Introduction to databases

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

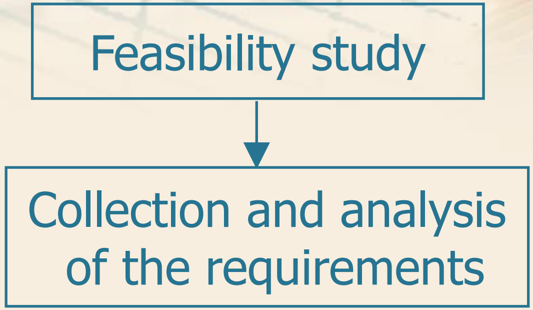
Life cycle of an information system

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

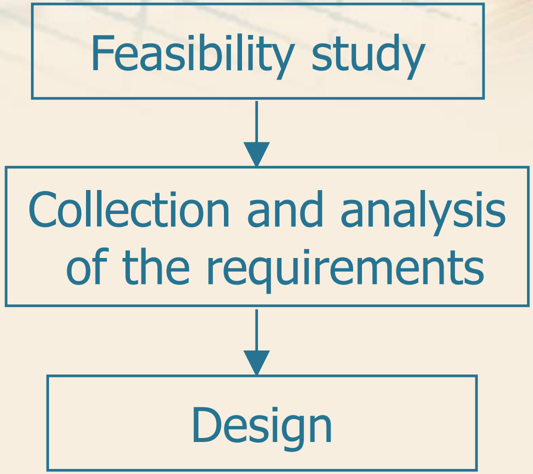
Life cycle of an information system



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

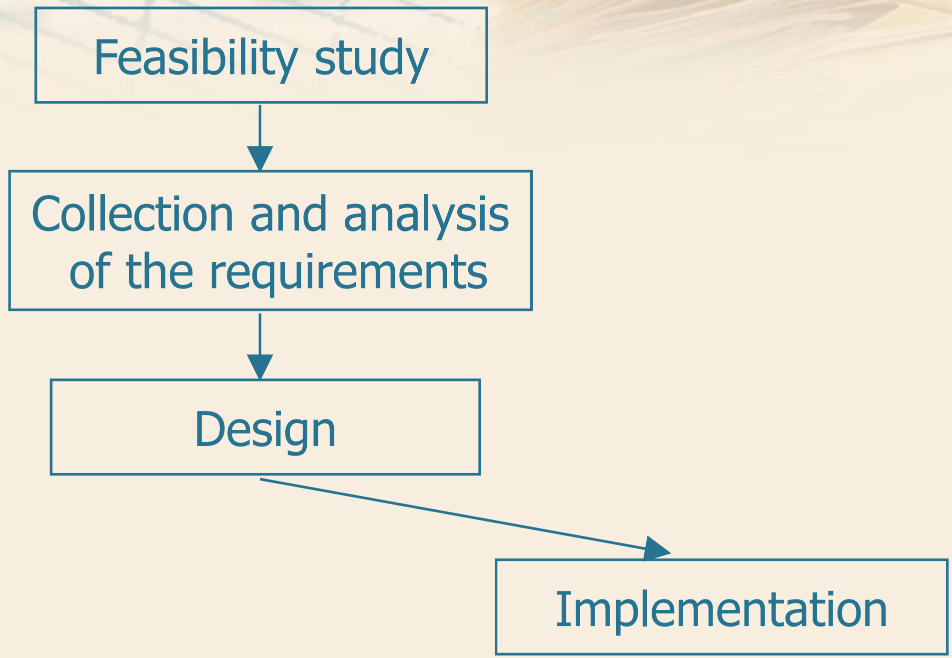
Life cycle of an information system



Life cycle of an information system

- Design
  - divided into data and application design
  - produces formal descriptions

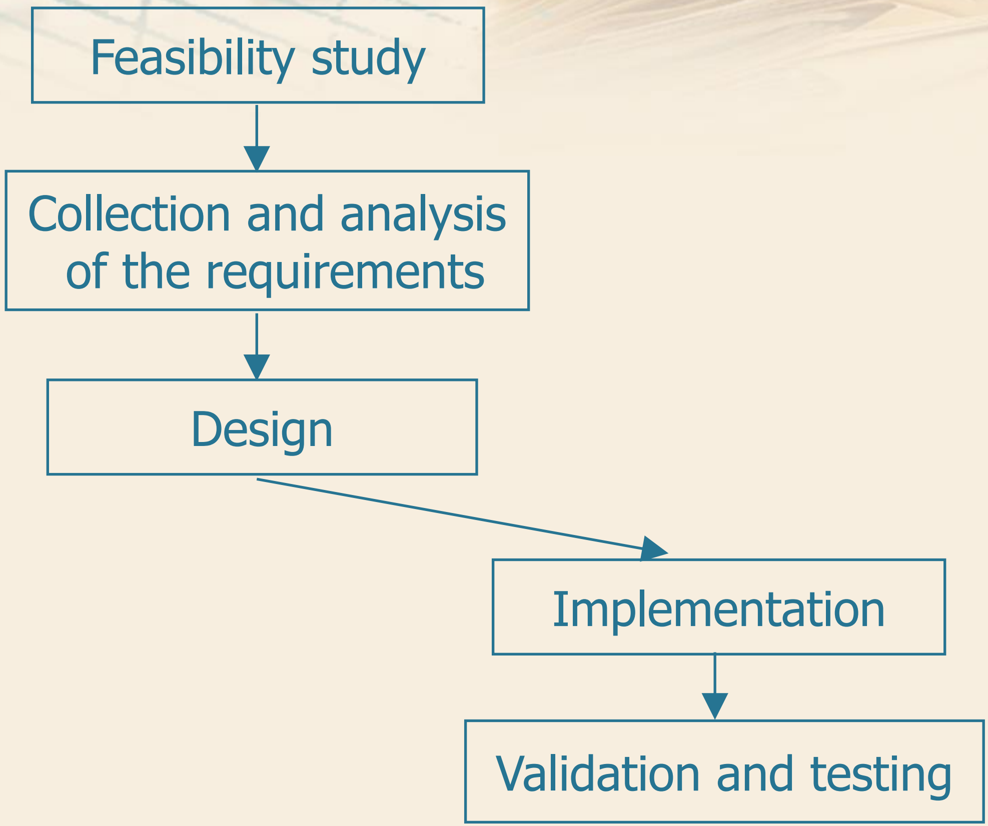
Life cycle of an information system



Life cycle of an information system

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

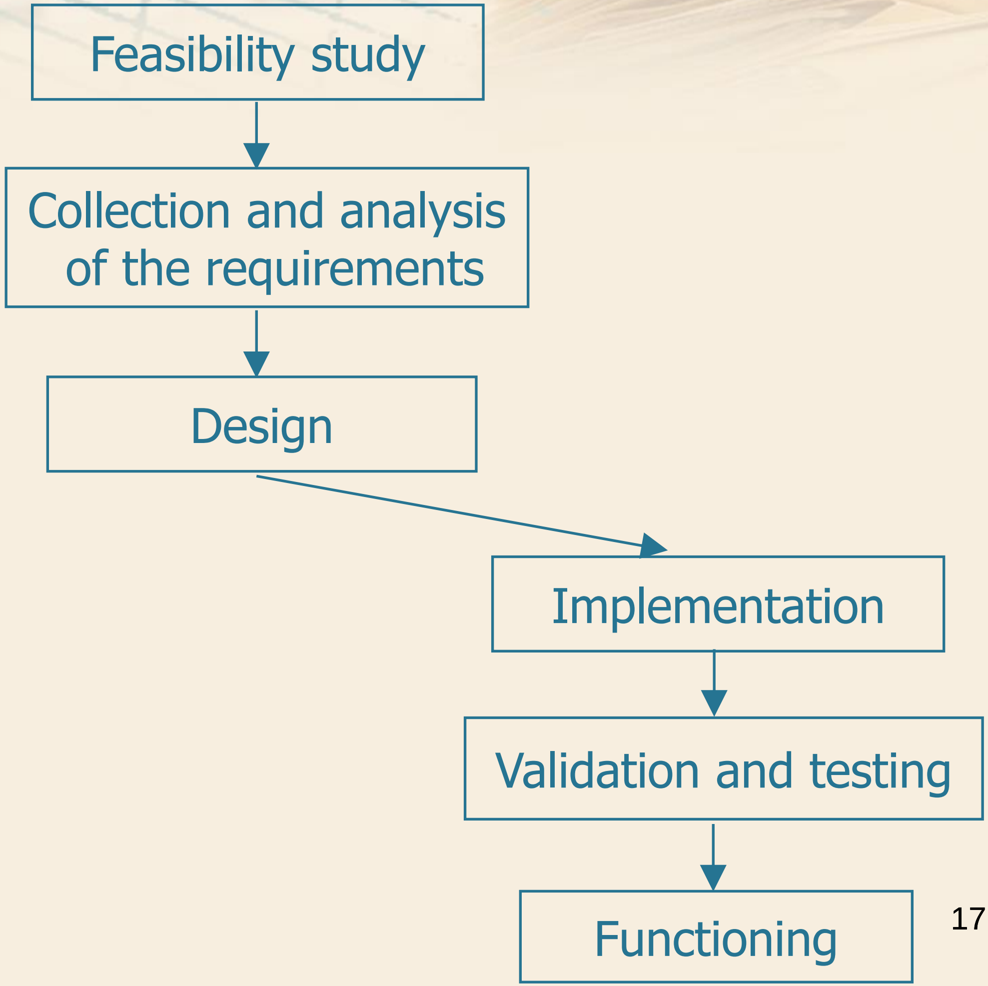
Life cycle of an information system



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

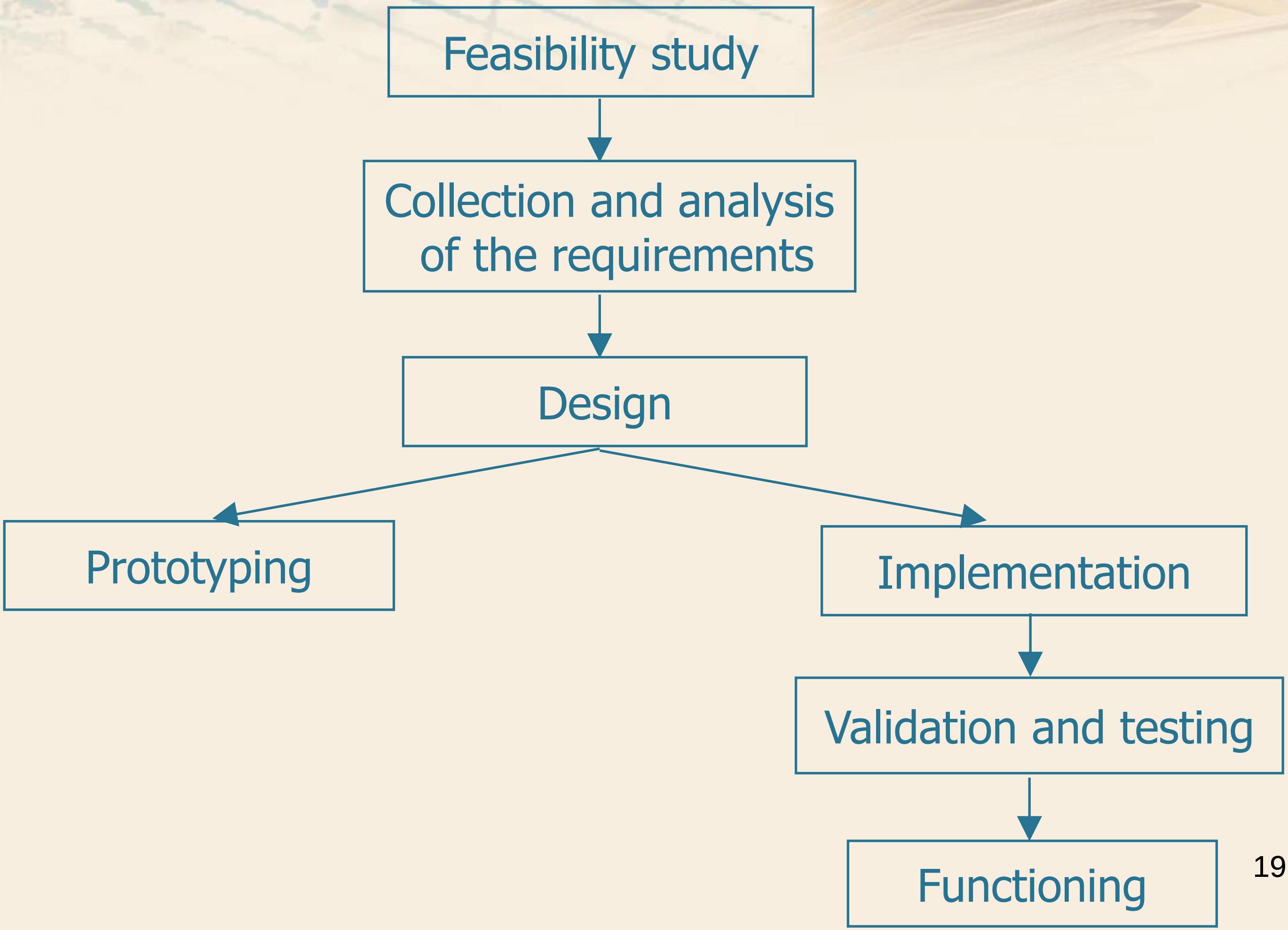
Life cycle of an information system



Life cycle of an information system

- Functioning
  - system operation
  - requires management and maintenance operations

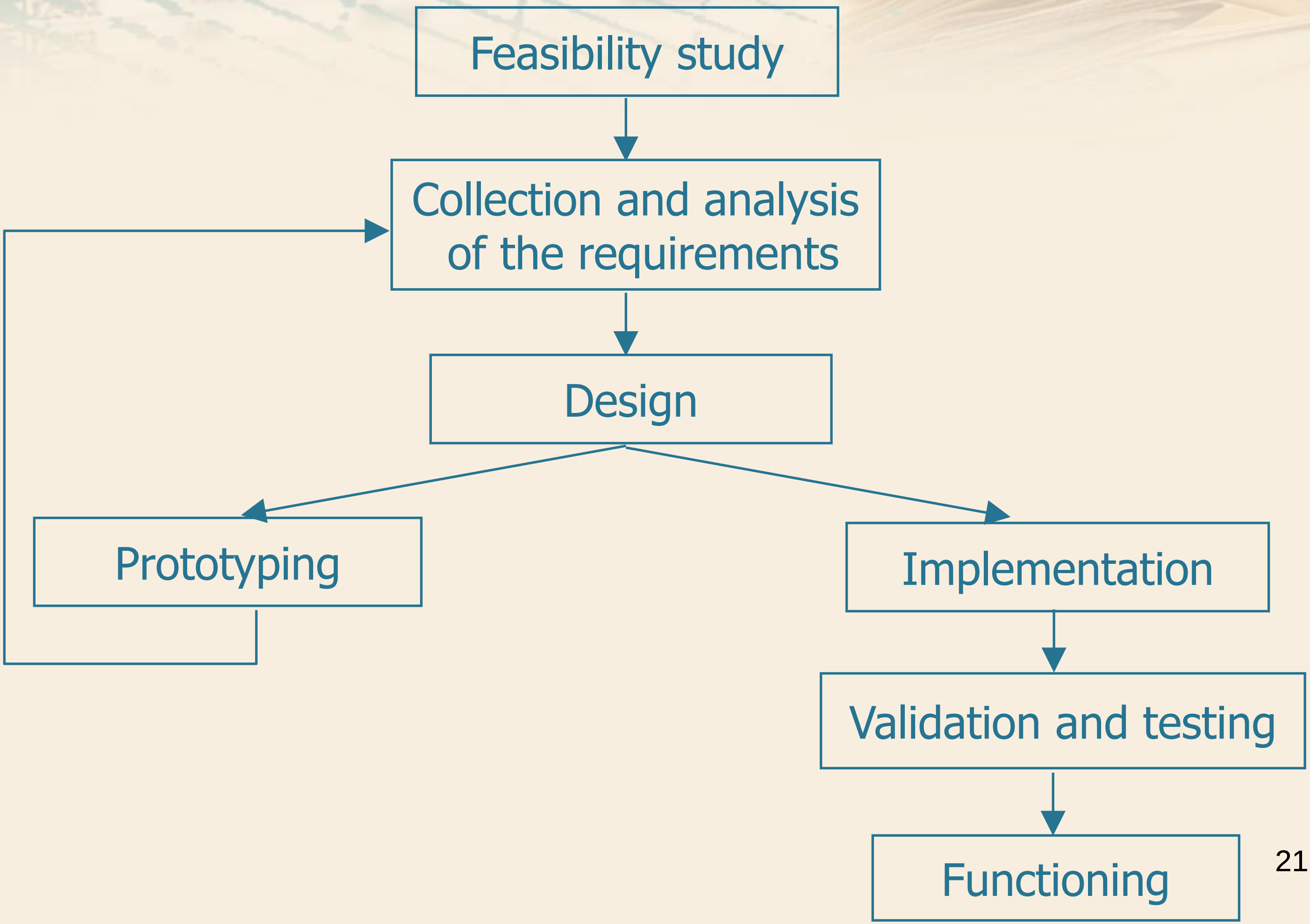
Life cycle of an information system



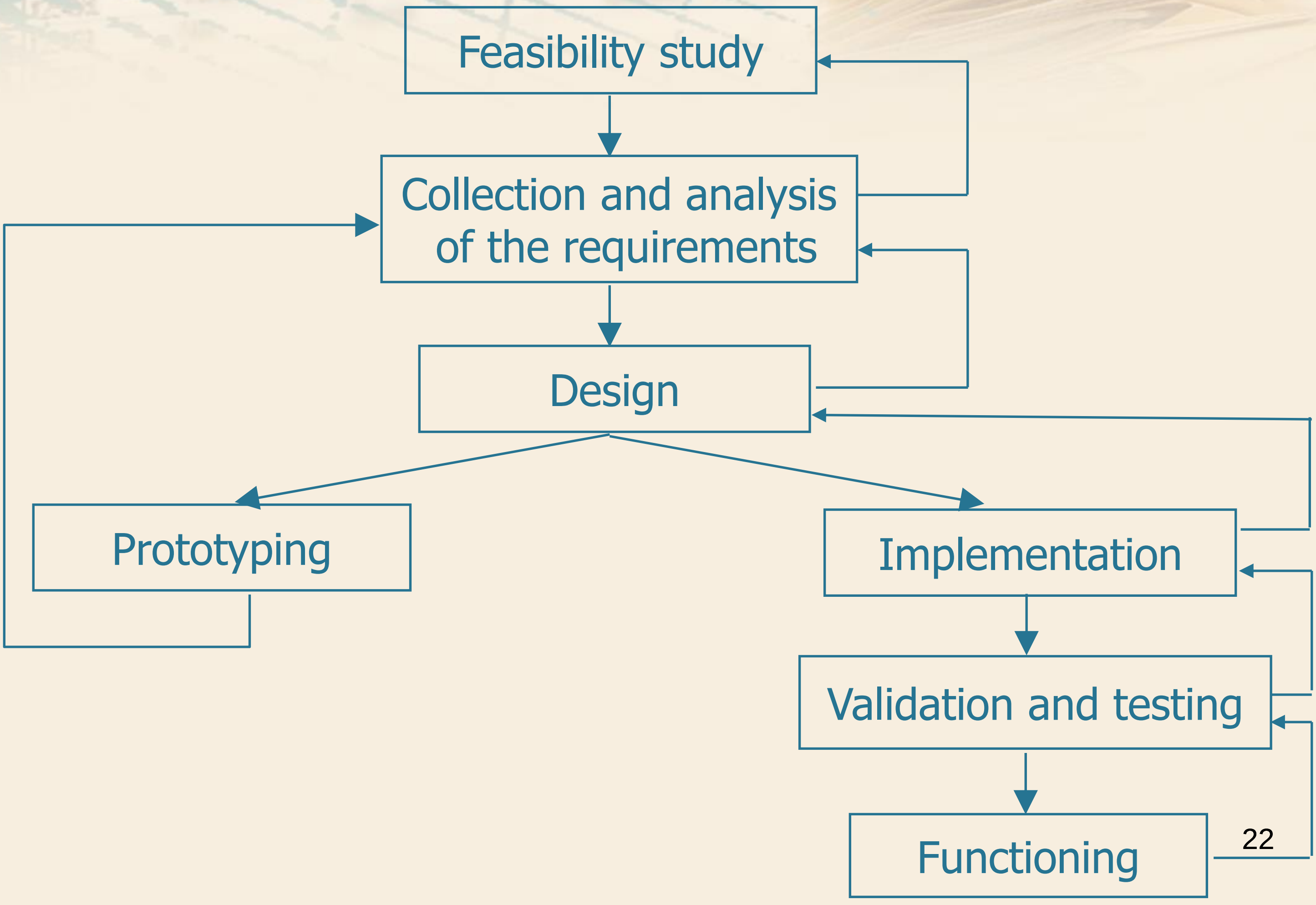
Life cycle of an information system

- 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) diet
       B) exercises 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: current weight, % of body fat

       output data: model of the body structure of a fit person

    2a. input data: fit person model

       output data: body structure model of the average athlete

Properties of the methodology

- Generality
  - same methodology regardless of the problem and the tools available
- Quality of the 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

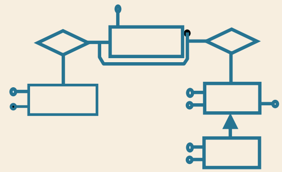
- Informal specifications of the reality of interest
  - application properties
  - application functionalities

Stages of database design

Application requirements

Conceptual design

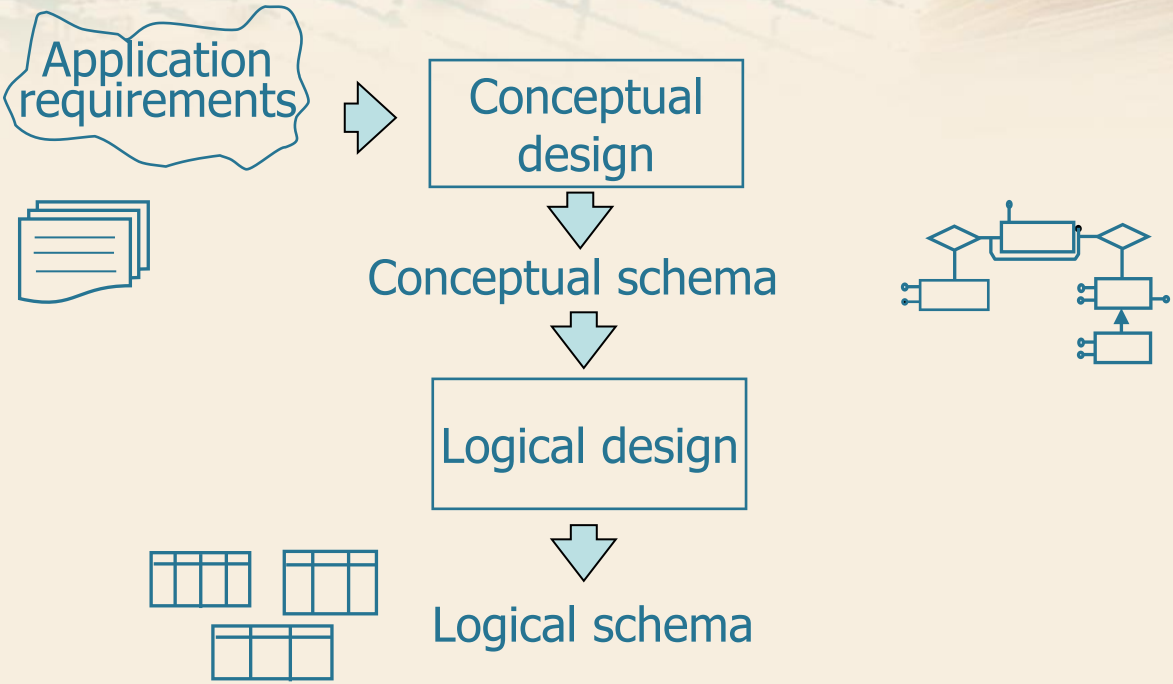
Conceptual schema



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

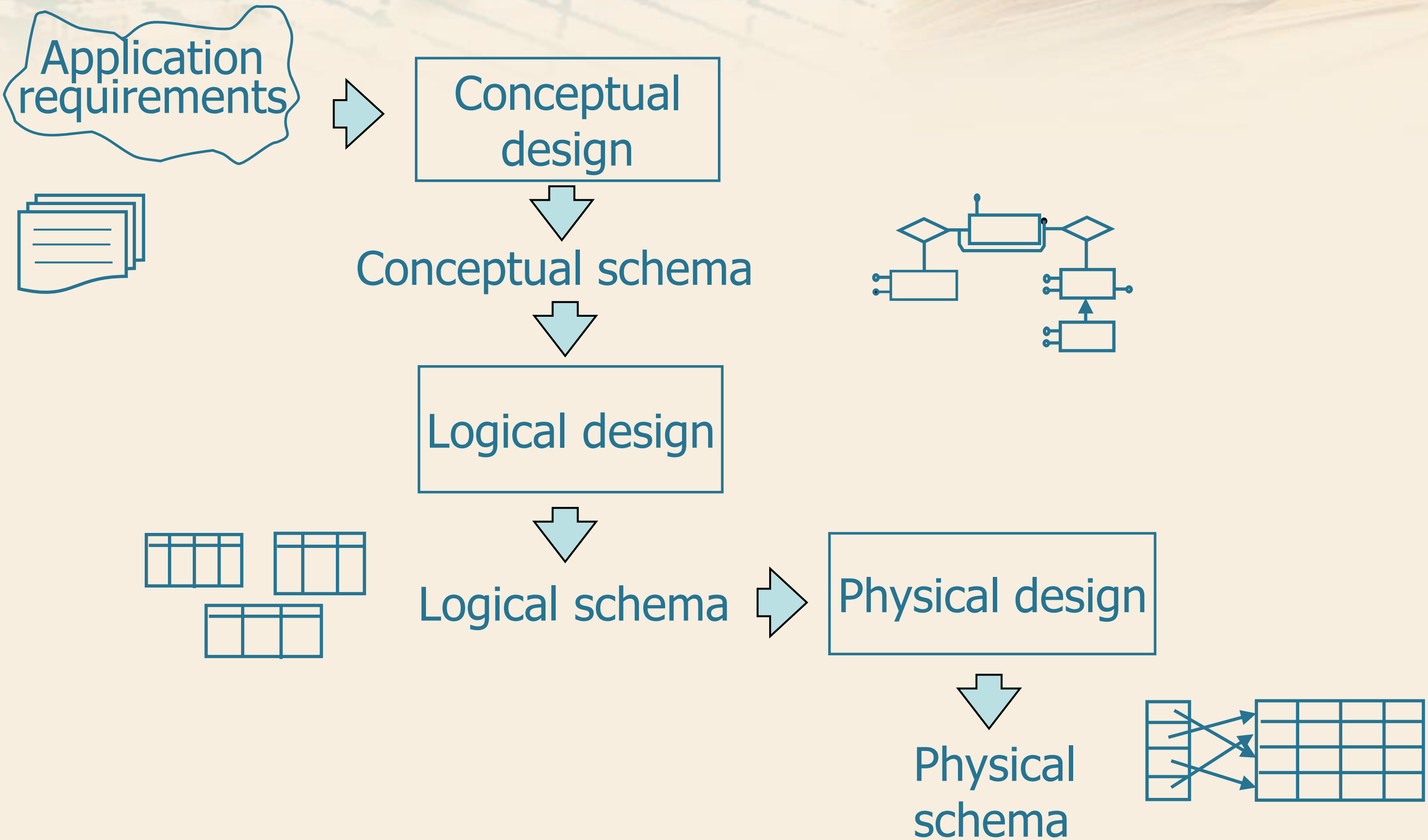
Stages of database design



Logical design

- 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)

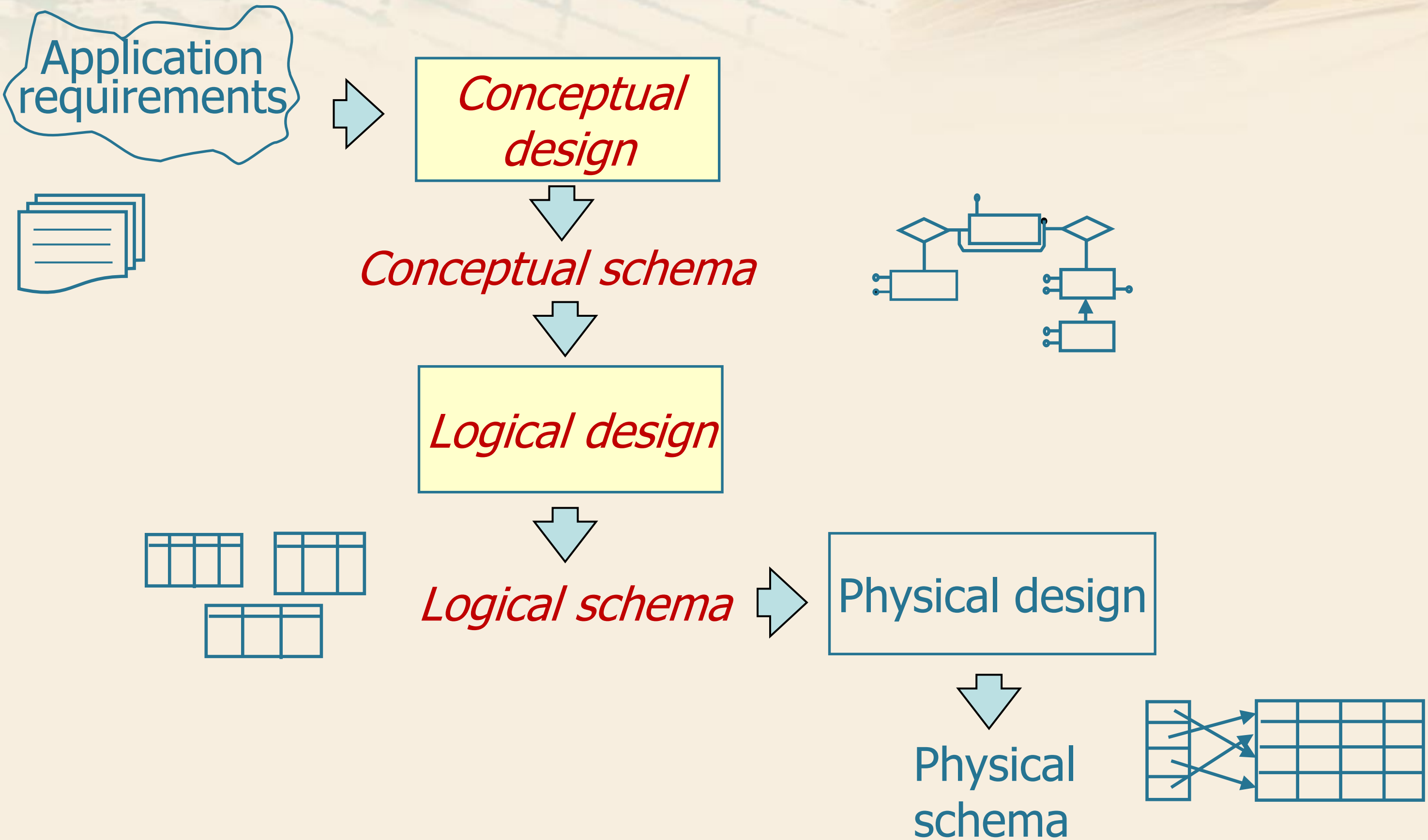
Stages of database design



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



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

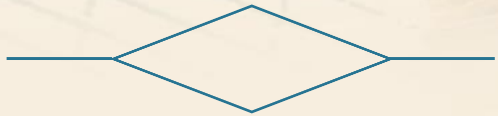
Main constructs of the E-R model

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

Entities

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

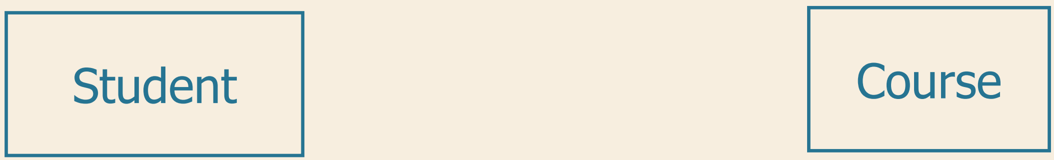
Relationship



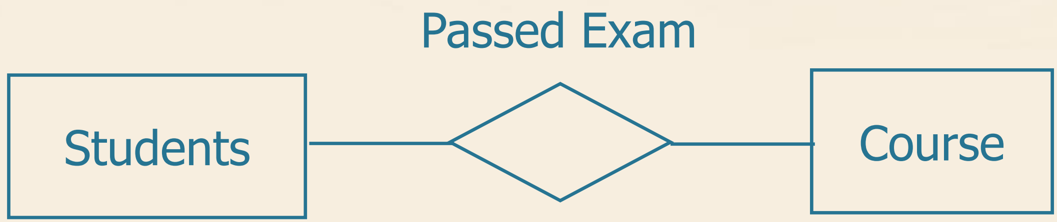
Relationship name

- 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

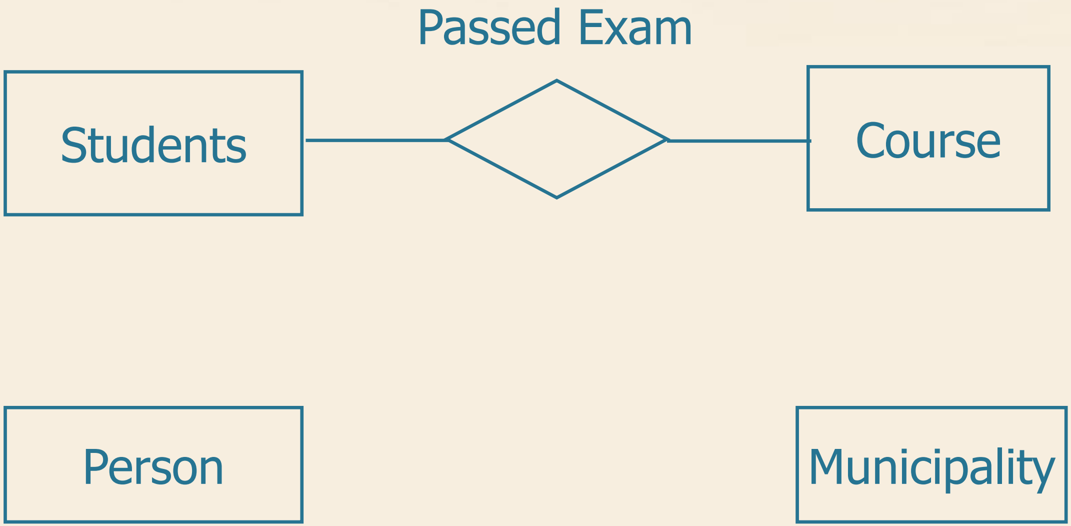
Relationships examples



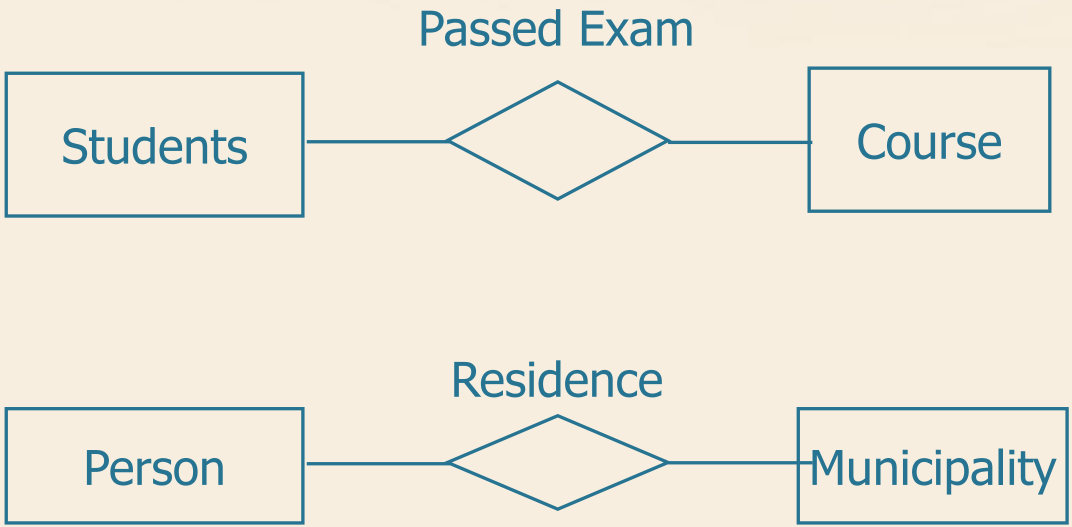
Relationships examples



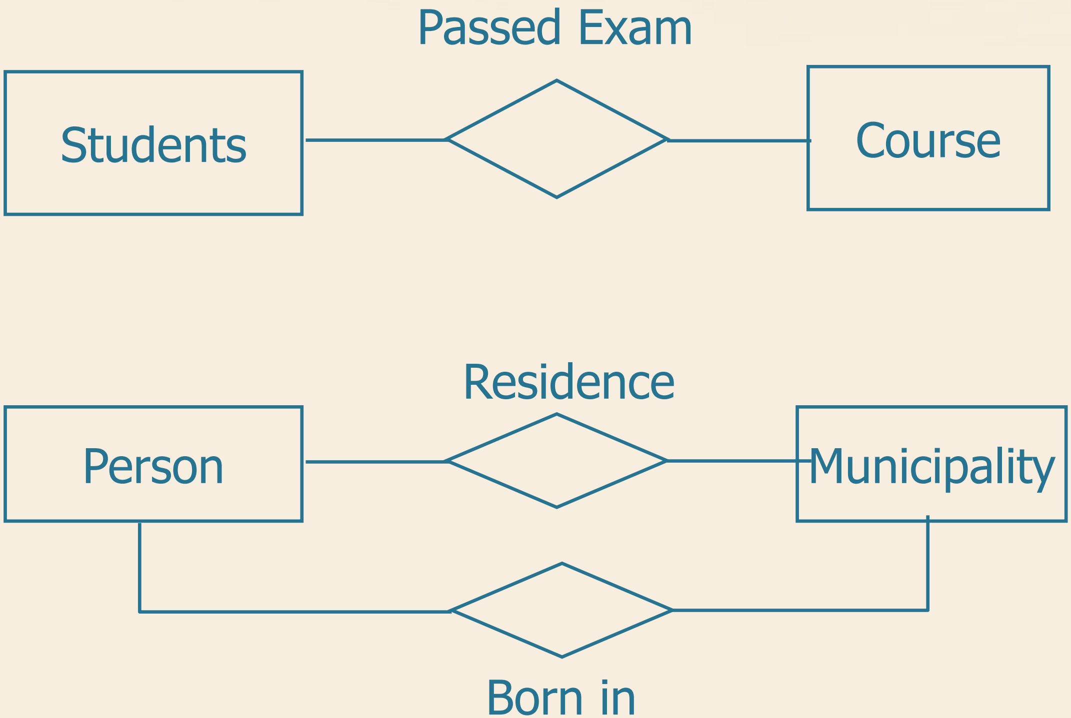
Relationships examples



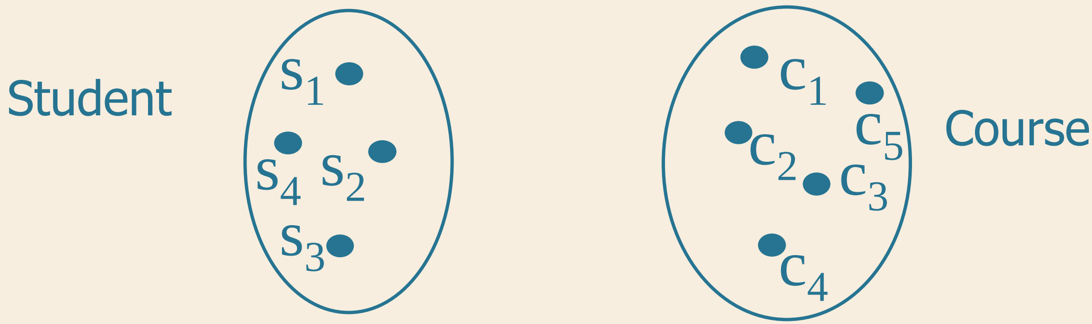
Relationships examples



Relationships examples

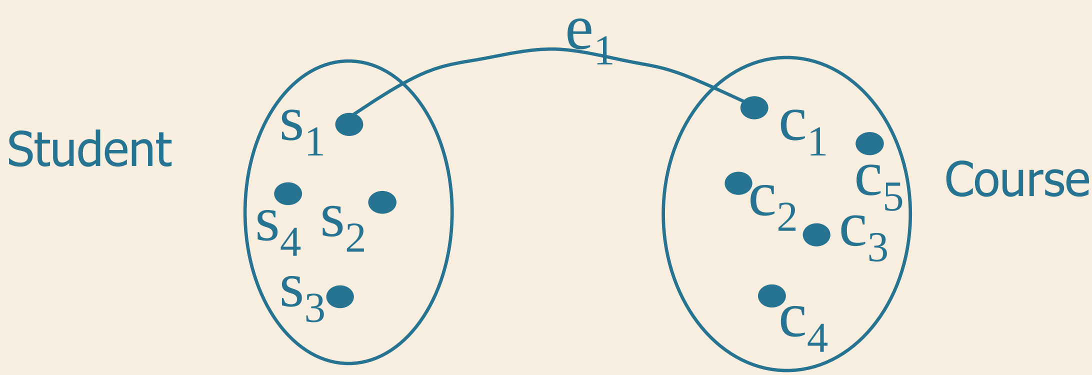


Occurrences of a relationship



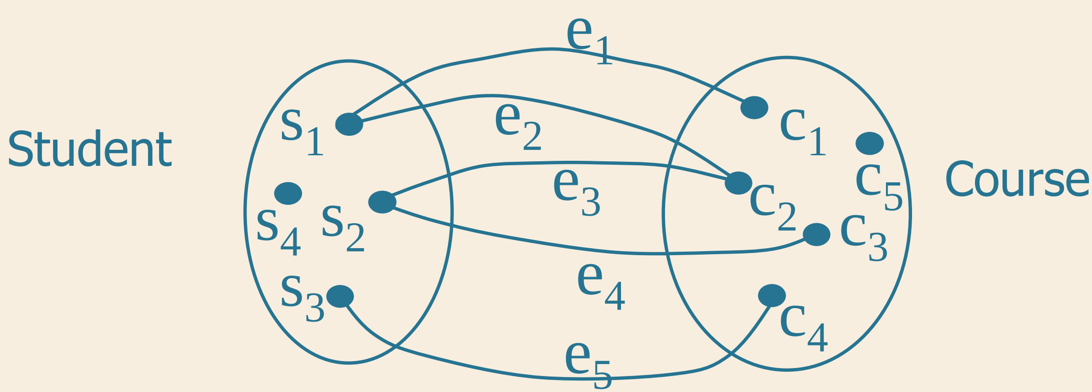
Occurrences of a relationship

- 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

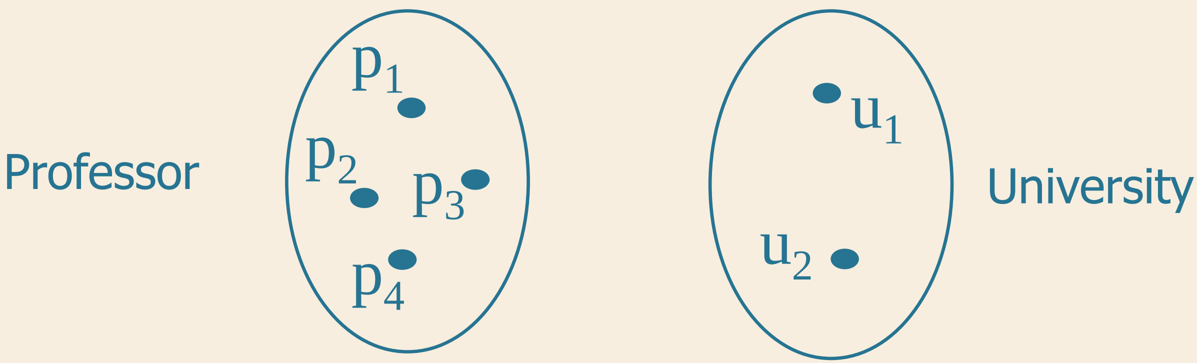


Cardinality of binary relationships

- Cardinalities are specified for each entity participating in a relationship
- They describe the maximum and minimum number of relationship occurrences to 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)

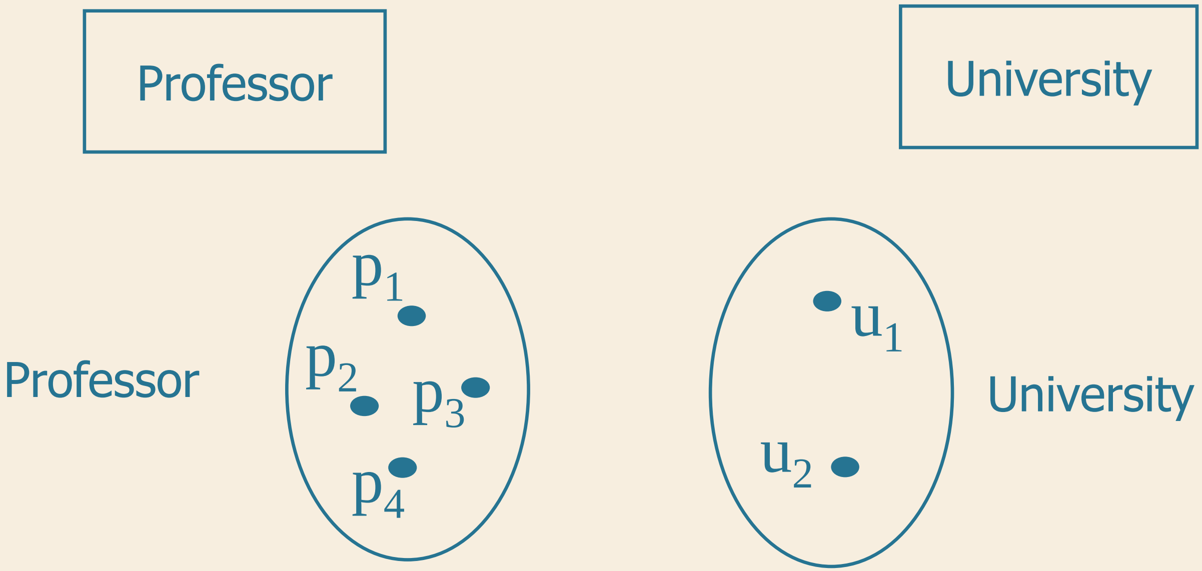
Cardinality of binary relationships

- 1 to 1 correspondence



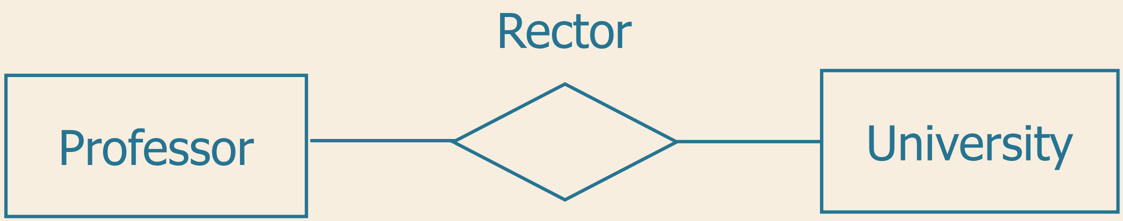
Cardinality of binary relationships

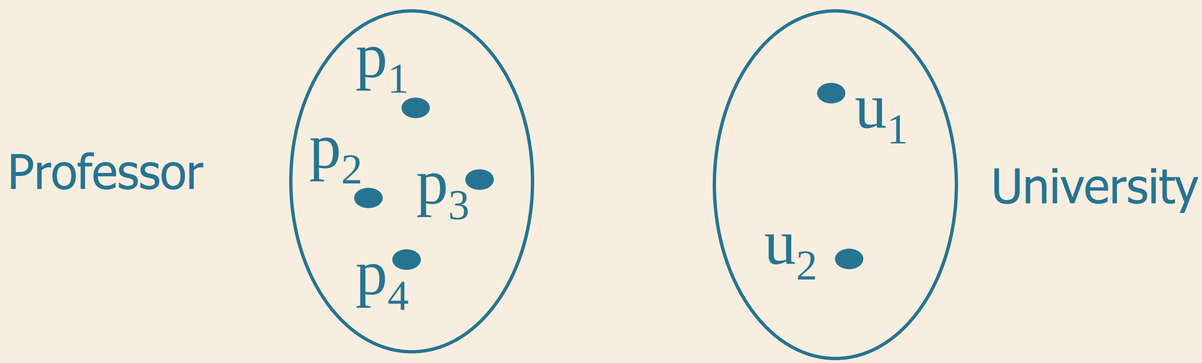
- 1 to 1 correspondence



Cardinality of binary relationships

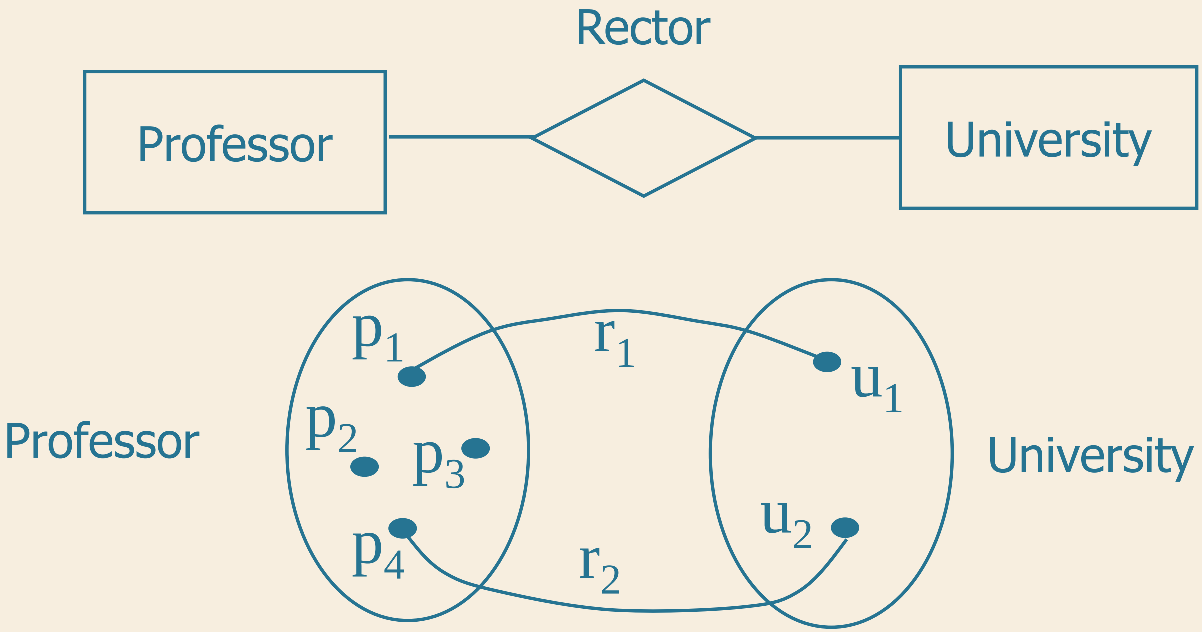
- 1 to 1 correspondence





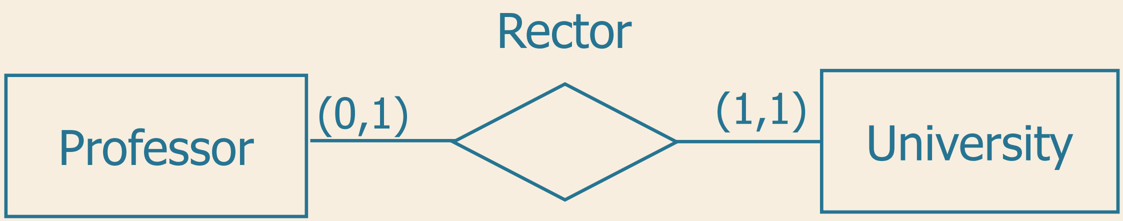
Cardinality of binary relationships

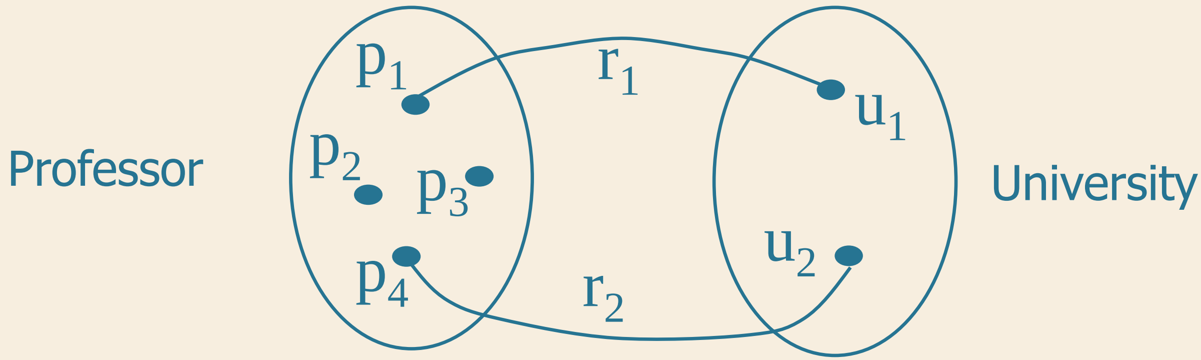
- 1 to 1 correspondence



Cardinality of binary relationships

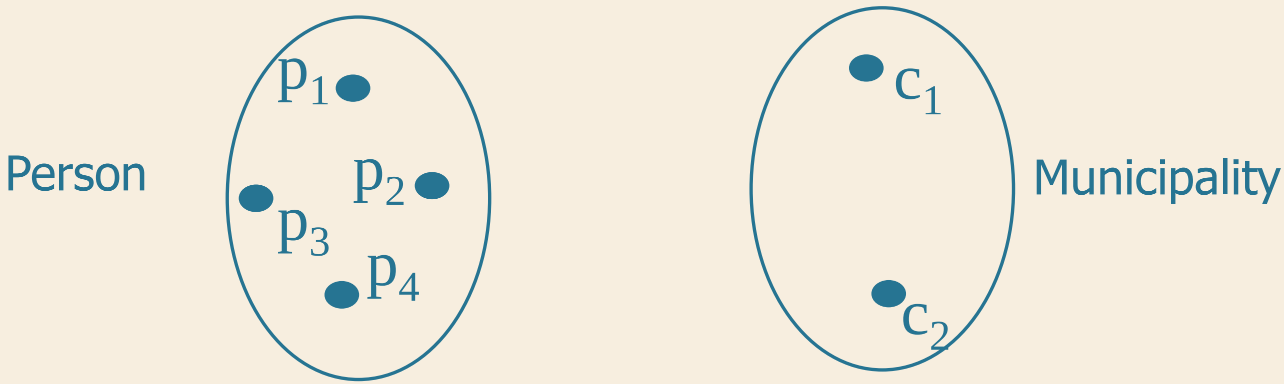
- 1 to 1 correspondence





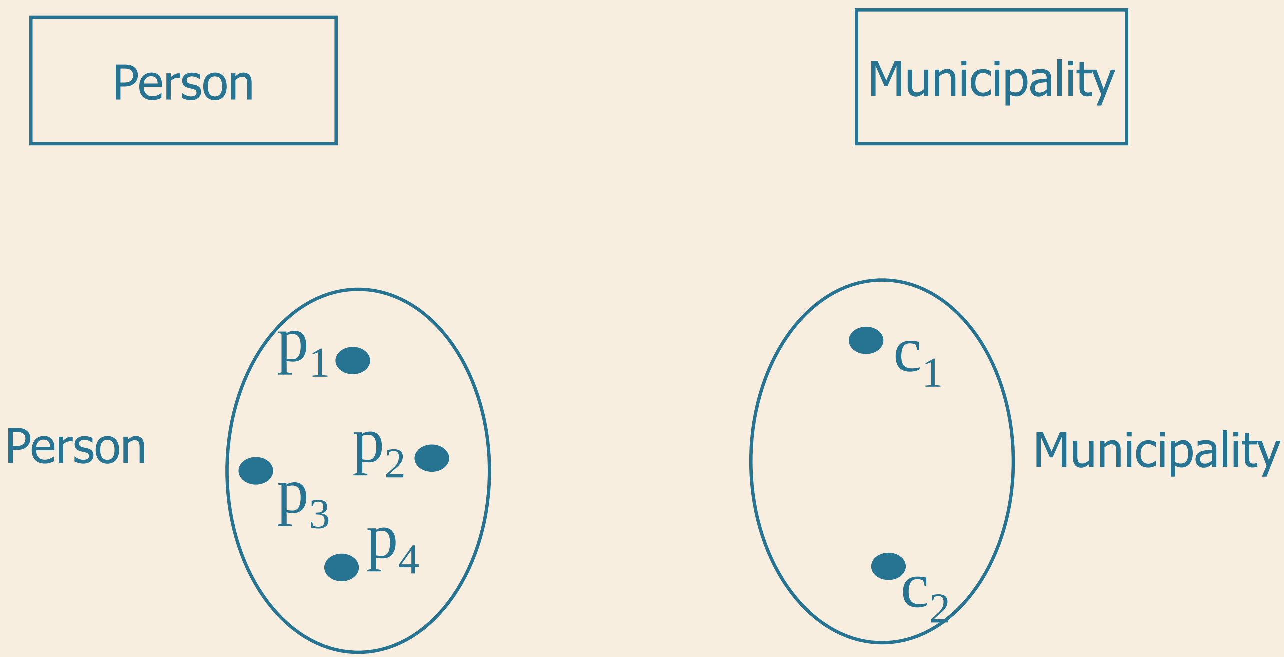
Cardinality of binary relationships

- 1 to N correspondence



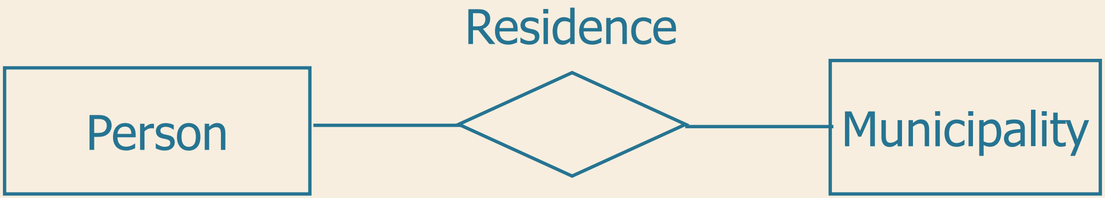
Cardinality of binary relationships

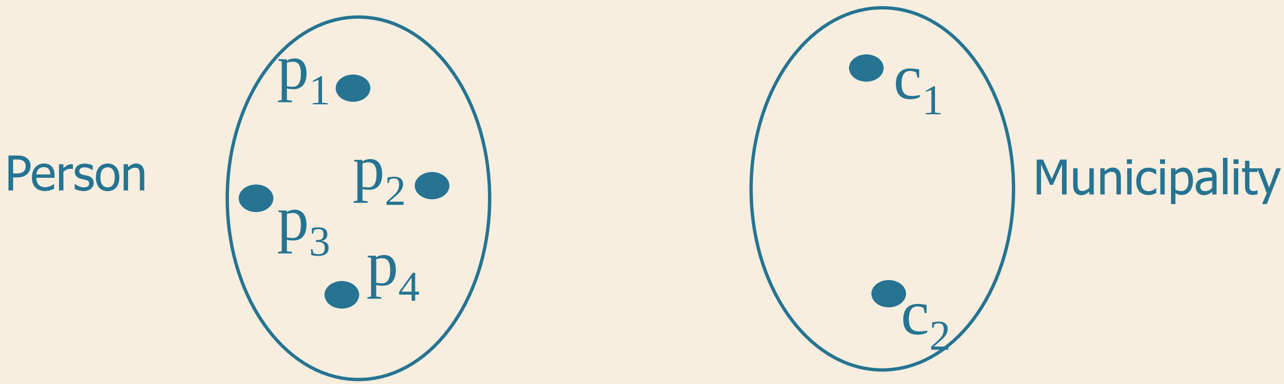
- 1 to N correspondence



Cardinality of binary relationships

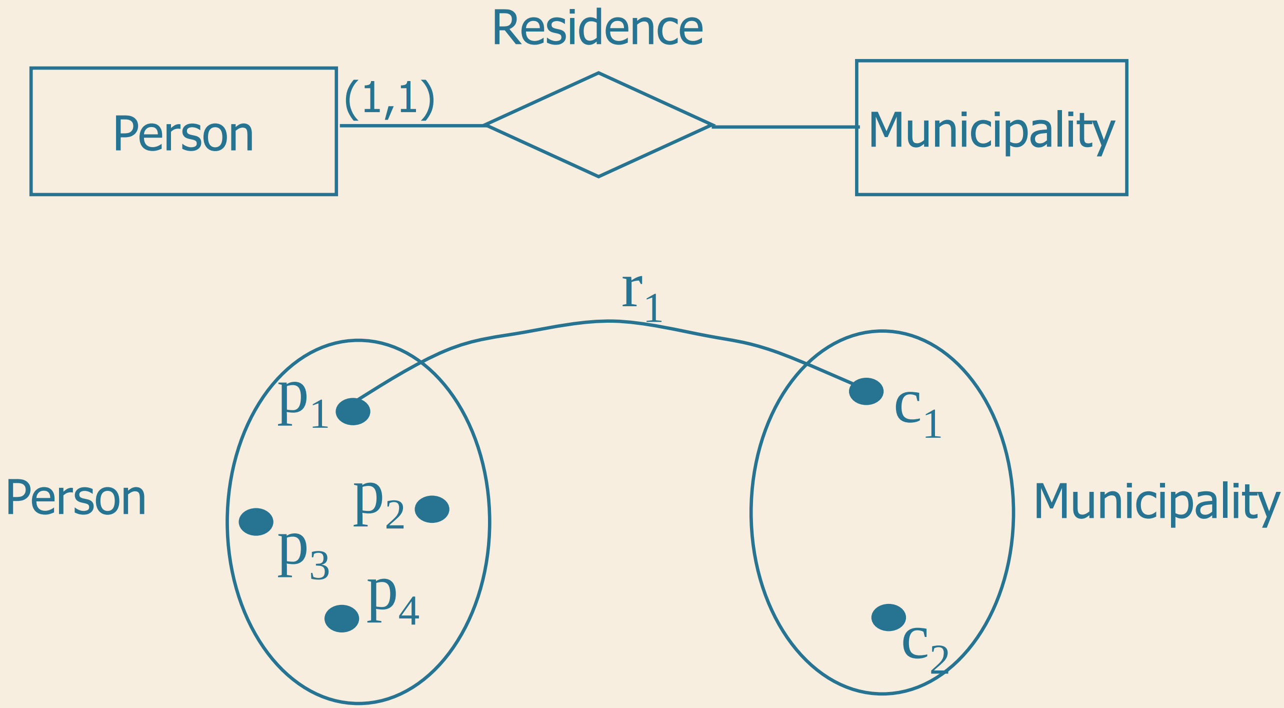
- 1 to N correspondence





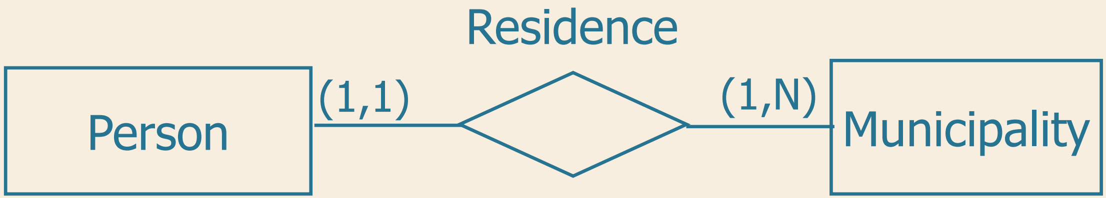
Cardinality of binary relationships

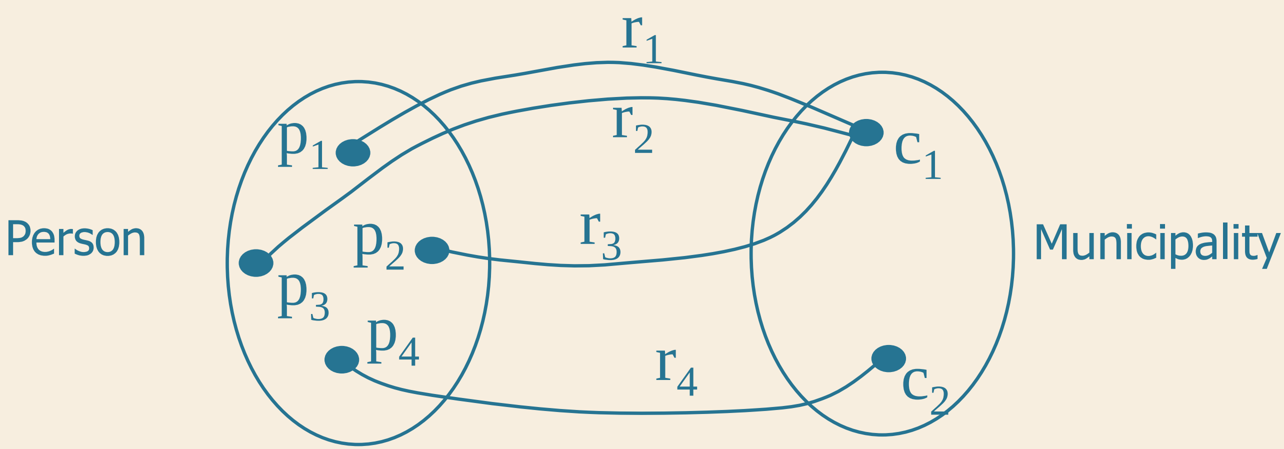
- 1 to N correspondence



Cardinality of binary relationships

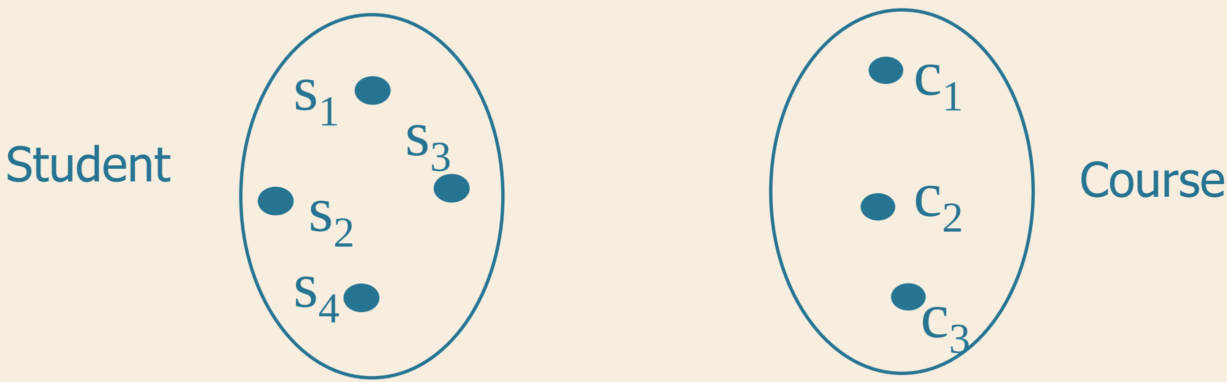
- 1 to N correspondence





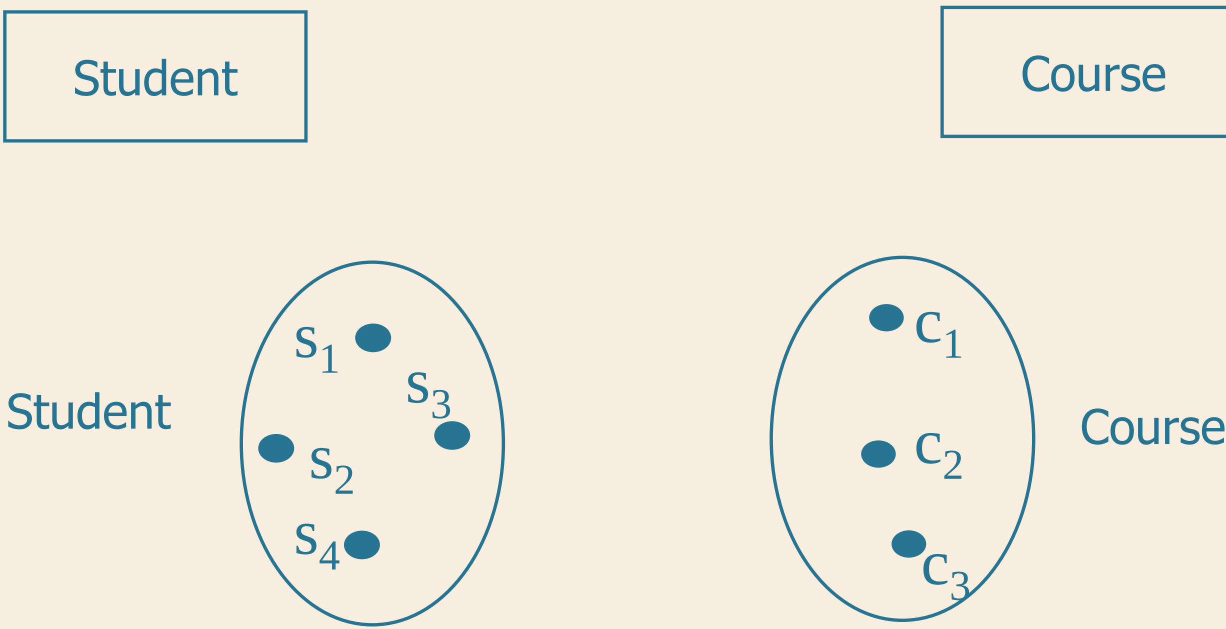
Cardinality of binary relationships

- N to N correspondence



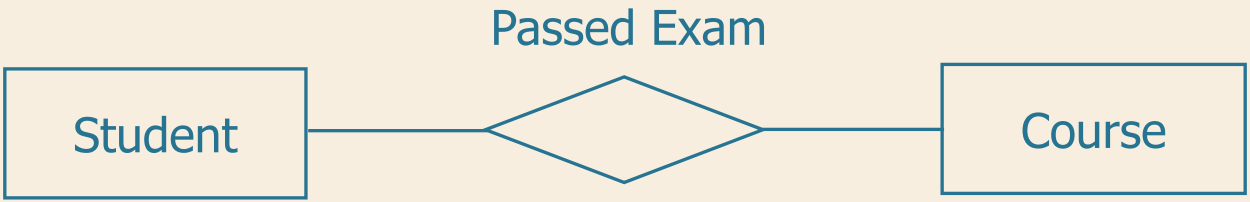
Cardinality of binary relationships

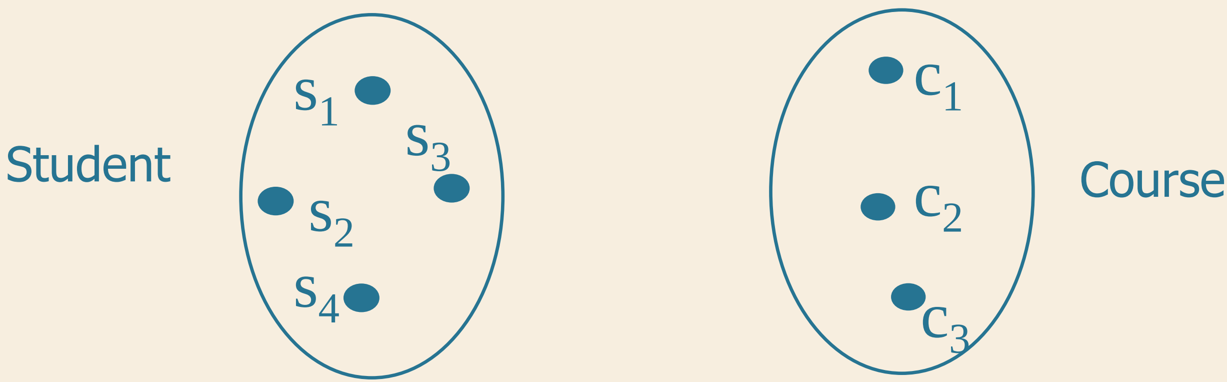
- N to N correspondence



Cardinality of binary relationships

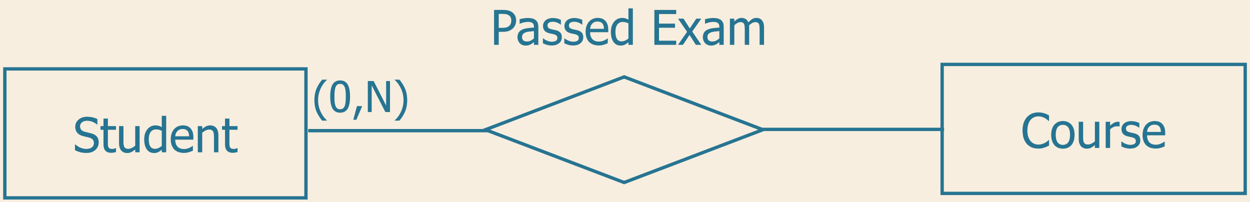
- N to N correspondence

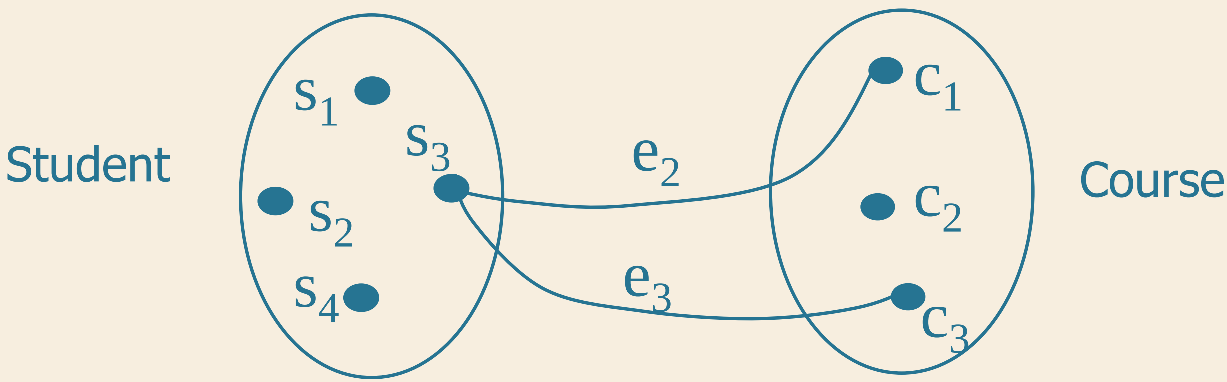




Cardinality of binary relationships

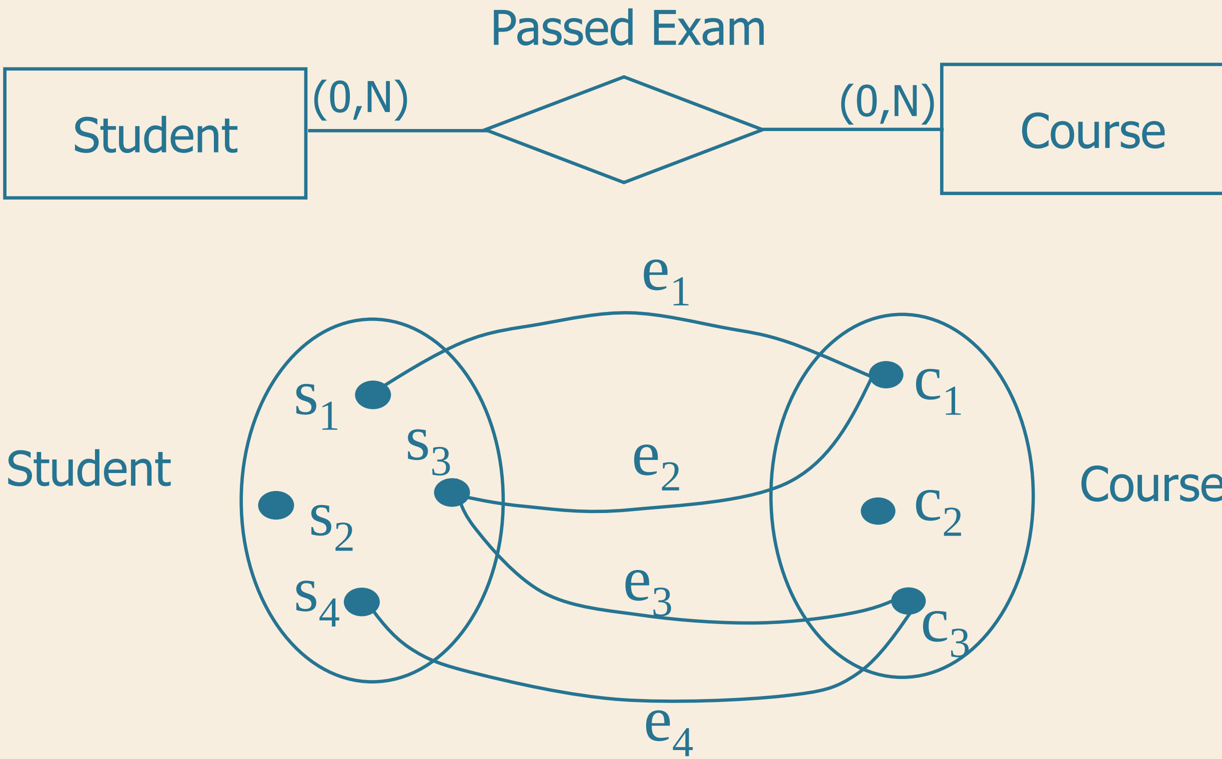
- N to N correspondence



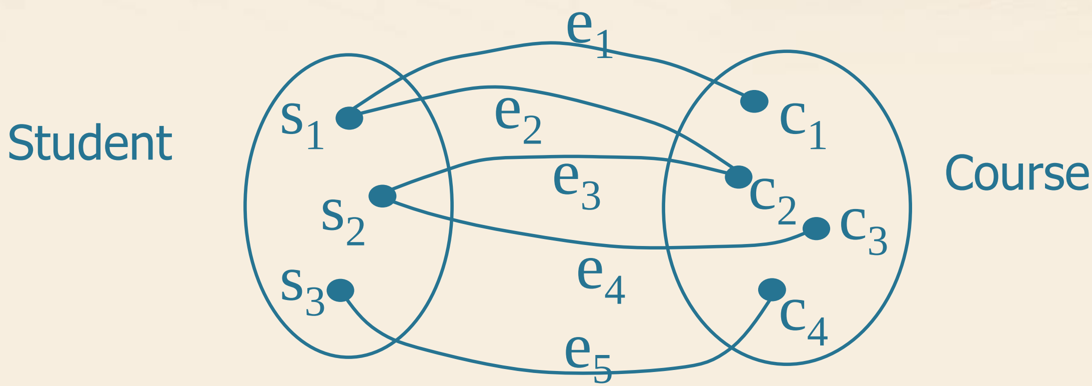


Cardinality of binary relationships

- 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

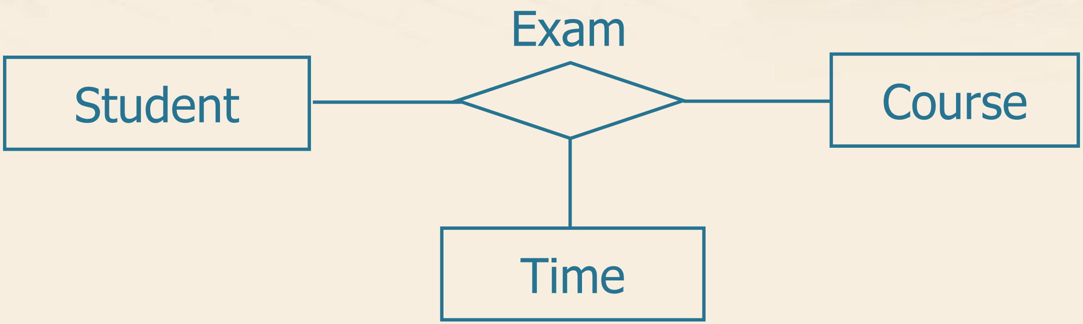
- A student can take the same exam more than once at different times
- Example of an exam instance

$s_1 \; c_1 \; t_1$

$s_1 \; c_1 \; t_2$

...

Ternary relationship



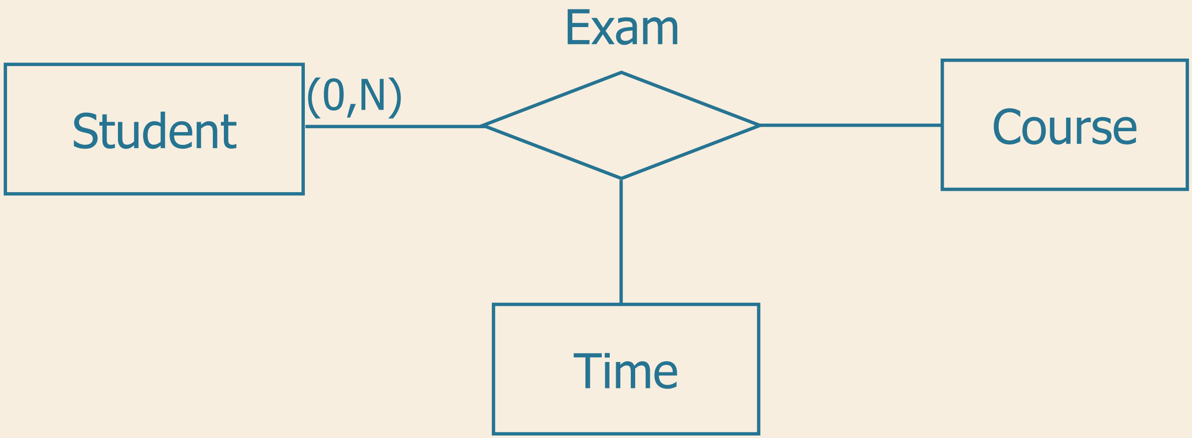
- A student can take the same exam more than once at different times
- Example of an exam instance

$s_1 \quad c_1 \quad t_1$

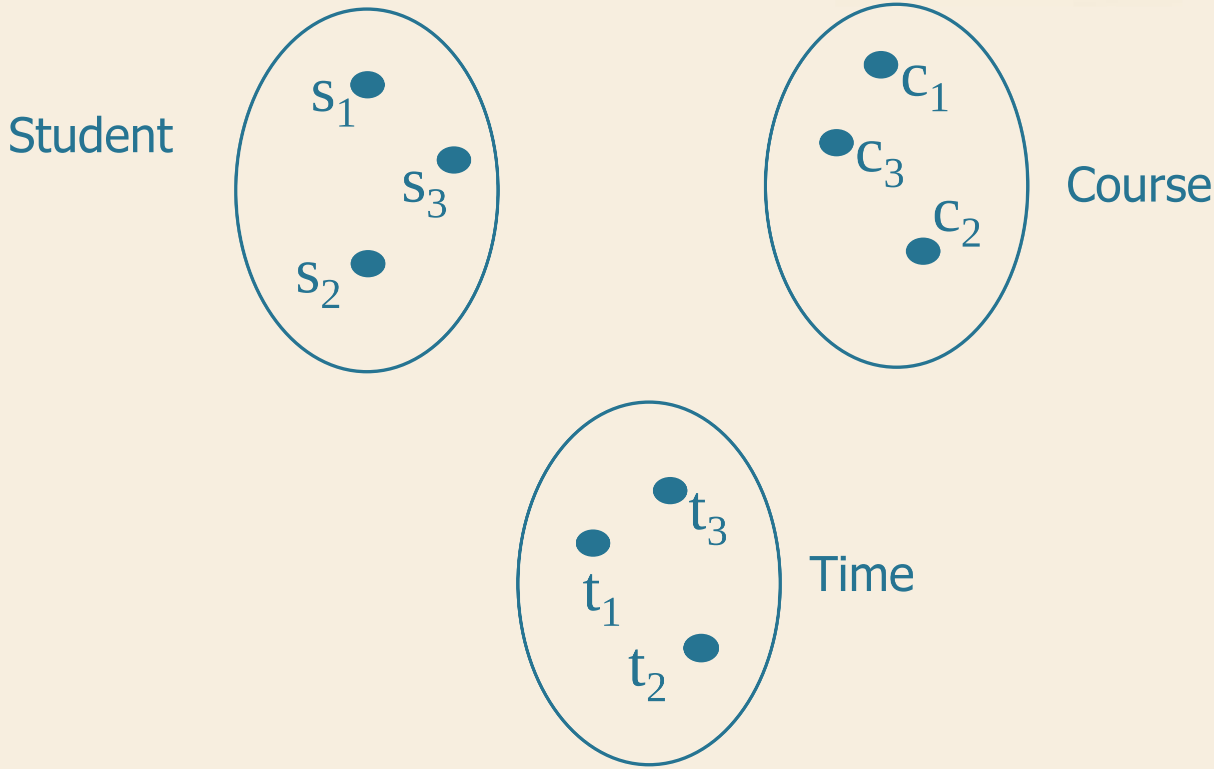
$s_1 \quad c_1 \quad t_2$

...

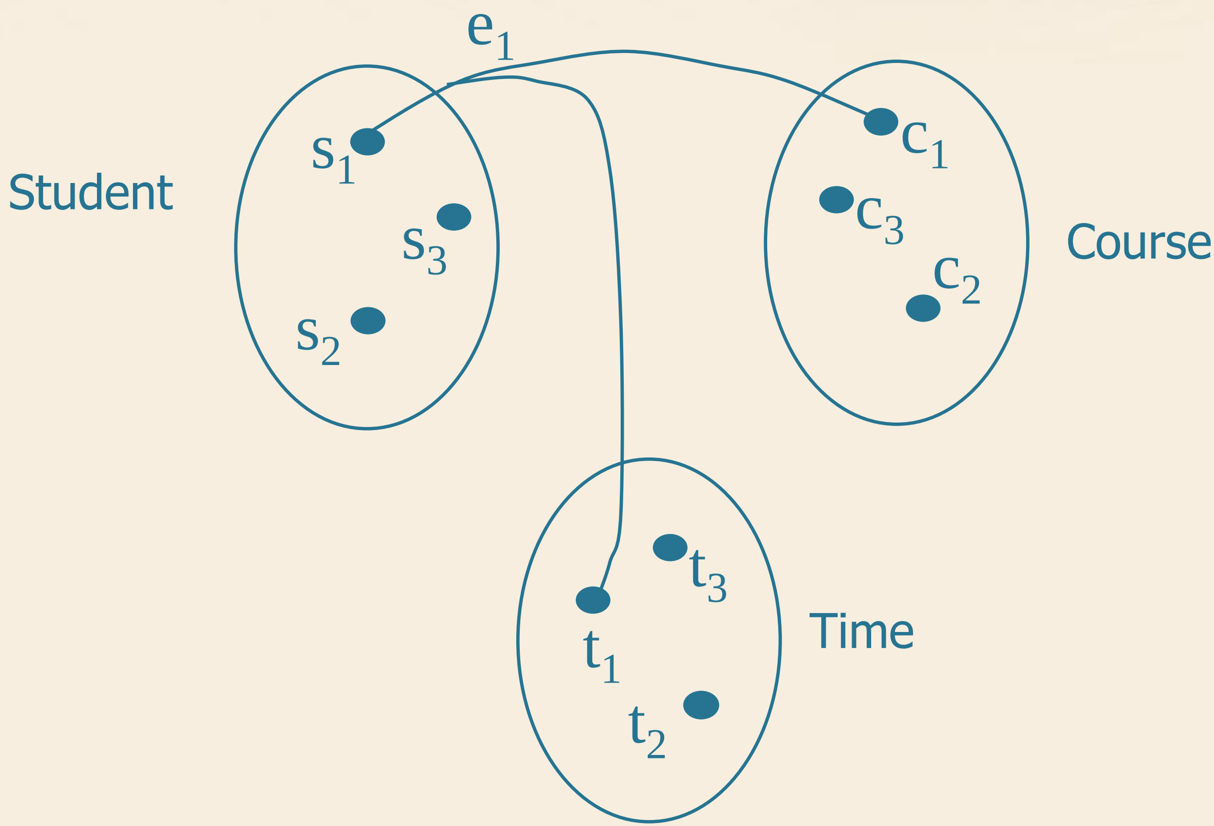
Cardinality of ternary relationships



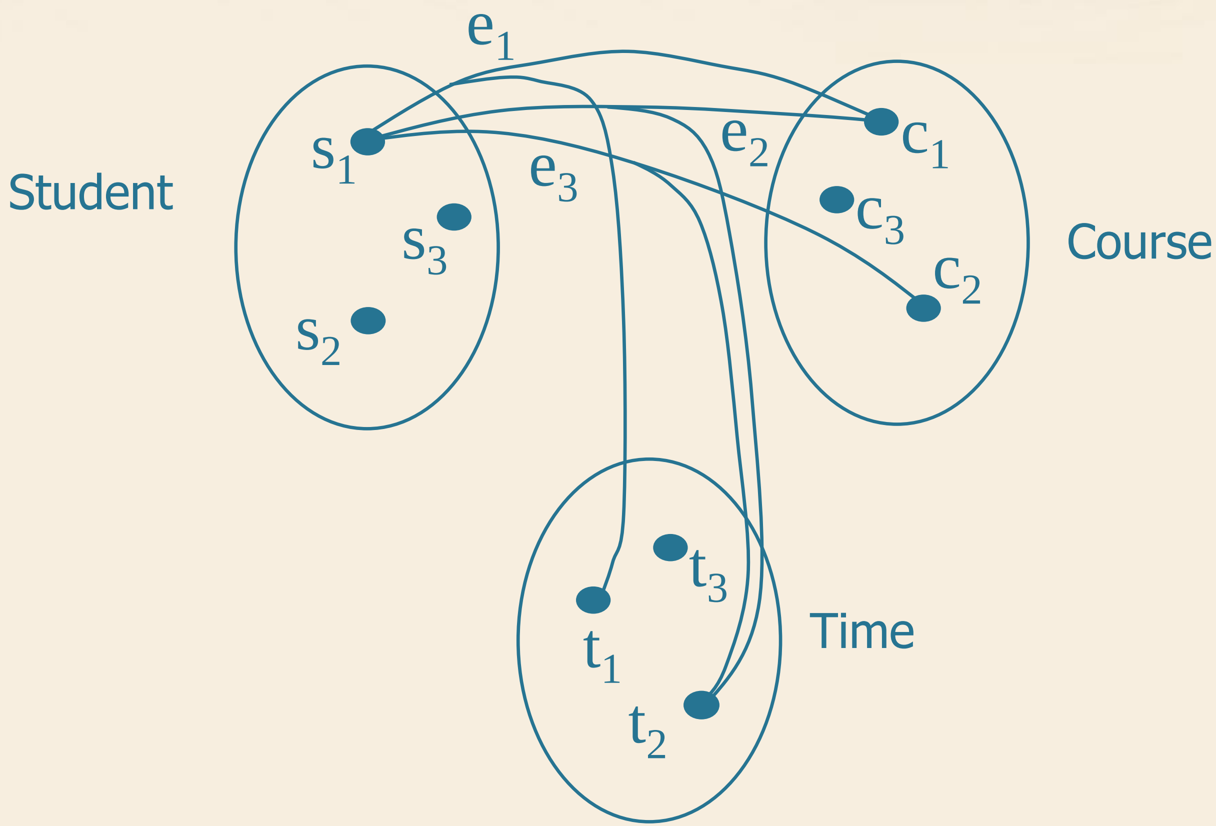
Occurences of a ternary relationship



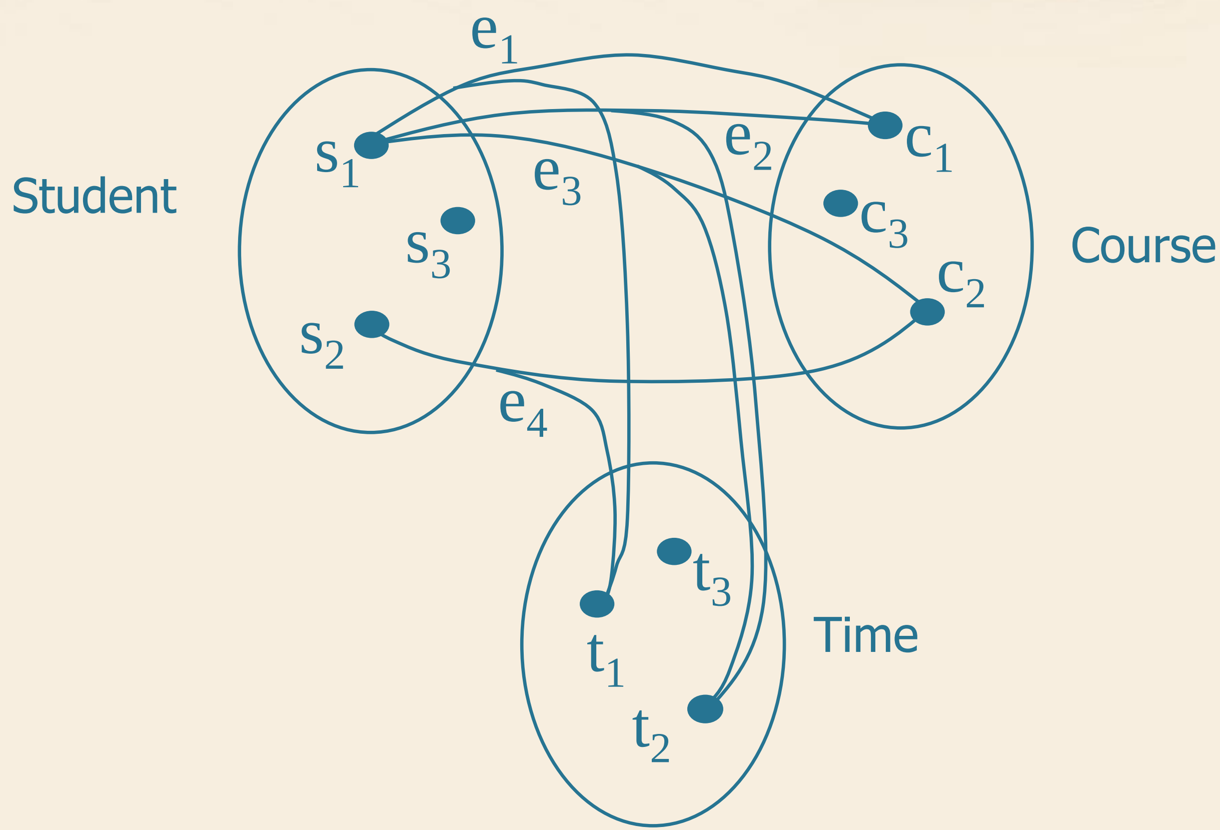
Occurences of a ternary relationship



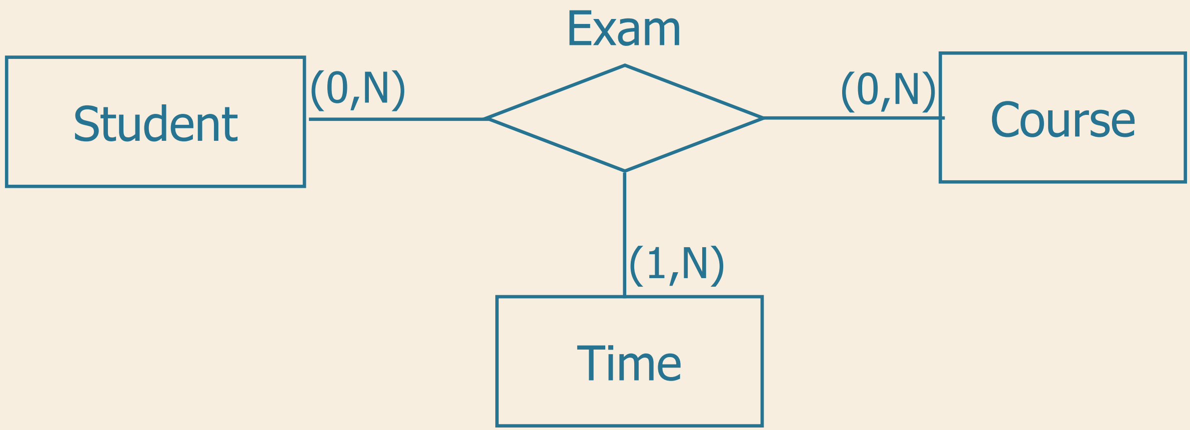
Occurences of a ternary relationship



Occurences of a ternary relationship



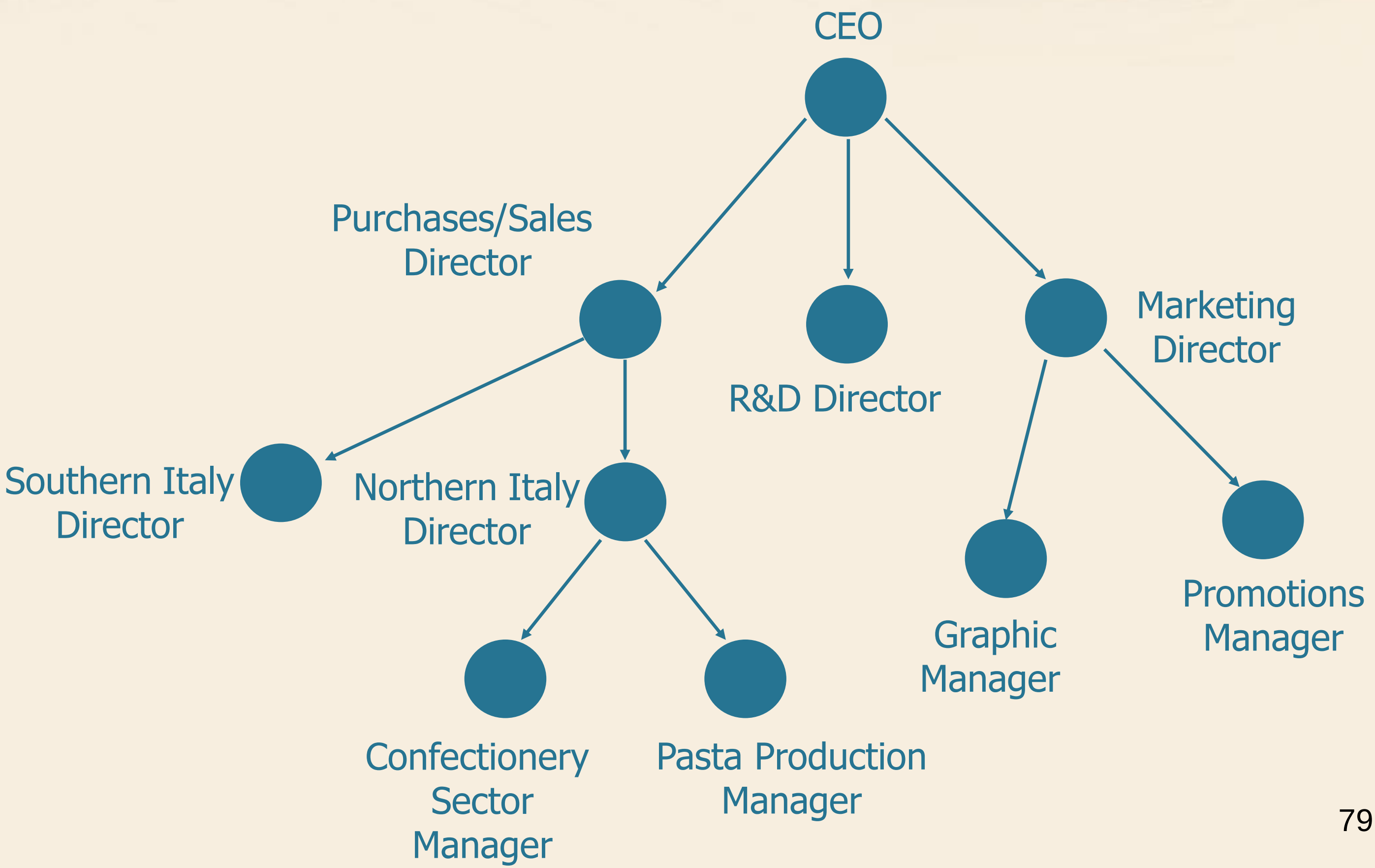
Cardinality of ternary relationships



Observations

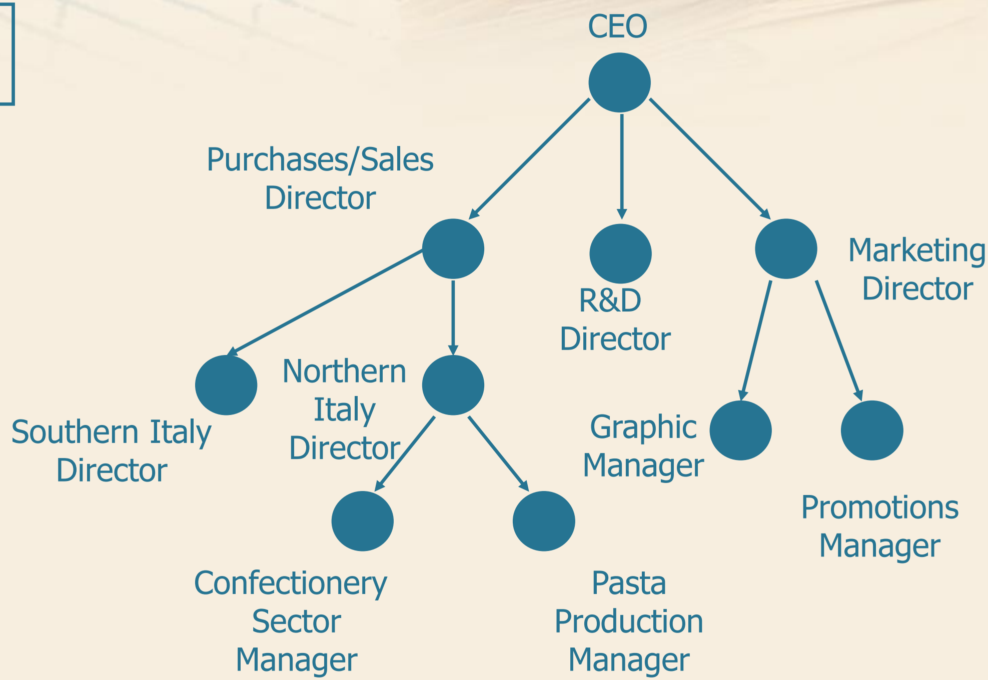
- Minimum cardinalities are rarely 1 for all entities involved in a relationship
- The maximum cardinalities of an n-ary relationship are (practically) always N
  - if the participation of an entity E has a maximum cardinality of 1, it is possible to eliminate the n-ary relationship and link entity E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

Recursive relationship

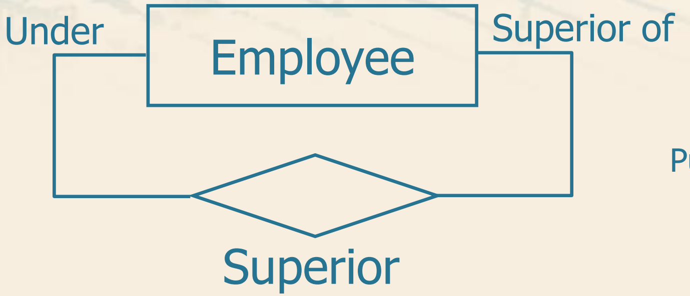


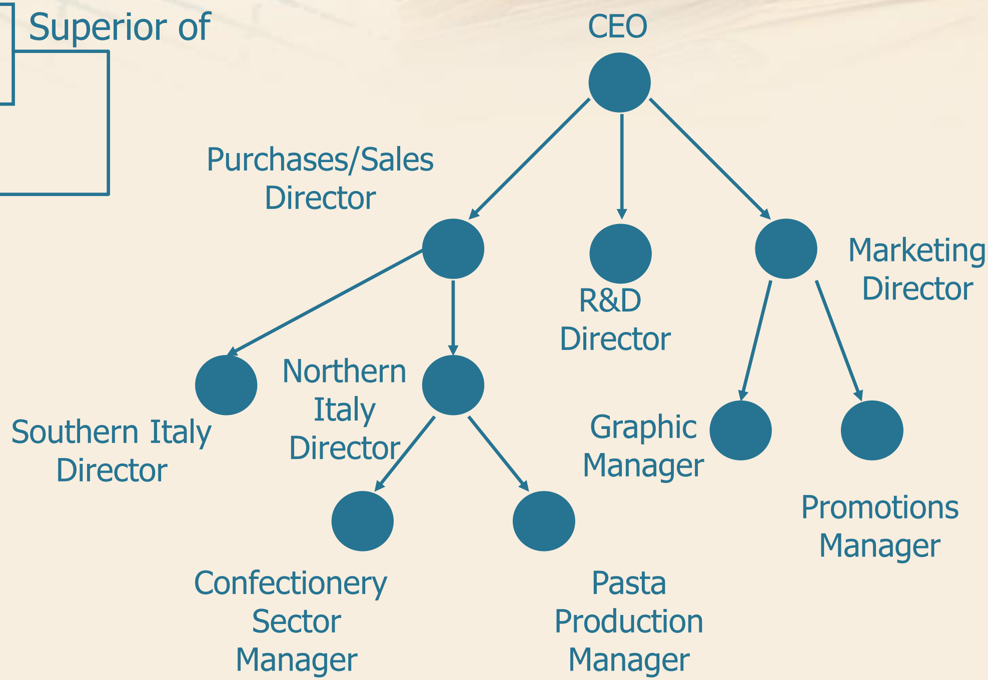
Recursive relationship

Employee



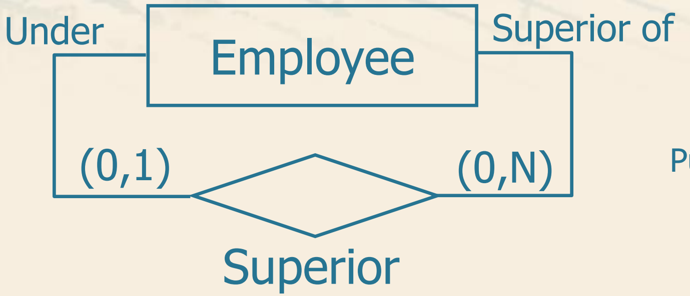
Recursive relationship

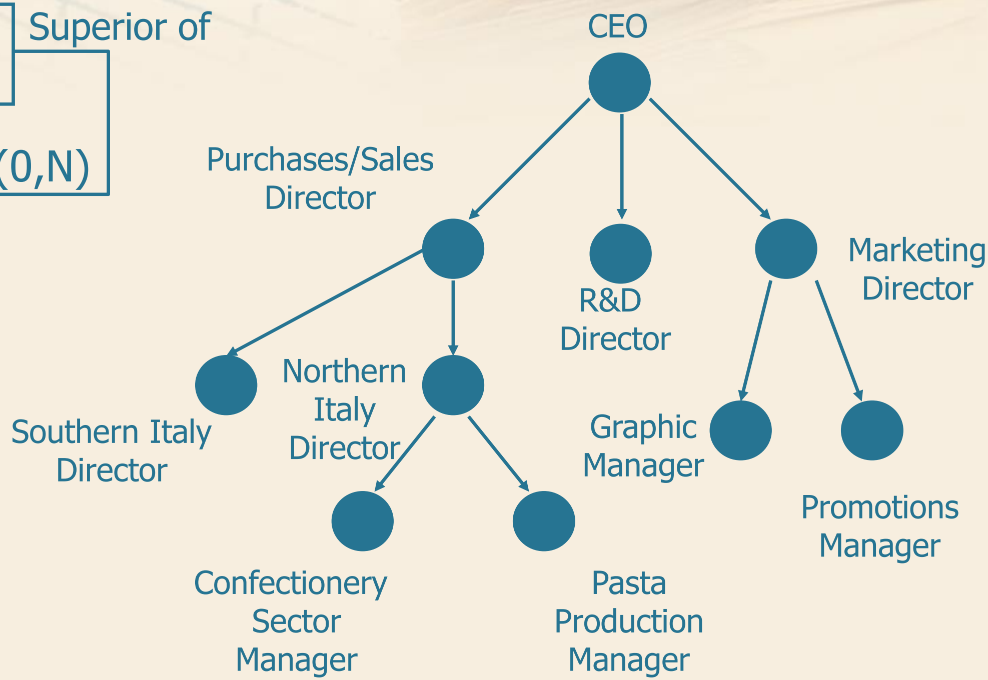




- Relationship between an entity and itself
- If the relationship is not symmetrical, the two roles of the entity must be defined

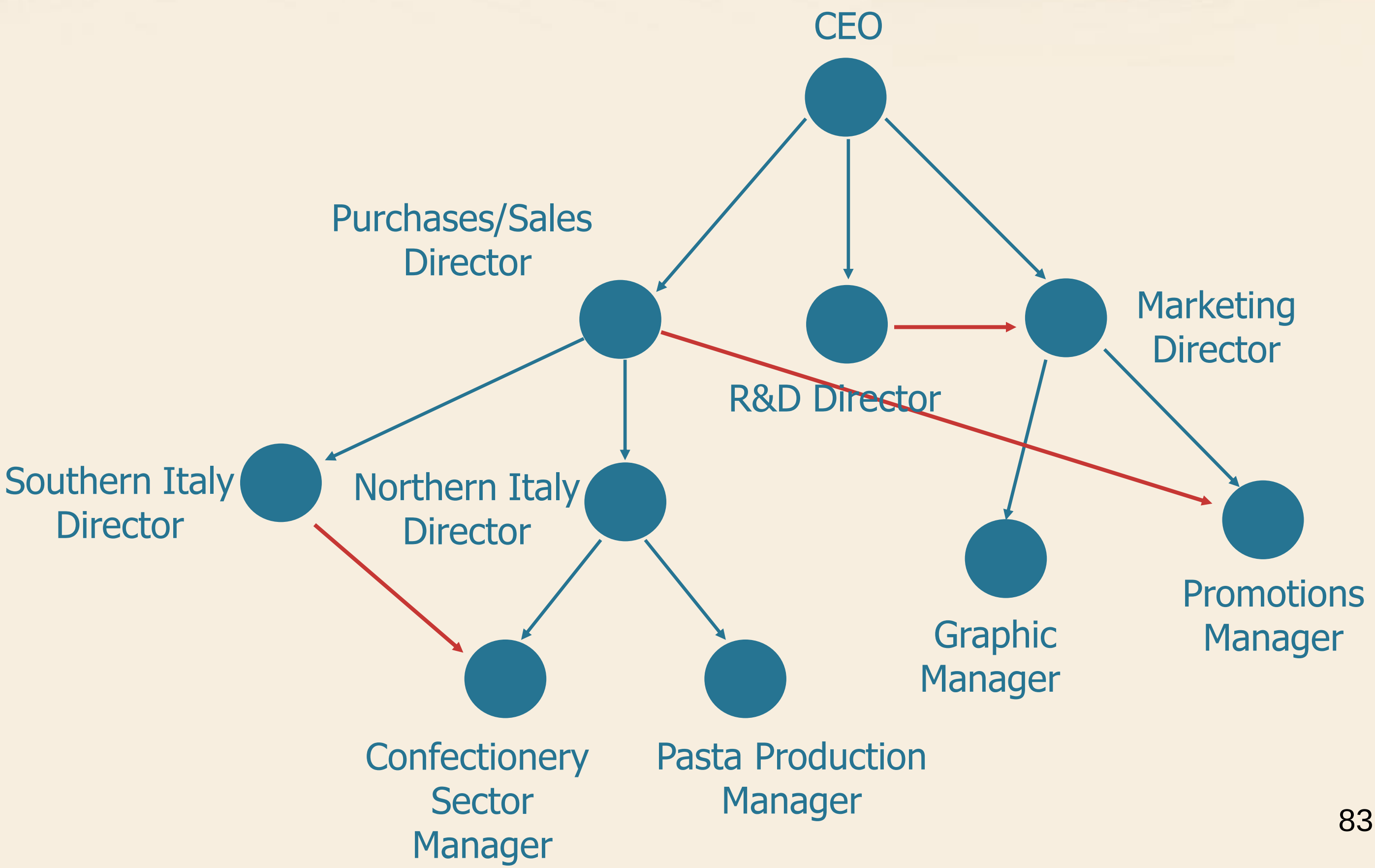
Recursive relationship





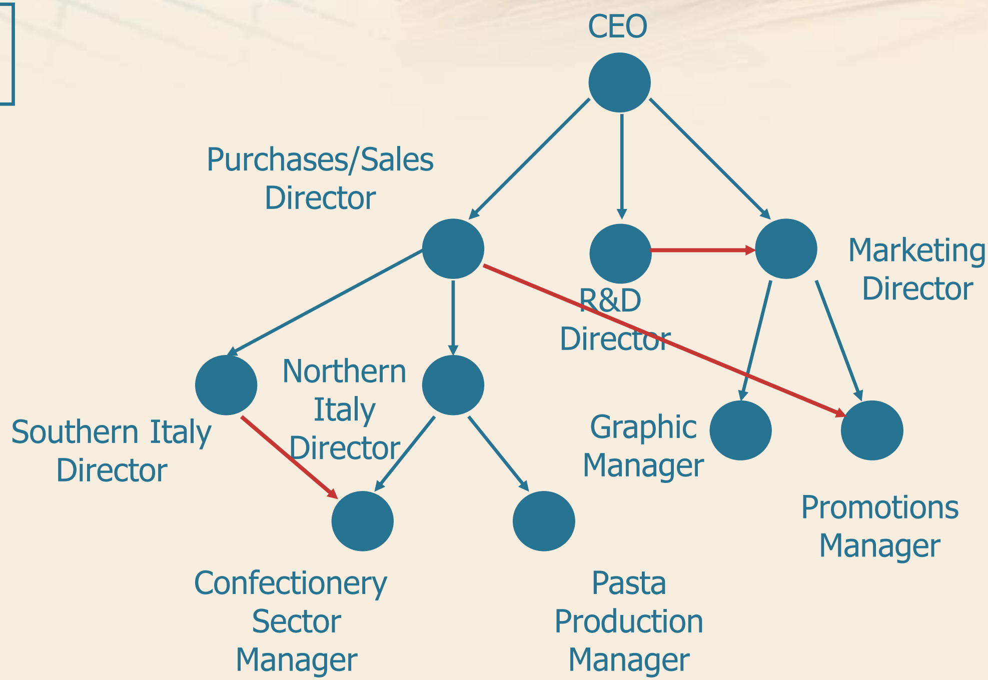
- Relationship between an entity and itself
- If the relationship is not symmetrical, the two roles of the entity must be defined

Recursive relationship

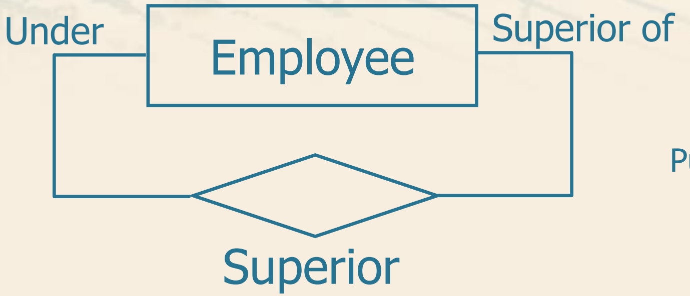


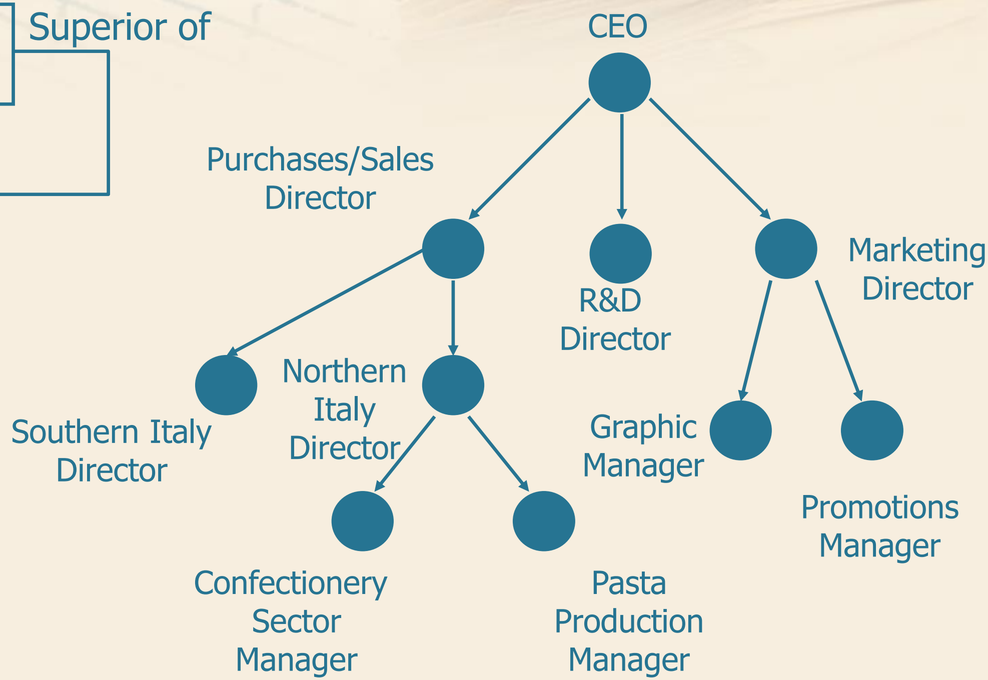
Recursive relationship

Employee



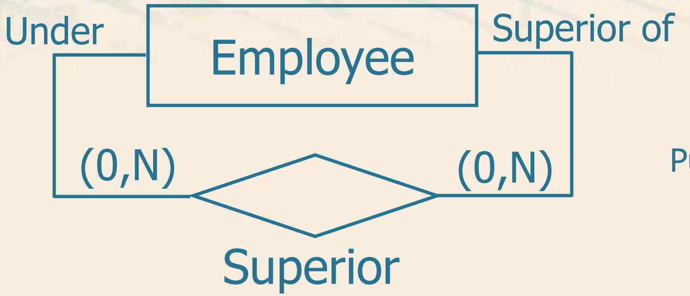
Recursive relationship

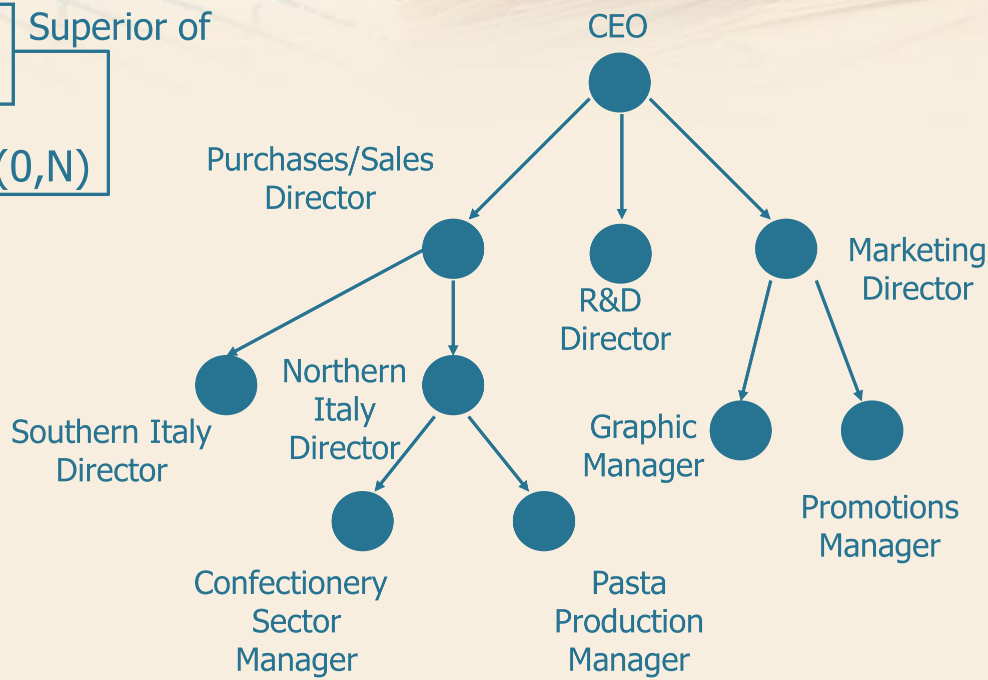




- An employee might have several superiors

Recursive relationship



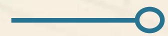


- An employee might have several superiors

Entity-Relationship model

Attributes

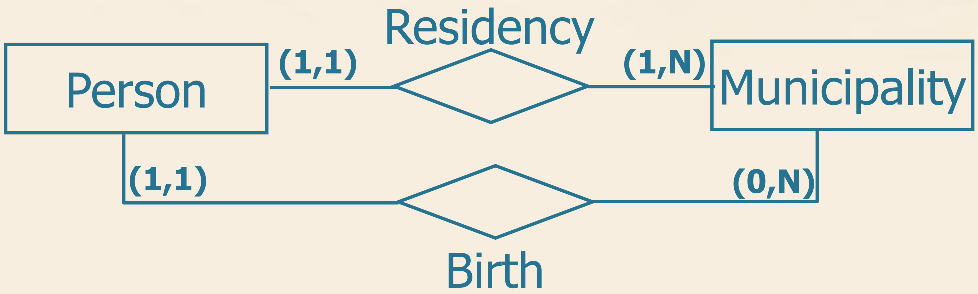
The attribute



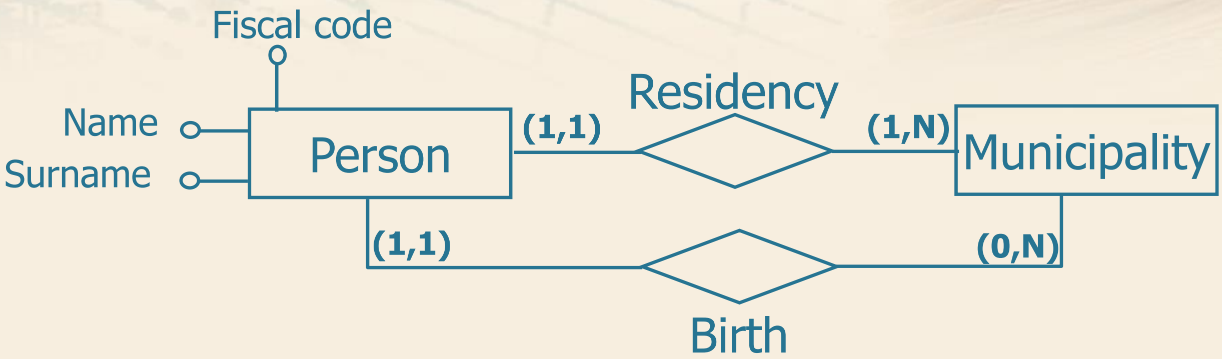
Name of the attribute

- It describes an elementary property of an entity or a relationship.
- Examples
  - Surname, name, student id are attributes that describe the entity student.
  - Grade is an attribute that describes the relationship exam.
- Each attribute is characterized by the *domain*, the set of eligible values for the attribute.

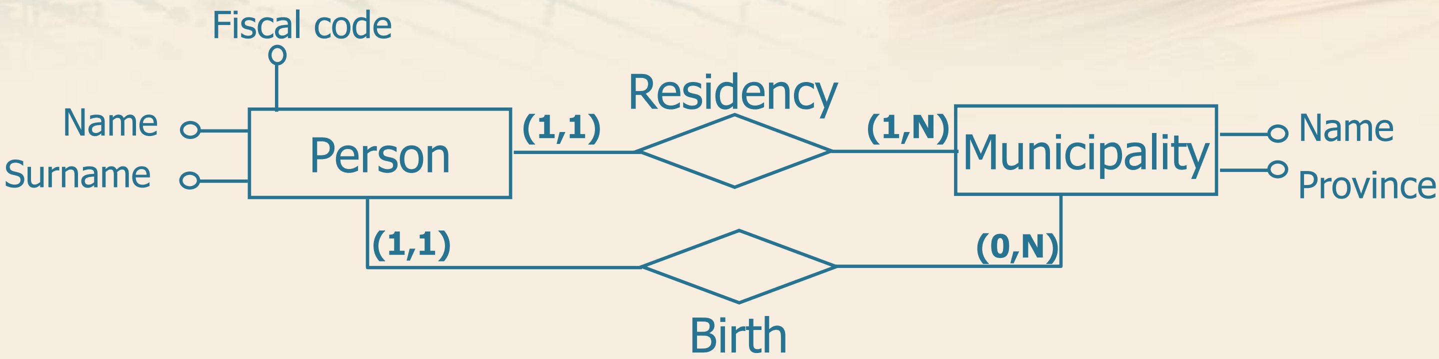
Examples of attributes



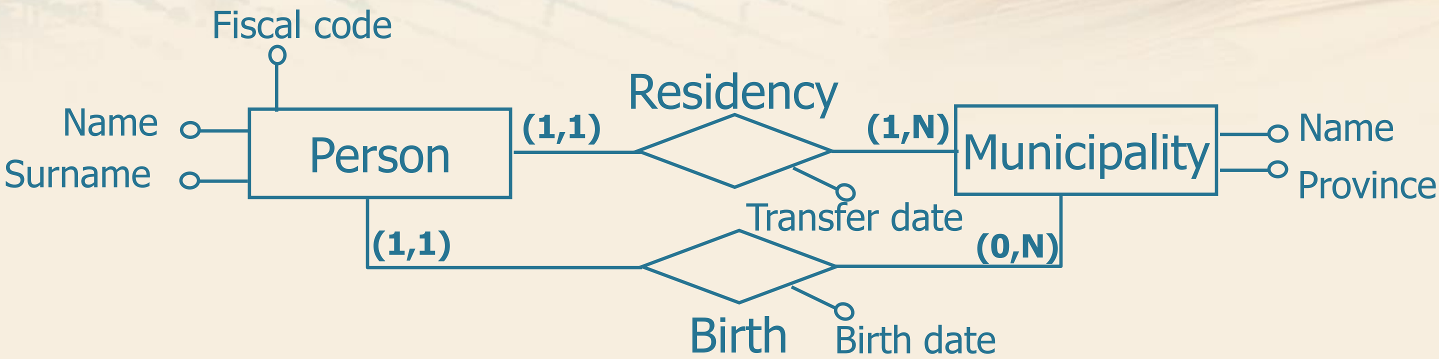
Examples of attributes



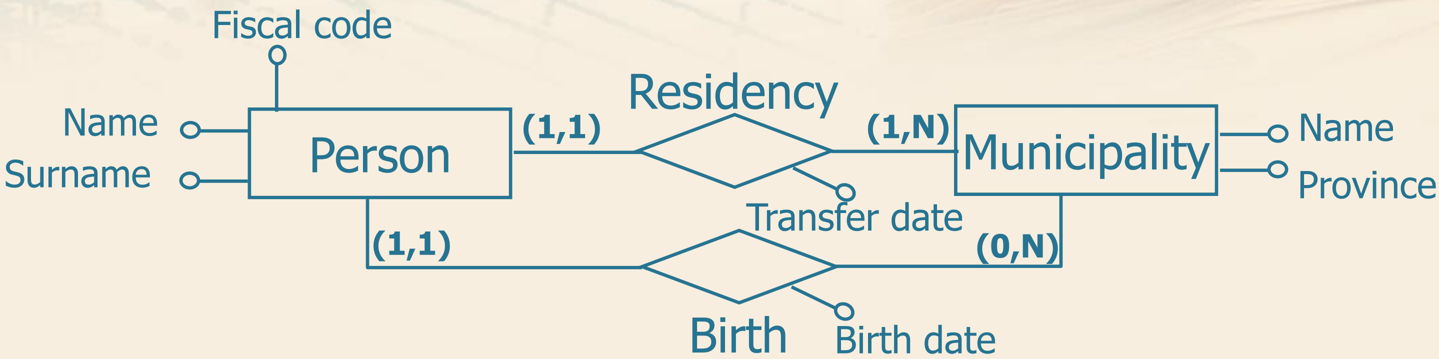
Examples of attributes

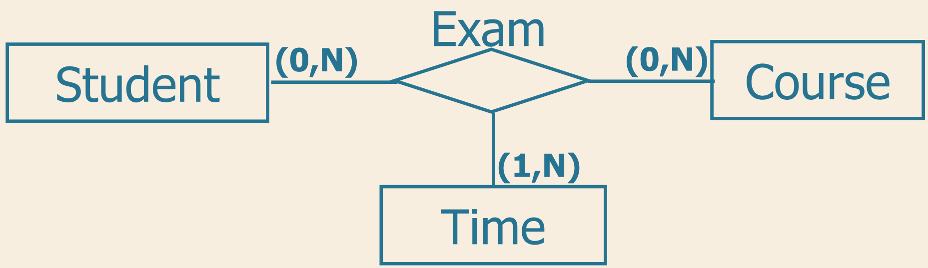


Examples of attributes

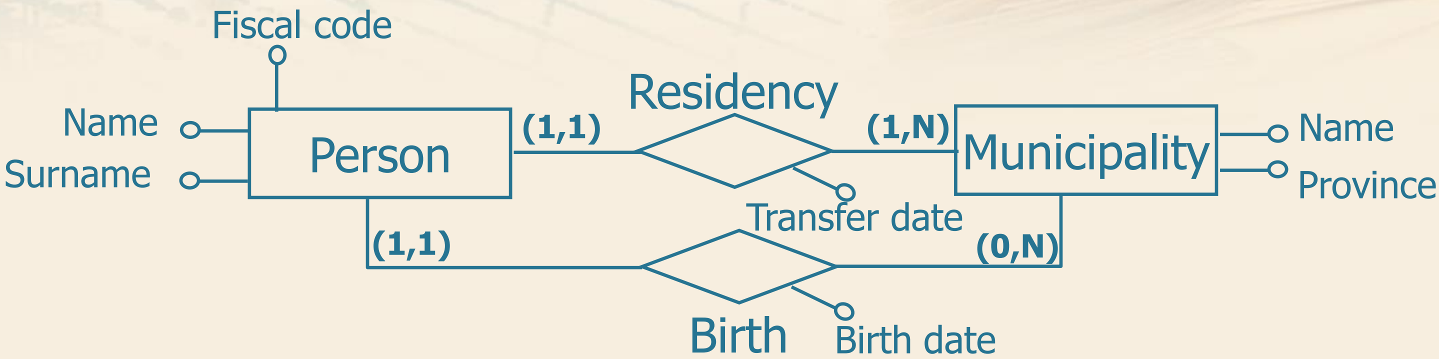


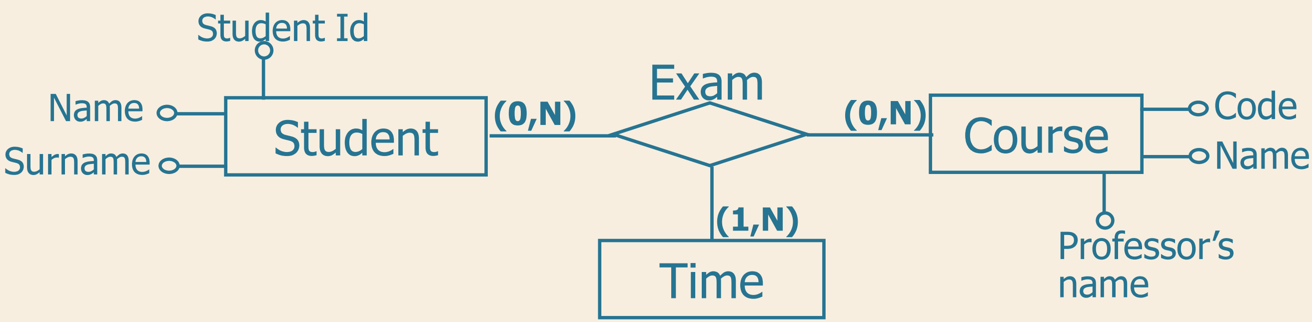
Examples of attributes



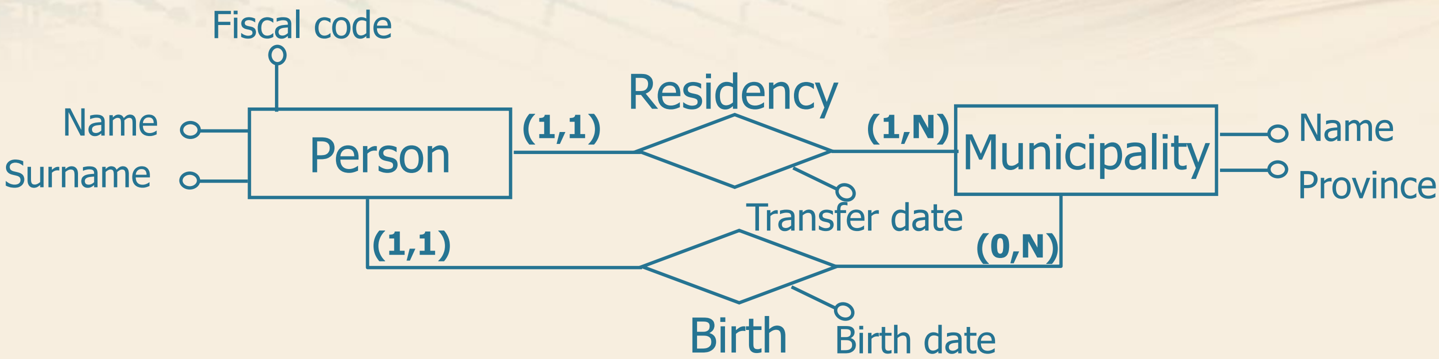


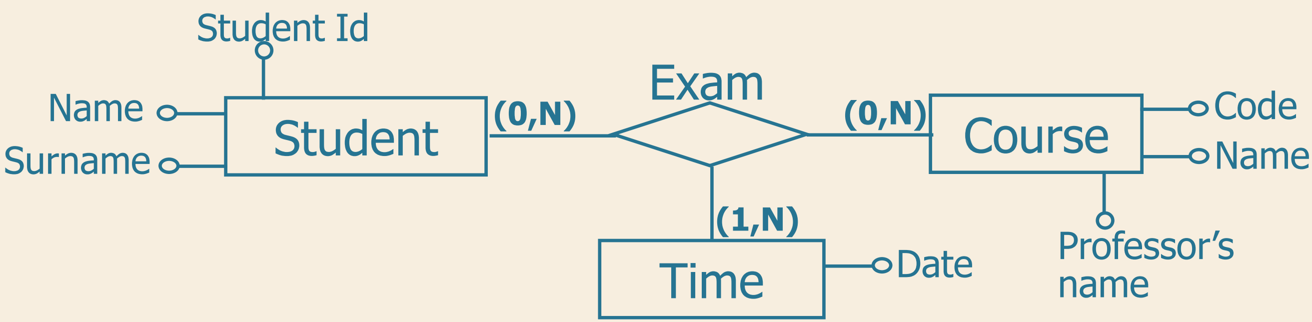
Examples of attributes



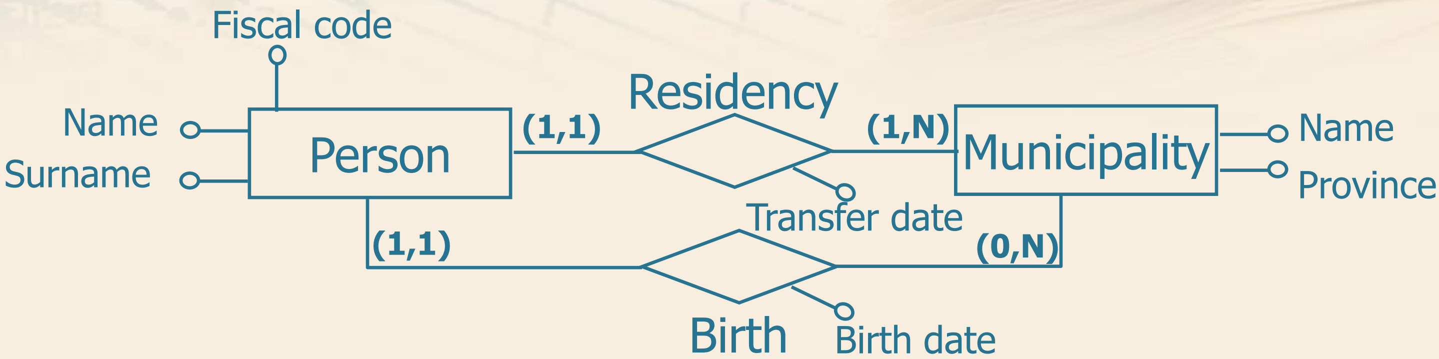


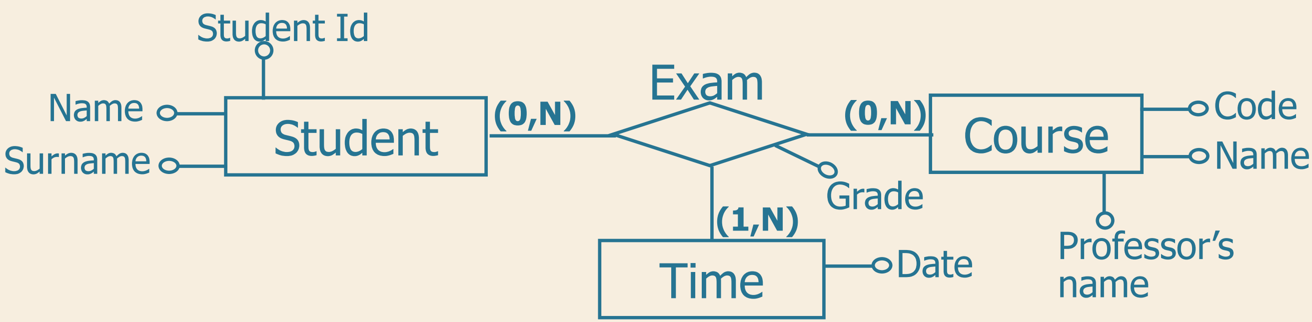
Examples of attributes



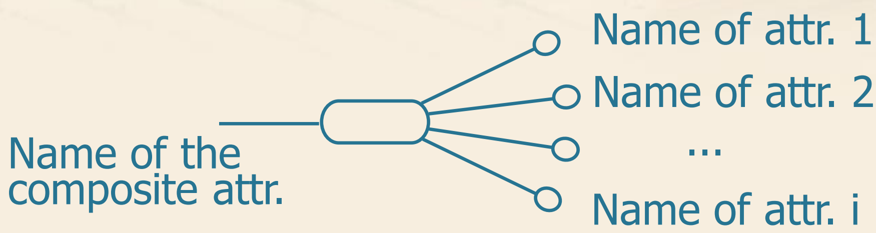


Examples of attributes



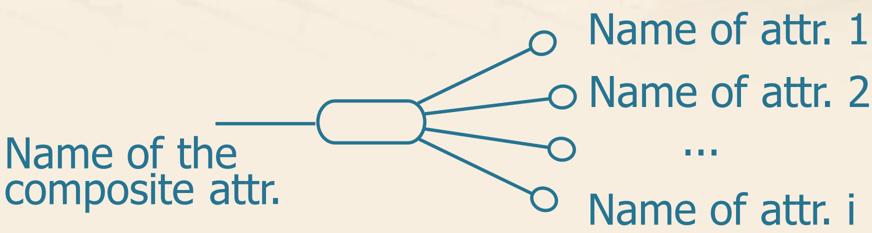


Composite attribute

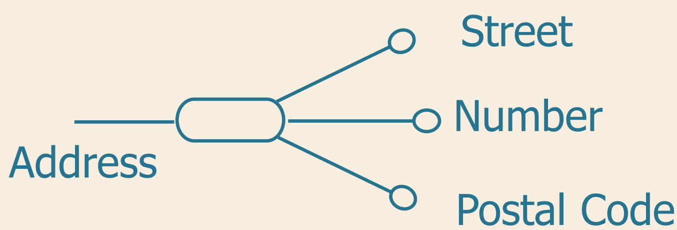


- Group of attributes that have closely connected meanings or uses.

Composite attribute



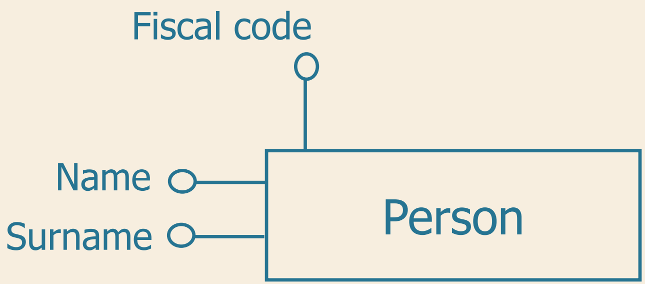
- Group of attributes that have closely connected meanings or uses.
- Example



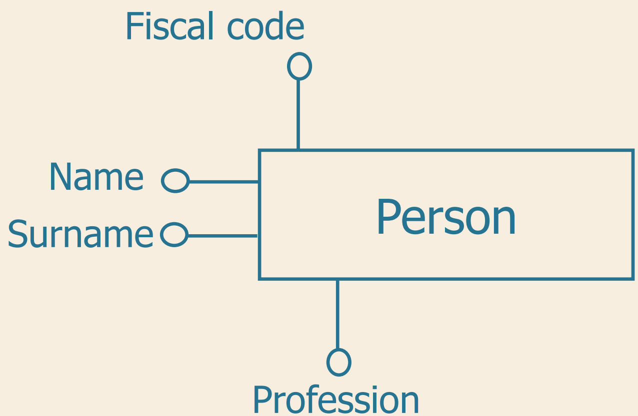
Cardinality of an attribute

- It can be specified for the attributes of entities or relationships
- It describes the minimum and maximum number of attribute's values associated to an instance of an entity or a relationship.
  - If omitted, it corresponds to (1,1)
  - minimum 0 corresponds to having an attribute that admits the null value
  - maximum N corresponds to having an attribute that can take more than one value for the same occurrence (multivalued attribute)

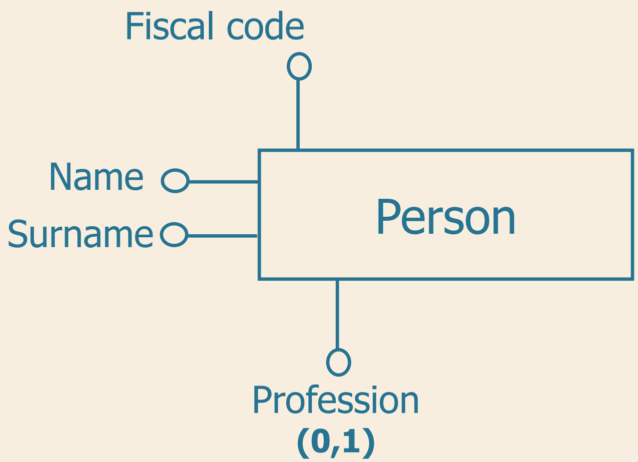
Cardinality of an attribute



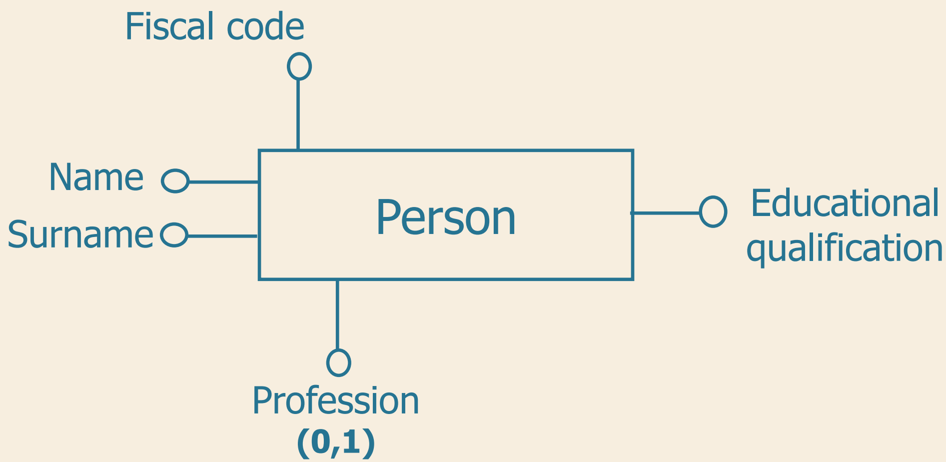
Cardinality of an attribute



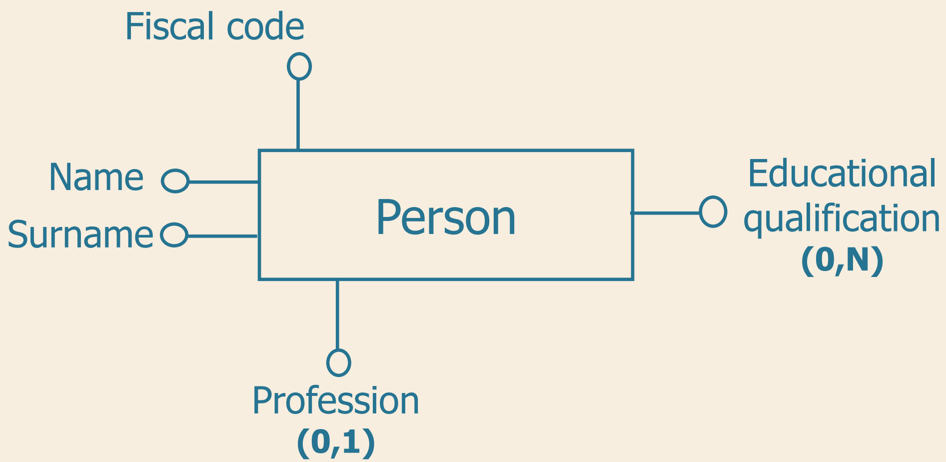
Cardinality of an attribute



Cardinality of an attribute



Cardinality of an attribute



The Entity-Relationship Model

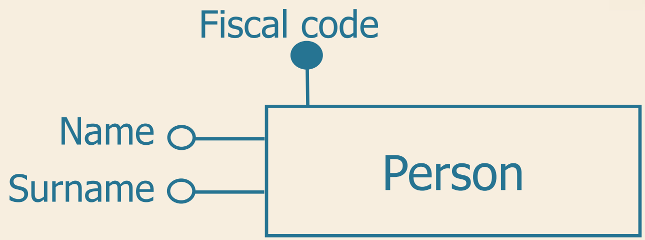
Identifiers

Identifier

- It is specified for each entity
- It describes concepts (attributes and/or entities) of the scheme that allow to identify uniquely the instances of an entity.
  - Each entity must have at least one identifier
  - There can be be more than one appropriated identifier for a given entity.

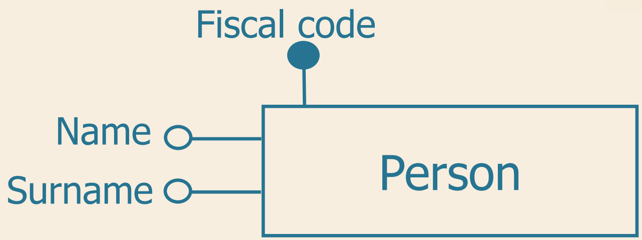
Internal Identifier

- Simple: consisting of only one attribute

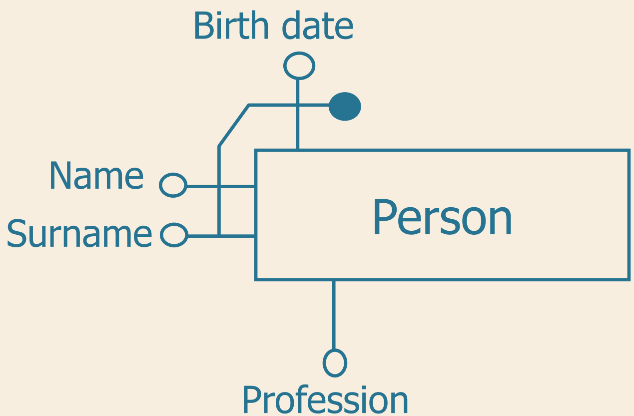


Internal Identifier

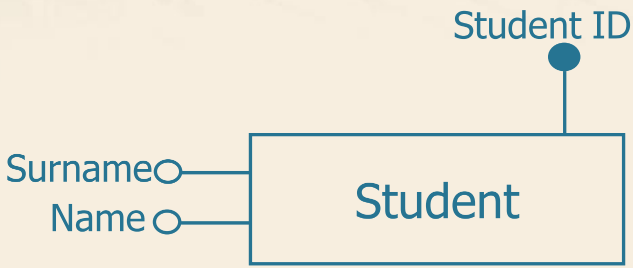
- Simple: consisting of only one attribute



- Composite: consisting of multiple attributes

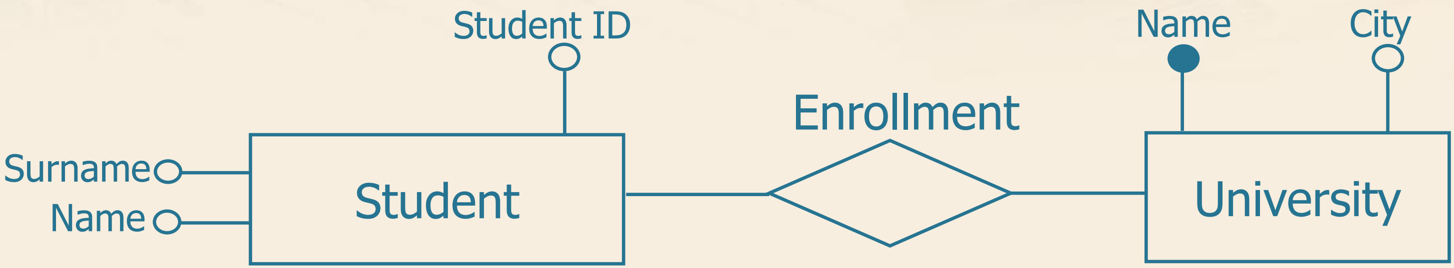


External Identifier

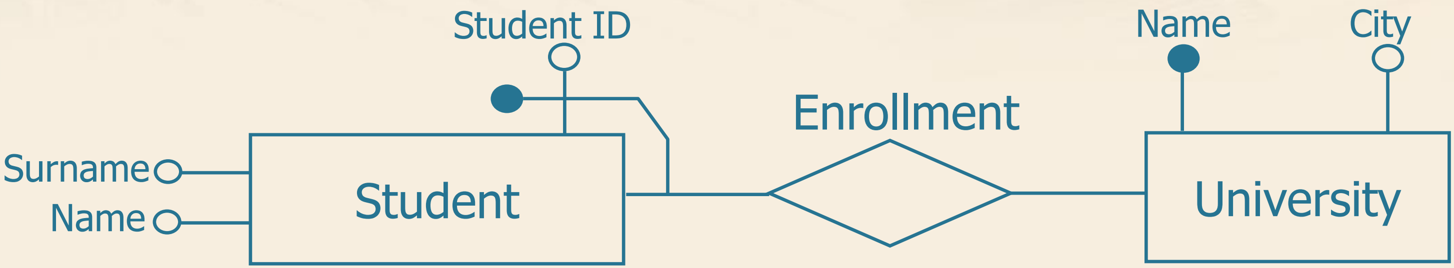


External Identifier

External Identifier

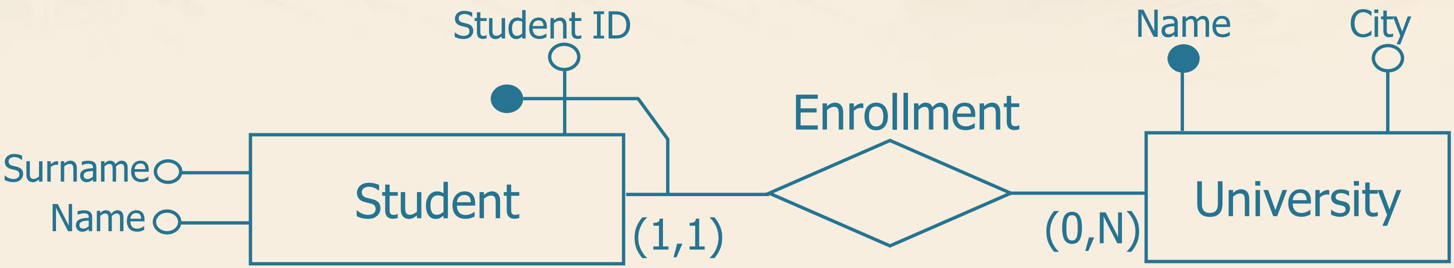


External Identifier



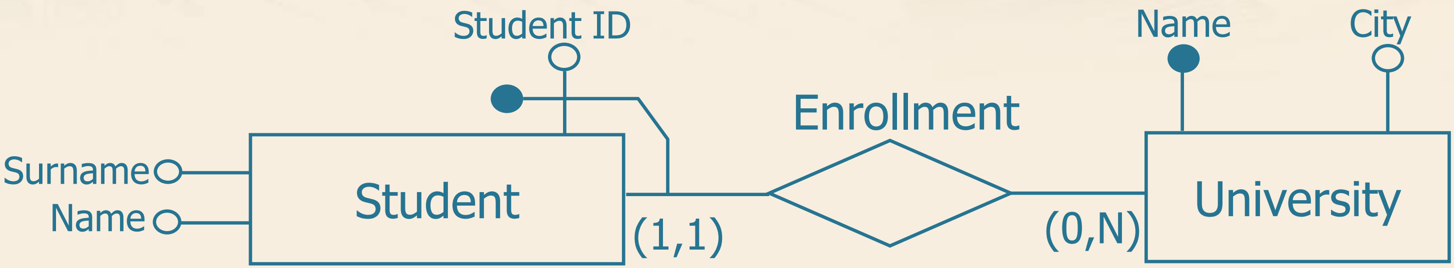
- One entity that does not have sufficient internal attributes able to define an identifier is called *weak entity*.

External Identifier



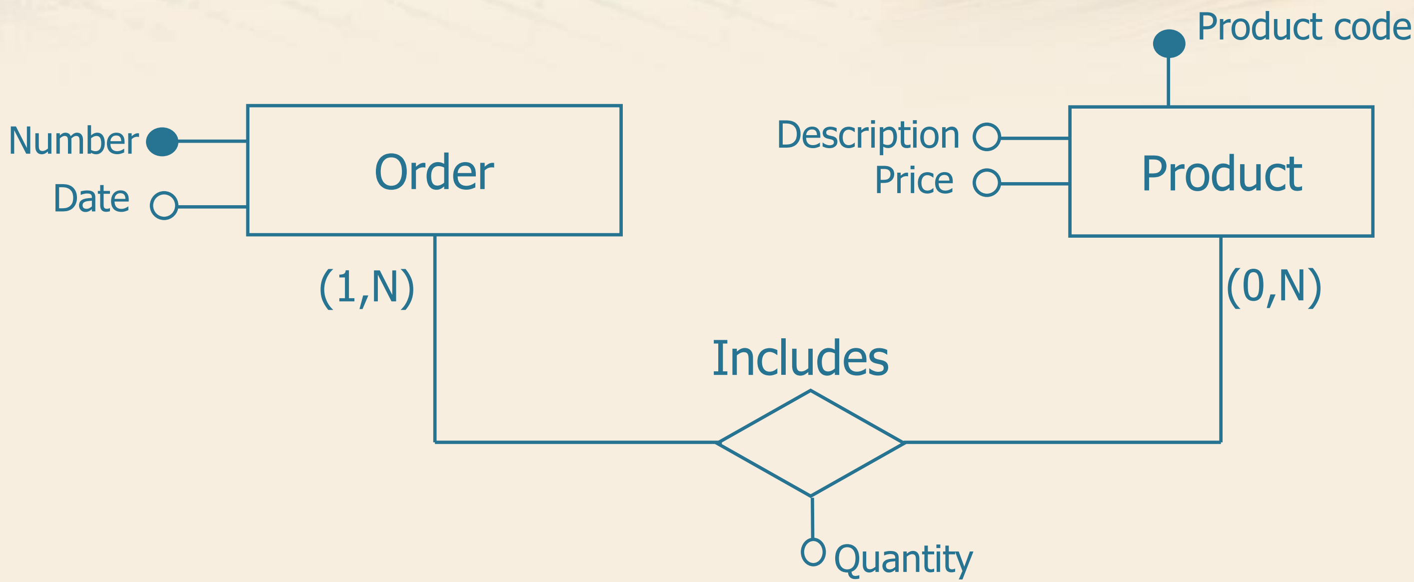
- One entity that does not have sufficient internal attributes able to define an identifier is called *weak entity*.

External Identifier

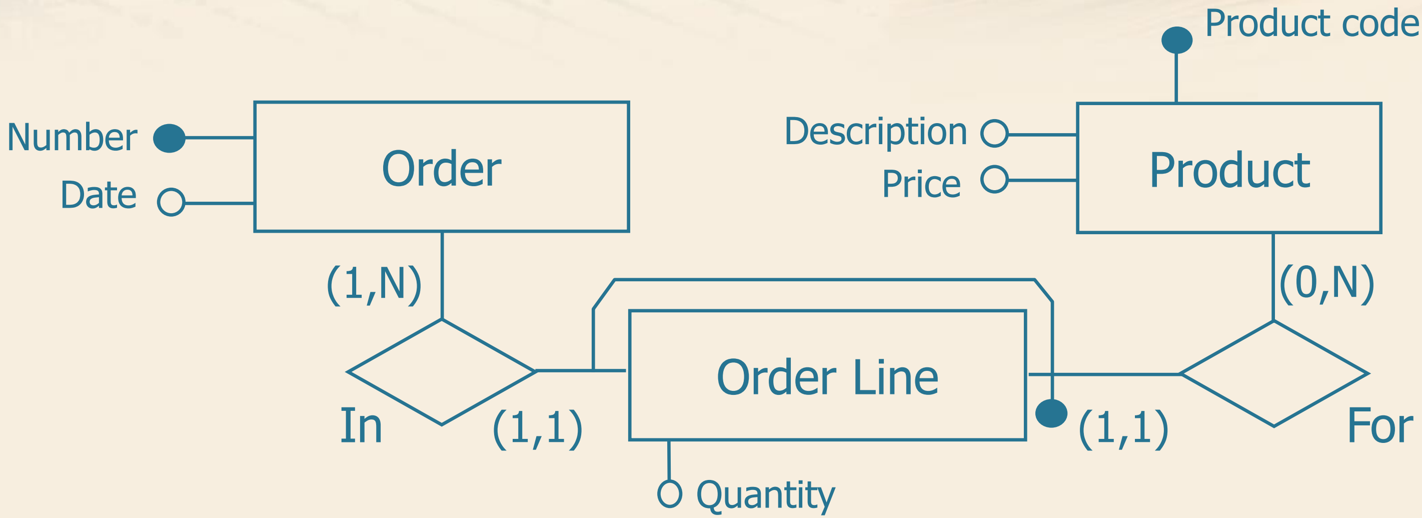


- One entity that does not have sufficient internal attributes able to define an identifier is called *weak entity*.
- A weak entity must partecipate with cardinality (1,1) in each of the relationships that provide part of the identifier

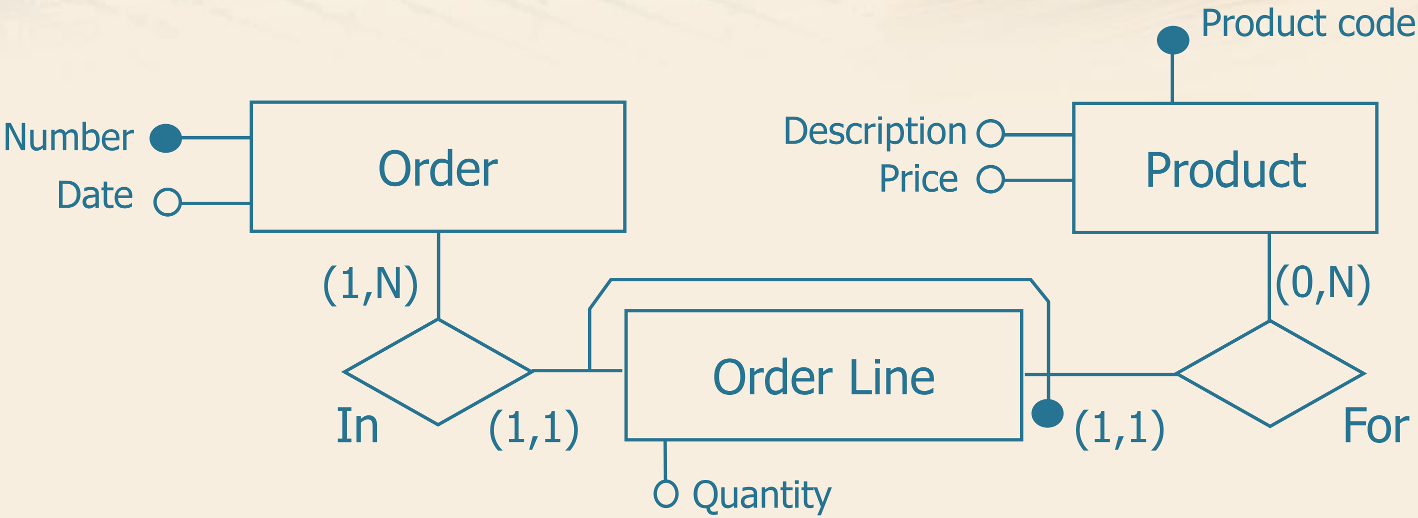
External Identifier



External Identifier

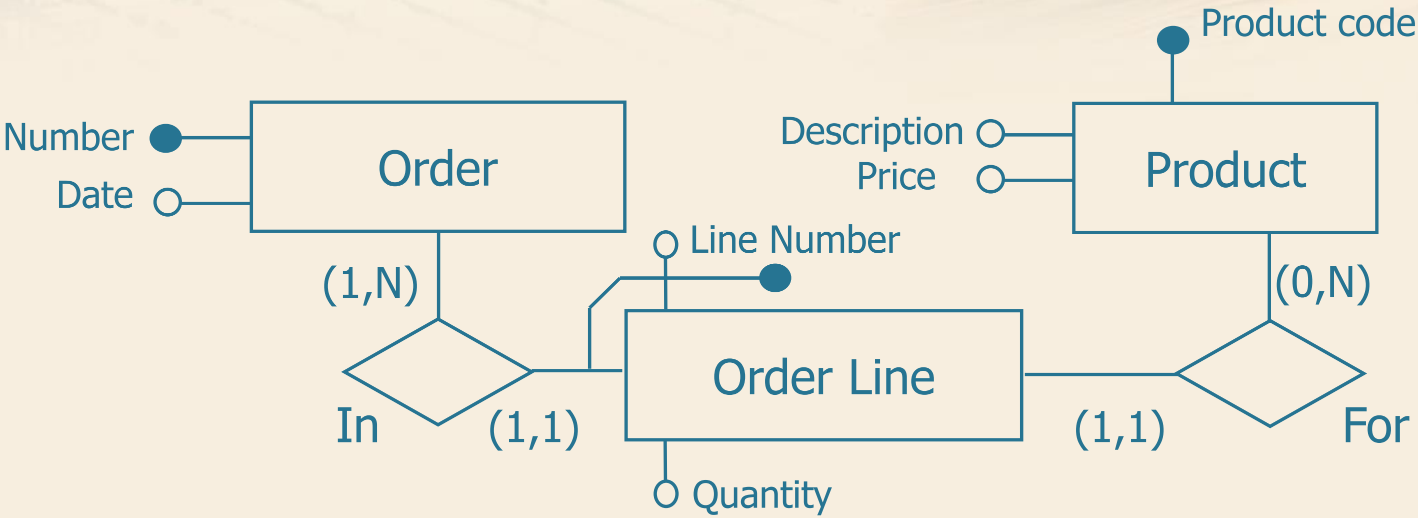


External Identifier



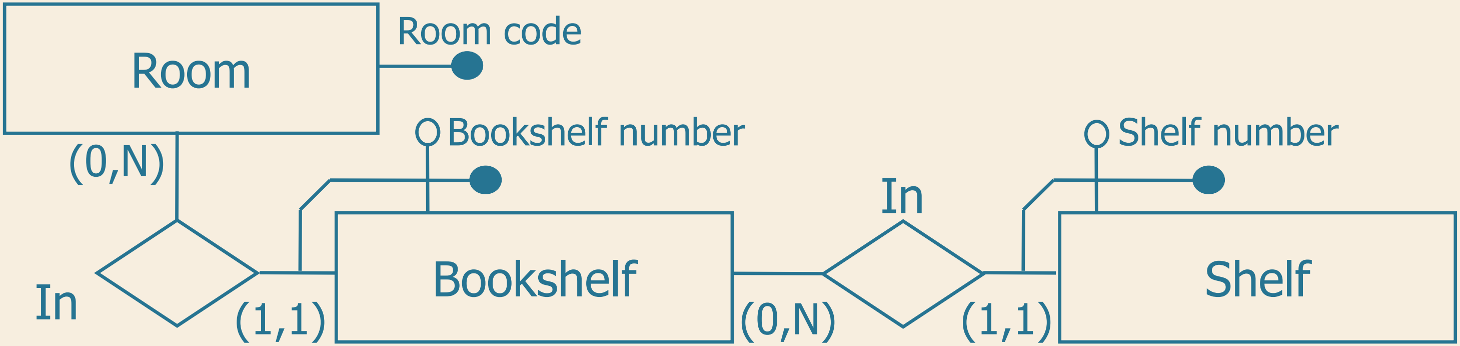
- Is it possible to represent in the same order more order lines for the same product?

External Identifier



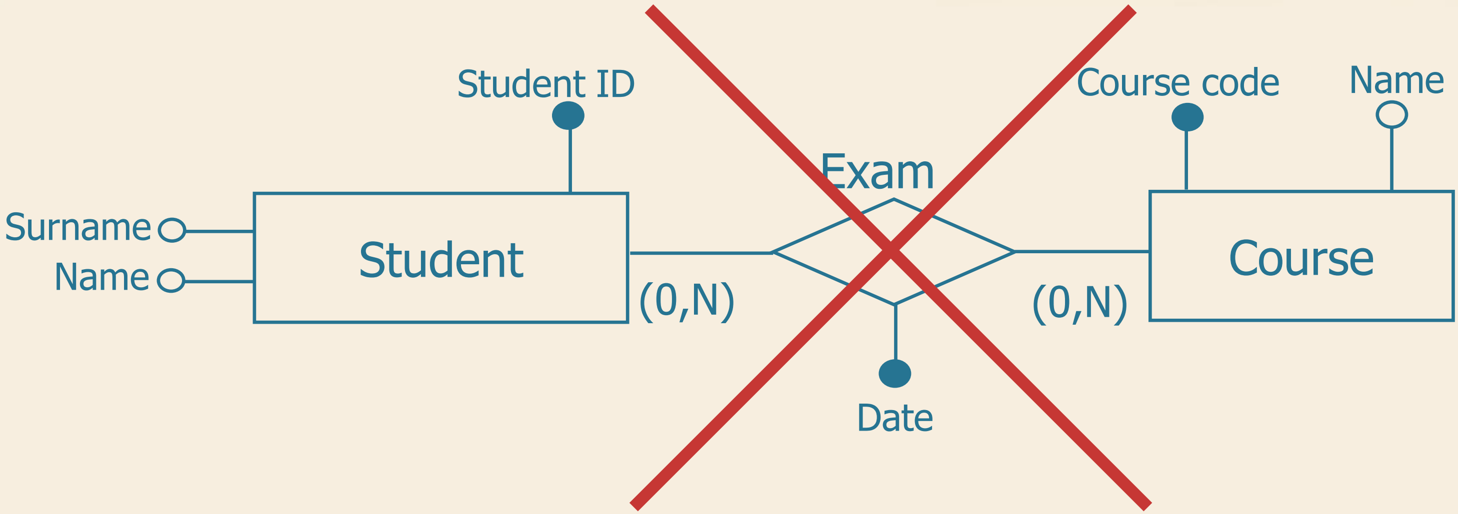
Remarks

- An external identifier may involve an entity that is itself externally identified
  - Identification cycles must not be generated



Remarks

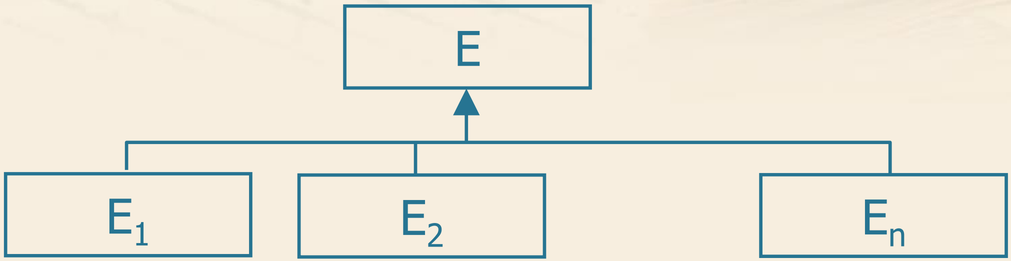
- Relationships do *not* have identifiers



The Entity-Relationship Model

Generalizations

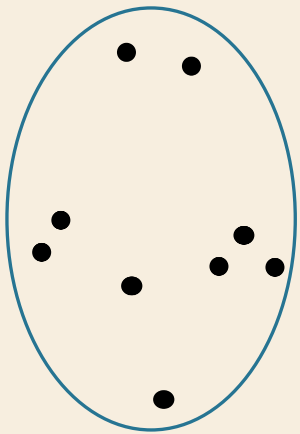
Generalization



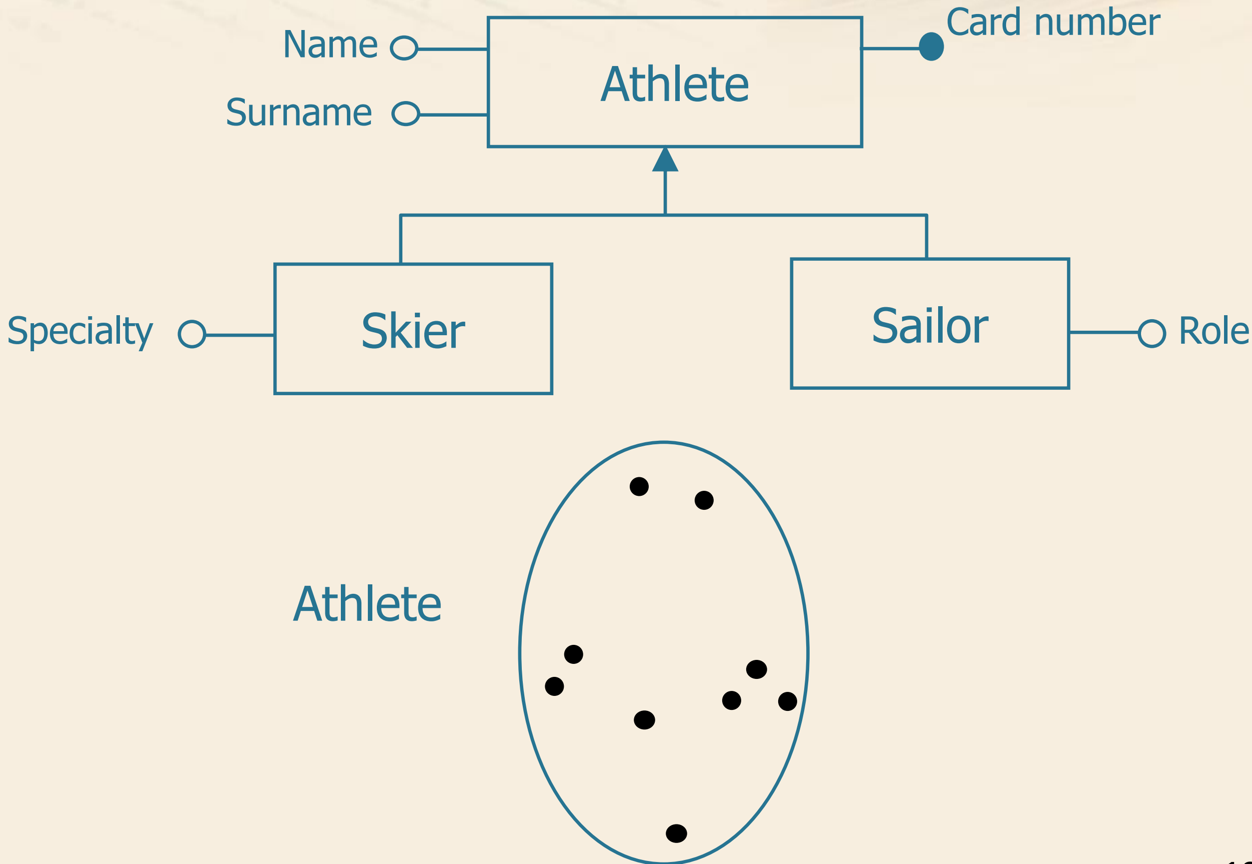
- It describes a logical link between an entity E and one or more entities E_1, E_2,…, E_n, that are particular cases of E.
  - E is called parent entity, and is a generalization of E_1, E_2,…, E_n
  - E_1, E_2,…, E_n are called child entities, and are specialization of E

Generalization: example

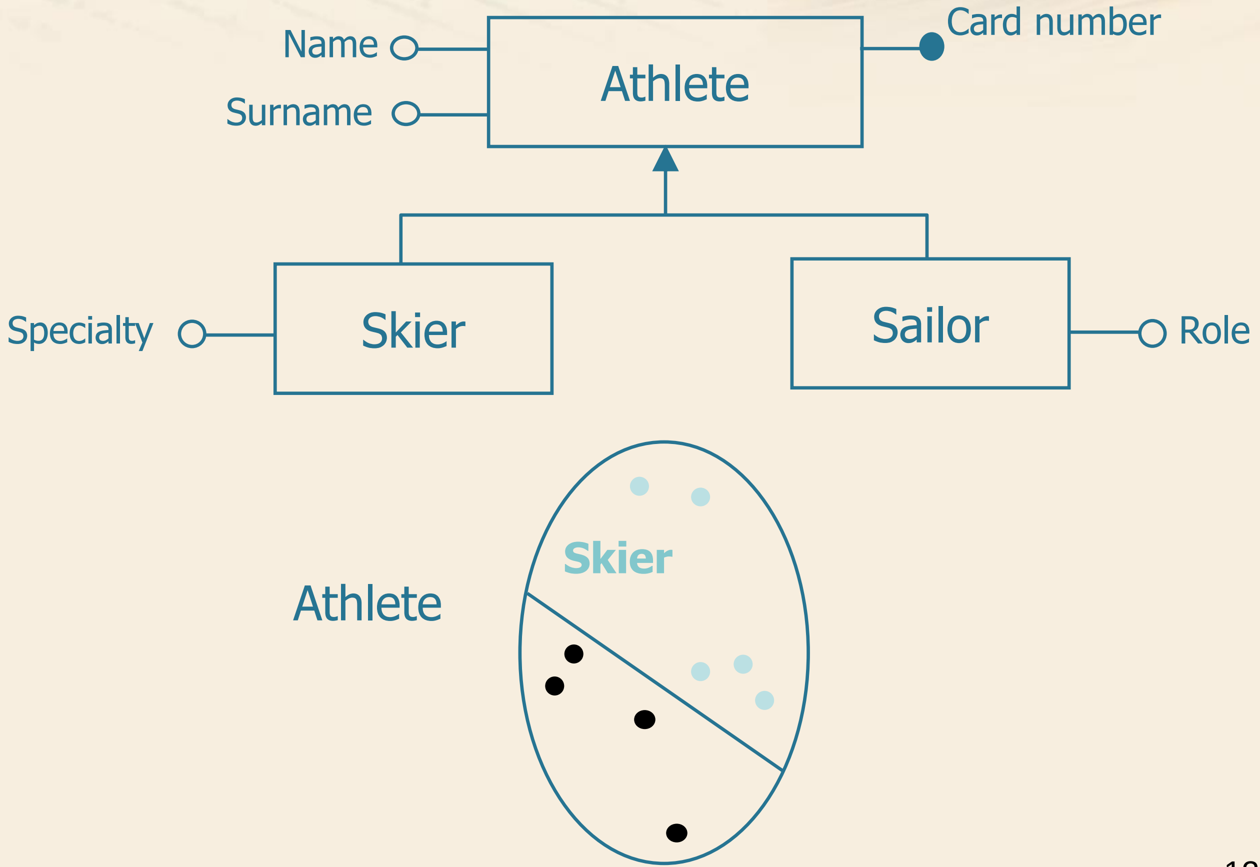
Athlete



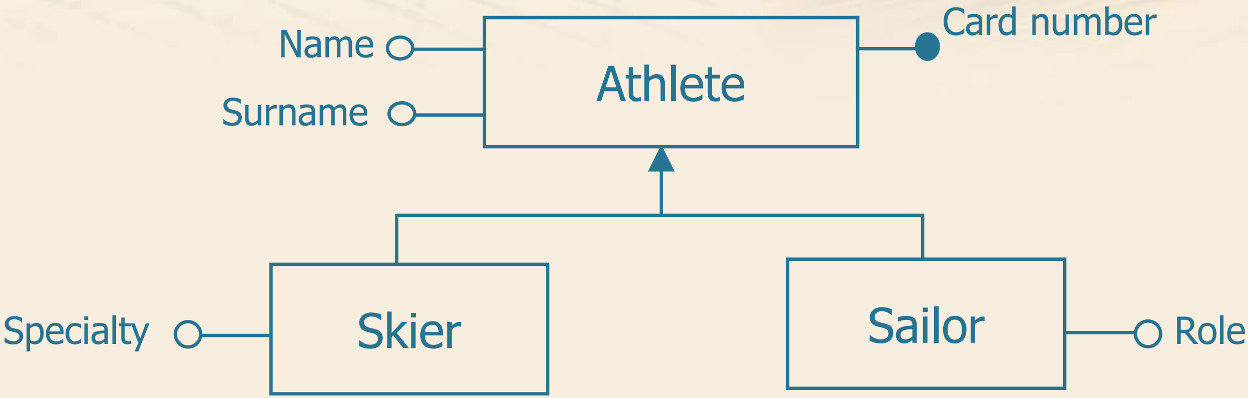
Generalization: example

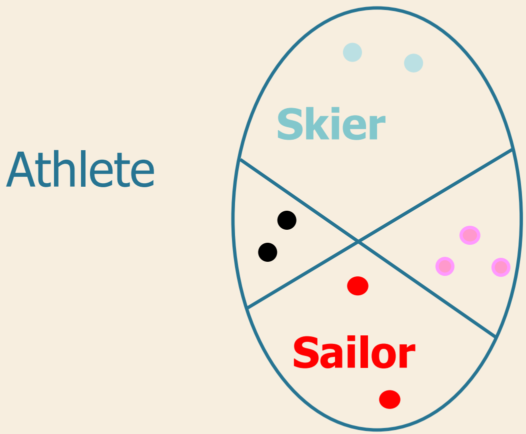


Generalization: example

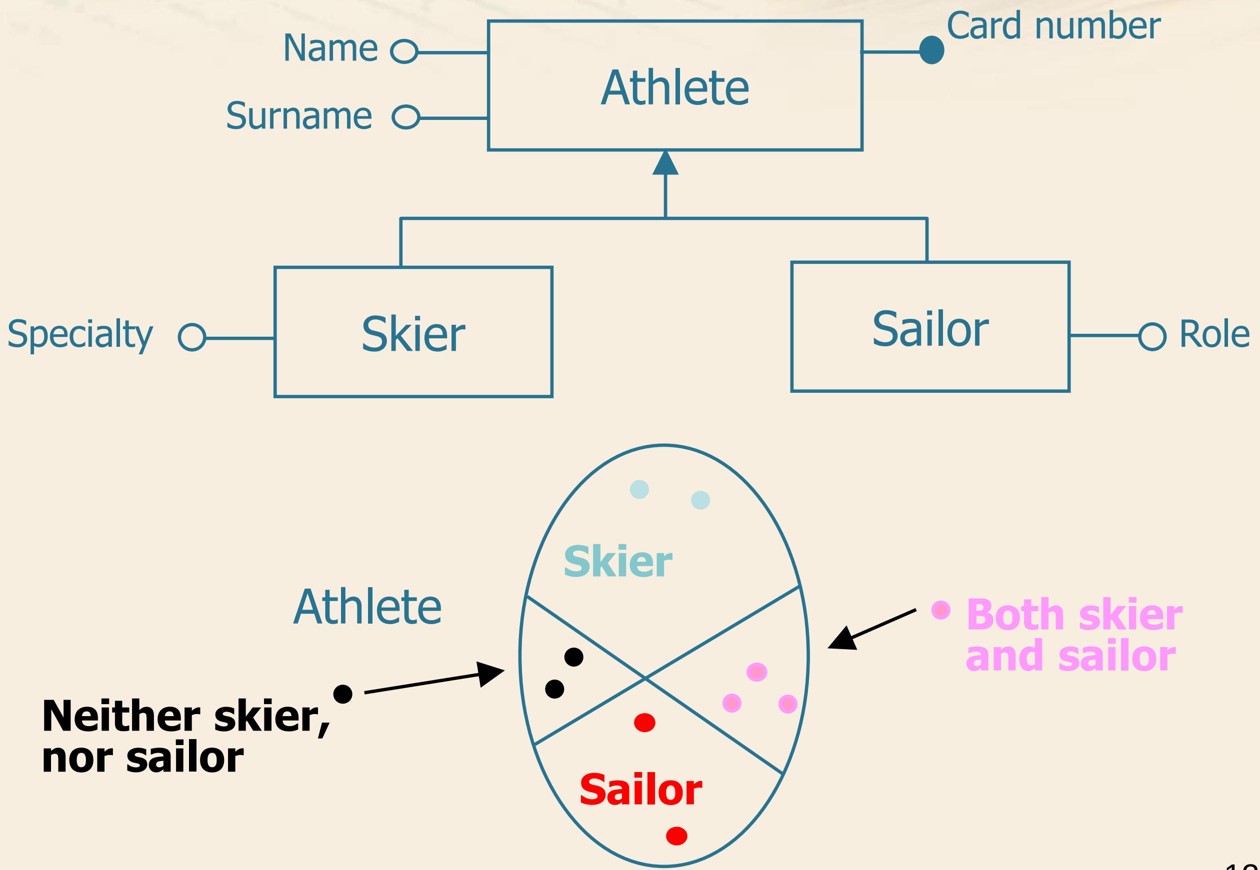


Generalization: example





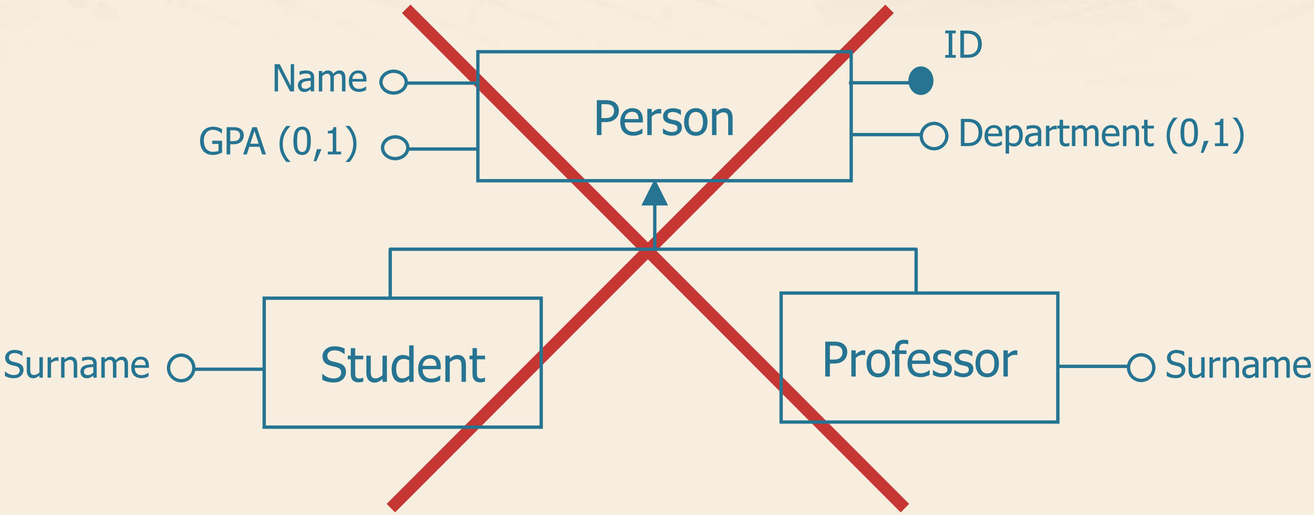
Generalization: example



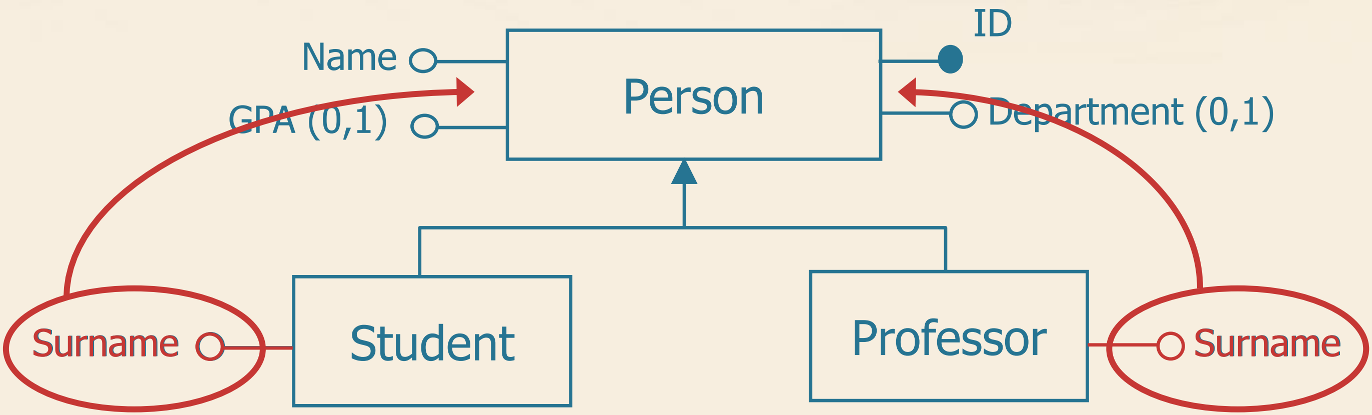
Generalization: property

- Each instance of a child entity is also an instance of the parent entity.
- Each property of the parent entity (attributes, identifiers, relationships, other generalizations) is also a property of each child entity.
  - Property known as *inheritance*
- One entity can be involved in several different generalizations.

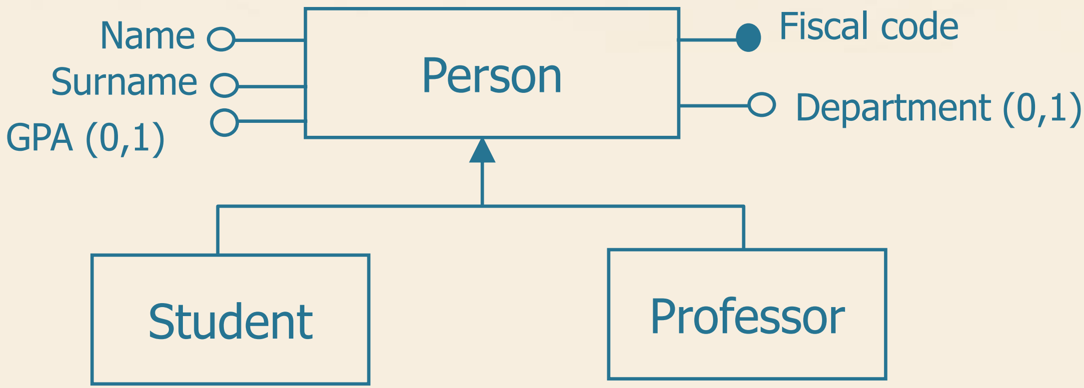
Generalization: incorrect example



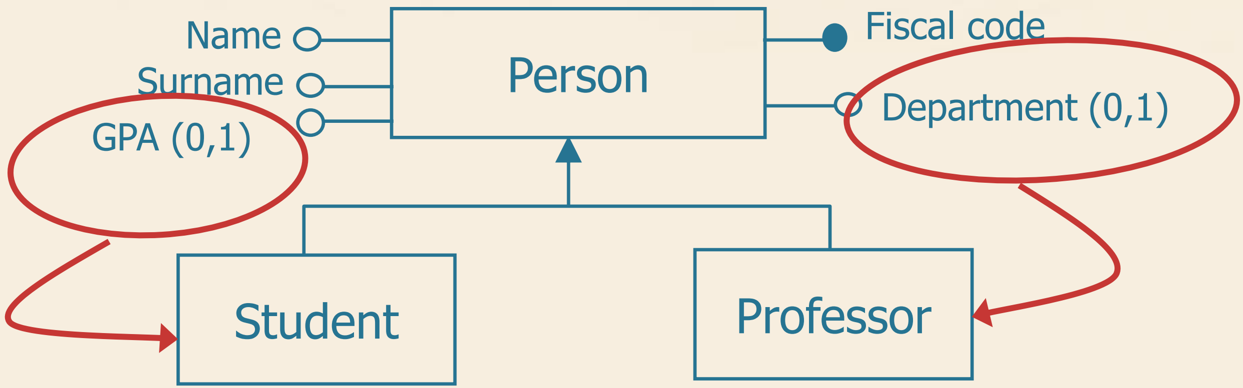
Generalization: incorrect example



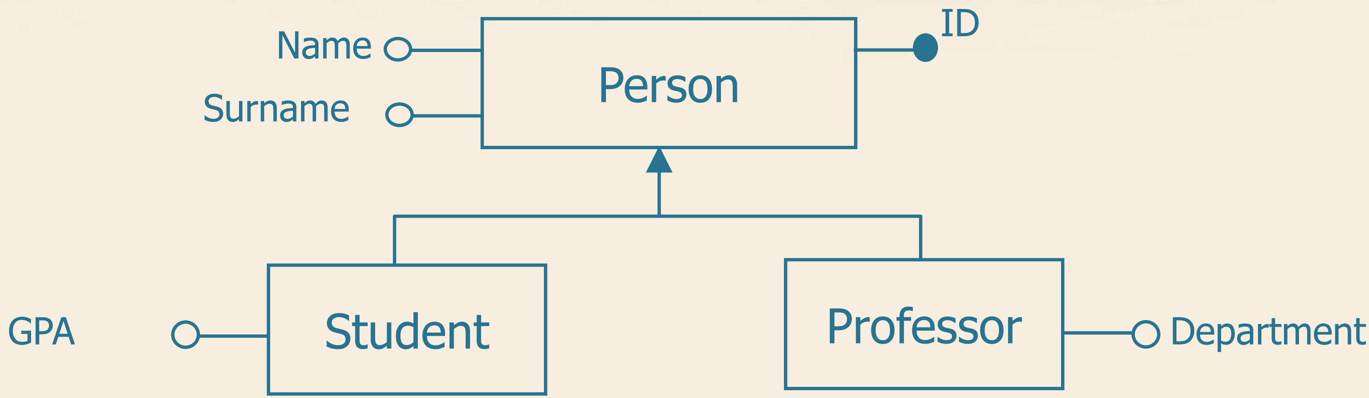
Generalization: incorrect example



Generalization: incorrect example



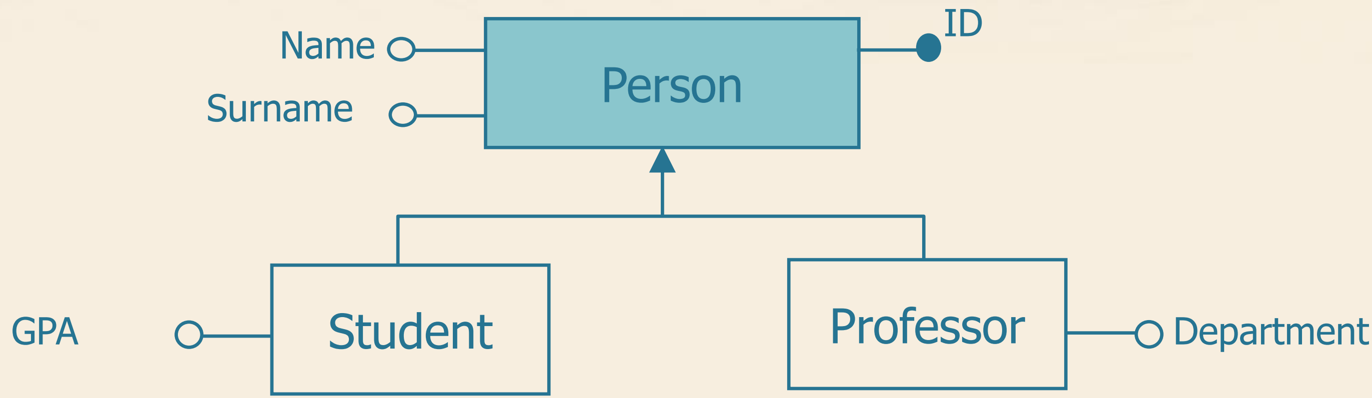
Generalization: correct example

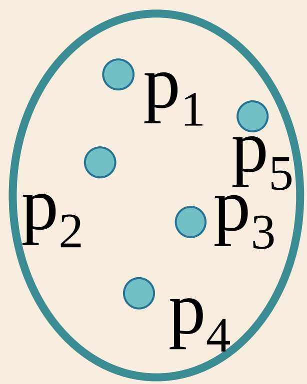


Generalization: property

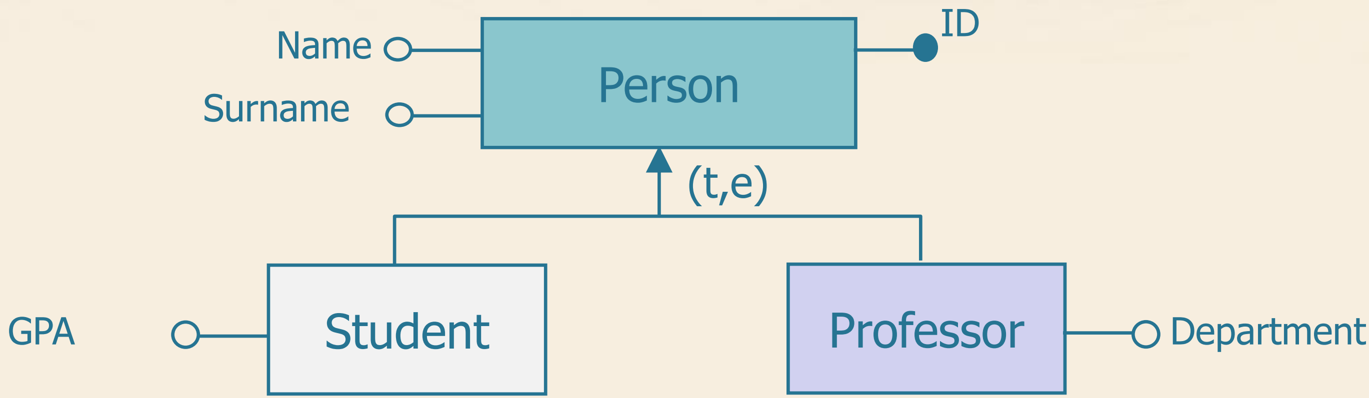
- Orthogonal characteristics
  - *total* generalization if each instance of the parent entity is an instance of at least one of the child entities, *partial* otherwise.
  - *exclusive* if each instance of the parent entity is at most one instance of one of the child entities, *overlapping* otherwise.

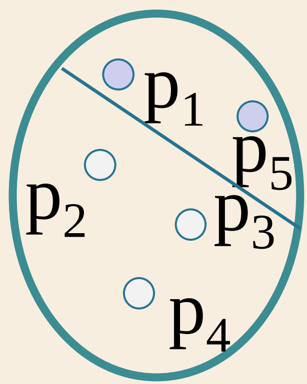
Generalization: example



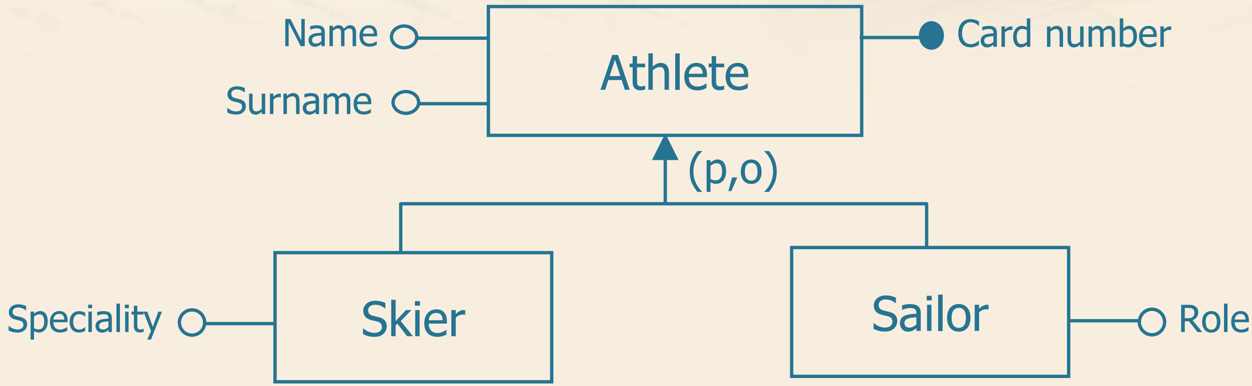


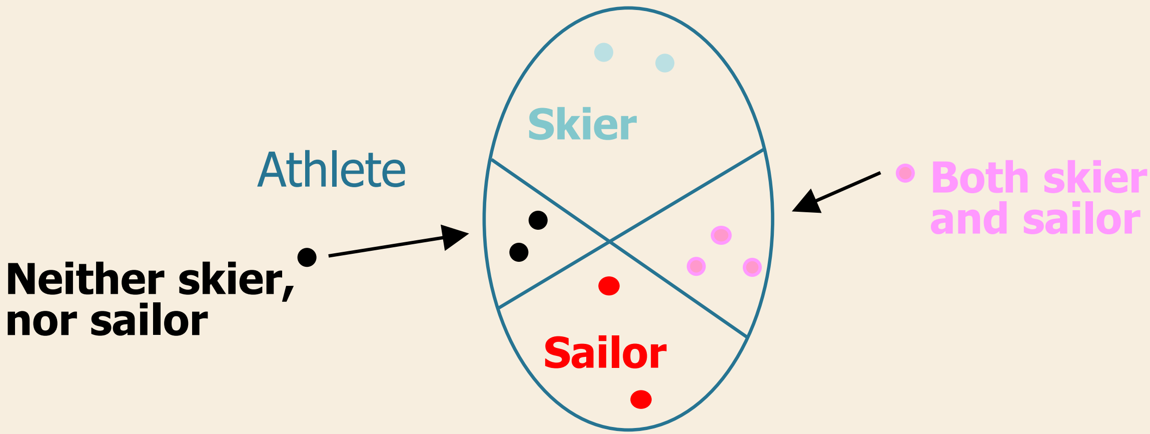
Generalization: example



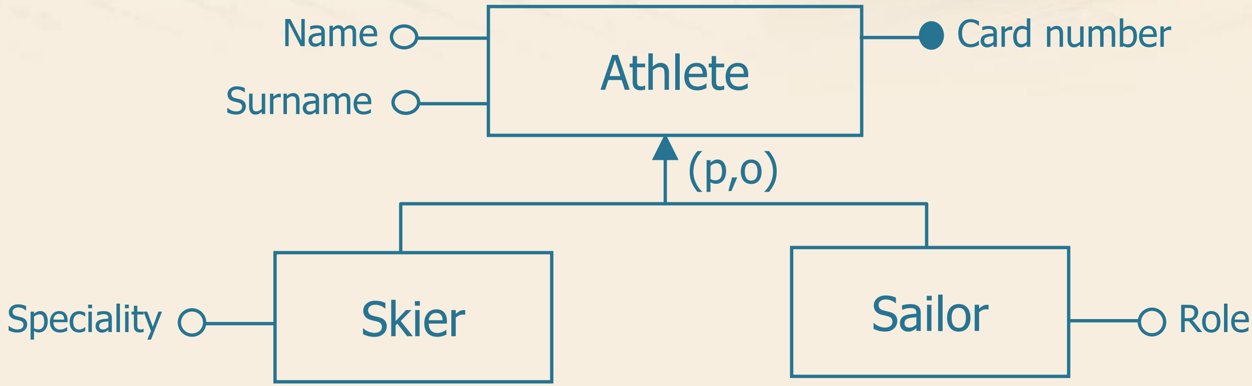


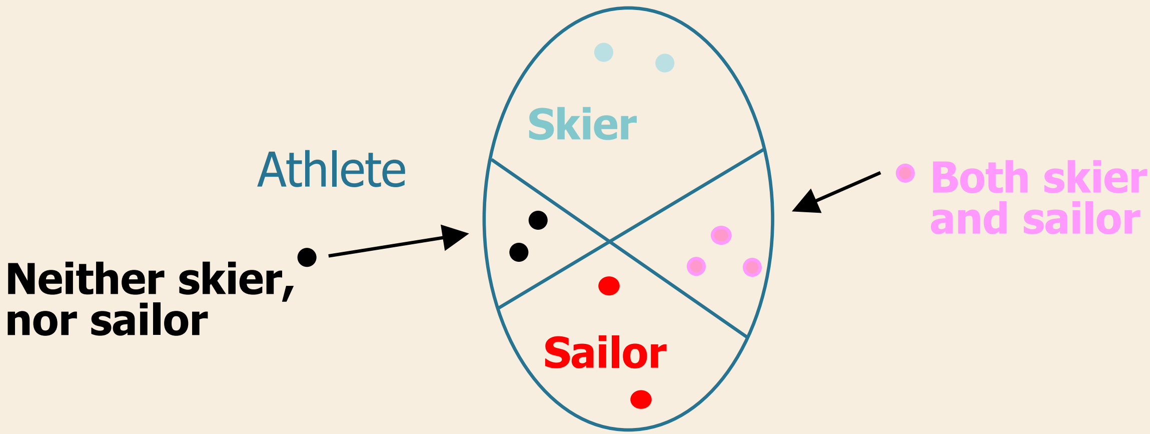
Generalization: example





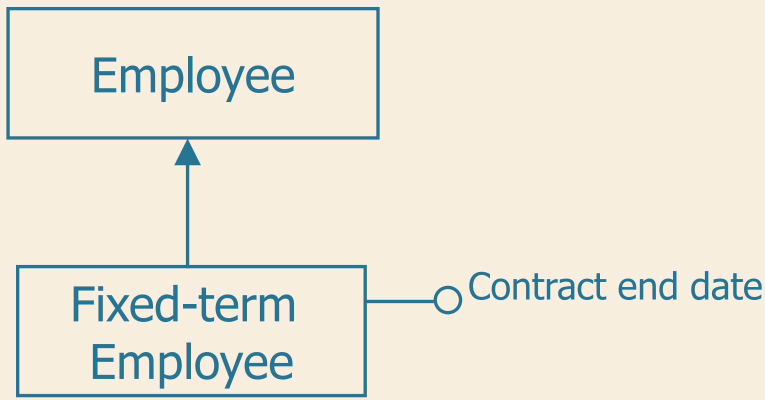
Generalization: example





Subset

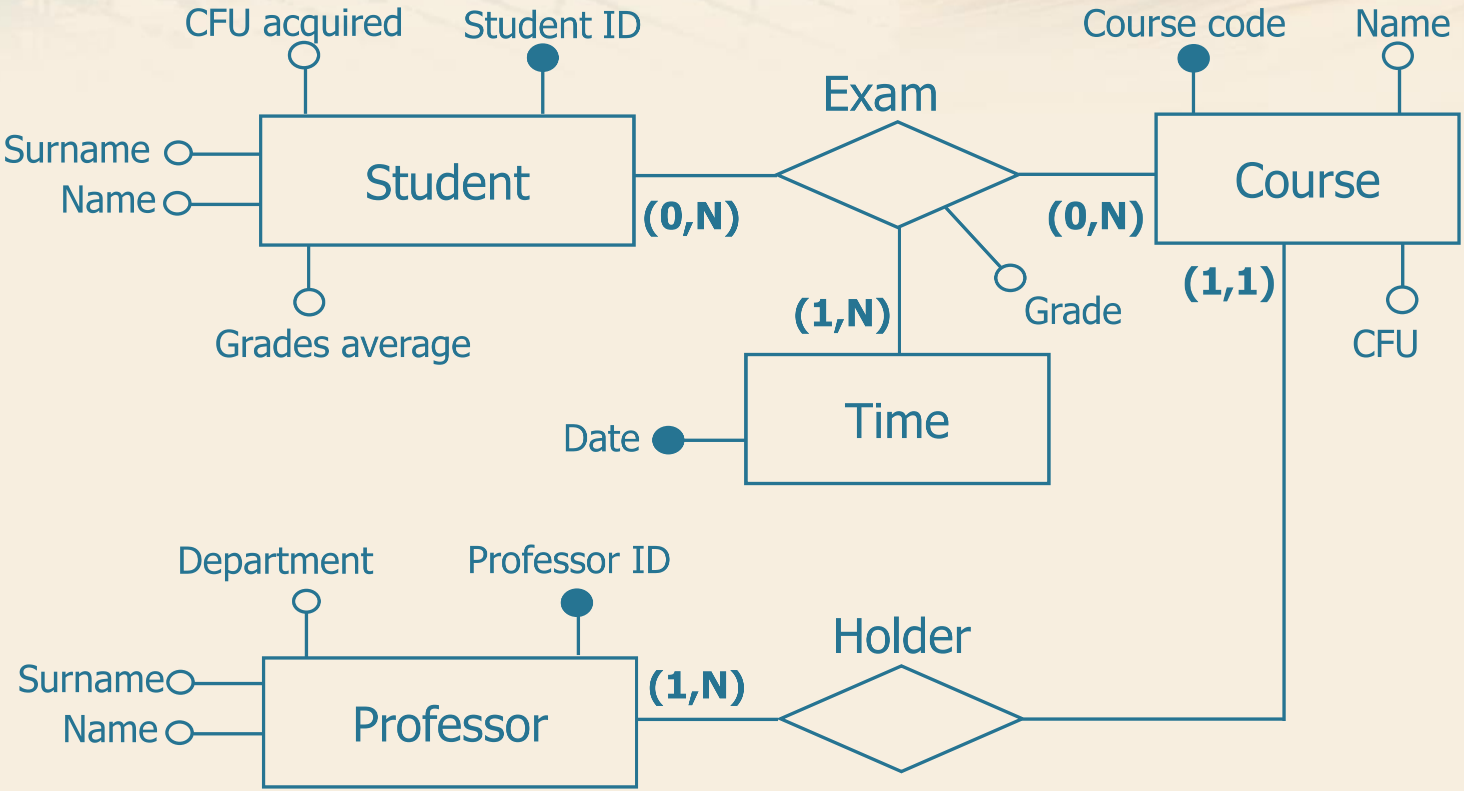
- Particular case of generalization with only one child entity
  - the generalization is always partial and exclusive.



The Entity-Relationship Model

Documentation of E-R schemes

Documentation of E-R schemes



Documentation of E-R schemes

- Data Dictionary
  - allows to enrich the E-R scheme with natural language description of entities, relationships and attributes

Data dictionary: example

Entity	Description	Attributes	Identifier
Student	University student	Student ID, Surname, Name, CFU acquired, Grades average	Student ID
Professor	University professor	Professor ID, Department, Surname, Name	Professor ID
Course	Courses offered by the university	Course code, Name, CFU	Course code
Time	Dates on which exams were taken	Date	Date

Data dictionary: example

Relationship	Description	Entities involved	Attributes
Exam	It associates a student to the exams taken and memorizes the mark obtained	Student (0,N), Course (0,N), Time (1,N)	Grade
Holder	It associates each course to the professor who teachers the course.	Course (1,1), Professor (0,N)	

Documentation of E-R schemes

- Data Dictionary
  - allows to enrich the E-R scheme with natural language description of entities, relationships and attributes
- Integrity constraints on data
  - may not always be explicitly indicated in an E-R scheme
  - can be described in natural language

Constraints of integrity on data: example

Constraints of integrity	
RV1	The grade of an exam can only take values between 0 and 30
RV2	Each student cannot pass the same exam twice
RV3	A student may not take more than three exams for the same course during the same academic year

Documentation of E-R schemes

- Data Dictionary
  - allows to enrich the E-R scheme with natural language description of entities, relationships and attributes
- Data integrity constraints
  - may not always be explicitly indicated in an E-R scheme
  - can be described in natural language
- Rules for deriving data
  - allow to define how a scheme concept can be obtained (by logical inference or arithmetic calculation) from other scheme concepts

Rules for deriving data: example

Derivation rules	
RD1	The number of credits acquired by a student is obtained by adding the number of credits of the courses for which the student has passed the exam
RD2	The average mark is obtained by calculating the average marks of the exams passed by a student

The Entity-Relationship Model

UML and E-R

UML and E-R

- UML (Unified Modeling Language)
  - is a model of a software application
    - structural and behavioural aspects (data, operations, processes and architectures)
  - rich formalism
    - Diagrams of classes, of actors, of sequence, of communication, of states,…
- E-R
  - is a model of a database
    - Structural aspects of an application
  - specific formalism for database modelling

UML and E-R

- Main charachteristics of UML that differs with respect to E-R diagrams
  - absence of standard notation to define identifiers
  - possibility to add notes to comment on diagrams
  - possibility to indicate the navigation direction of an association (not relevant in the design of a database)

UML and E-R

- Different formalisms
- The class diagram of an application is different from the E-R scheme of the database
- The class diagram, even if designed for a different use, may be adapted for the description of the conceptual design of a database