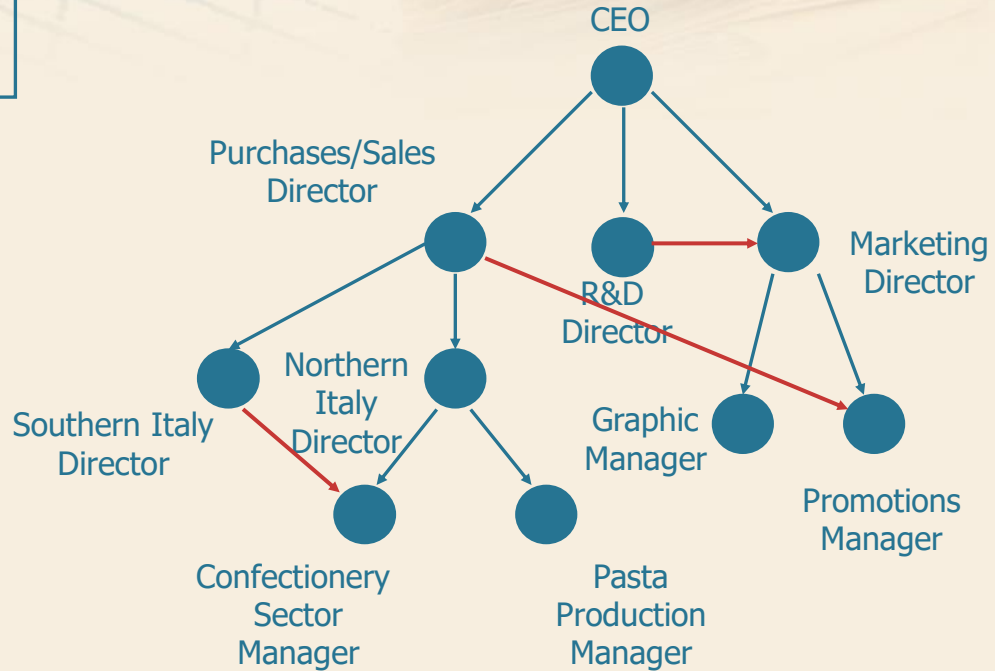
- Relationship between an entity and itself
- If the relationship is not symmetrical, the two roles of the entity must be defined

Recursive relationship

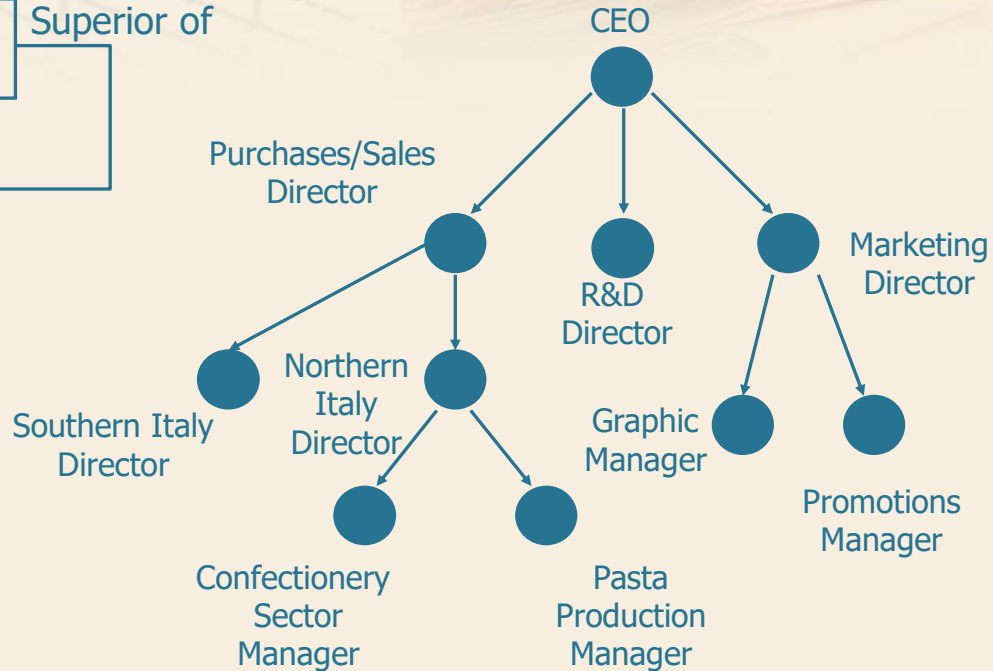
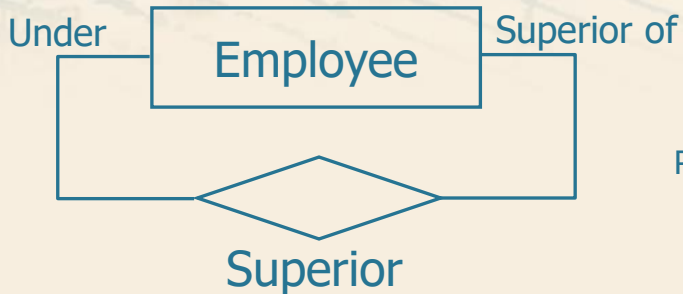


Recursive relationship

Employee

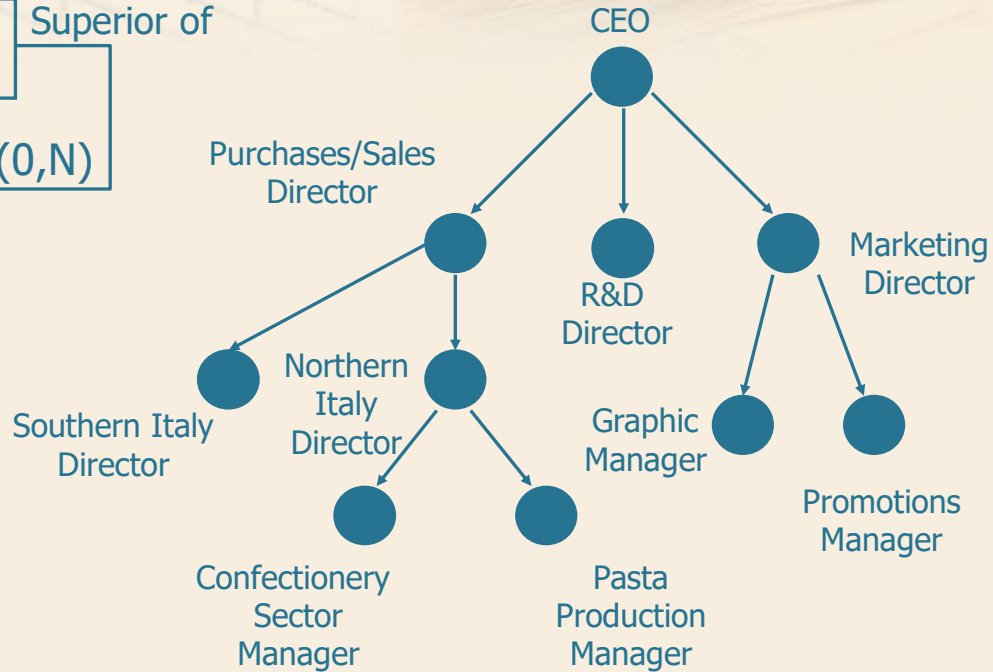
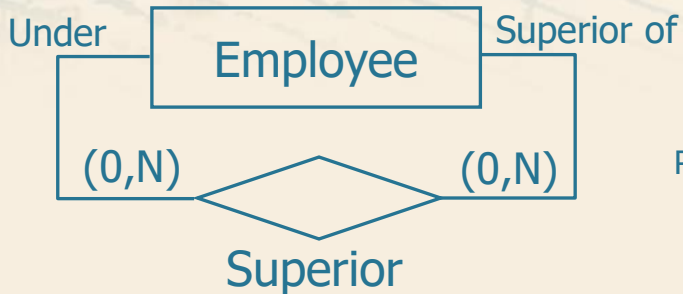


Recursive relationship



➤ An employee might have several superiors

Recursive relationship



➤ An employee might have several superiors