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

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

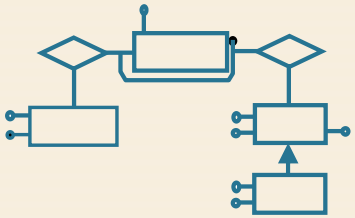
- Informal specifications of the reality of interest
  - application properties
  - application functionalities

Stages of database design

Application requirements

Conceptual design

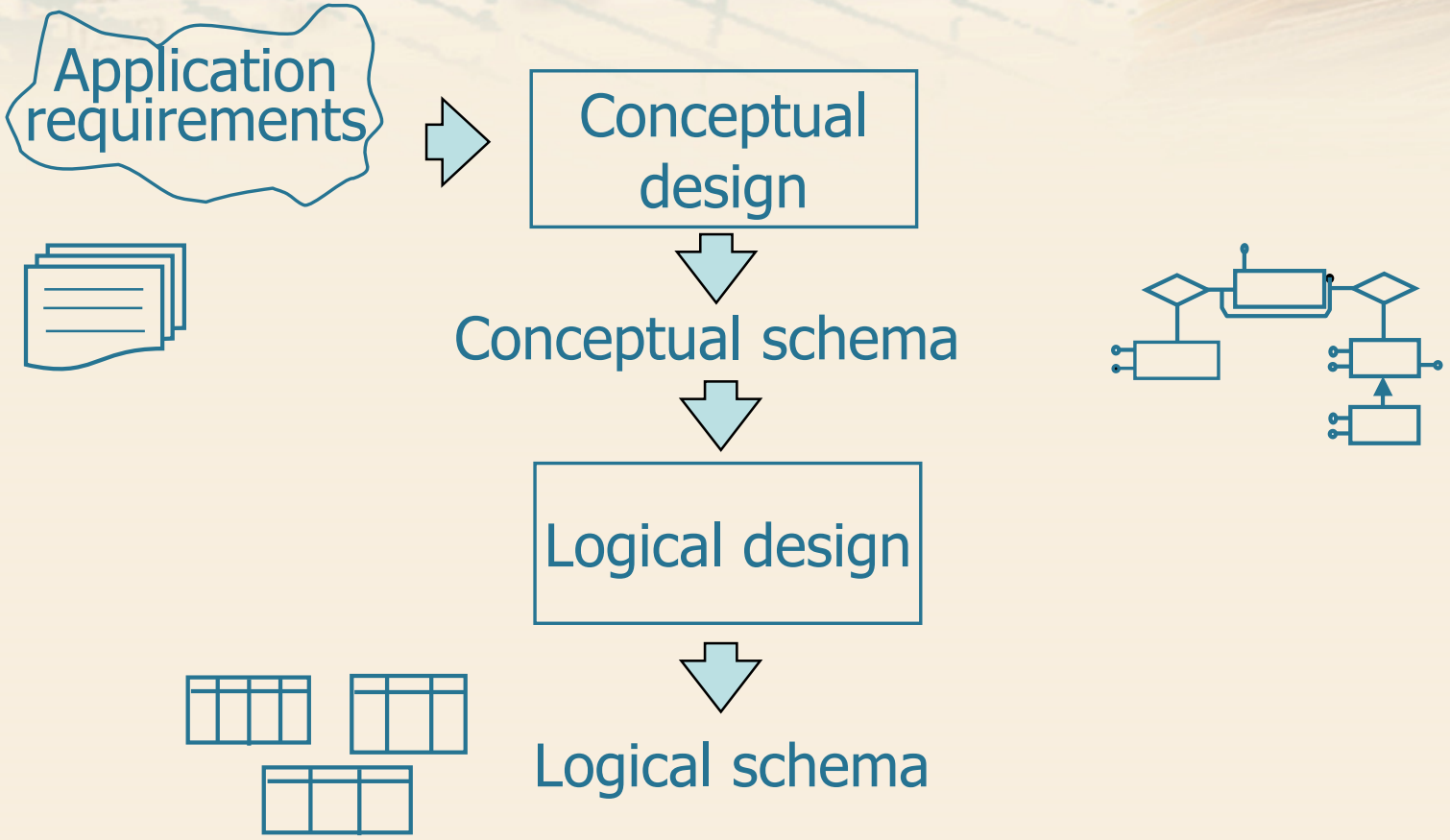
Conceptual schema



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

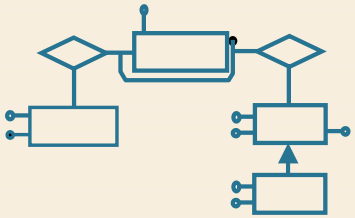
- Translation of the conceptual schema into the logical schema
  - refers to the chosen logical data model
  - criteria are used to optimize the operations which must be performed on the data
  - quality of the schema verified by formal techniques (normalization)

Stages of database design

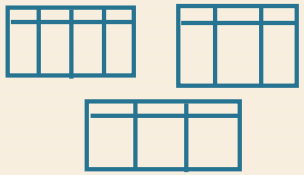
Application requirements

Conceptual design

Conceptual schema



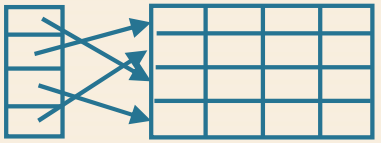
Logical design



Logical schema

Physical design

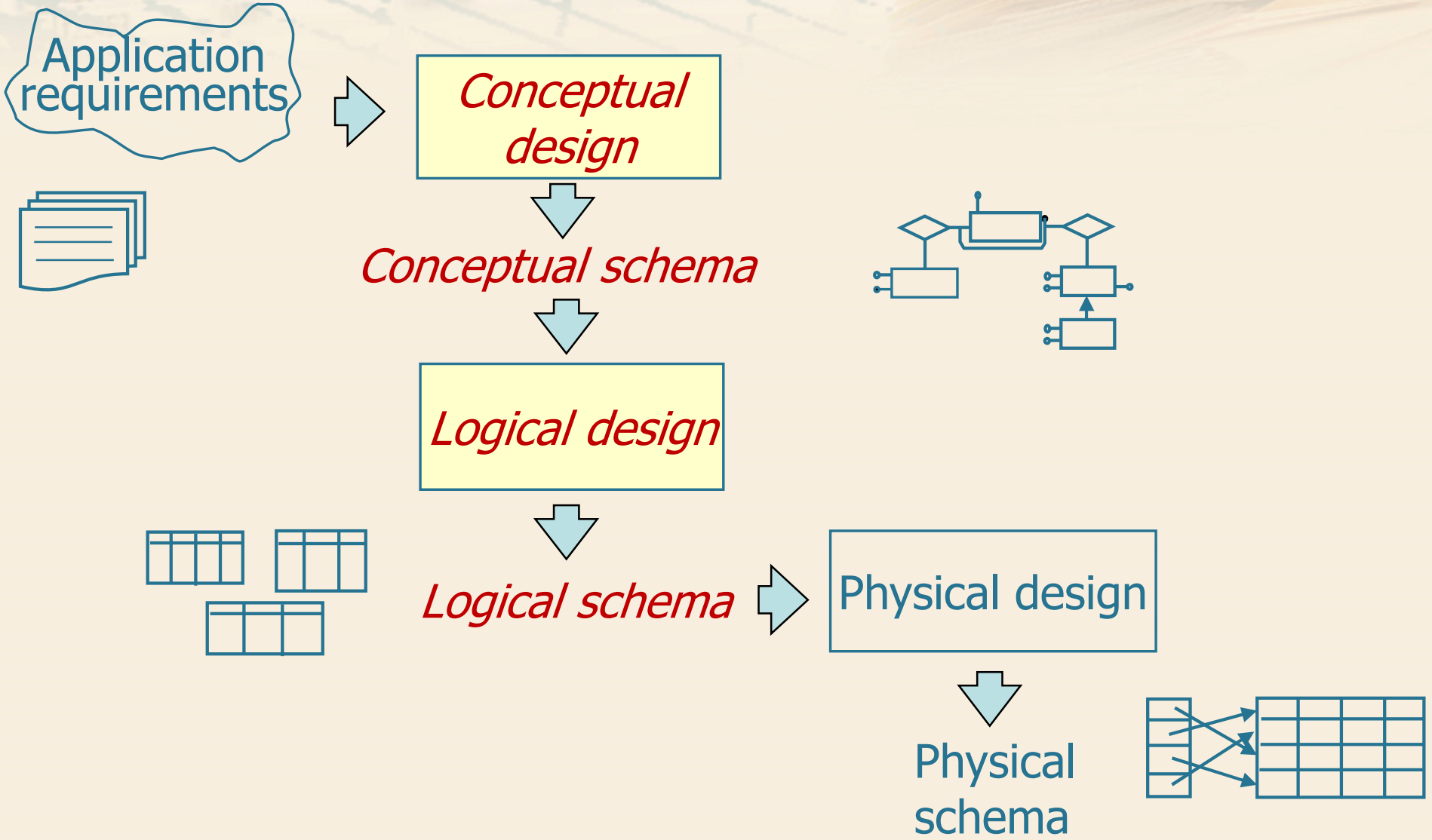
Physical schema



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



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

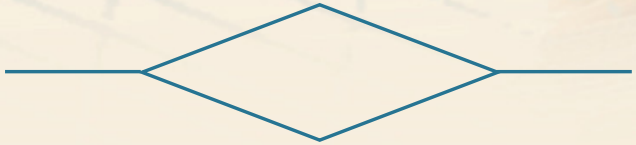
Main constructs of the E-R model

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

Entities

- 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

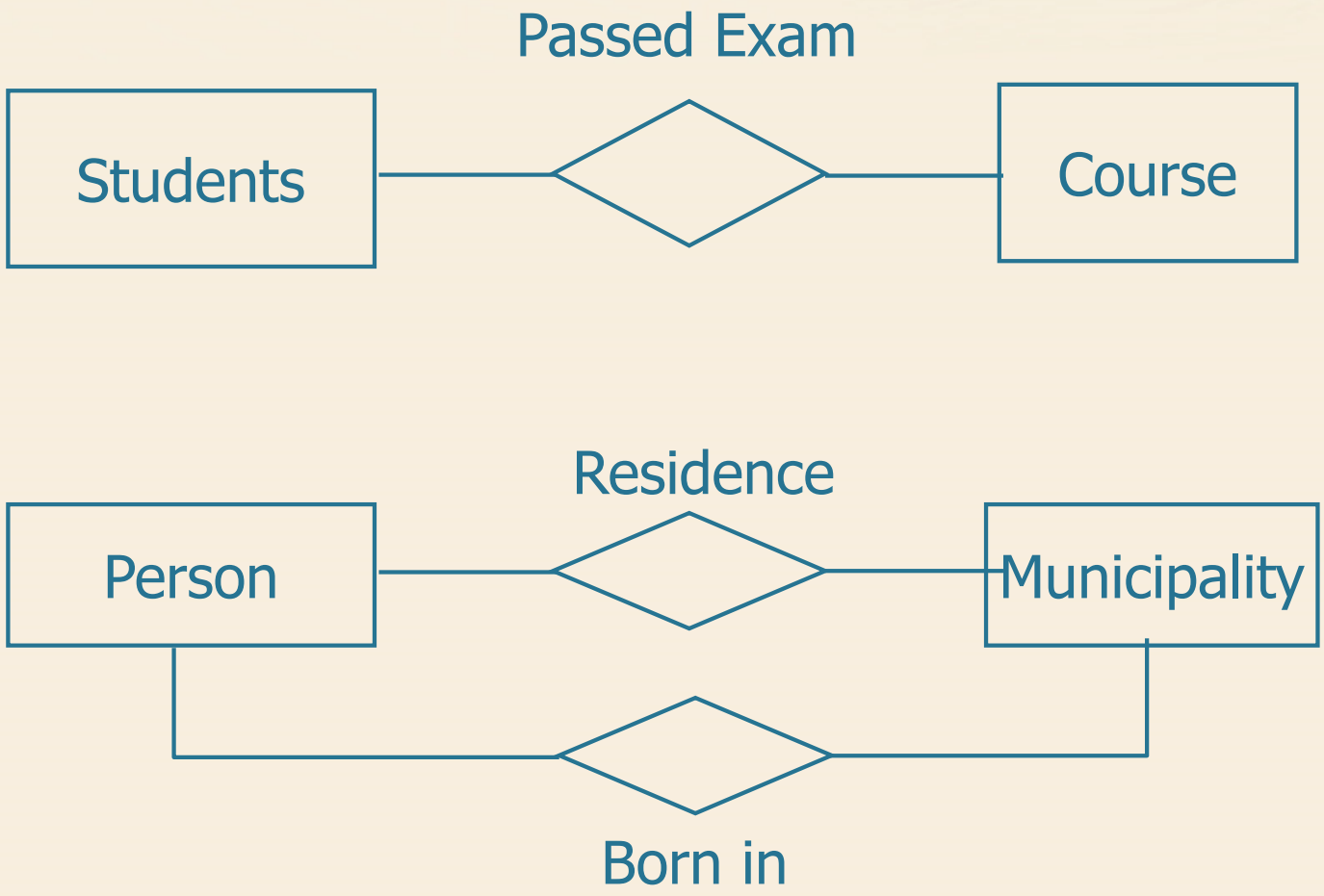
Relationship



Relationship name

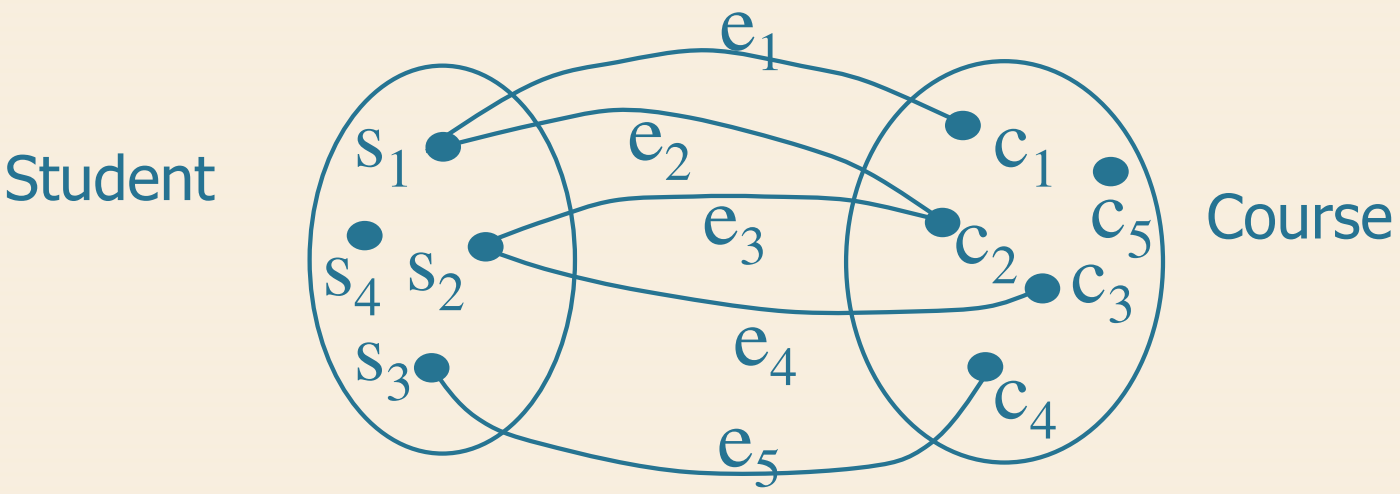
- Represents a logical link between two or more entities
- Examples: exam between student and course, residence between person and municipality
- Not to be confused with the relationship of the relational model
  - sometimes it is named association

Relationships examples



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

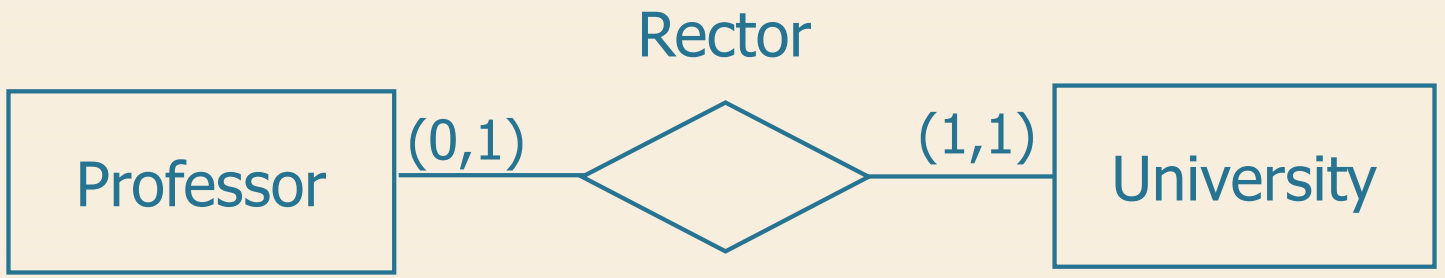


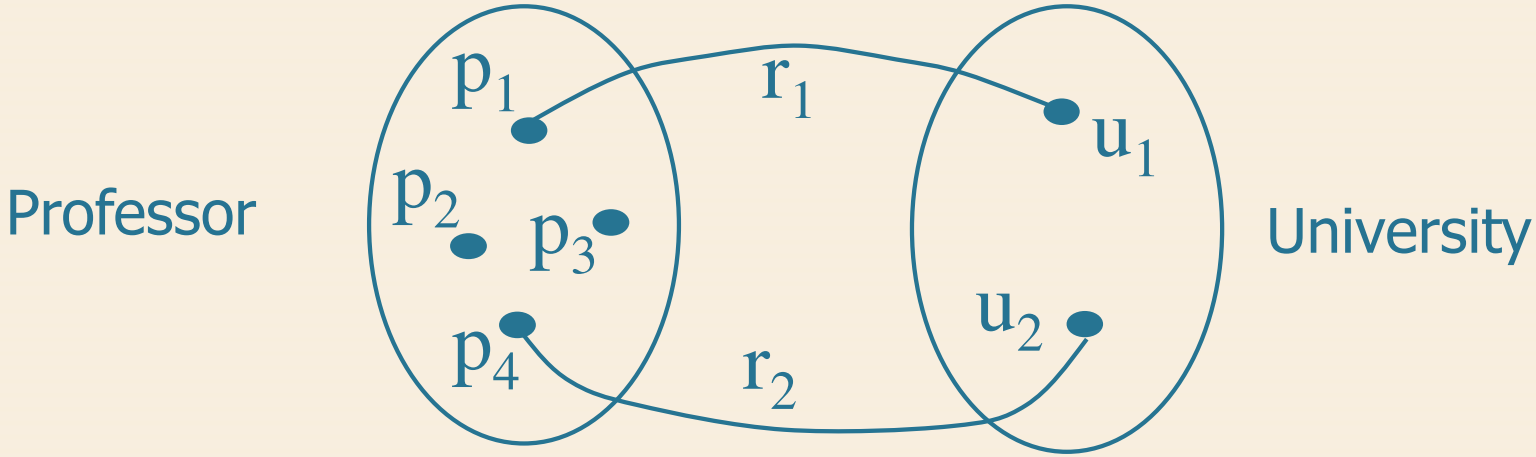
Cardinality of binary relationships

- 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

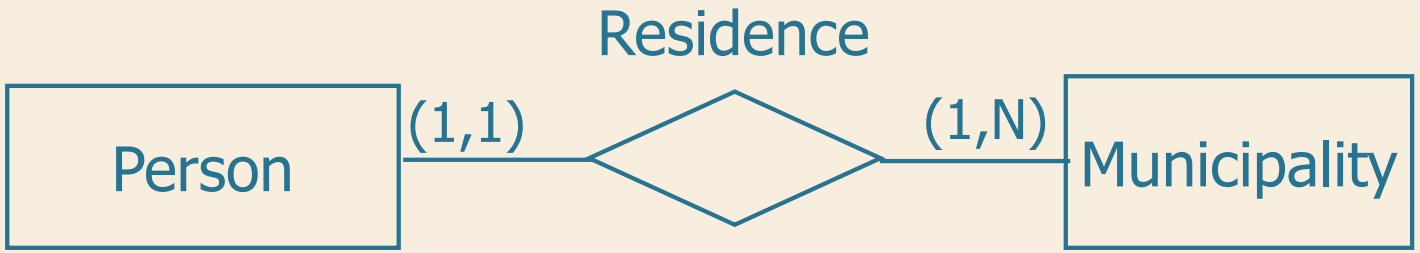
- 1 to 1 correspondence

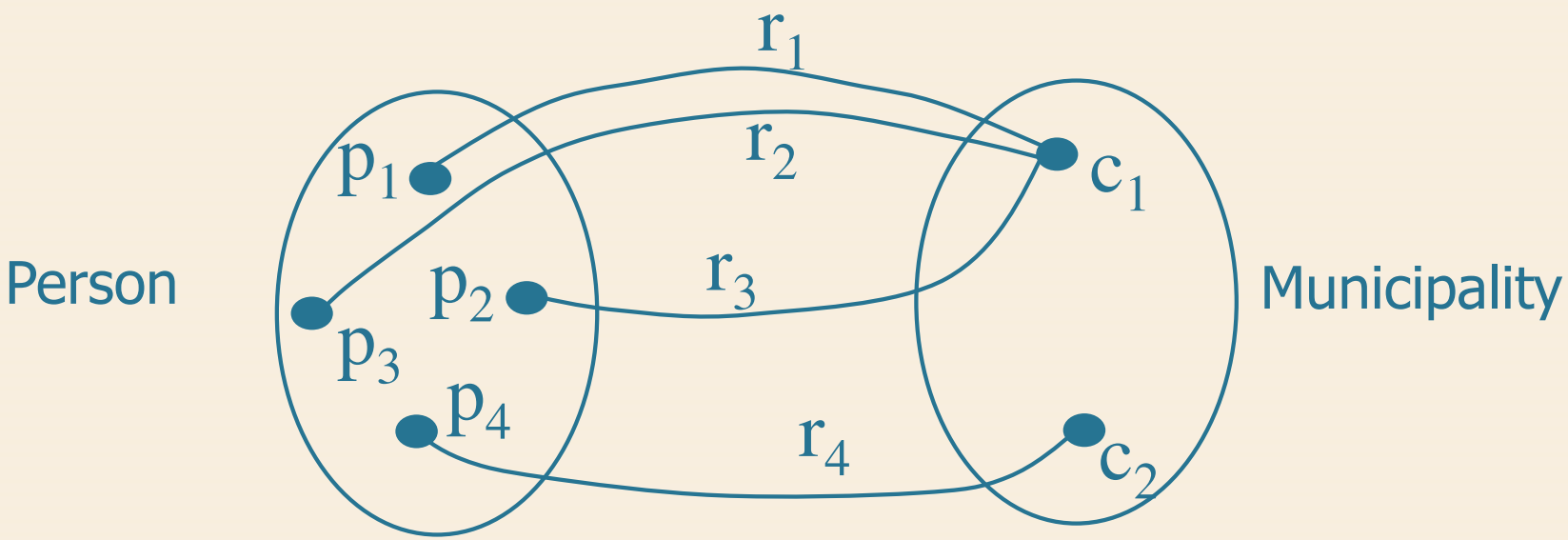




Cardinality of binary relationships

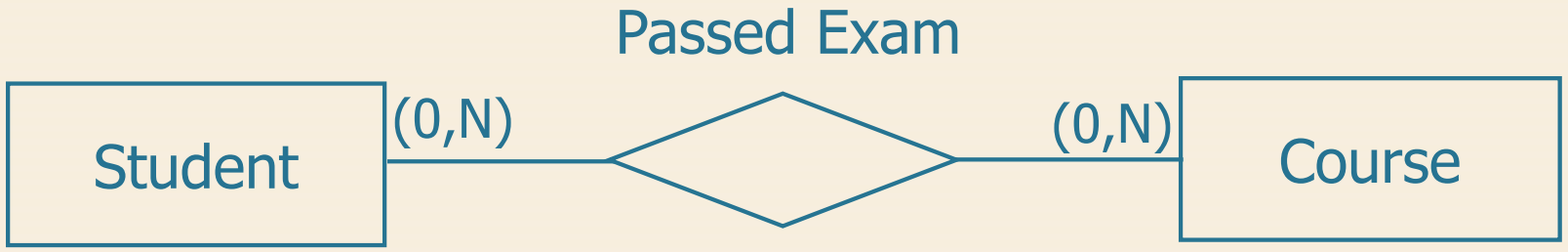
- 1 to N correspondence

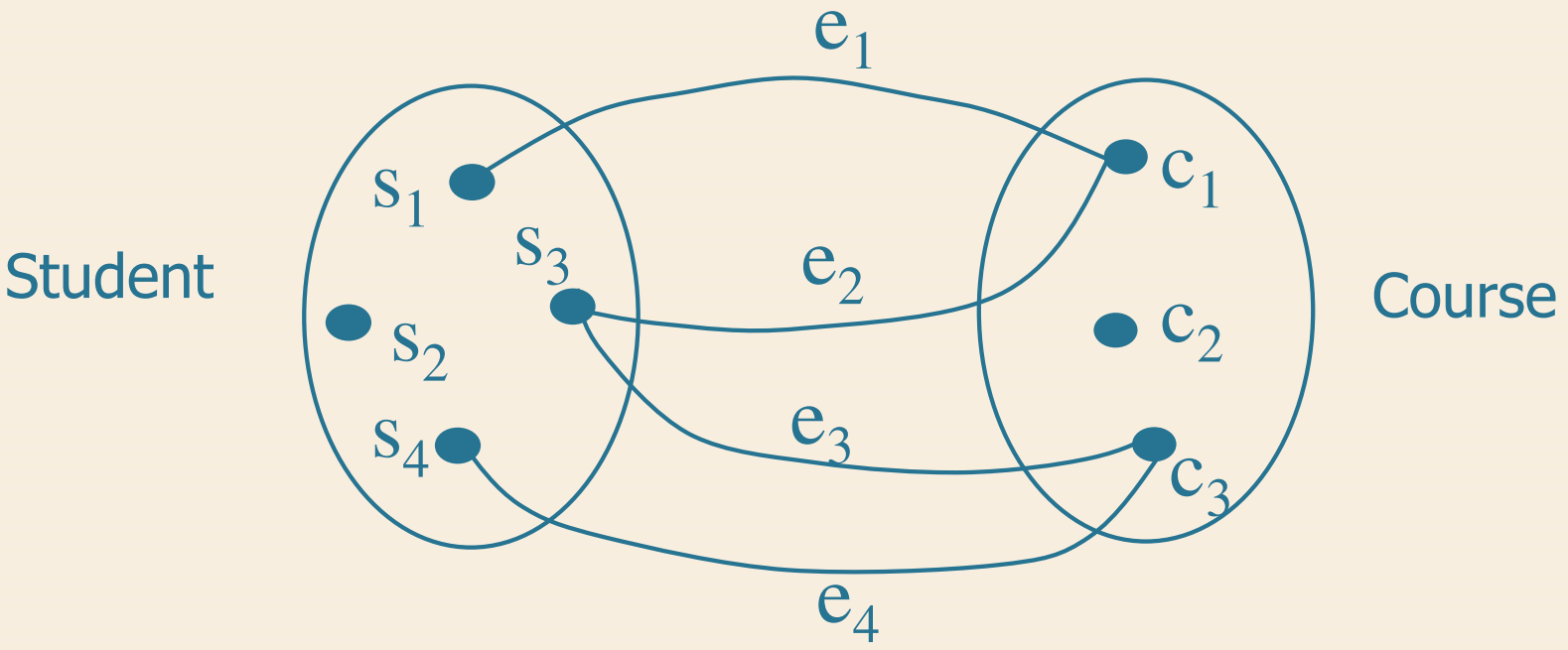




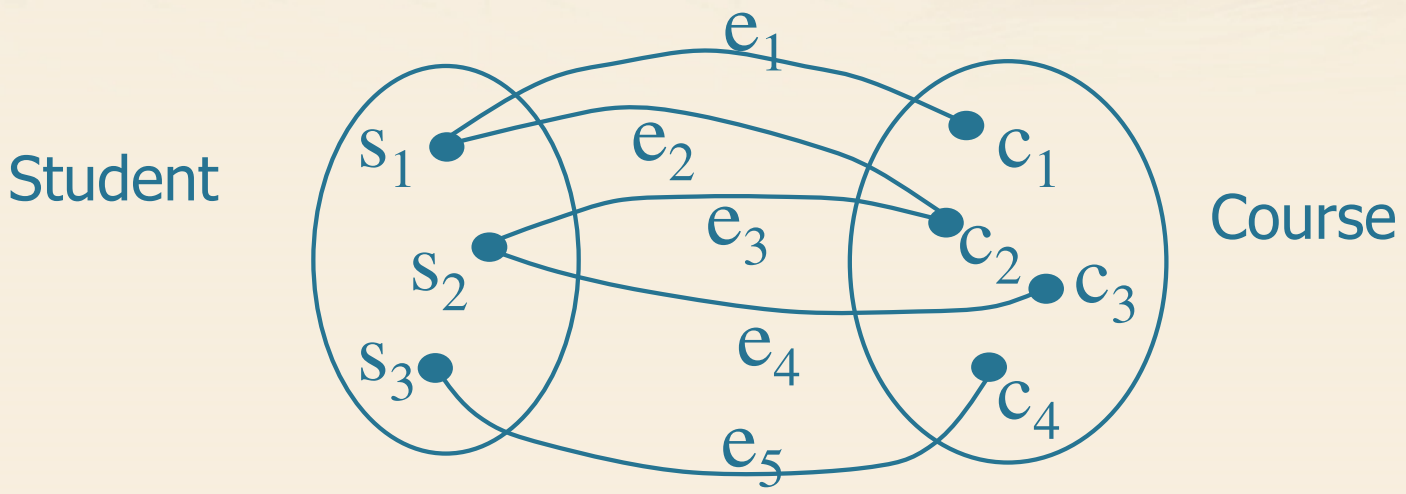
Cardinality of binary relationships

- N to N correspondence





Limitations of a binary relationship

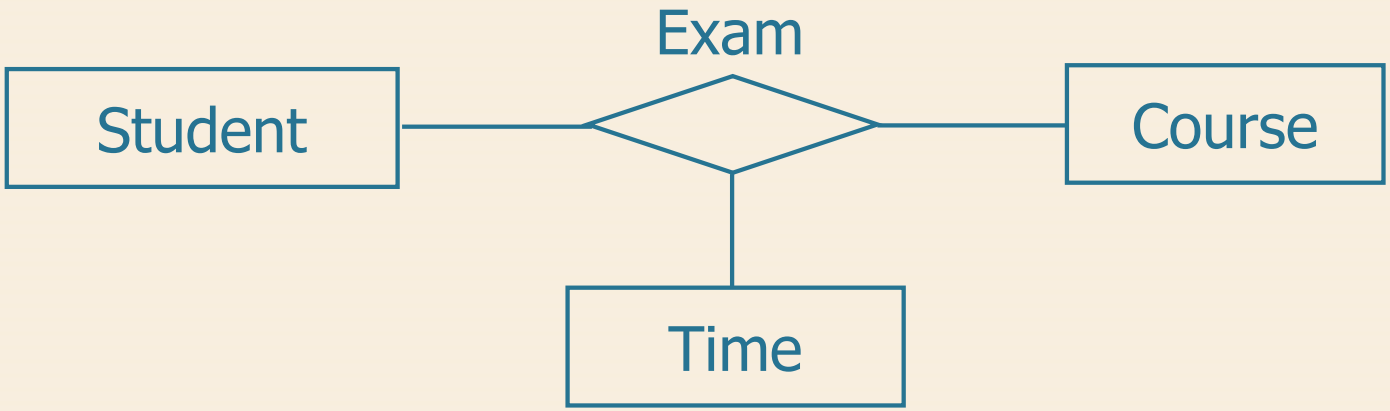


- It is not possible that a student takes the same exam more than once

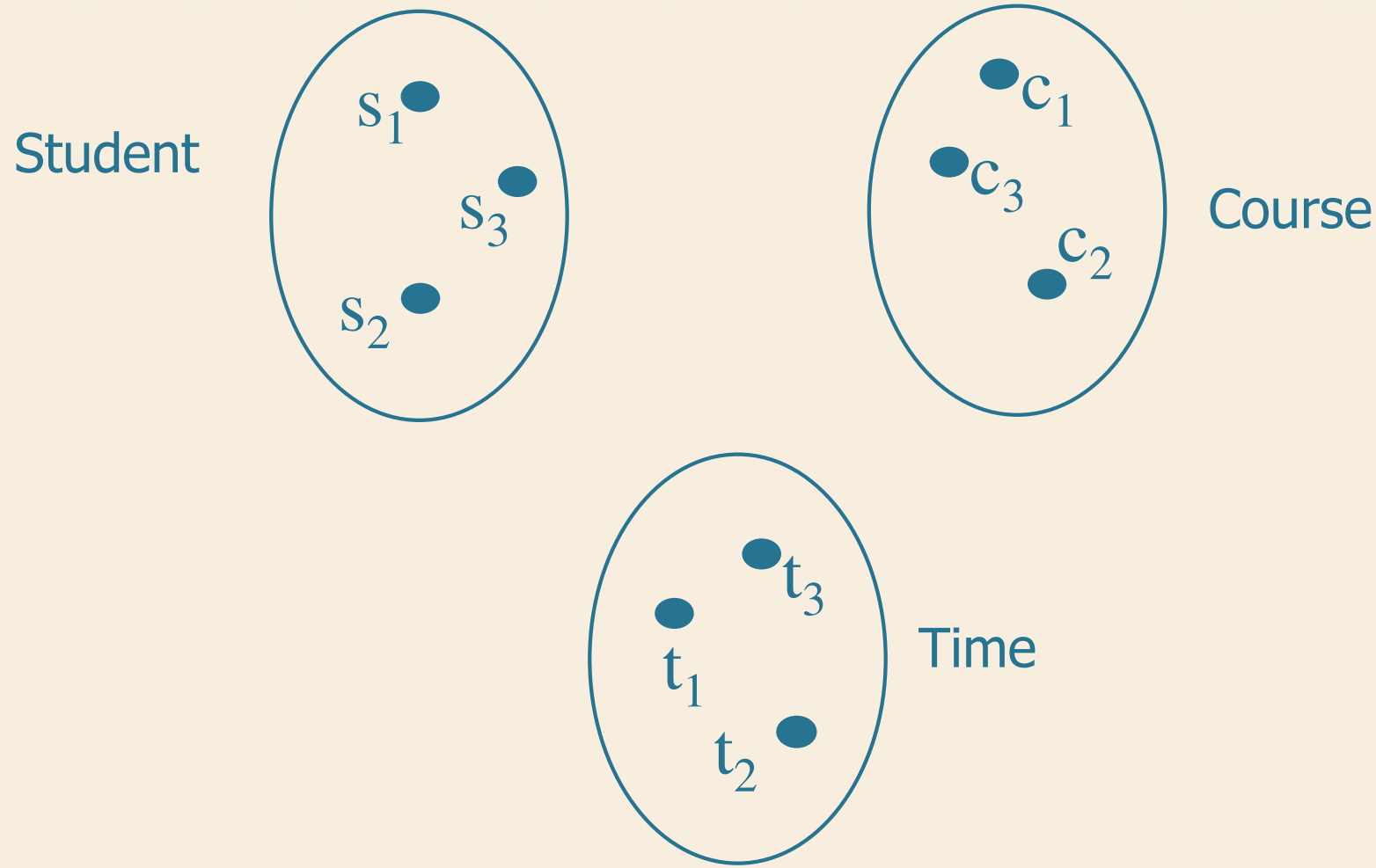
Ternary relationship

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

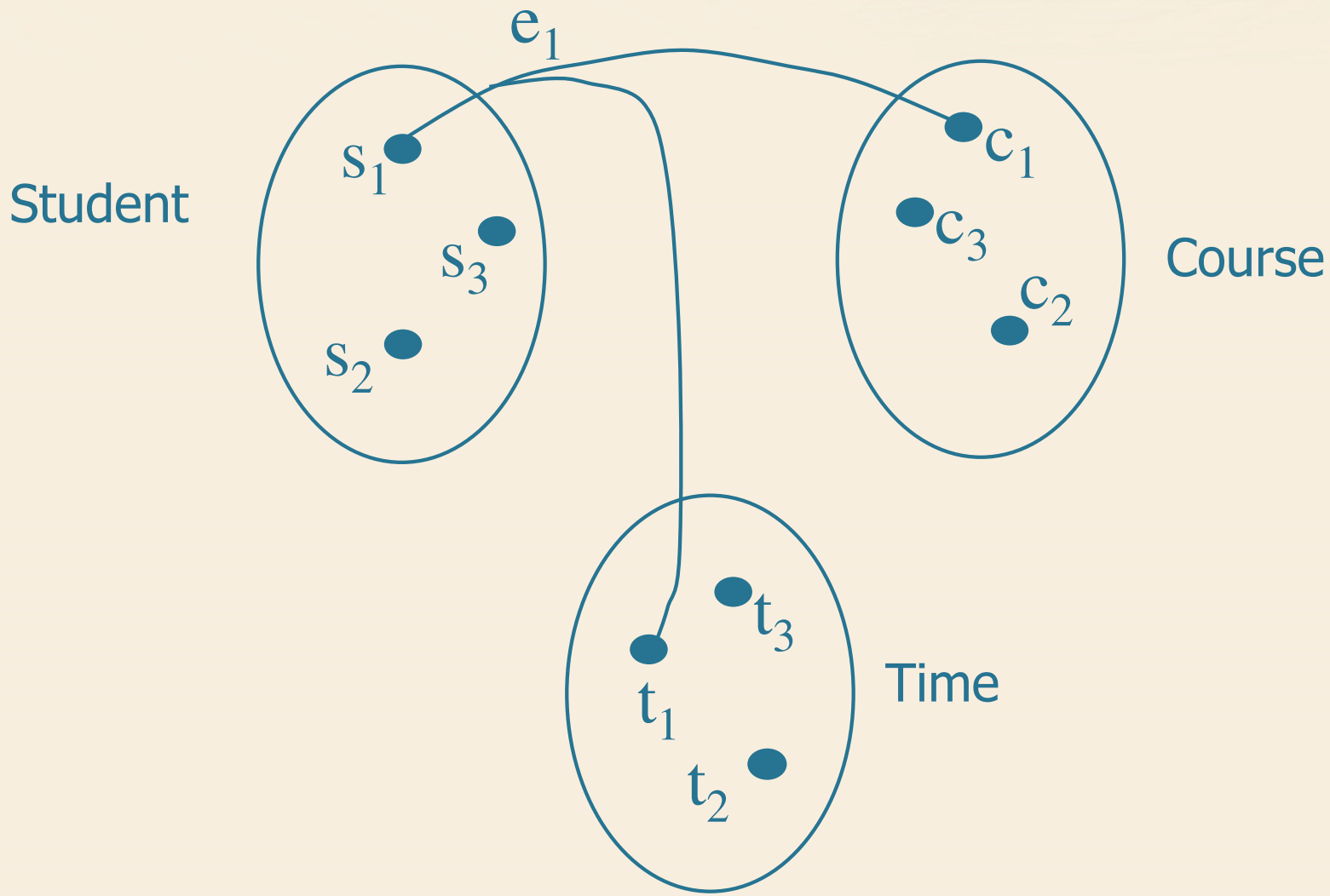
s_1 c_1 t_1
s_1 c_1 t_2
...



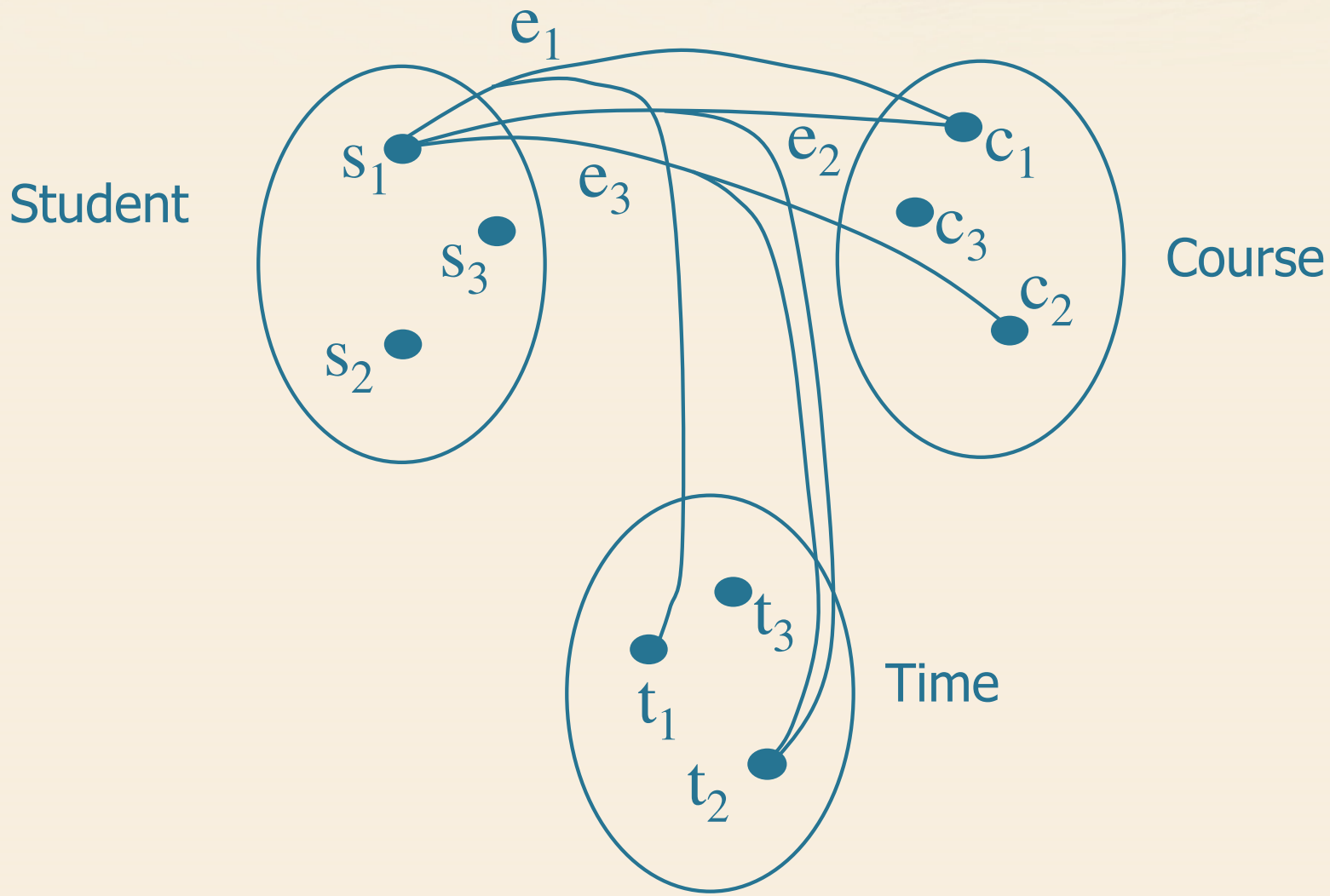
Occurences of a ternary relationship



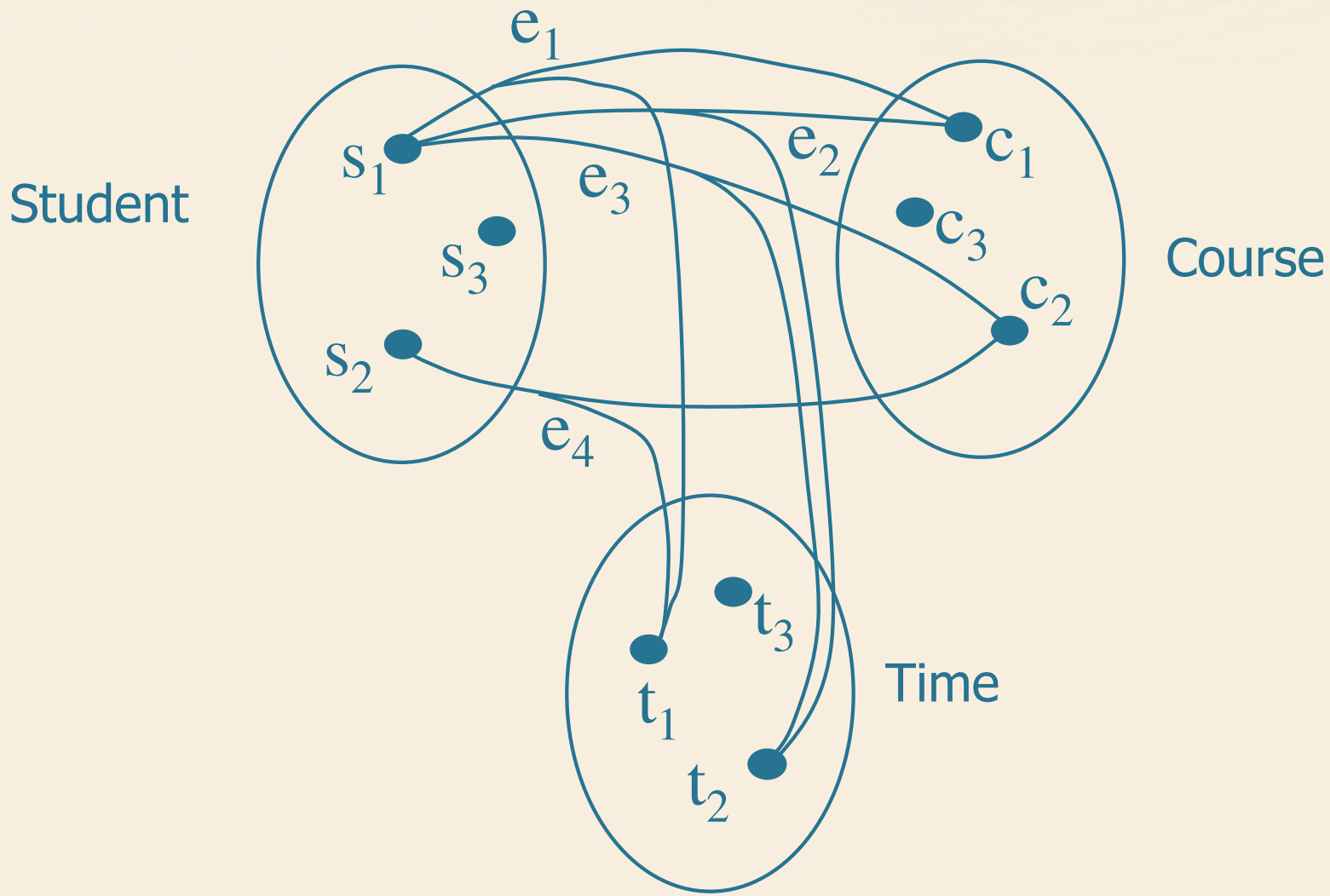
Occurences of a ternary relationship



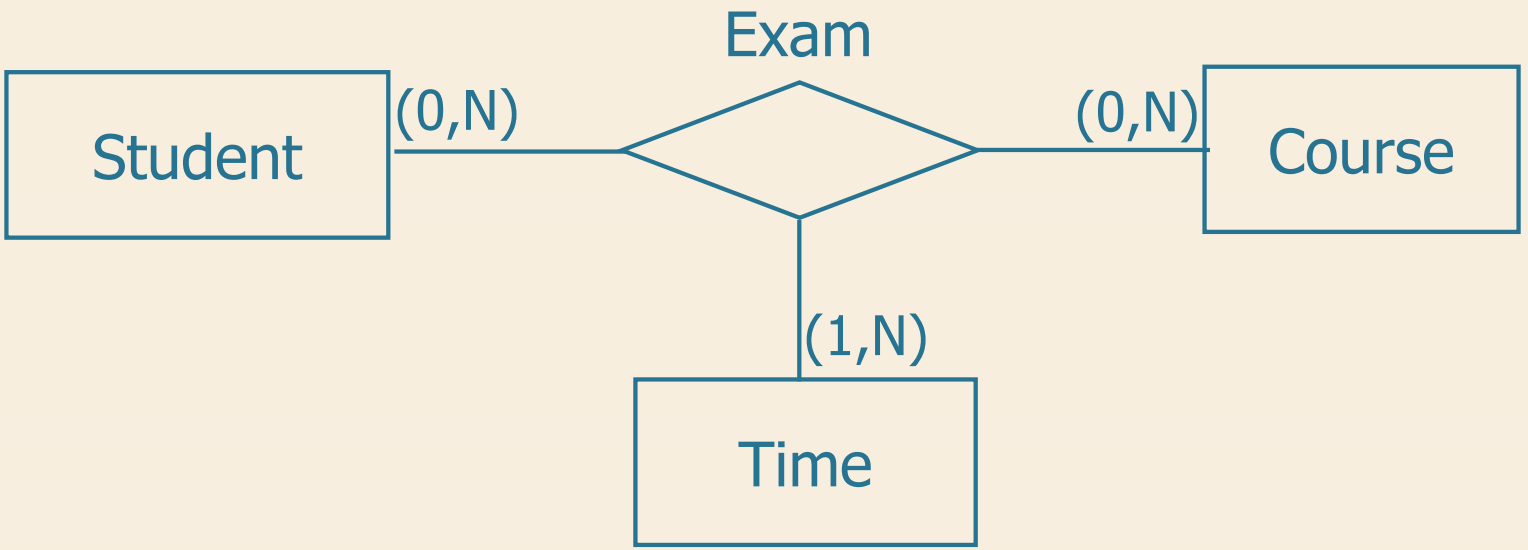
Occurences of a ternary relationship



Occurences of a ternary relationship



Cardinality of ternary relationships



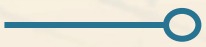
Observations

- 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

The Entity-Relationship Model

Attributes

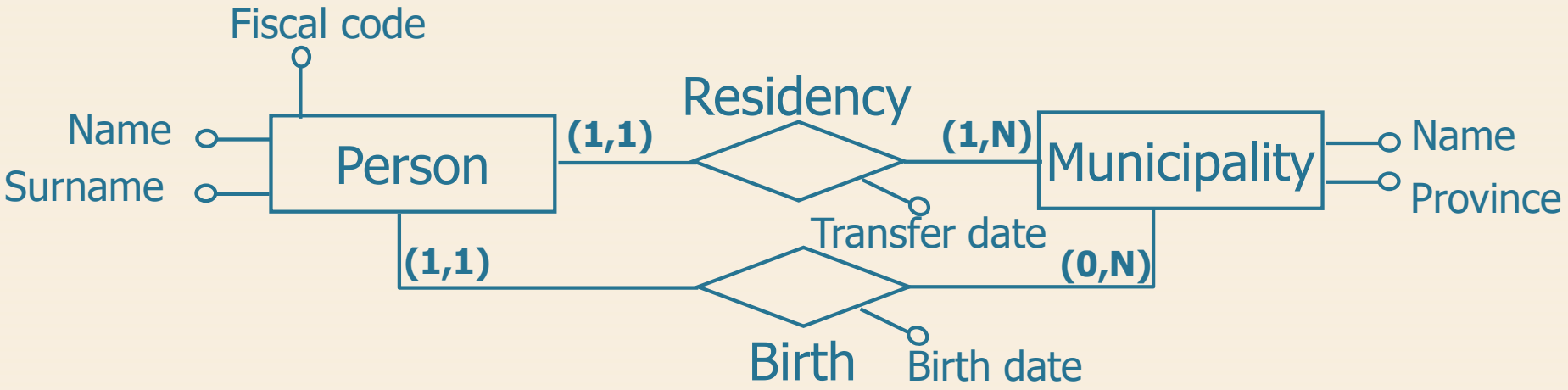
The attribute



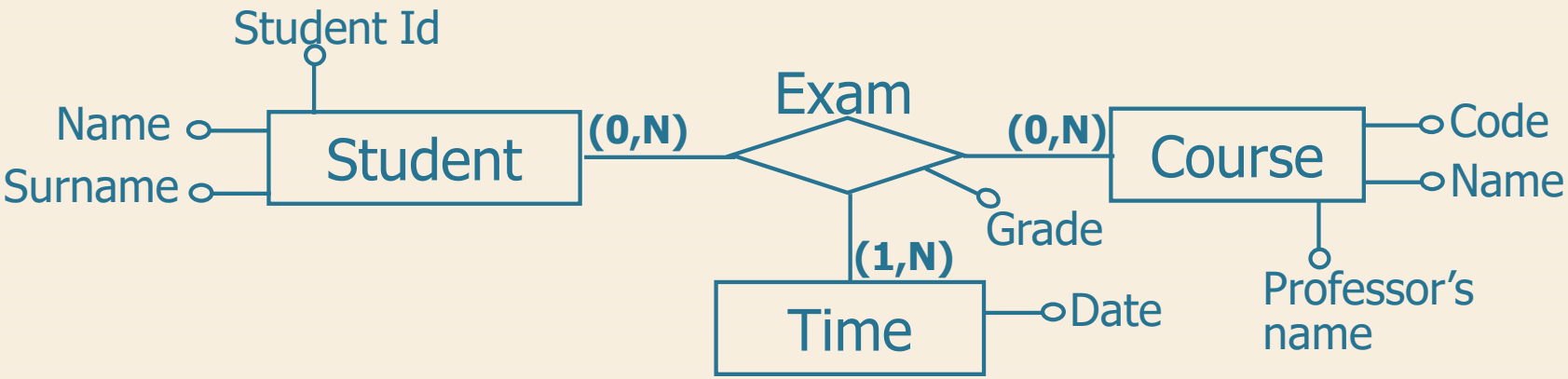
Name of the attribute

- 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.

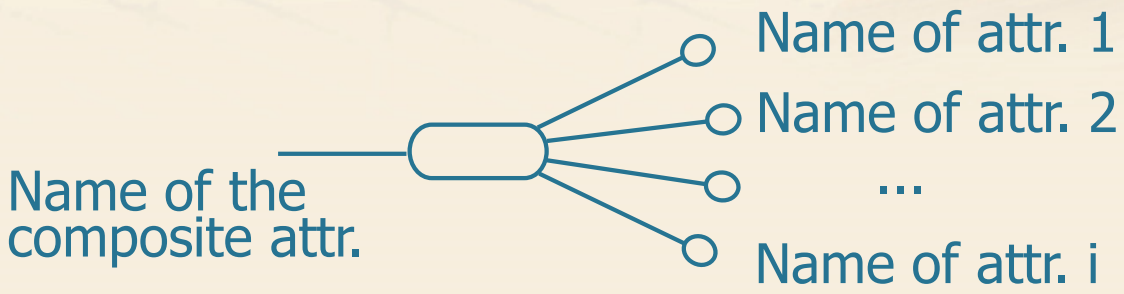
Examples of attributes



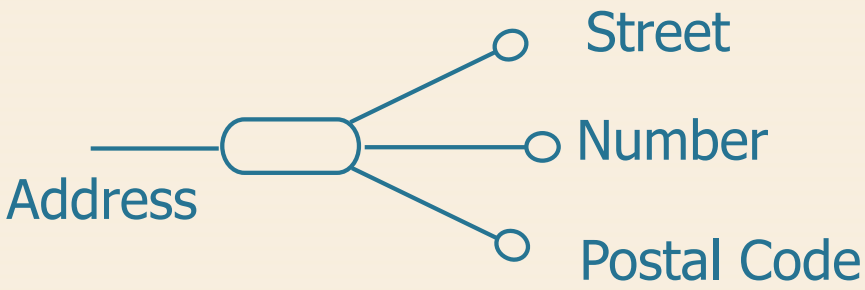
Examples of attributes



Composite attribute



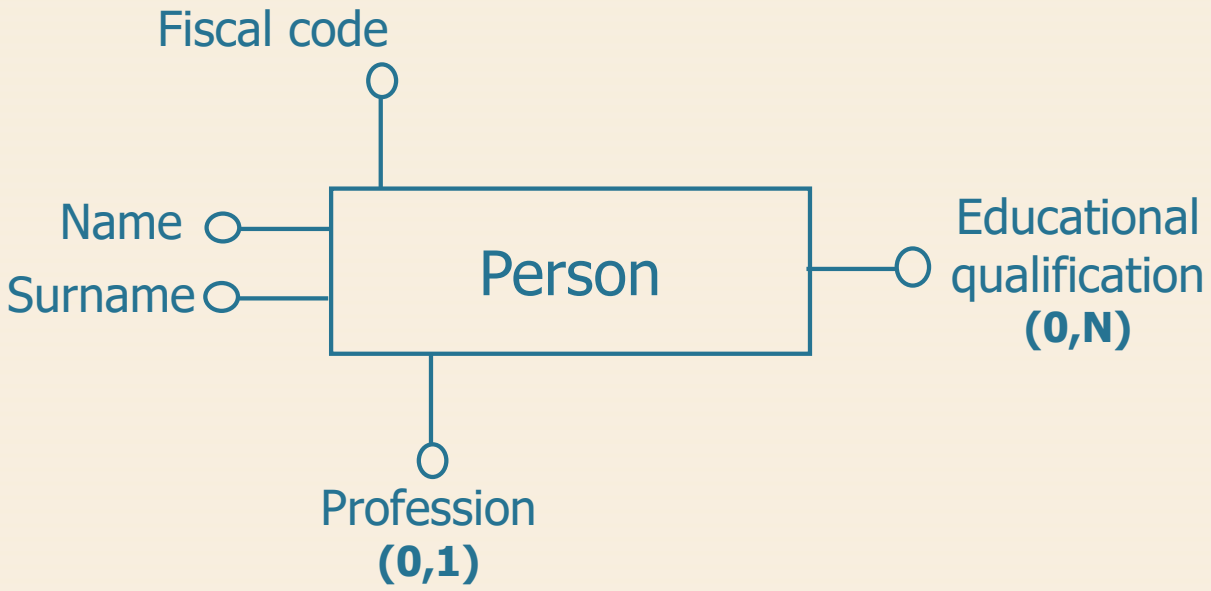
- Group of attributes that have closely connected meanings or uses.
- Example



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



The Entity-Relationship Model

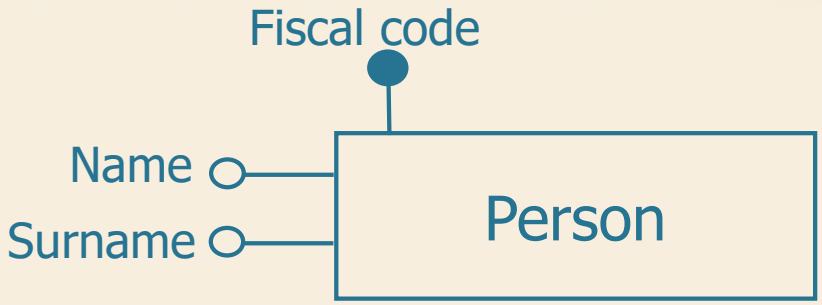
Identifiers

Identifier

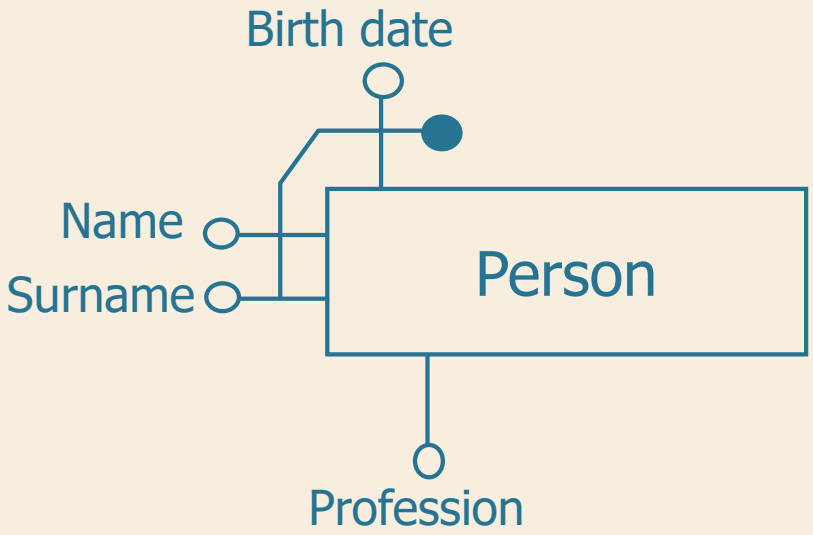
- It is specified for each entity
- It describes concepts (attributes and/or entities) of the scheme that allow to identify uniquely the instances of an entity.
  - Each entity must have at least on identifier
  - It can exists more than one appropiated identifier for a given entity.

Internal Identifier

- Simple: consisting of only one attribute

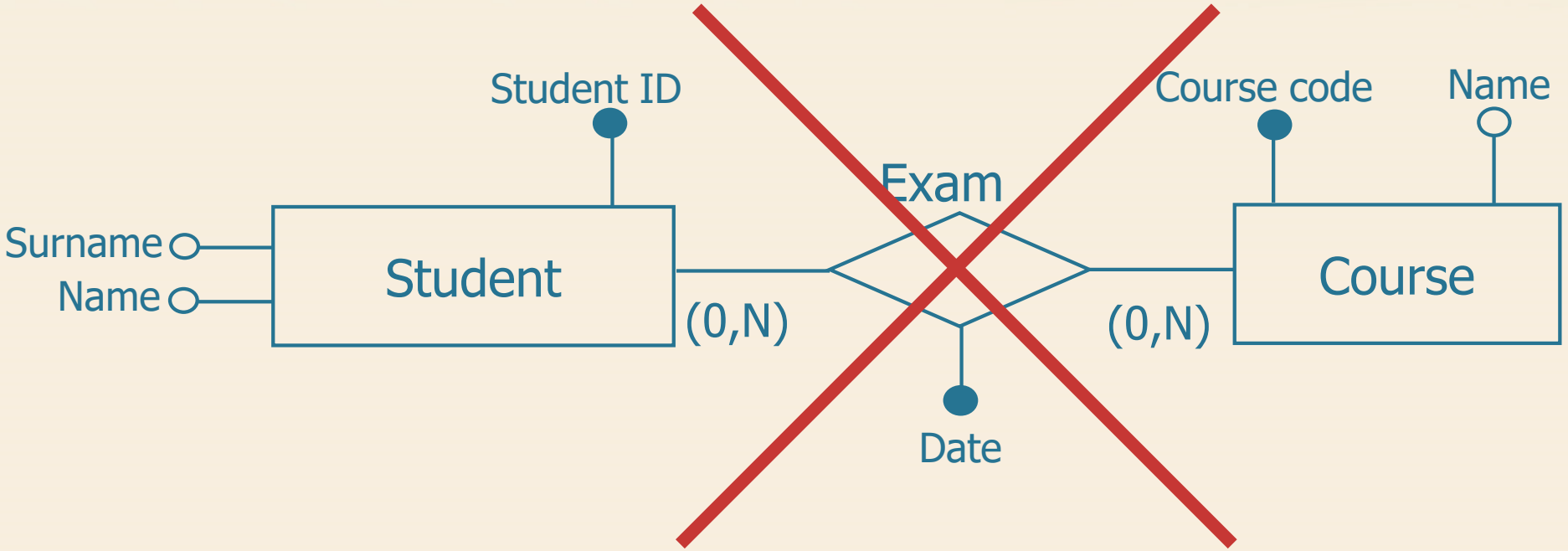


- Composite: consisting of multiple attributes



Remarks

- Relationships do *not* have identifiers



The Entity-Relationship Model

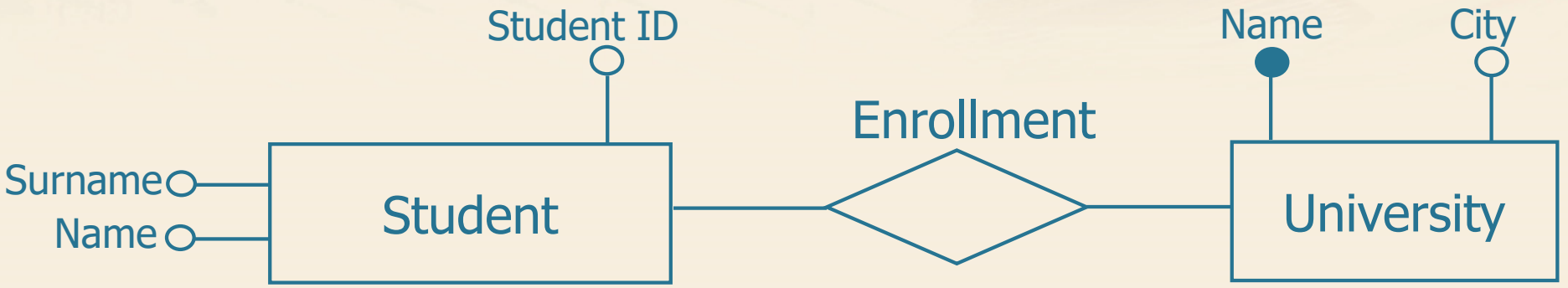
External Identifiers

Identifier

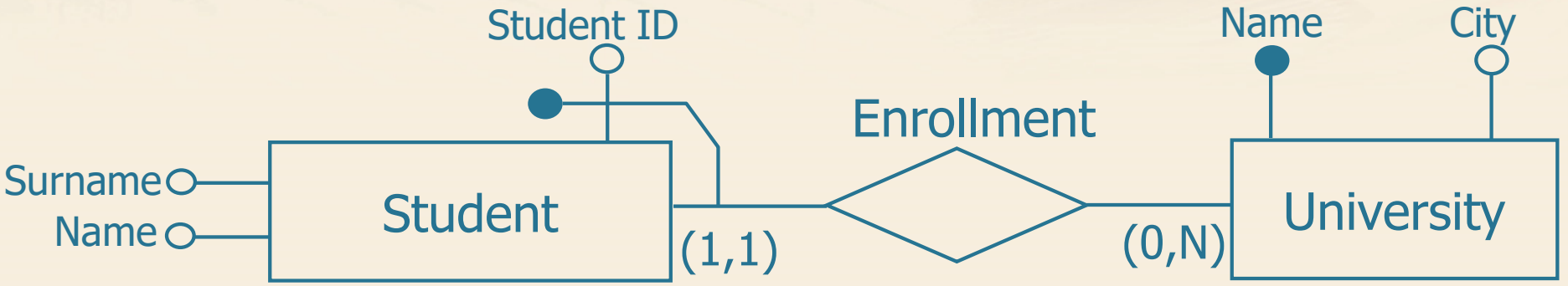
- It is specified for each entity
- It describes concepts (attributes and/or entities) of the scheme that allow to identify uniquely the instances of an entity.
  - Each entity must have at least on identifier
  - It can exists more than one appropiated identifier for a given entity.

External Identifier

External Identifier



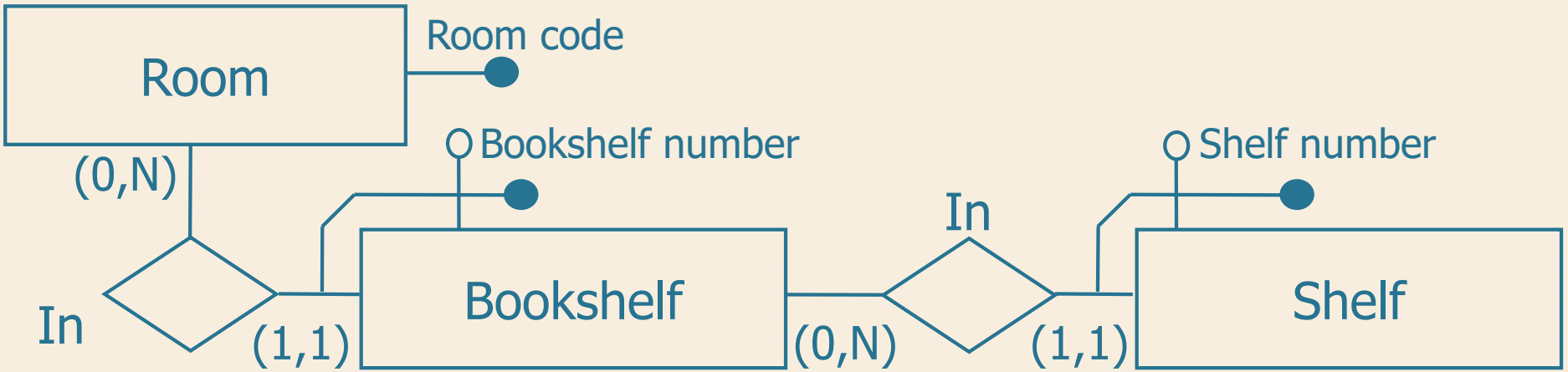
External Identifier



- 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

Remarks

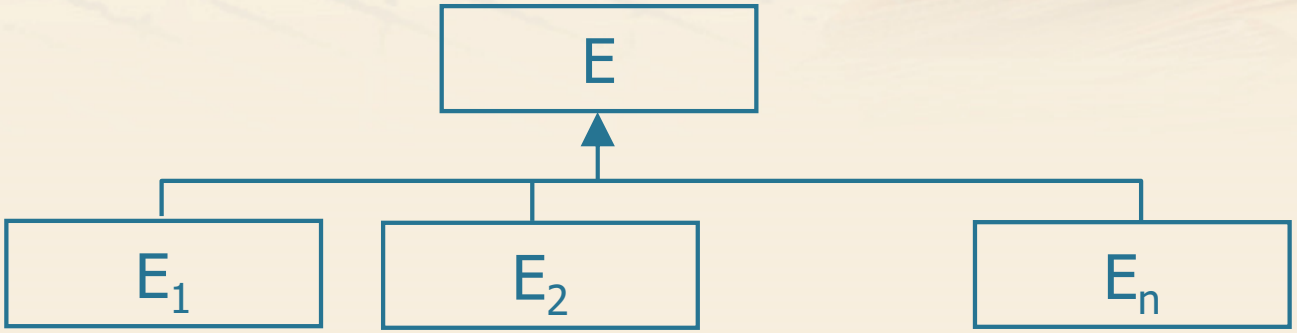
- An external identifier may involve an entity that is itself externally identified
  - Identification cycles must not be generated



The Entity-Relationship Model

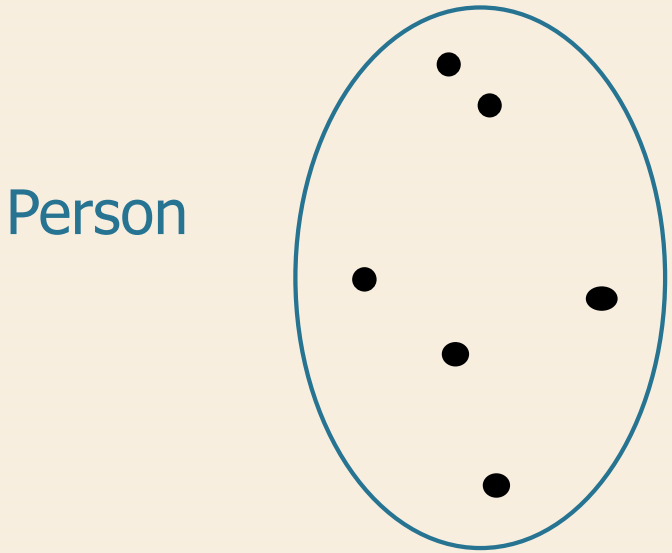
Generalizations

Generalization

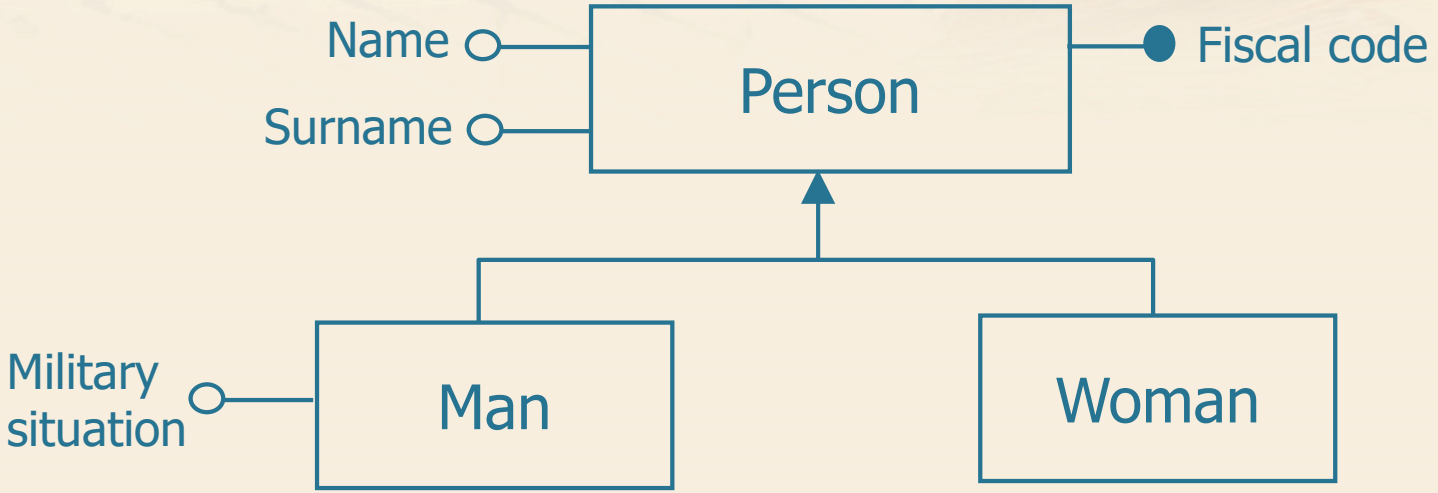


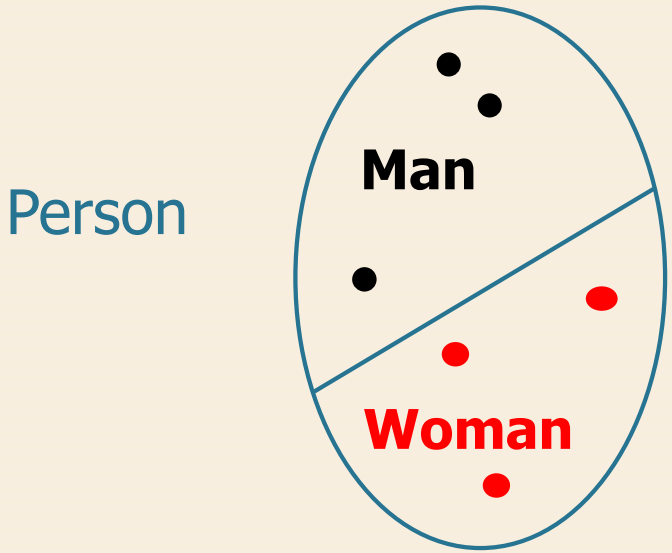
- It describes a logical link between an entity E and one or more entities E_1, E_2,…, E_n, that are particular cases of E.
  - E is called parent entity, is a generalization of E_1, E_2,…, E_n
  - E_1, E_2,…, E_n are called child entities, are specialization of E

Generalization: example

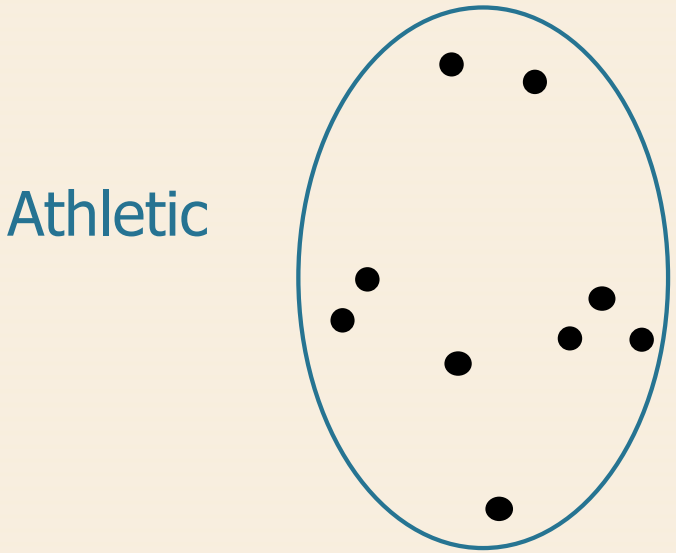


Generalization: example

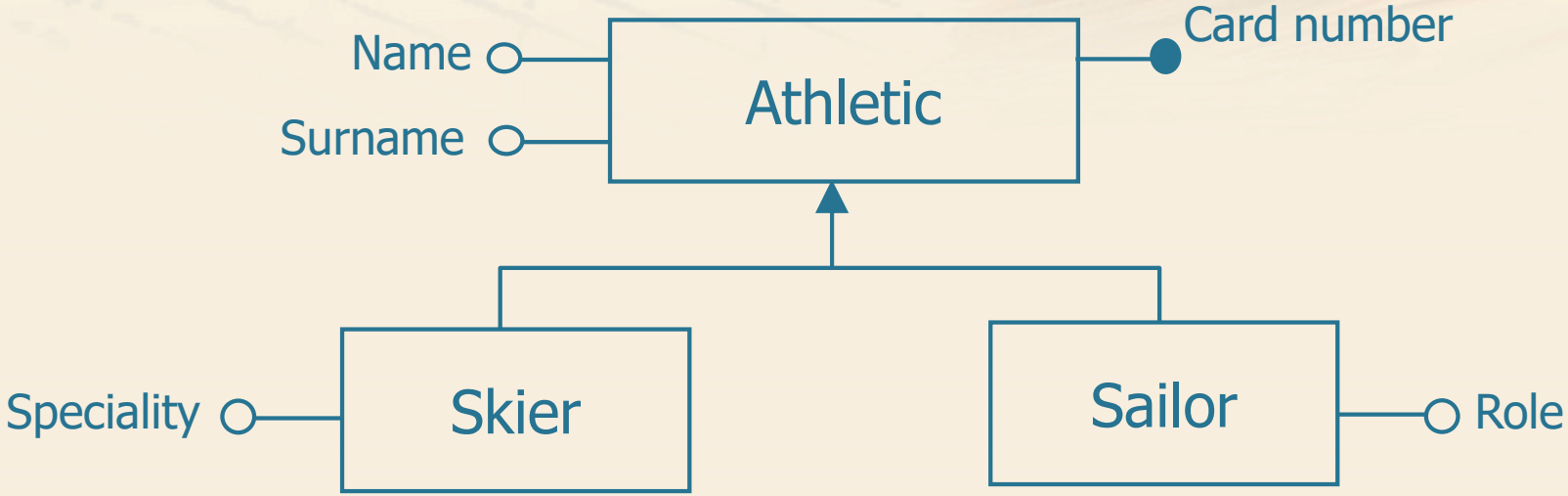


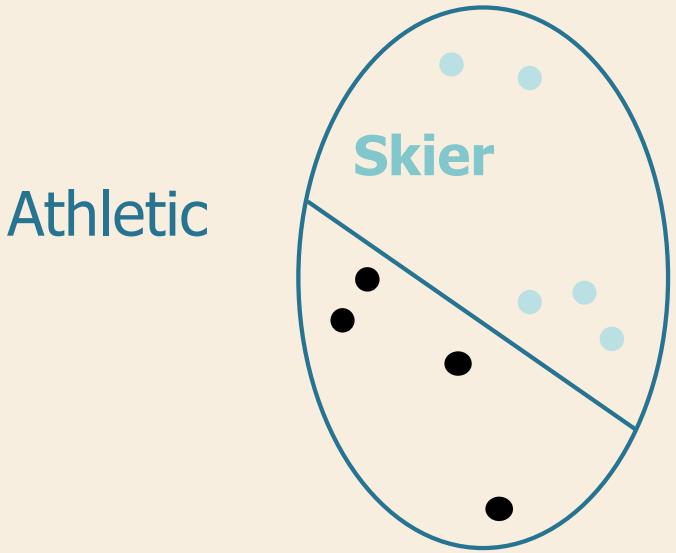


Generalization: example

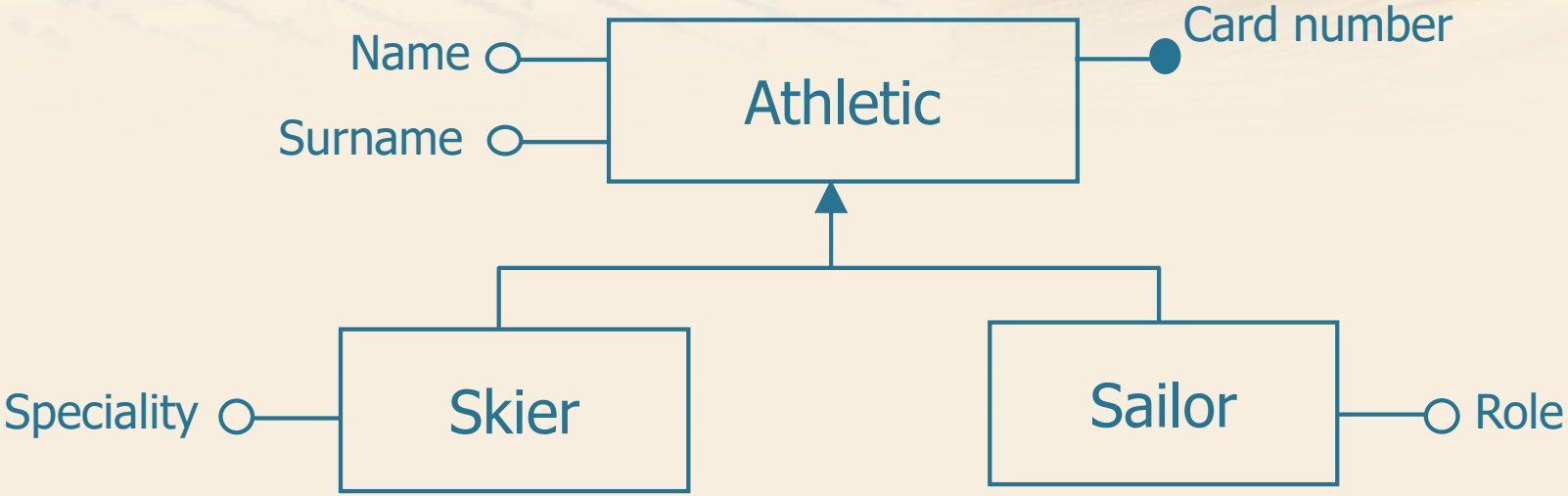


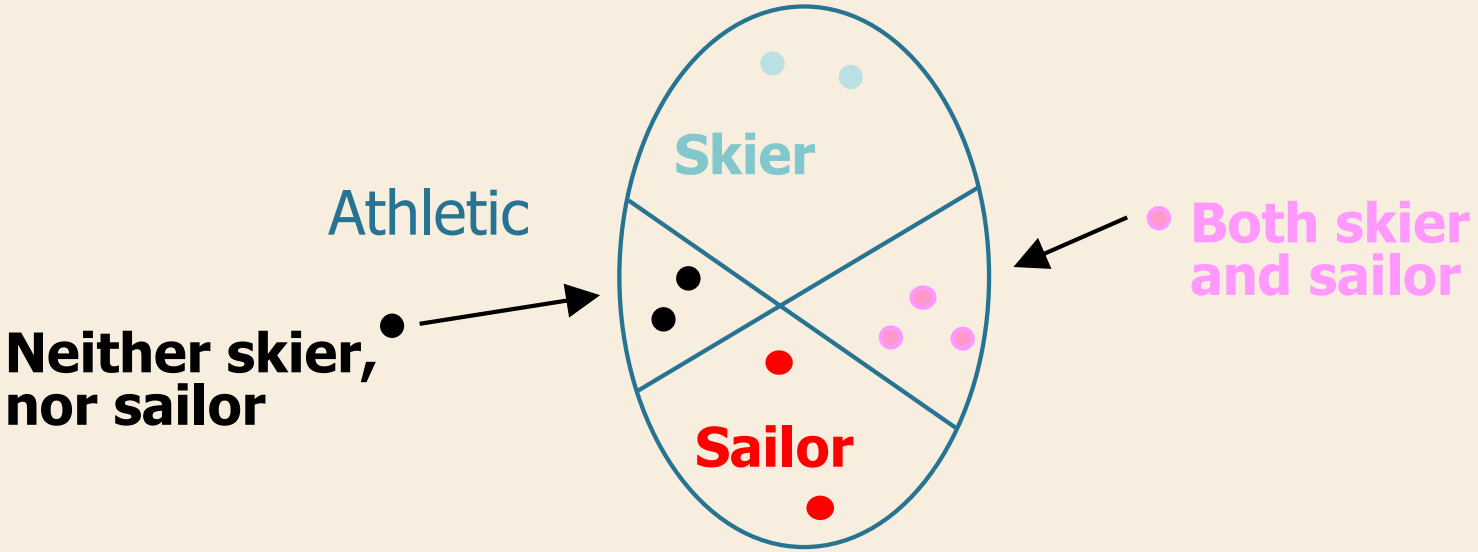
Generalization: example





Generalization: example

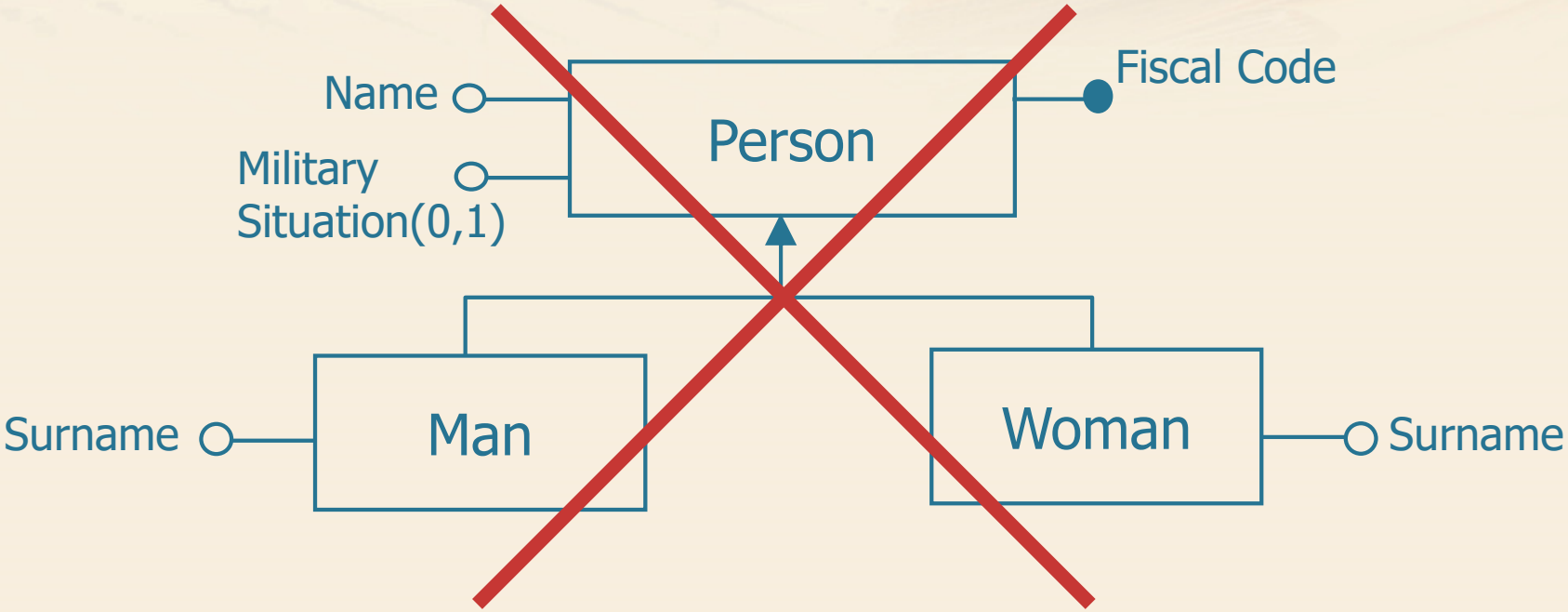




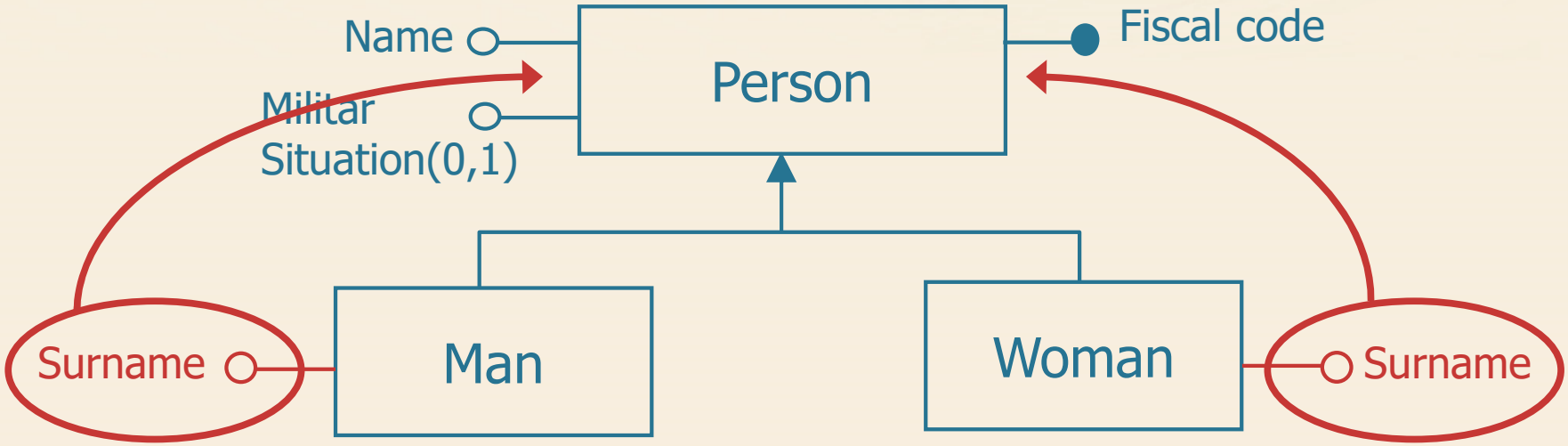
Generalization: property

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

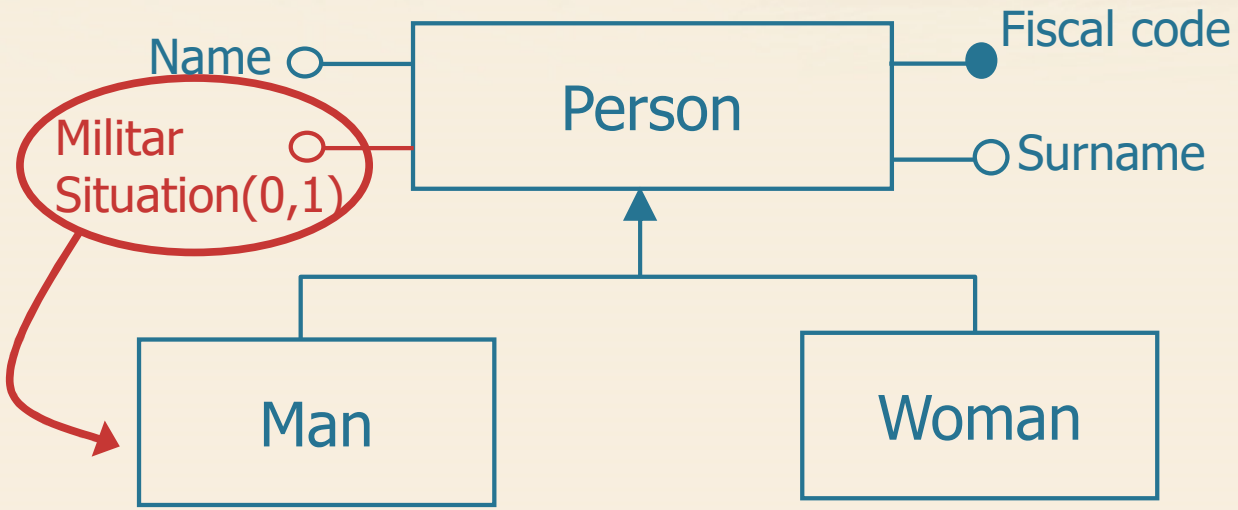
Generalizzation: incorrect example



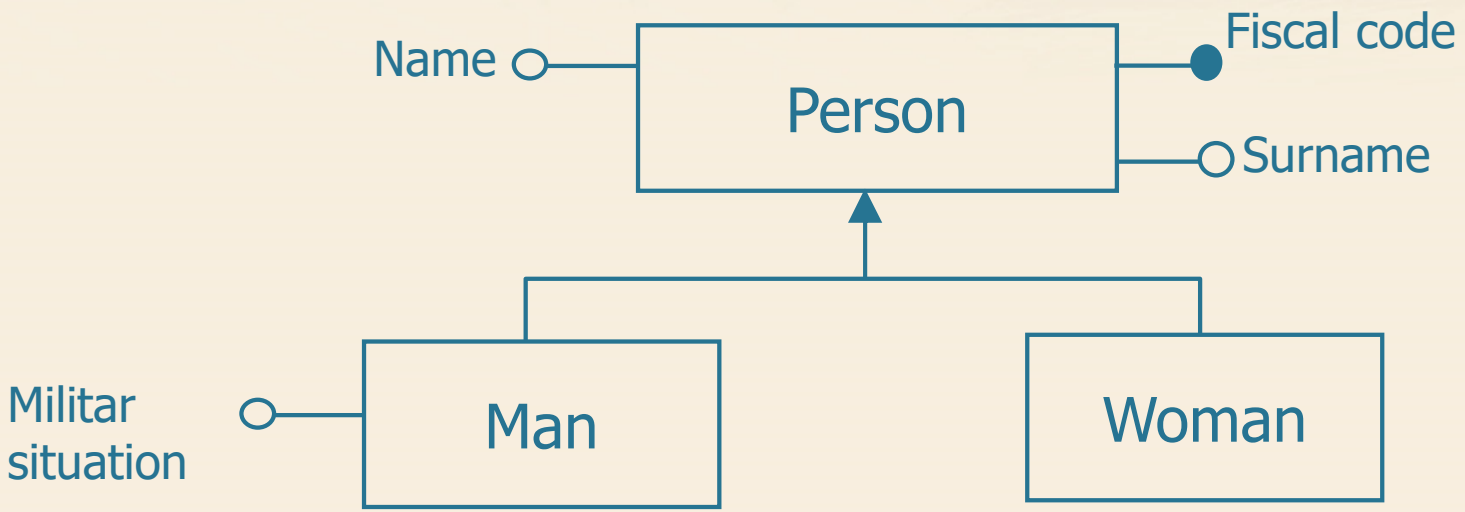
Generalizzation: incorrect example



Generalizzation: incorrect example



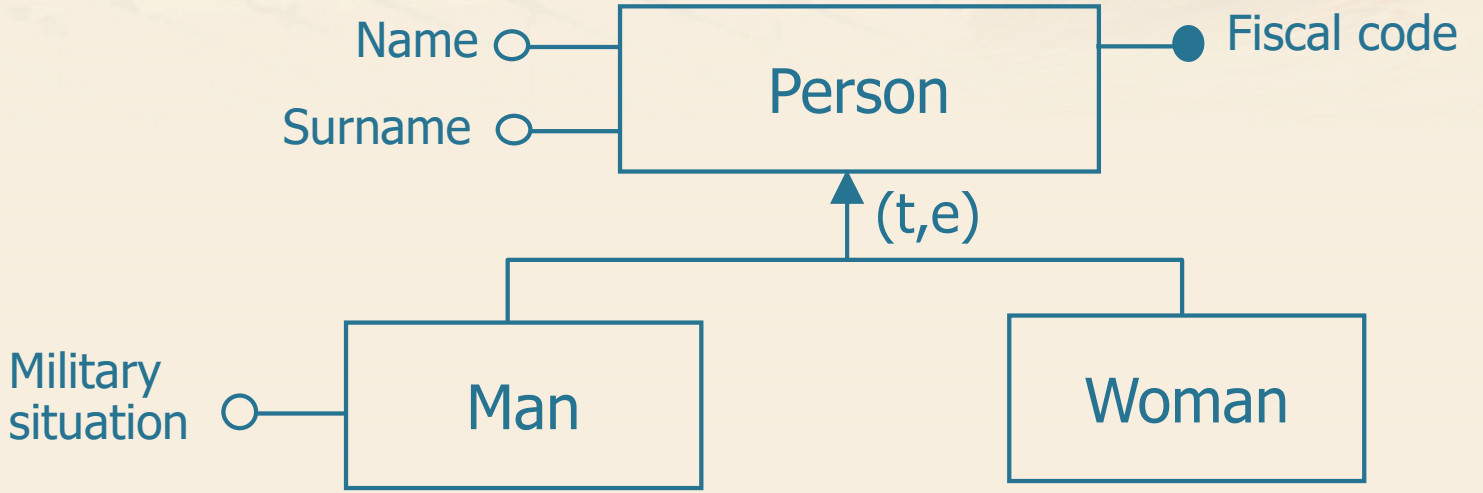
Generalizzation: correct example

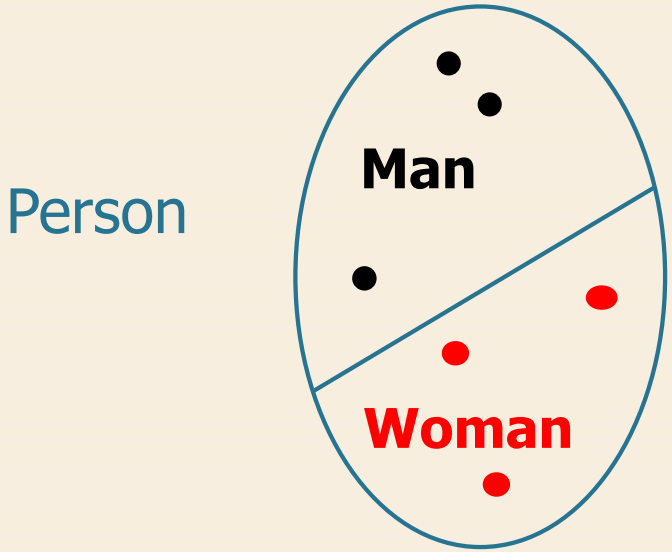


Generalization: property

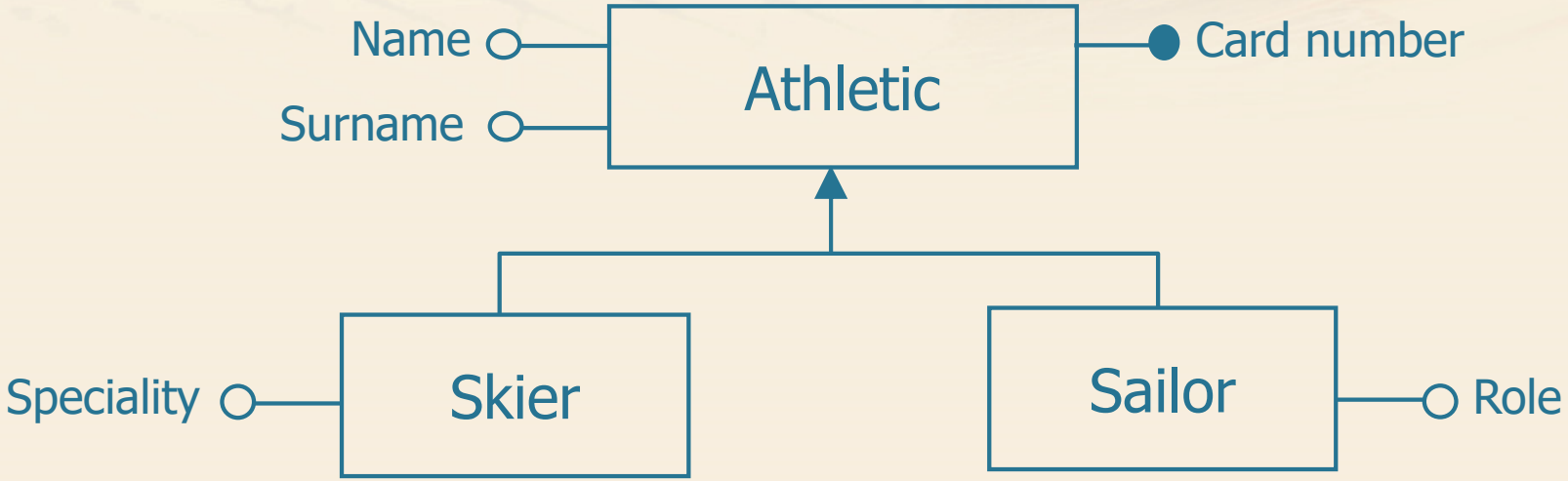
- Orthogonal characteristics
  - *total* generalization if each instance of the parent entity is an instance of at least one of the child entities, *partial* otherwise.
  - *exclusive* if each instance of the parent entity is at most one instance of one of the child entities, *overlapping* otherwise.

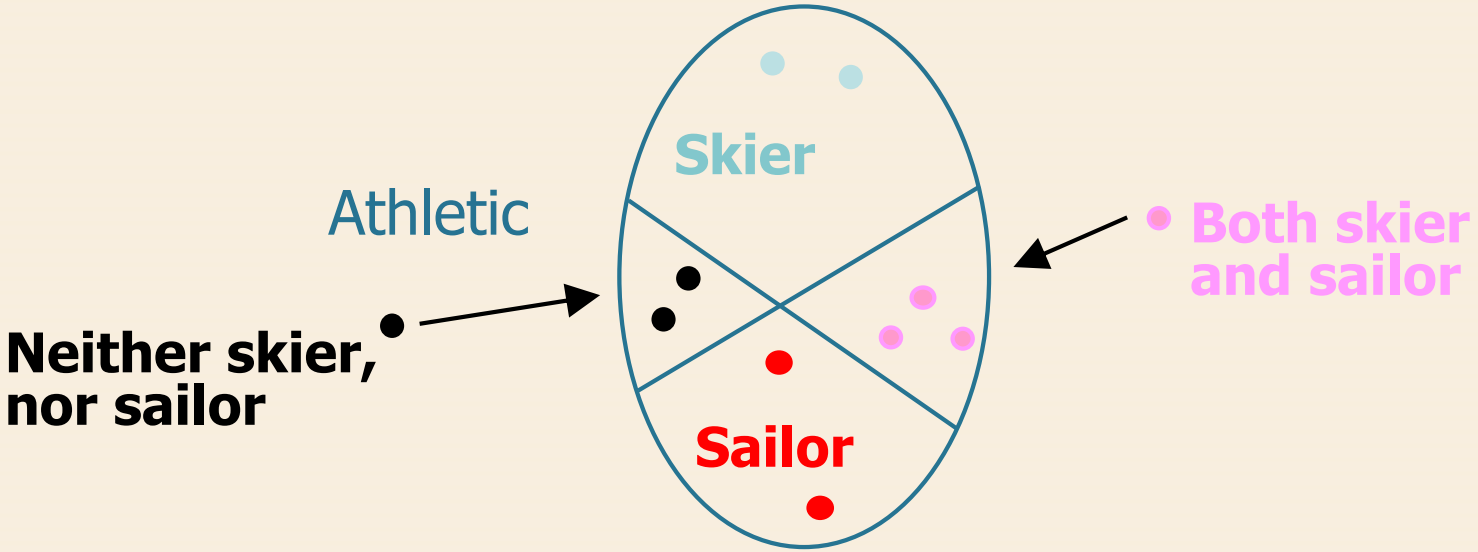
Generalization: example





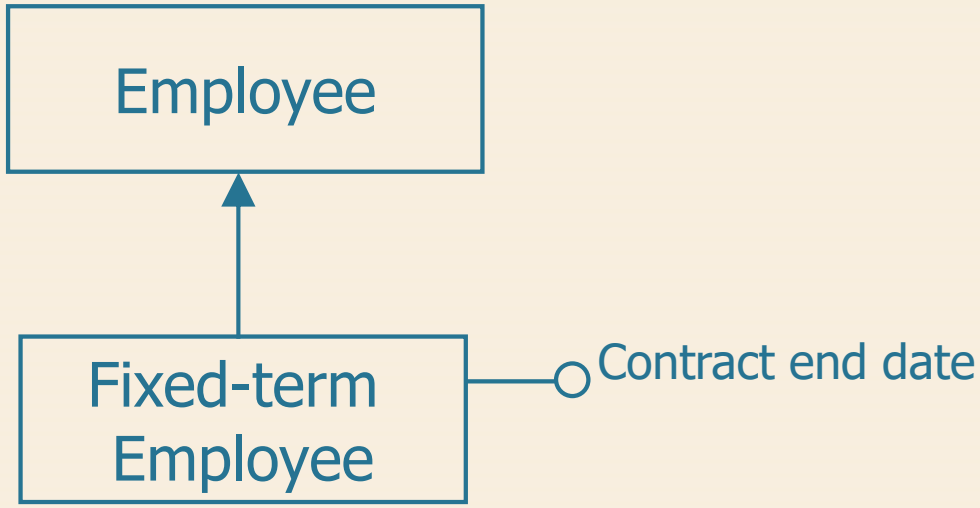
Generalization: example





Subset

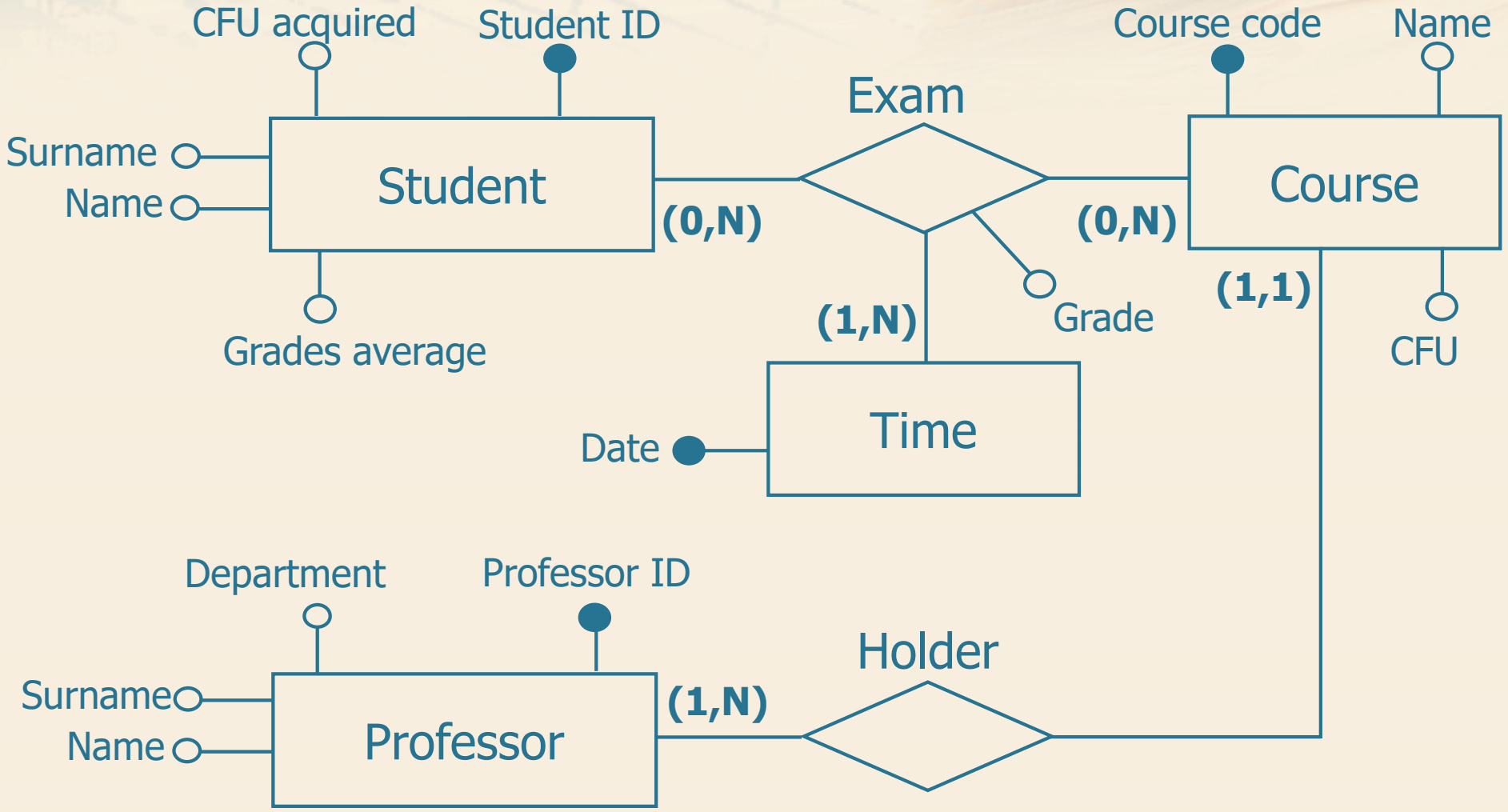
- Particular case of generalization with only one child entity
  - the generalization is always partial and exclusive.



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 memorize the mark obtained	Student (0,N), Course (0,N), Time (1,N)	Grade
Holder	It associates each course to its holder professor.	Course (1,1), Professor (0,N)	

Documentation of E-R schemes

- Data Dictionary
  - It allows to enrich the E-R scheme with natural language description of entities, relationships and attributes
- Constraints of integrity 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
  - It allows to enrich the E-R scheme with natural language description of entities, relationships and attributes
- Constraints of data integrity
  - may not always be explicitly indicated in an E-R scheme
  - can be described in natural language
- Rules for deriving data
  - allow to explain that a scheme concept can be obtained (by 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 marks of a student is obtained by calculating the average marks of the exams passed by the student

The Entity-Relationship Model

UML and E-R

UML and E-R

- UML (Unified Modeling Language)
  - is a modeling of a software application
    - structural and behavioural aspects (data, operations, processes and architectures)
  - rich formalism
    - Diagrams of classes, of actors, of sequence, of comunication, of the states,...
- E-R
  - is a modeling of a data base
    - Structural aspects of an application
  - useful constructs for the modelling of databases

UML and E-R

- Main charachteristics of UML that differs with respect to ER
  - 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 different use, may be adapted for the description of the conceptual design of a database