Politecnico di Torino

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
- The methodology for database design is based on the separation of decisions
  - *what* to represent in the database
    - conceptual design
  - *how* to represent it
    - logical and physical design

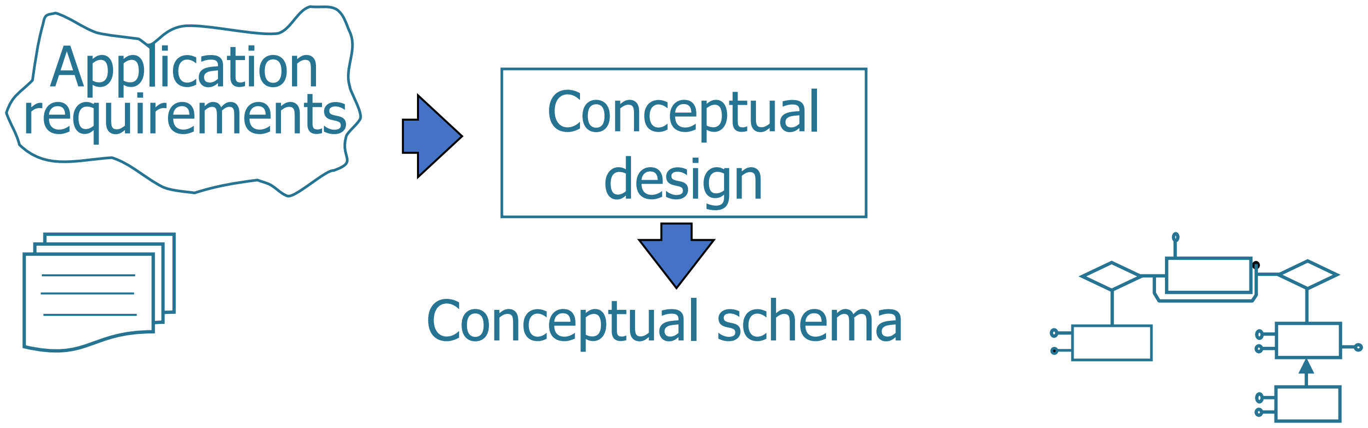
Stages of database design

Application requirements

Application requirements

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

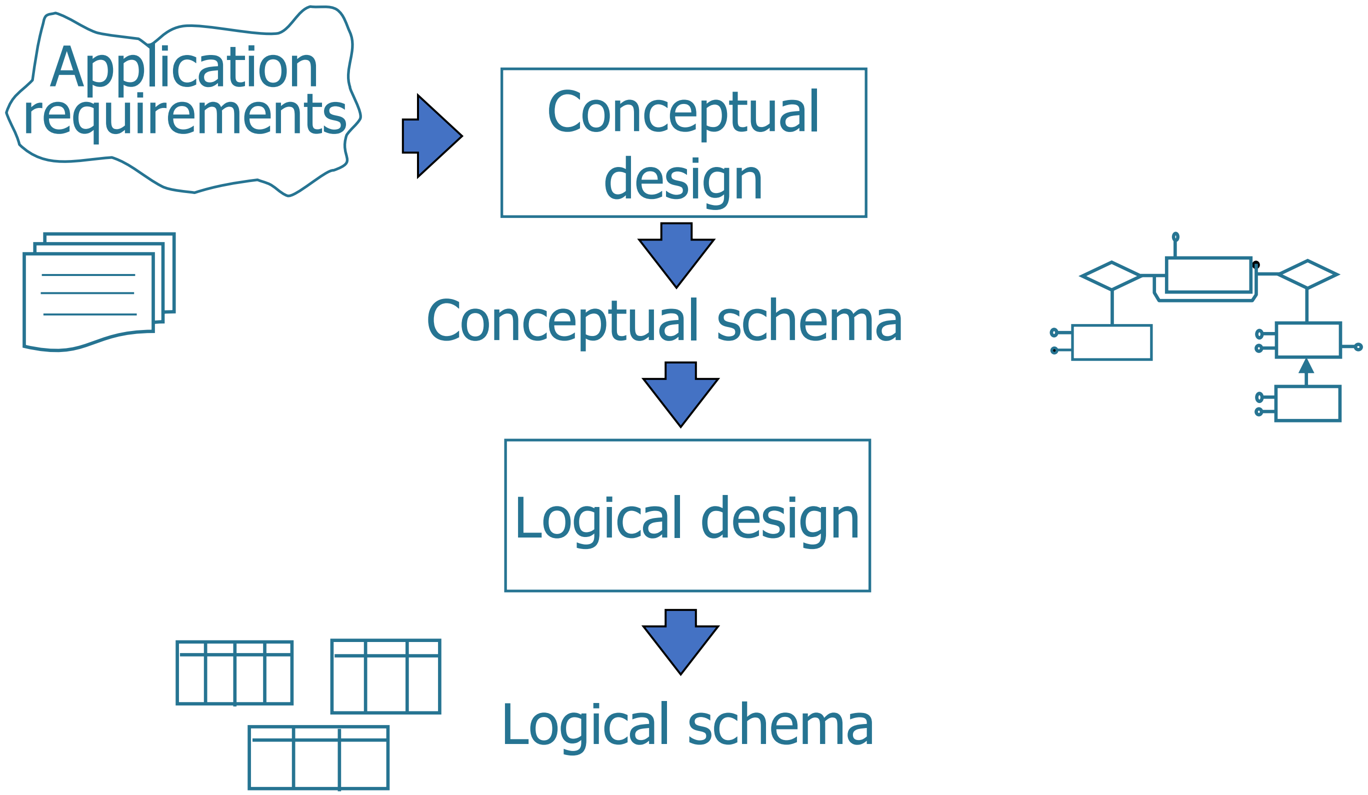
Stages of database design



Conceptual design

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

Stages of database design



Application requirements

Conceptual design

Conceptual schema

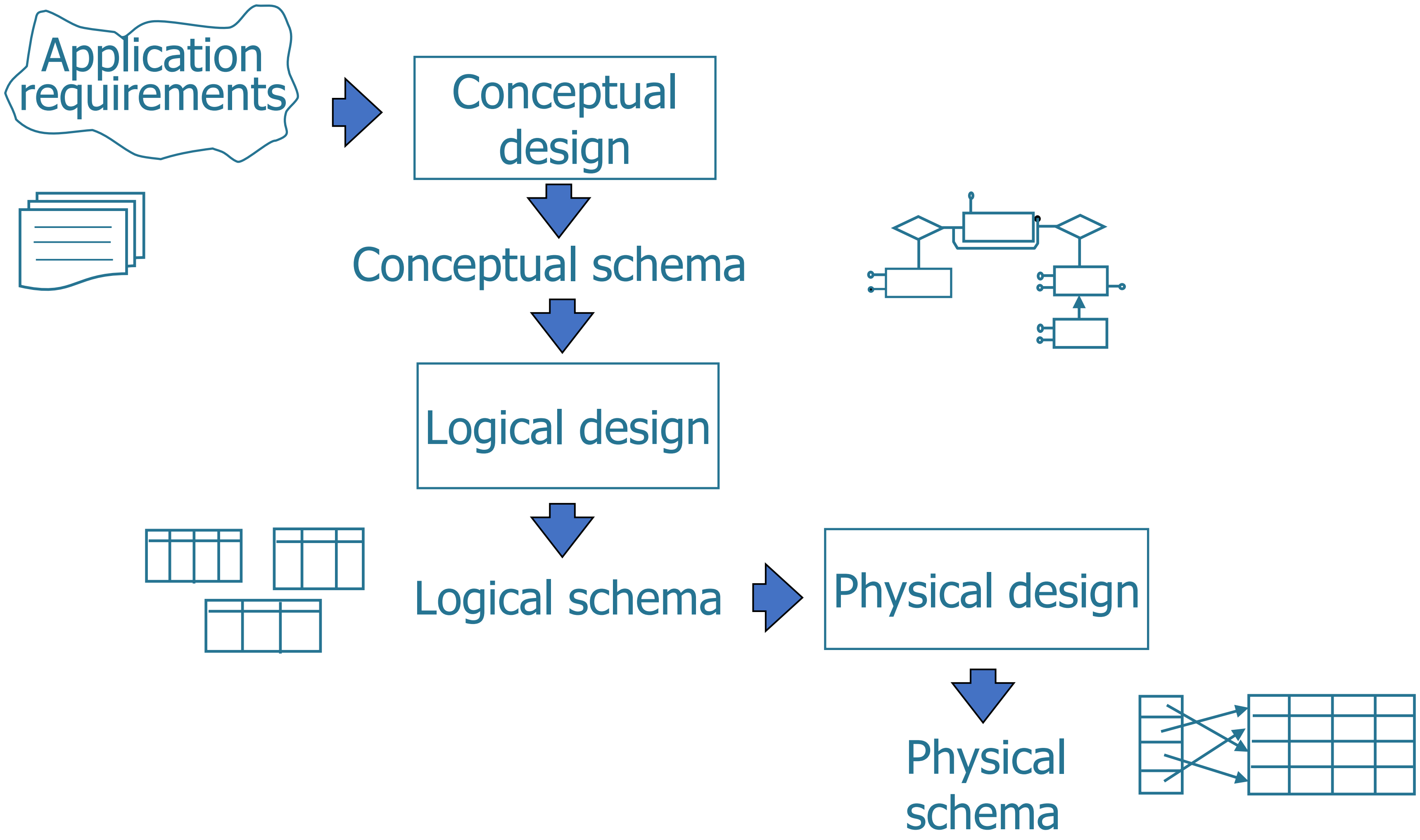
Logical design

Logical schema

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 logical schema verified by formal techniques (normalization)

Stages of database design



Application requirements

Conceptual design

Conceptual schema

Logical design

Logical schema

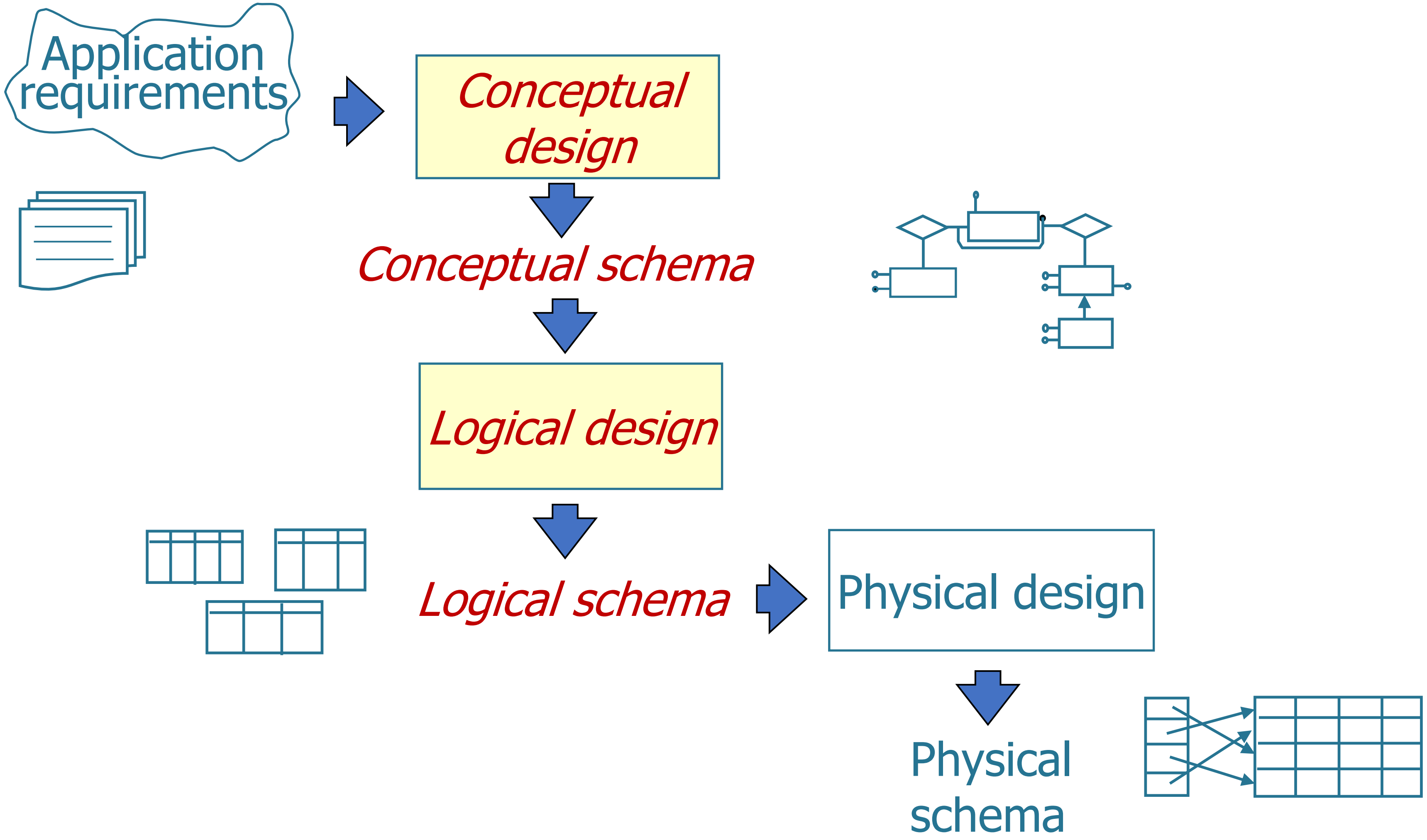
Physical design

Physical schema

Physical design

- Specification of physical data storage parameters (organization of data 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

Database design

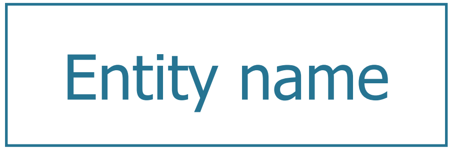
E-R model (Entity-Relationship)

- 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 logical data model, which can be chosen later
- There are numerous variations

Main constructs of the E-R model

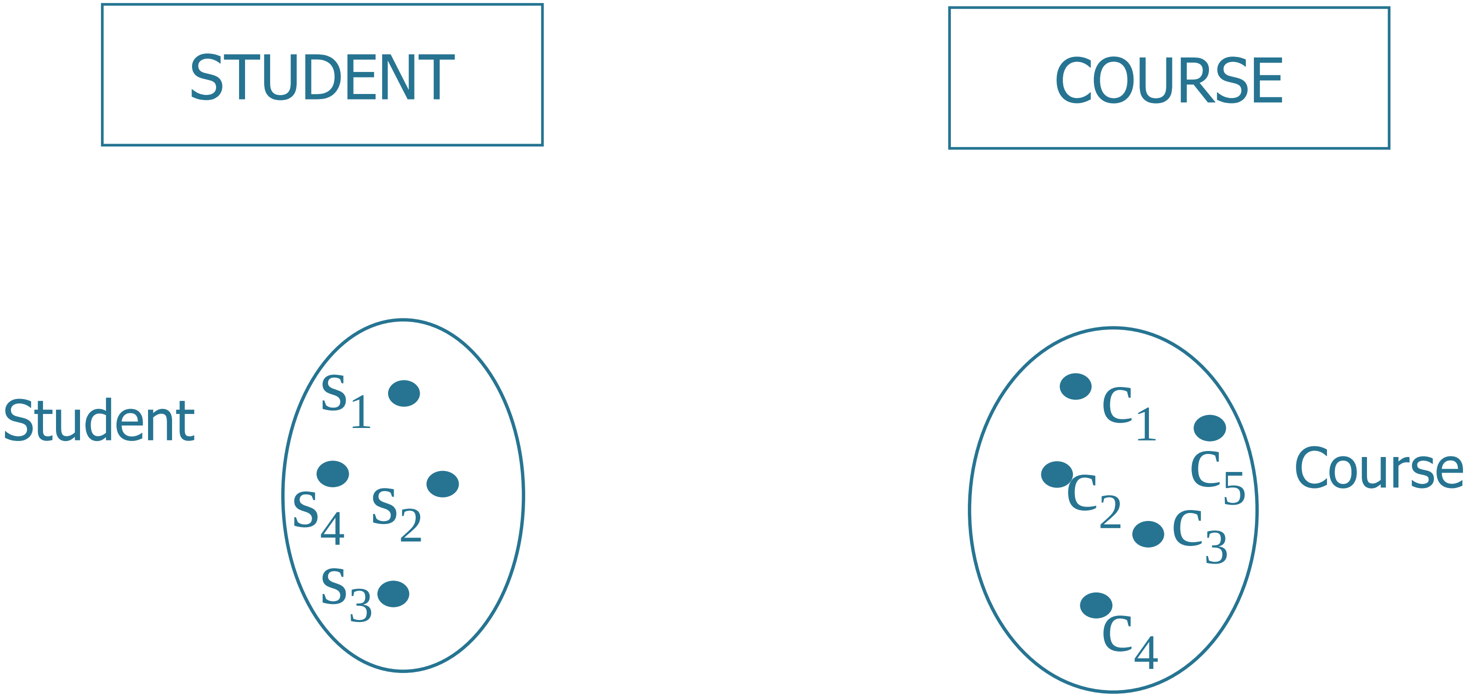
- Entity
- Relationship
- Attribute
- Identifier
- Generalization and subset

Entity

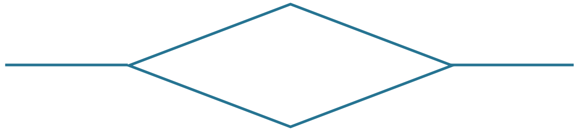


- Represents classes of real-world objects (people, things, events, ...), which they have
  - common properties
  - autonomous existence
- Examples: employee, student, ….
- An occurrence of an entity is an object of the class that the entity represents

Entity: Examples



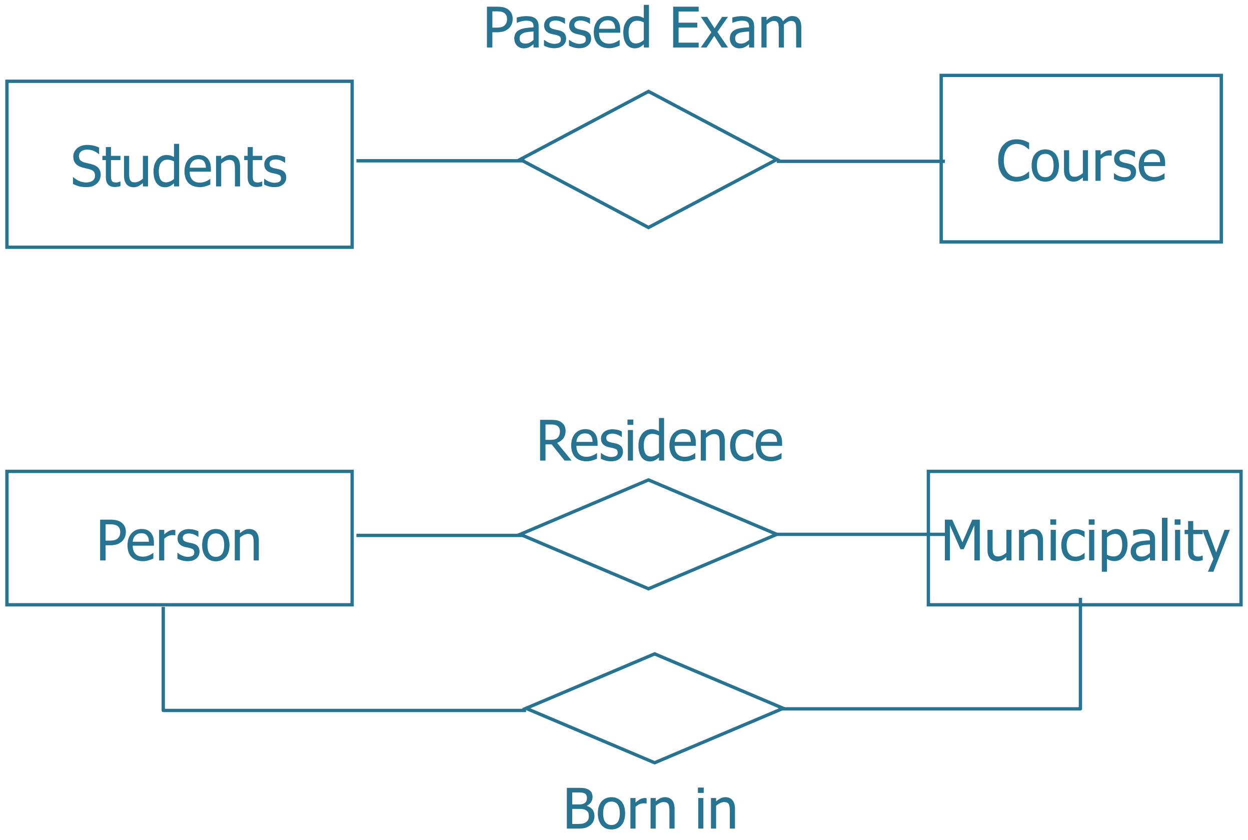
Relationship



Relationship name

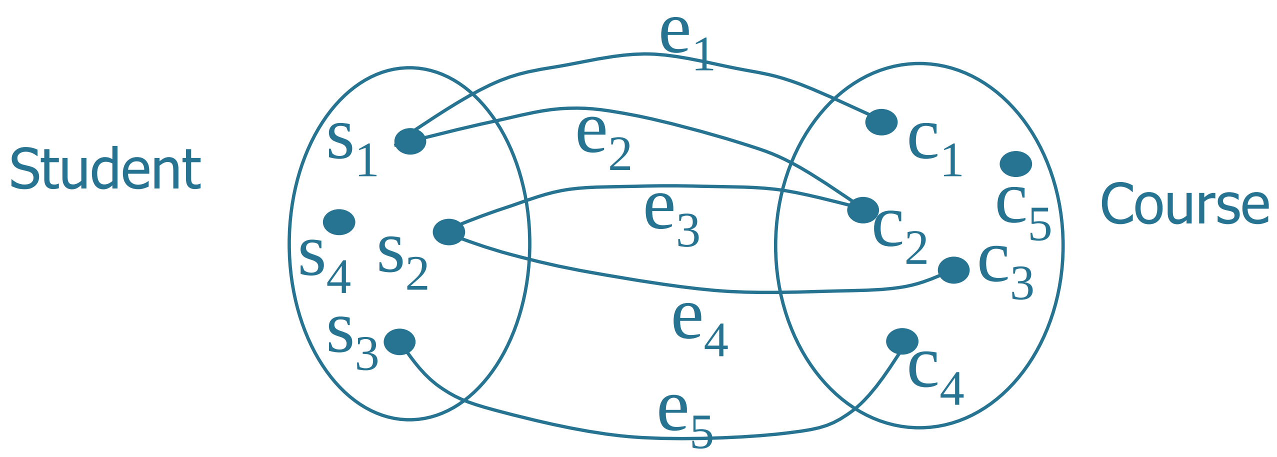
- Represents a logical link between two or more entities
- Examples: exam between student and course, residence between person and municipality
- Not to be confused with the relation of the relational model
  - sometimes it is named association

Relationship: 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 involved entities, one occurrence for each of the entities involved
- No identical n-tuples are allowed

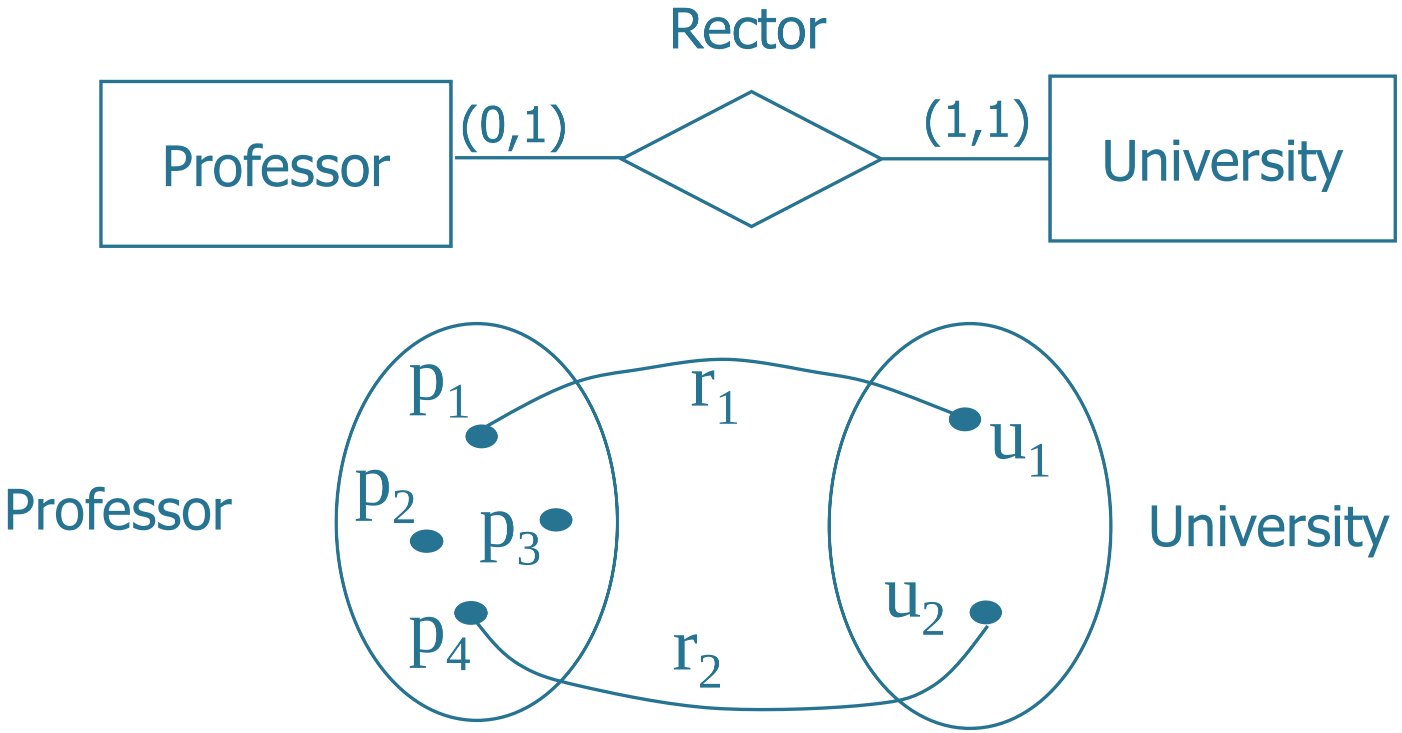


Cardinality of binary relationships

- It is specified for each entity participating in a relationship
- It describes 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)

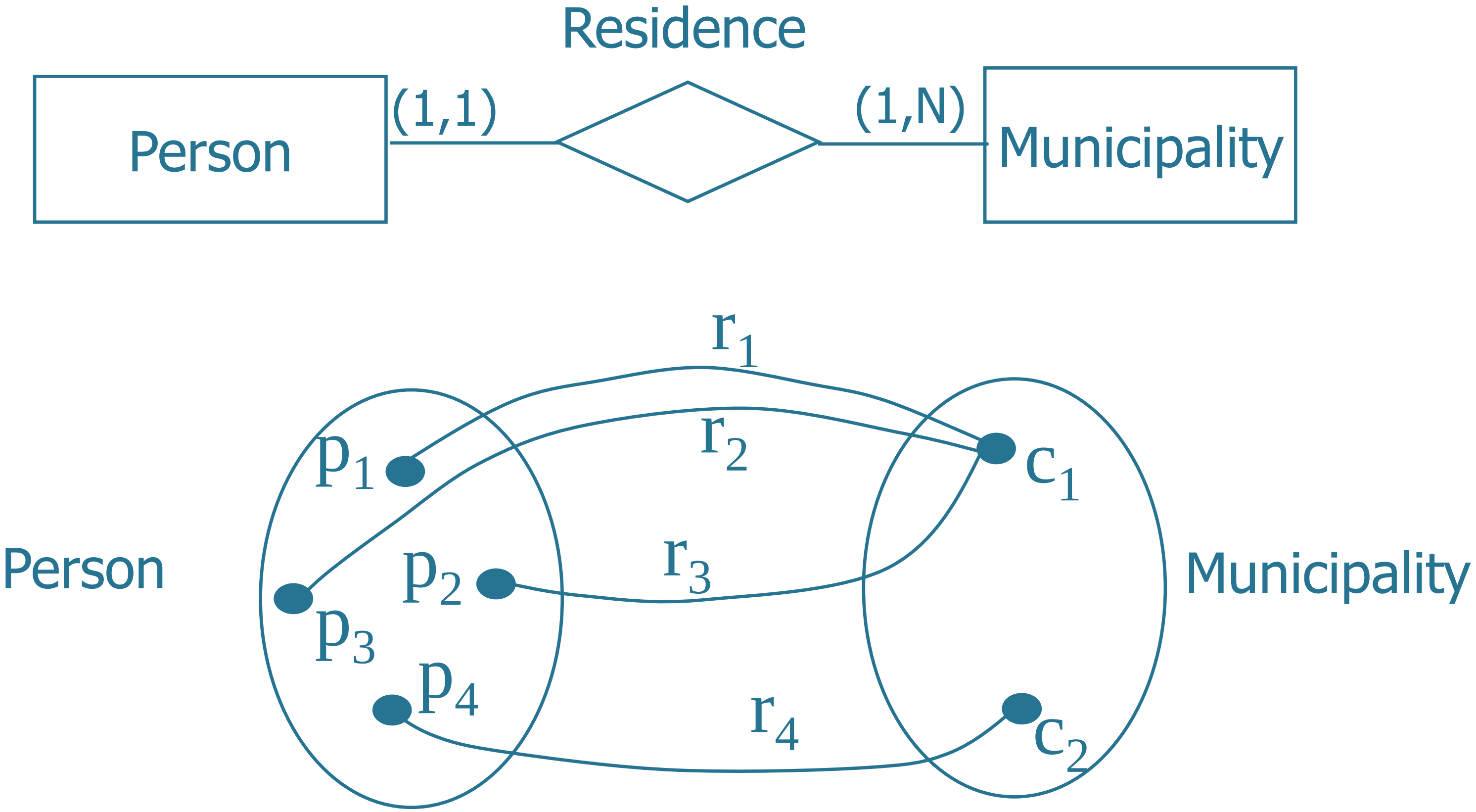
Cardinality of binary relationships

- 1 to 1 correspondence



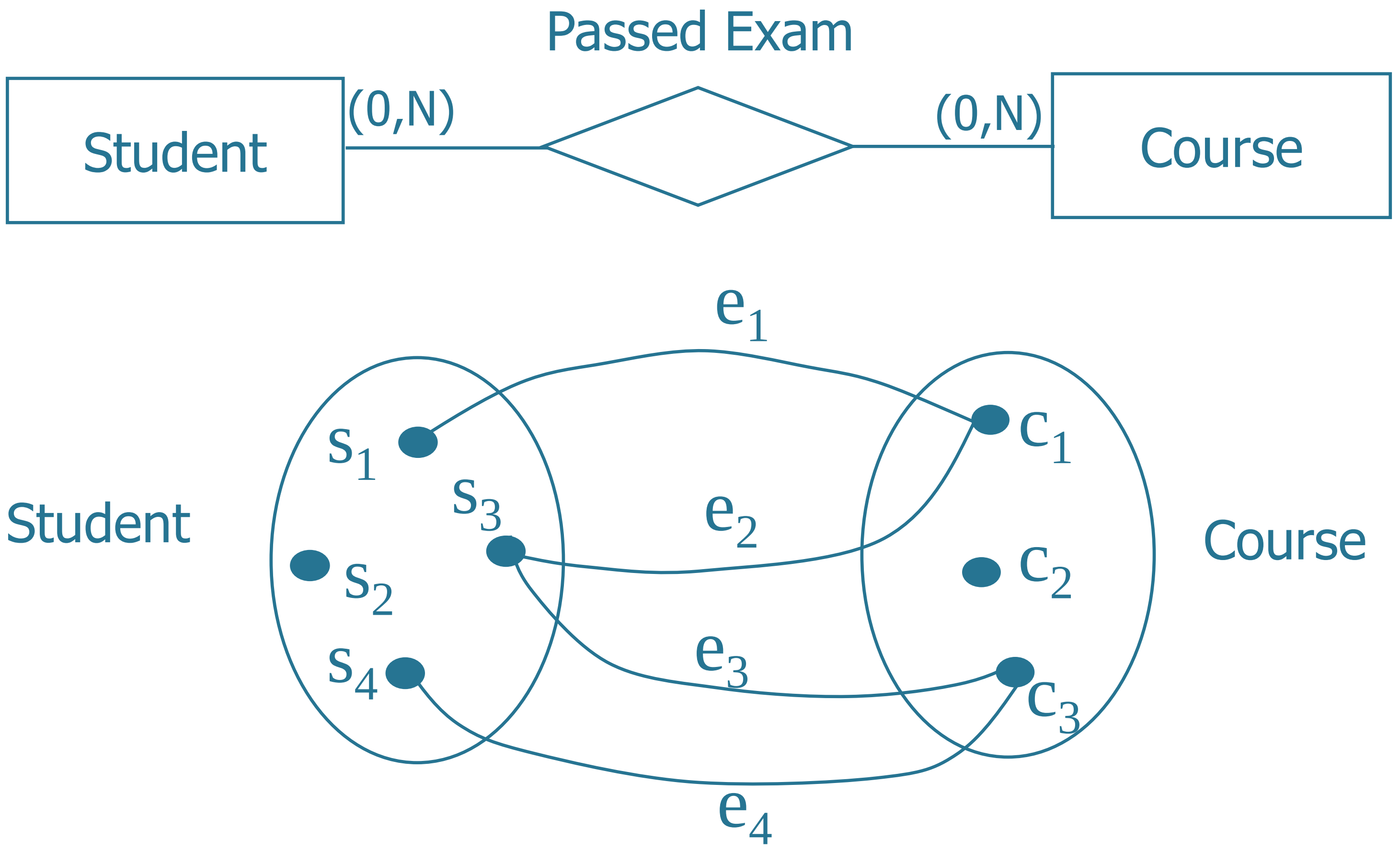
Cardinality of binary relationships

- 1 to N correspondence

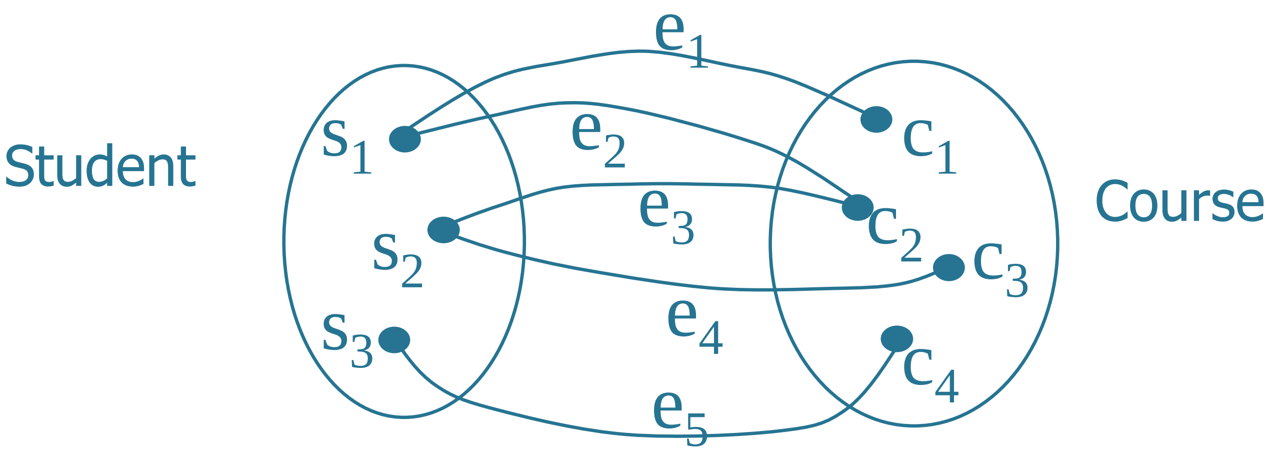


Cardinality of binary relationships

- N to N correspondence



Limitations of a binary relationship

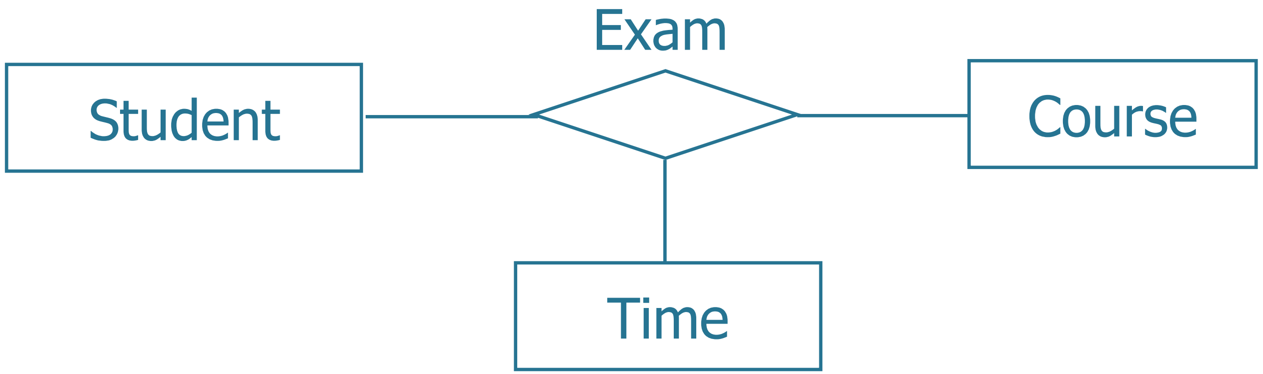


- A student cannot take the same exam more than once.

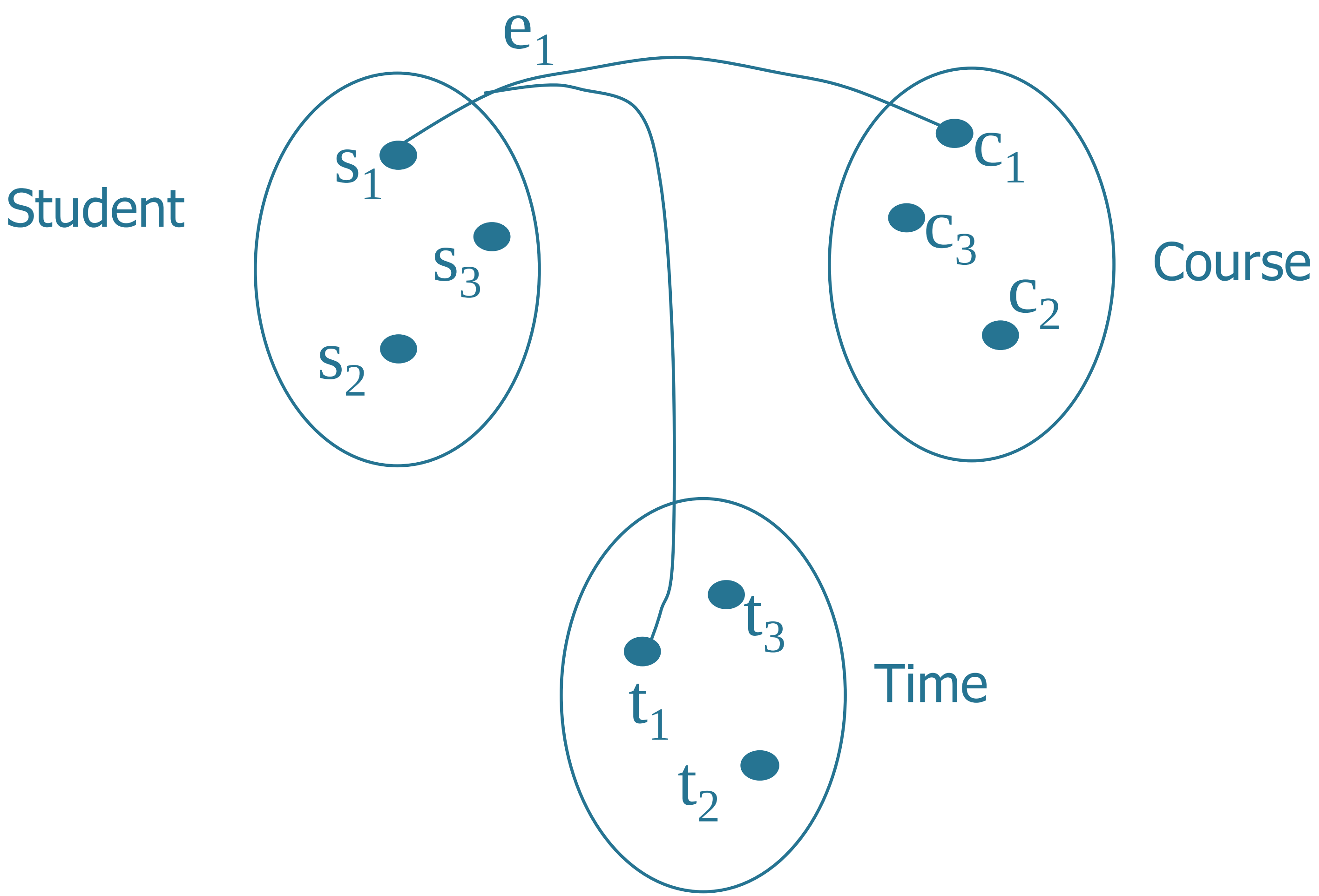
Ternary relationship

- 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
…



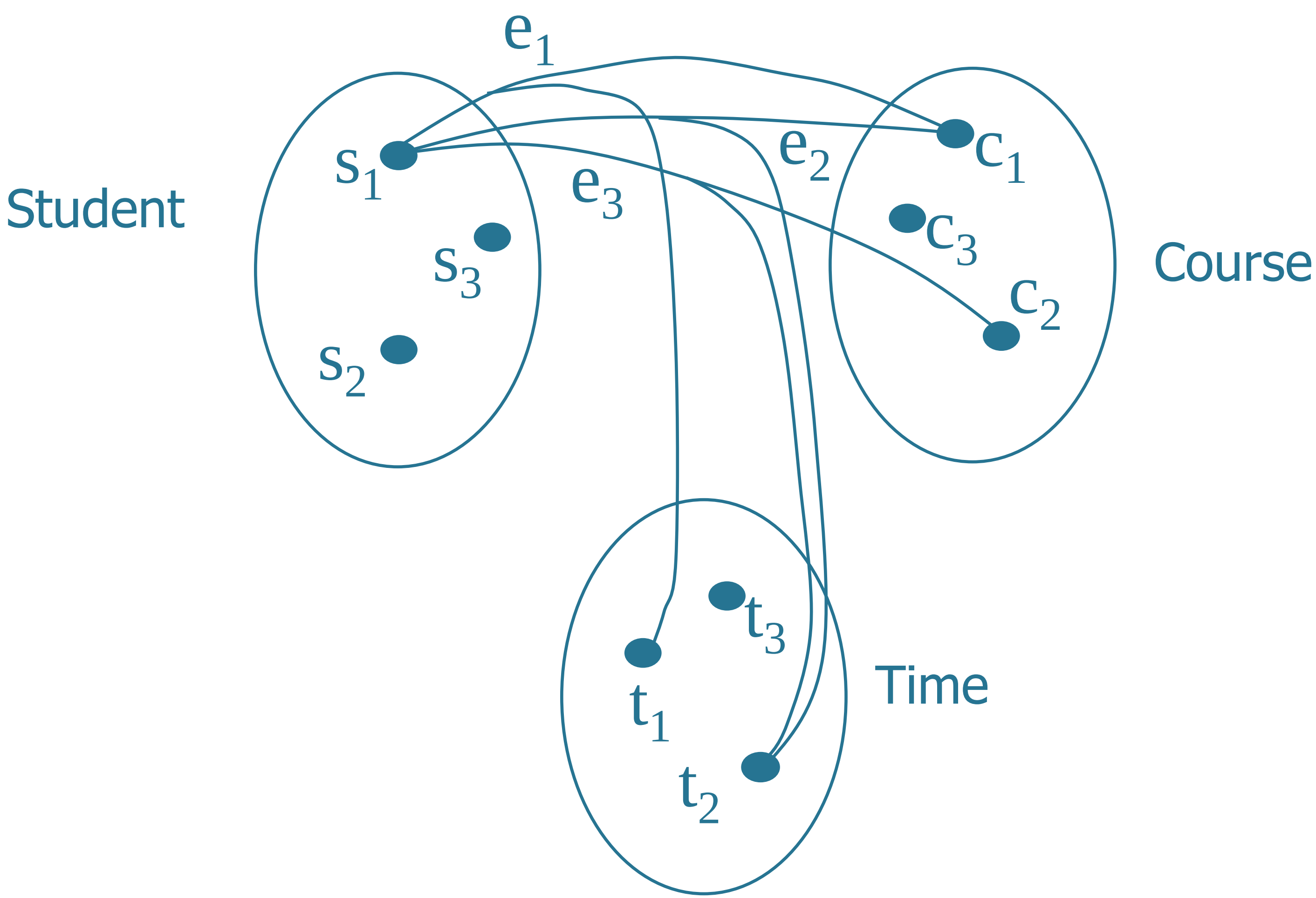
Occurences of a ternary relationship



EXAM occurrences

s_1 c_1 t_1

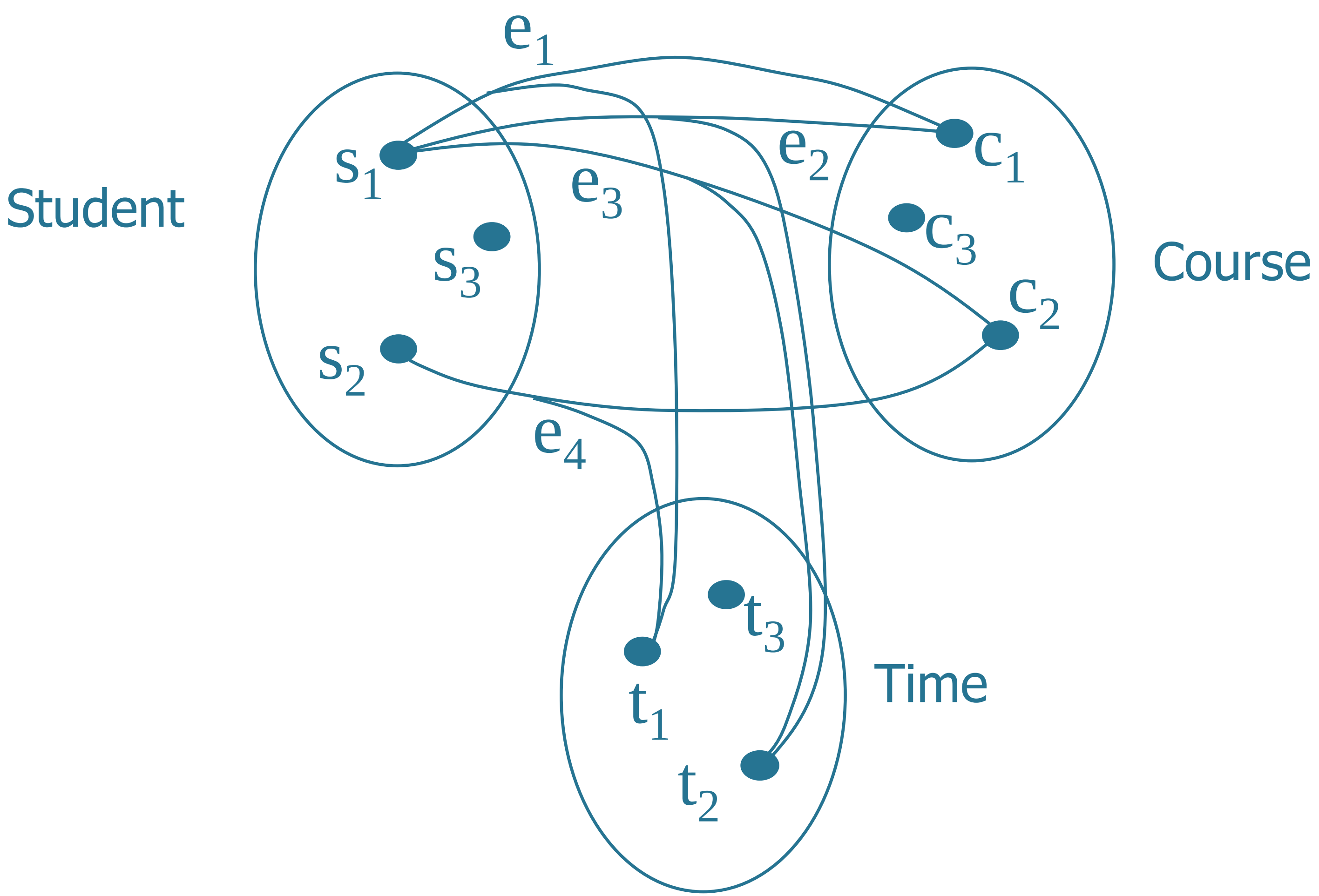
Occurences of a ternary relationship



EXAM occurrences

s_1	c_1	t_1
s_1	c_1	t_2
s_1	c_2	t_2

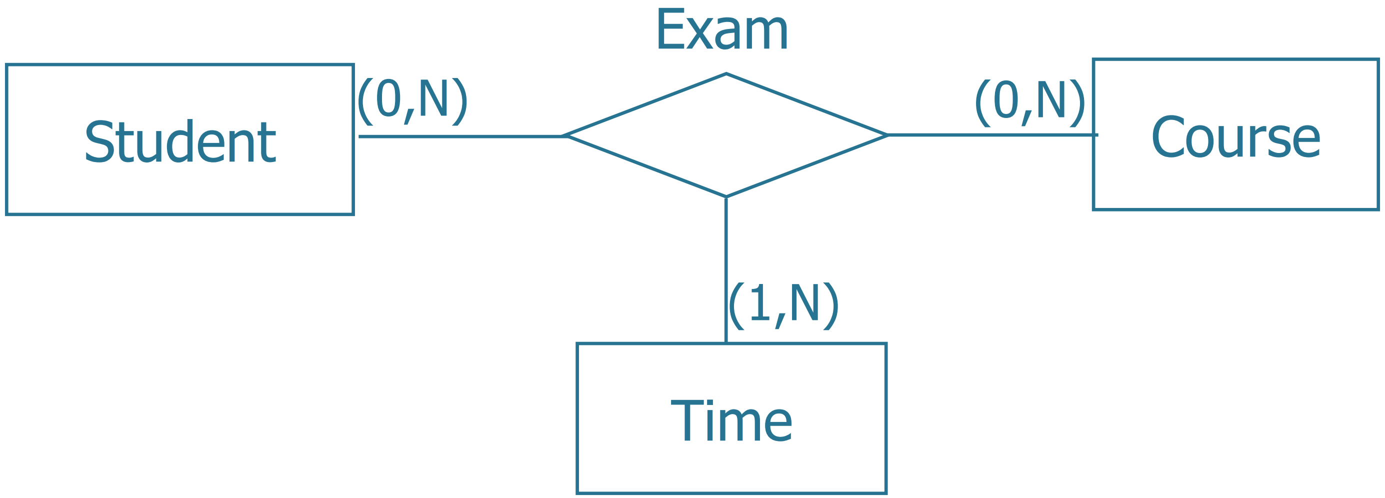
Occurences of a ternary relationship



EXAM occurrences

s_1	c_1	t_1
s_1	c_1	t_2
s_1	c_2	t_2
s_2	c_2	t_1

Cardinality of ternary relationships



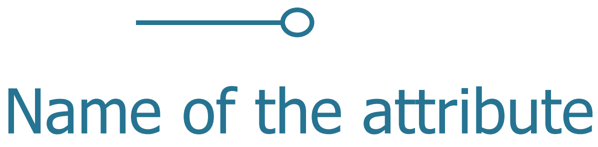
Observations

- Minimum cardinalities are rarely 1 for all entities involved in a n-ary 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

Attribute

Entity-Relationship model

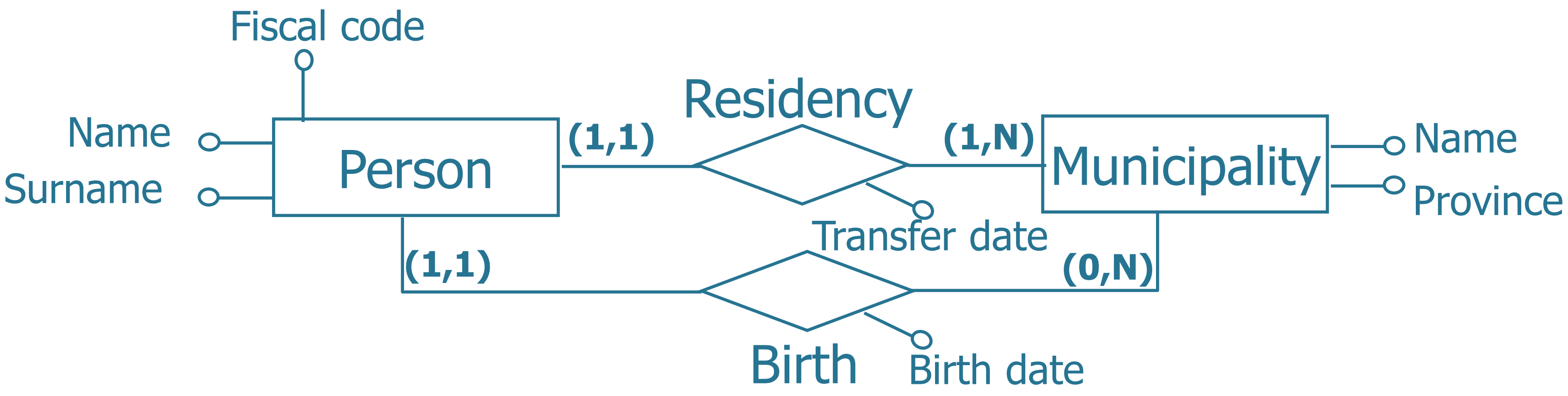
Attribute



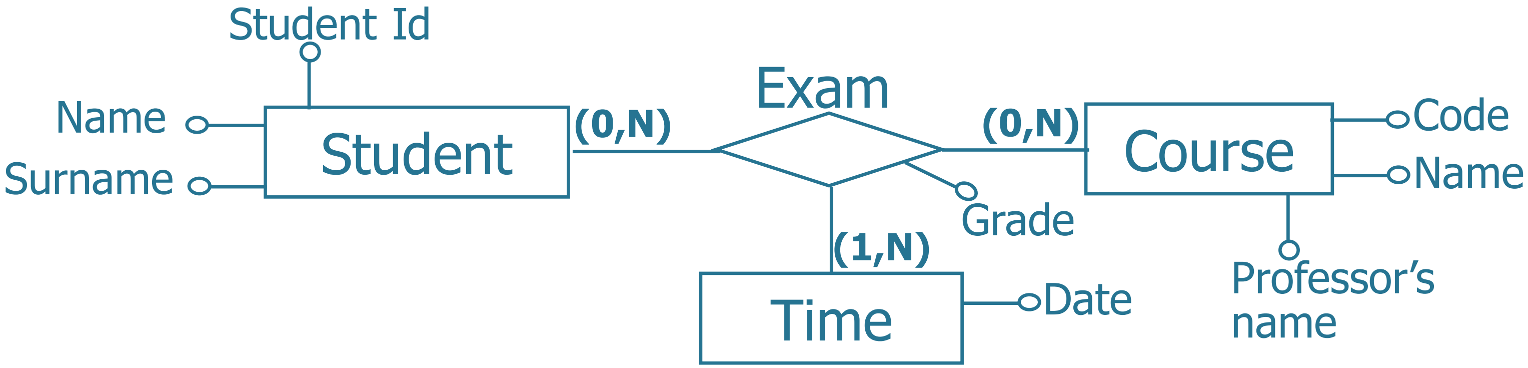
Name of the attribute

- 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.

Example of attributes

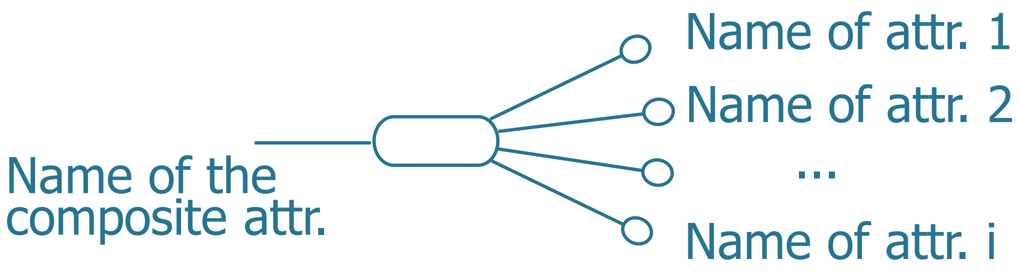


Example of attributes

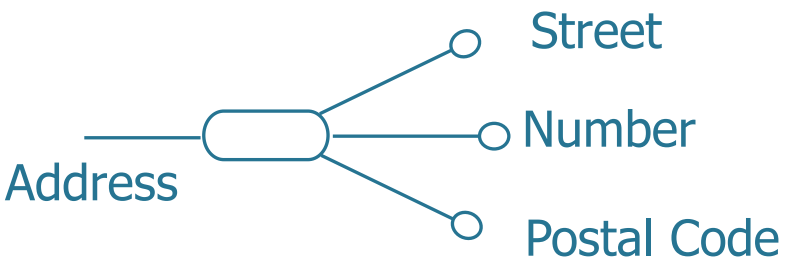


Composite attribute

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



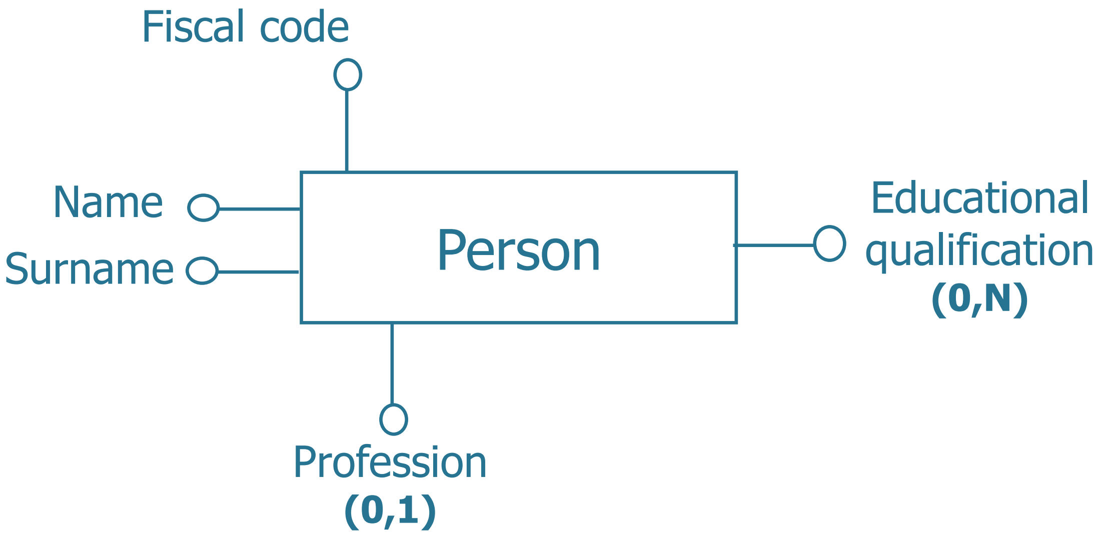
- Example



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 optional attribute, i.e., the attribute value may not be specified for some occurrences of the entity or relationship. This attribute 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



Identifier

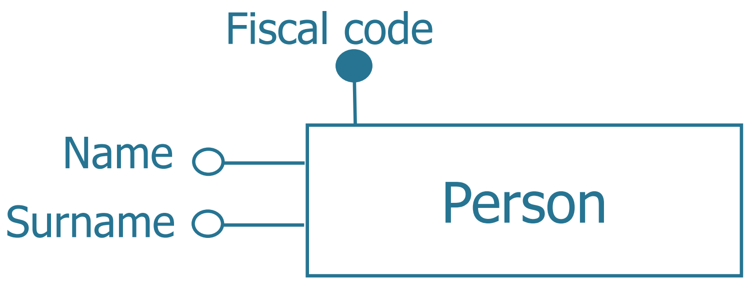
Entity-Relationship model

Identifier

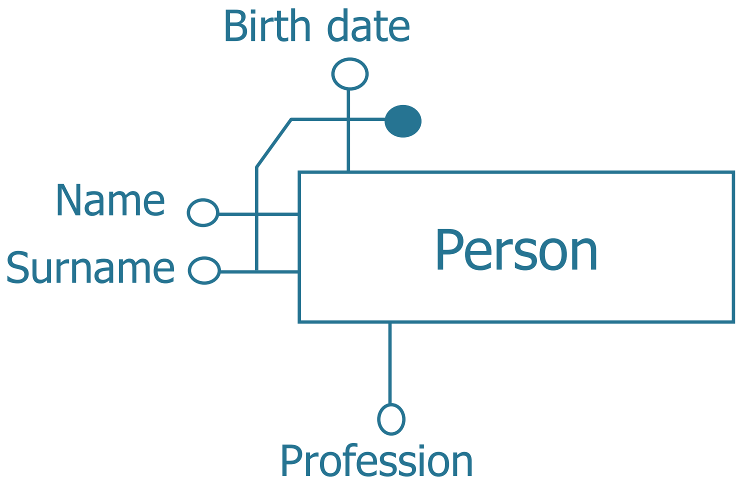
- It is specified for each entity
- It describes concepts (attributes and/or entities) of the schema that allow to identify uniquely the occurrences of an entity.
  - Each entity must have at least on identifier
  - It can exists more than one appropiated identifier for a given entity.

Internal Identifier

- Simple: consisting of one attribute

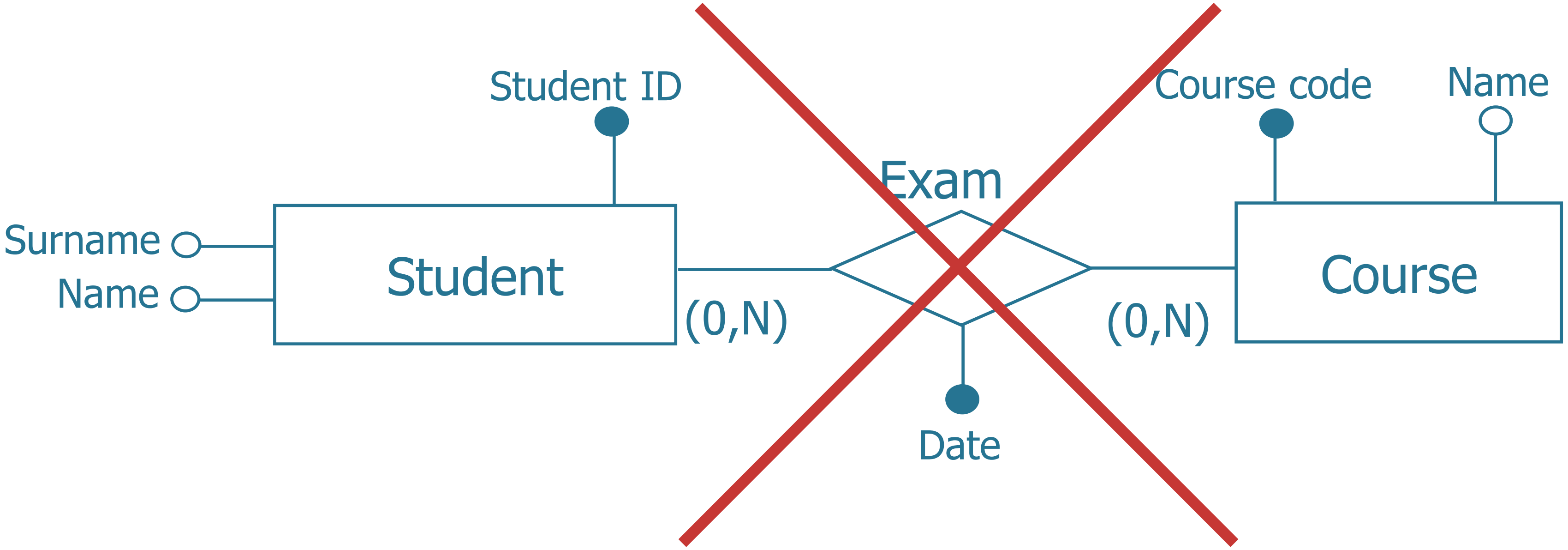


- Composite: consisting of multiple attributes

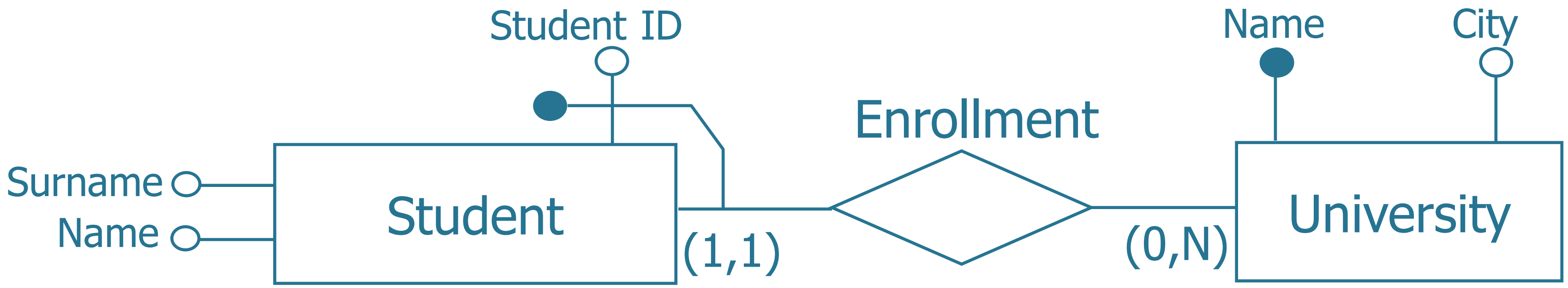


Remarks

- Relationships do *not* have identifiers



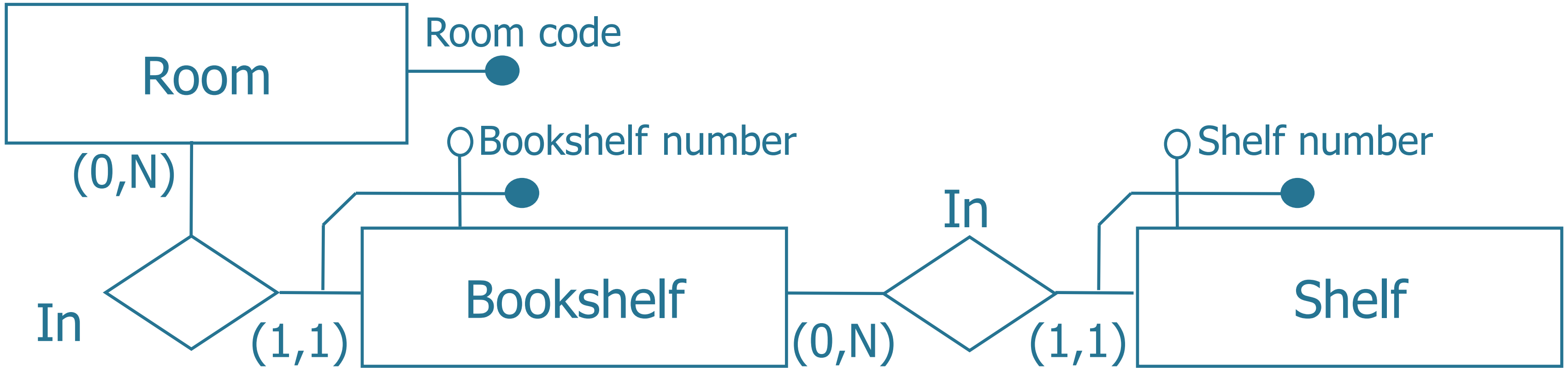
External Identifier



- The identifier for the STUDENT entity in this schema is made up of the attribute StudentID and of the UNIVERSITY entity. This is called an *external identifier*.
- One entity without sufficient internal attributes 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 its identifier.

Remarks

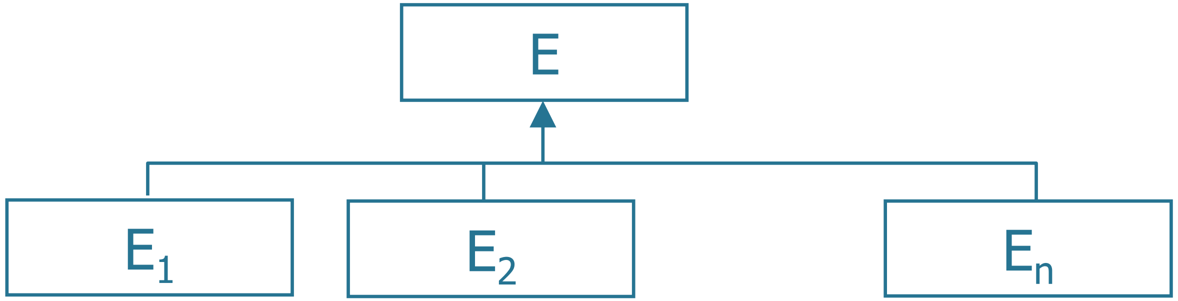
- An external identifier may involve an entity that is itself externally identified
  - Identification cycles must not be generated



Generalization

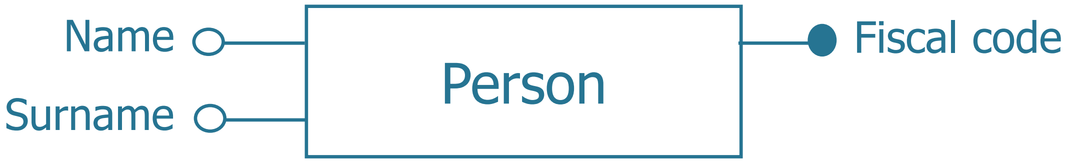
Entity-Relationship model

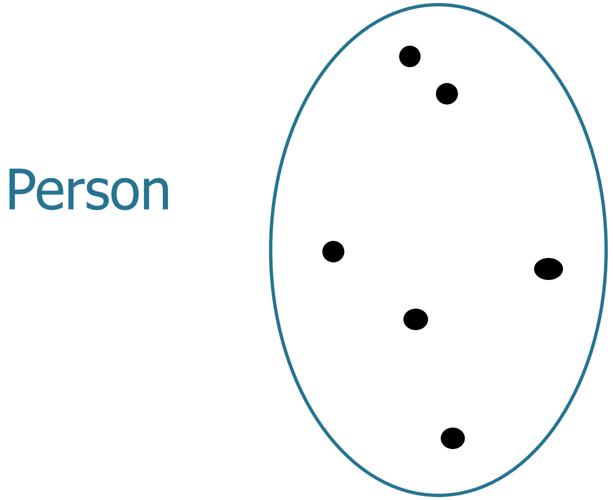
Generalization



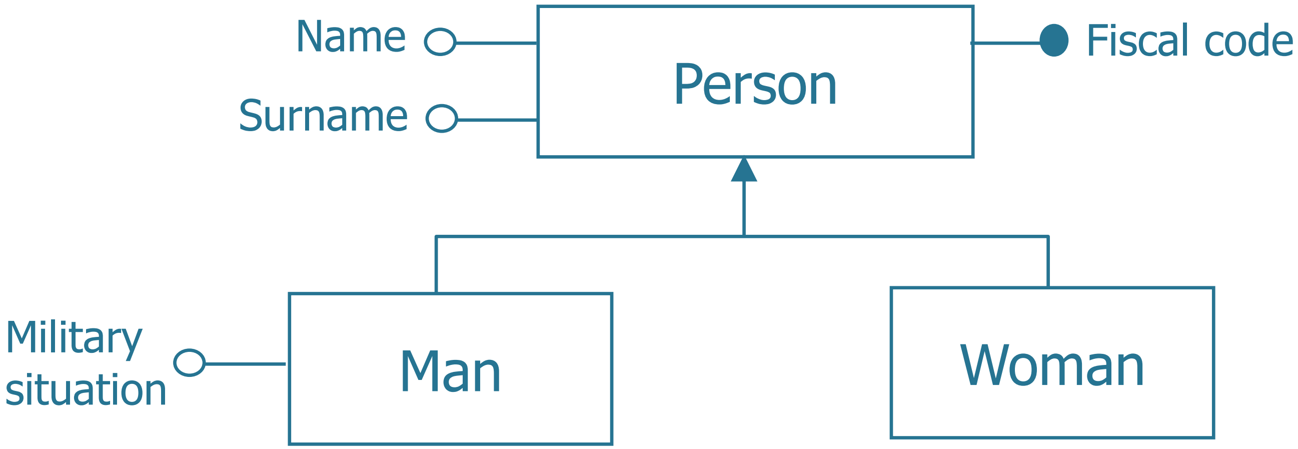
- It describes a logical link between an entity E and one or more entities E_1, E_2,…, E_n, that are particular cases of E.
    - E is called parent entity, is a generalization of E_1, E_2,…, E_n
    - E_1, E_2,…, E_n are called child entities, are specialization of E

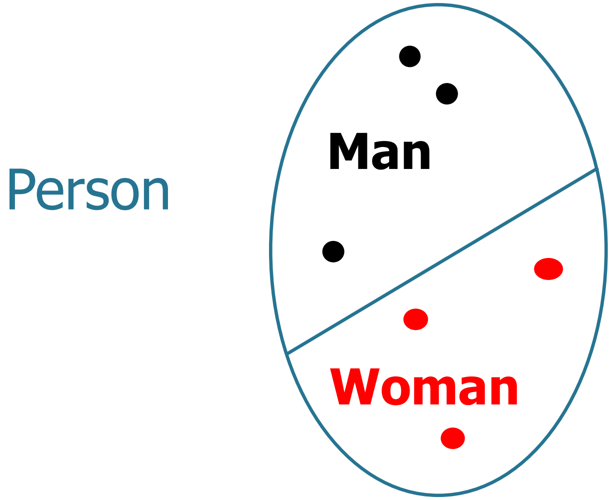
Generalization: example



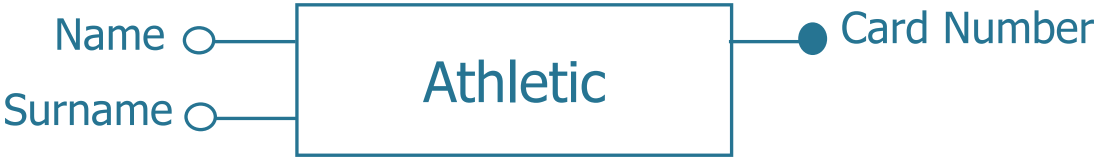


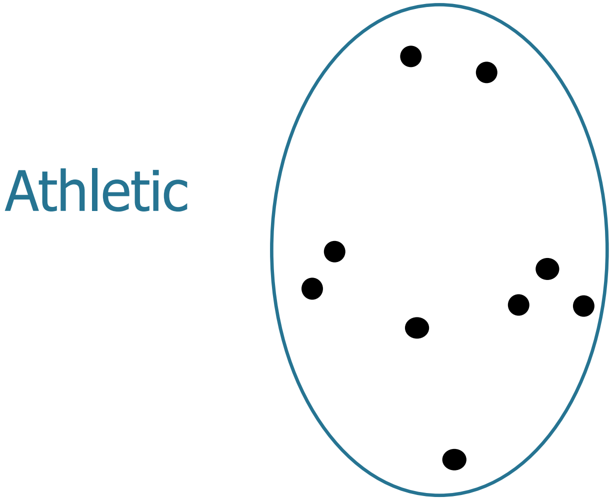
Generalization: example



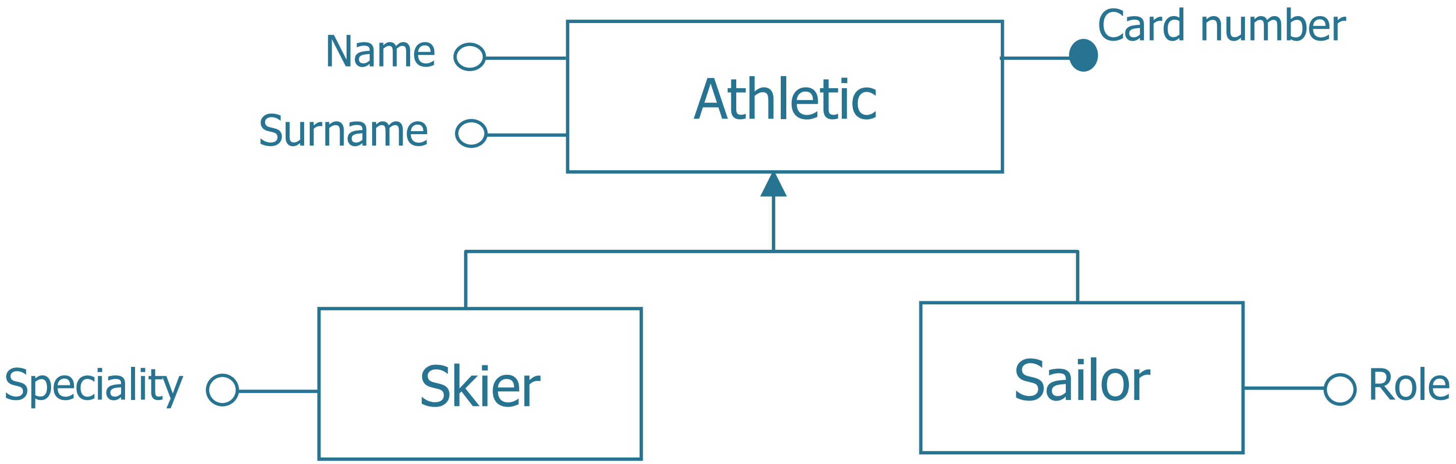


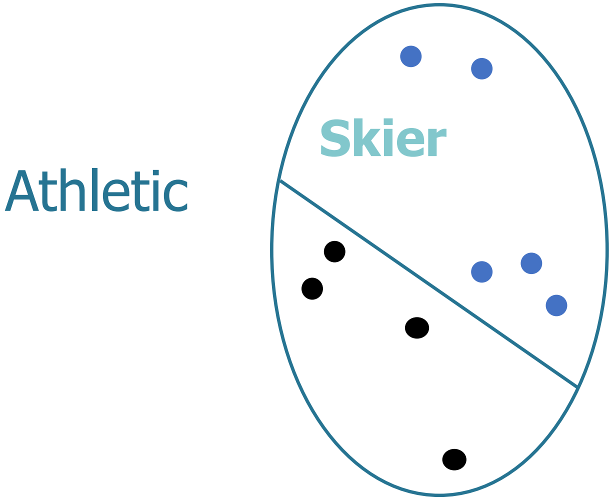
Generalization: example



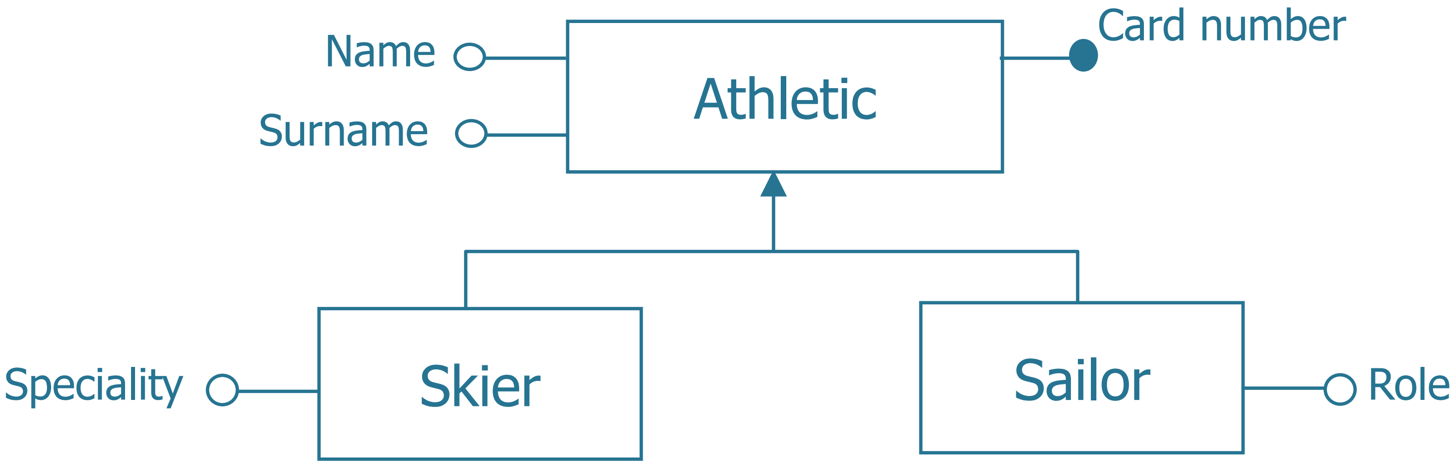


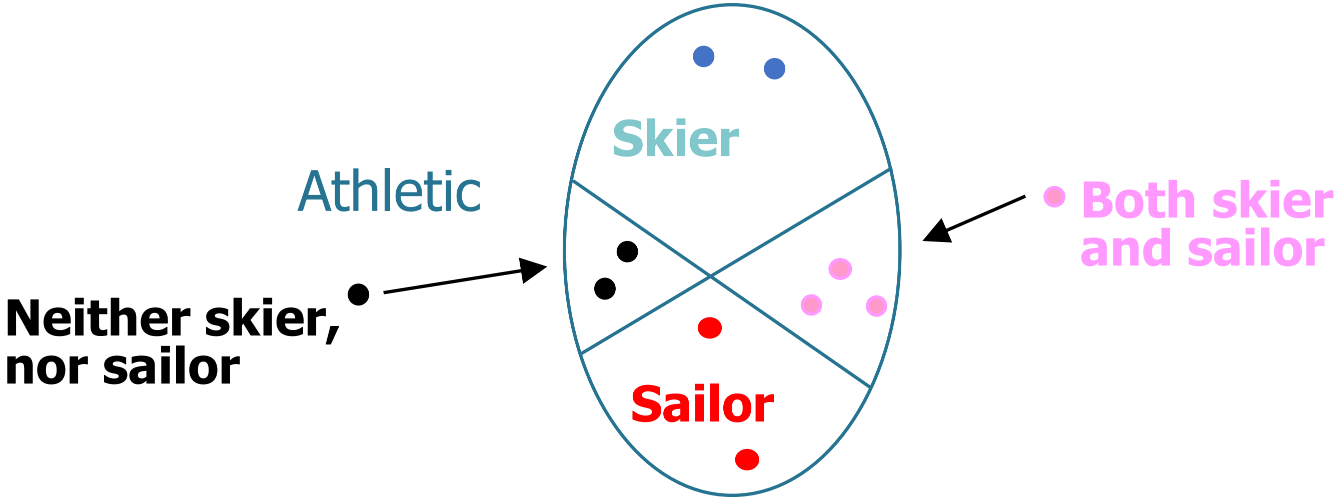
Generalization: example





Generalization: example

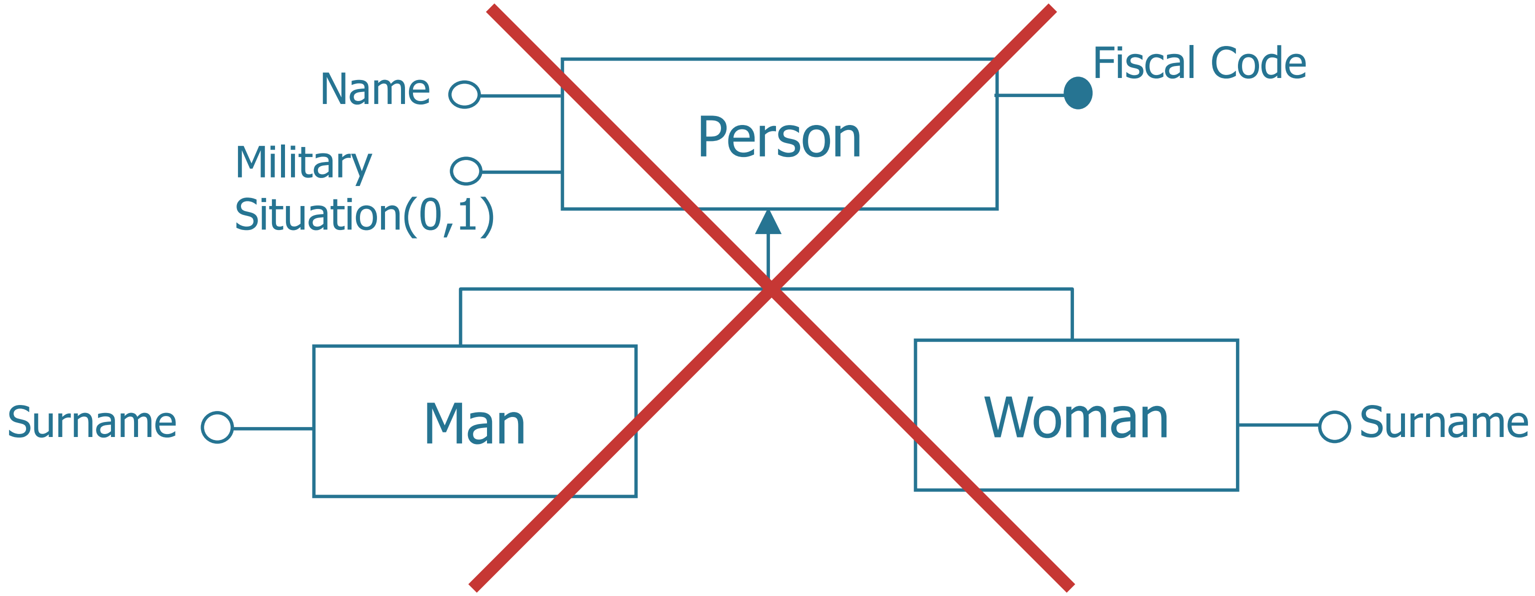




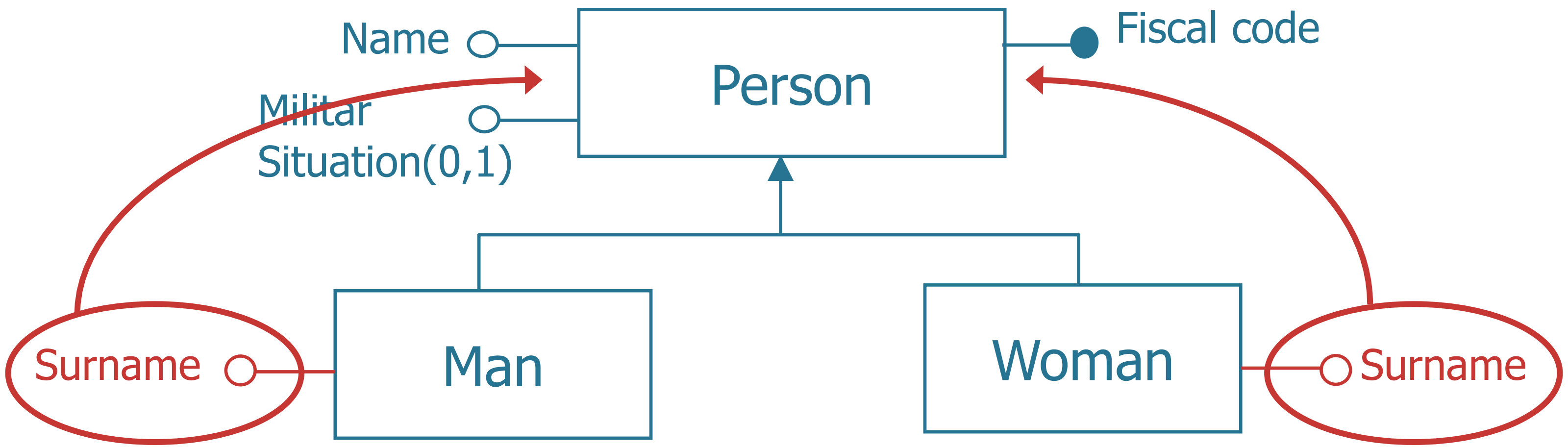
Generalization: property

- Each occurence of a child entity is also an occurence 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.

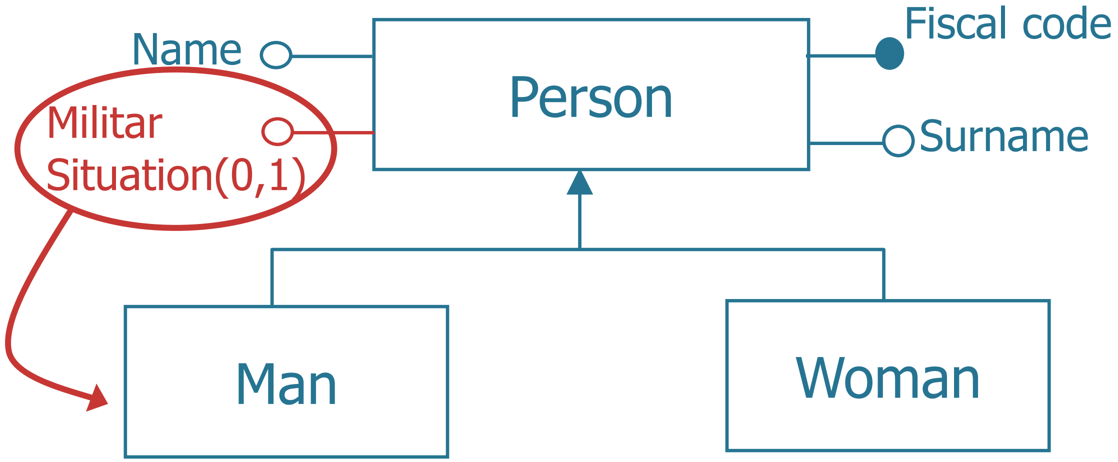
Generalizzation: incorrect example



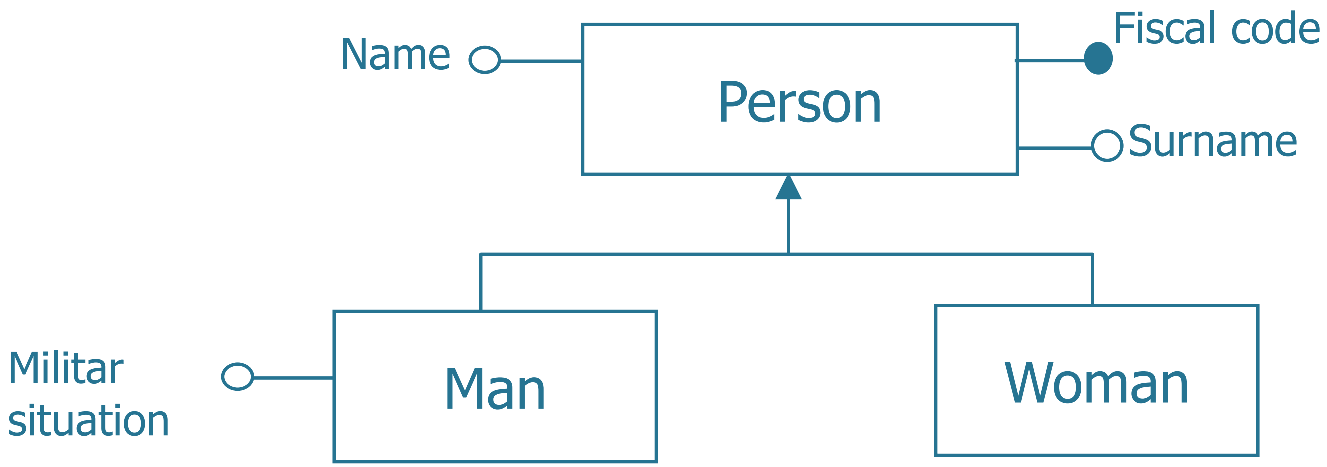
Generalizzation: incorrect example



Generalizzation: incorrect example



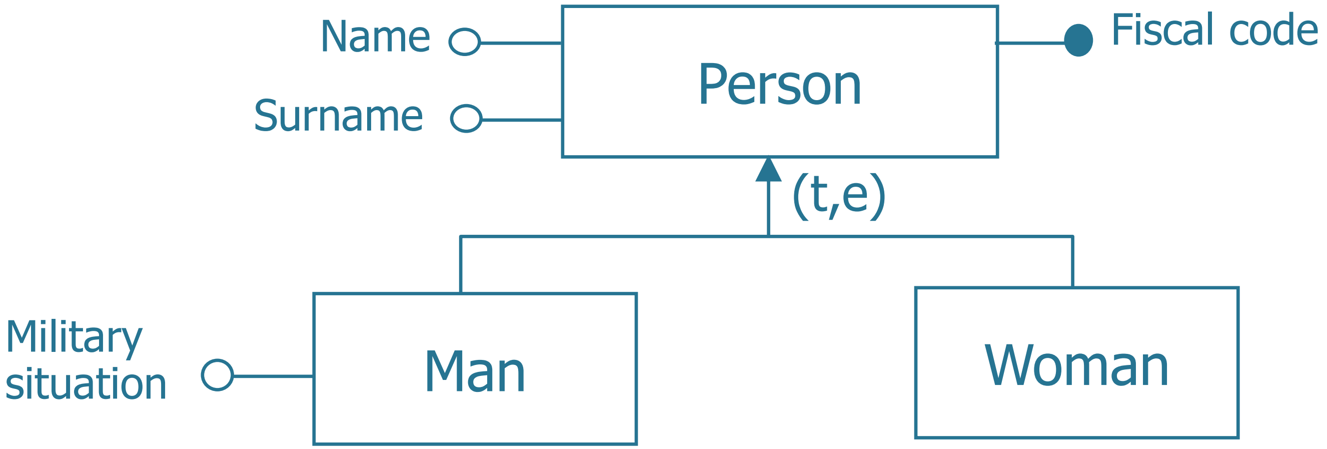
Generalizzation: correct example

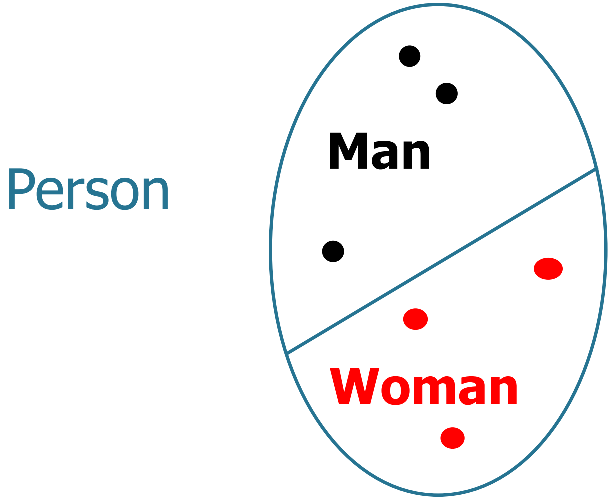


Generalization: property

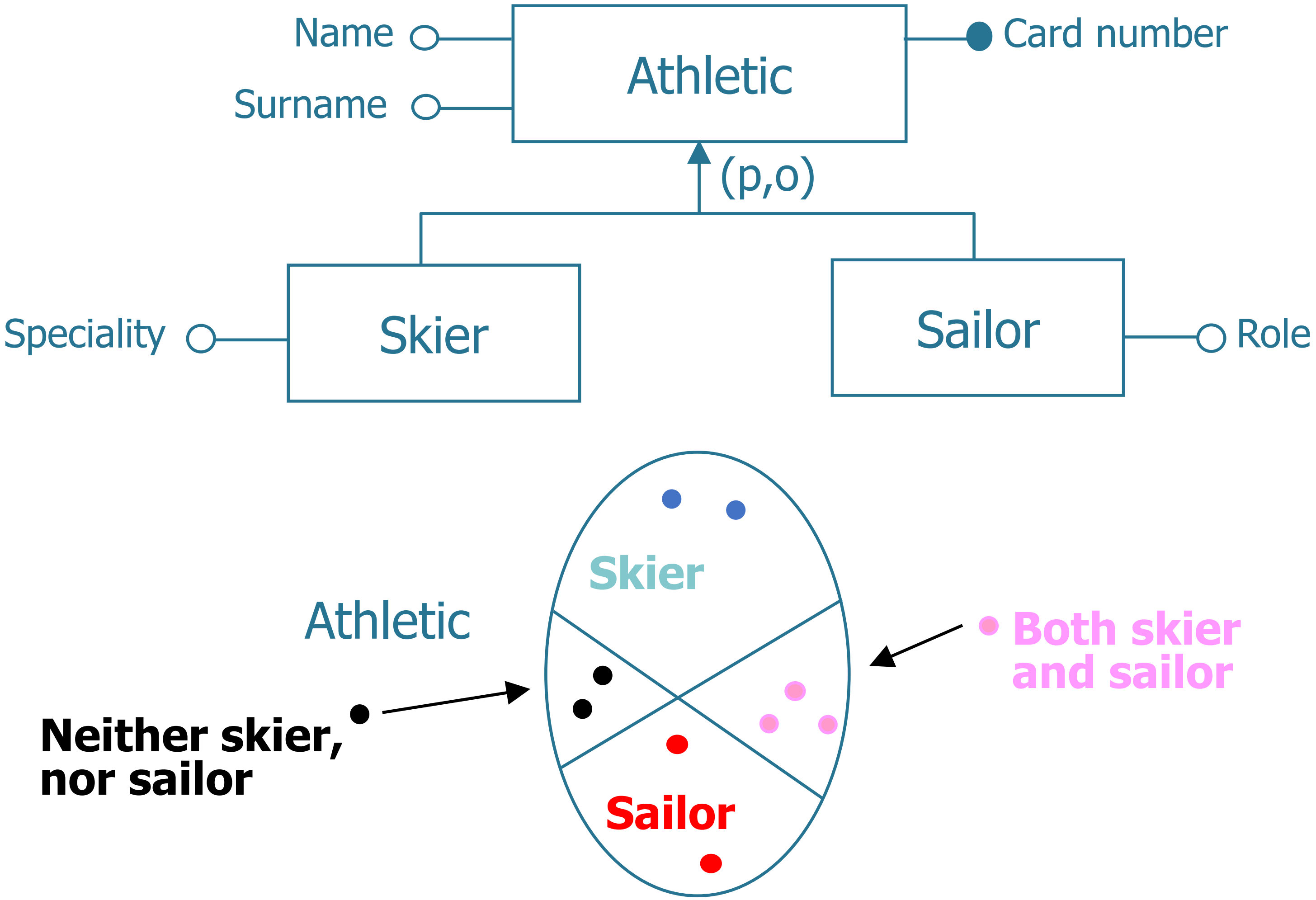
- Orthogonal characteristics
  - *total* generalization if each occurrence of the parent entity is an occurrence of at least one of the child entities, *partial* otherwise.
  - *exclusive* if each occurrence of the parent entity is at most one occurrence of one of the child entities, *overlapping* otherwise.

Generalization: example



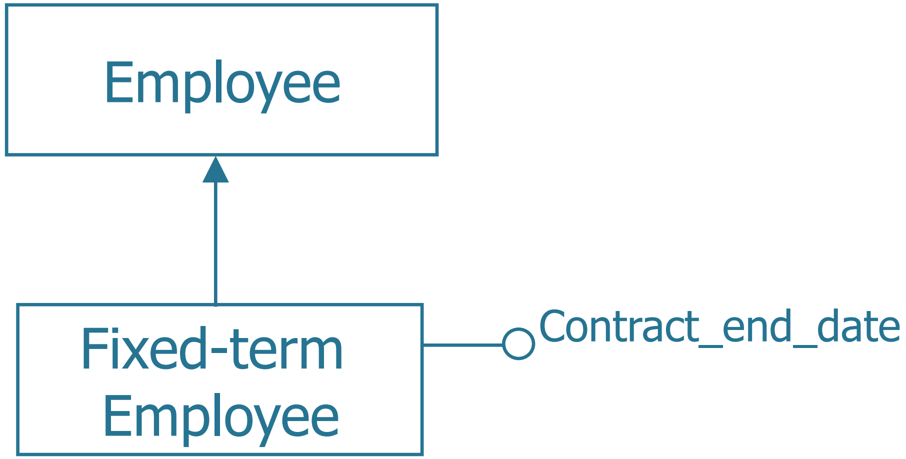


Generalization: example



Subset

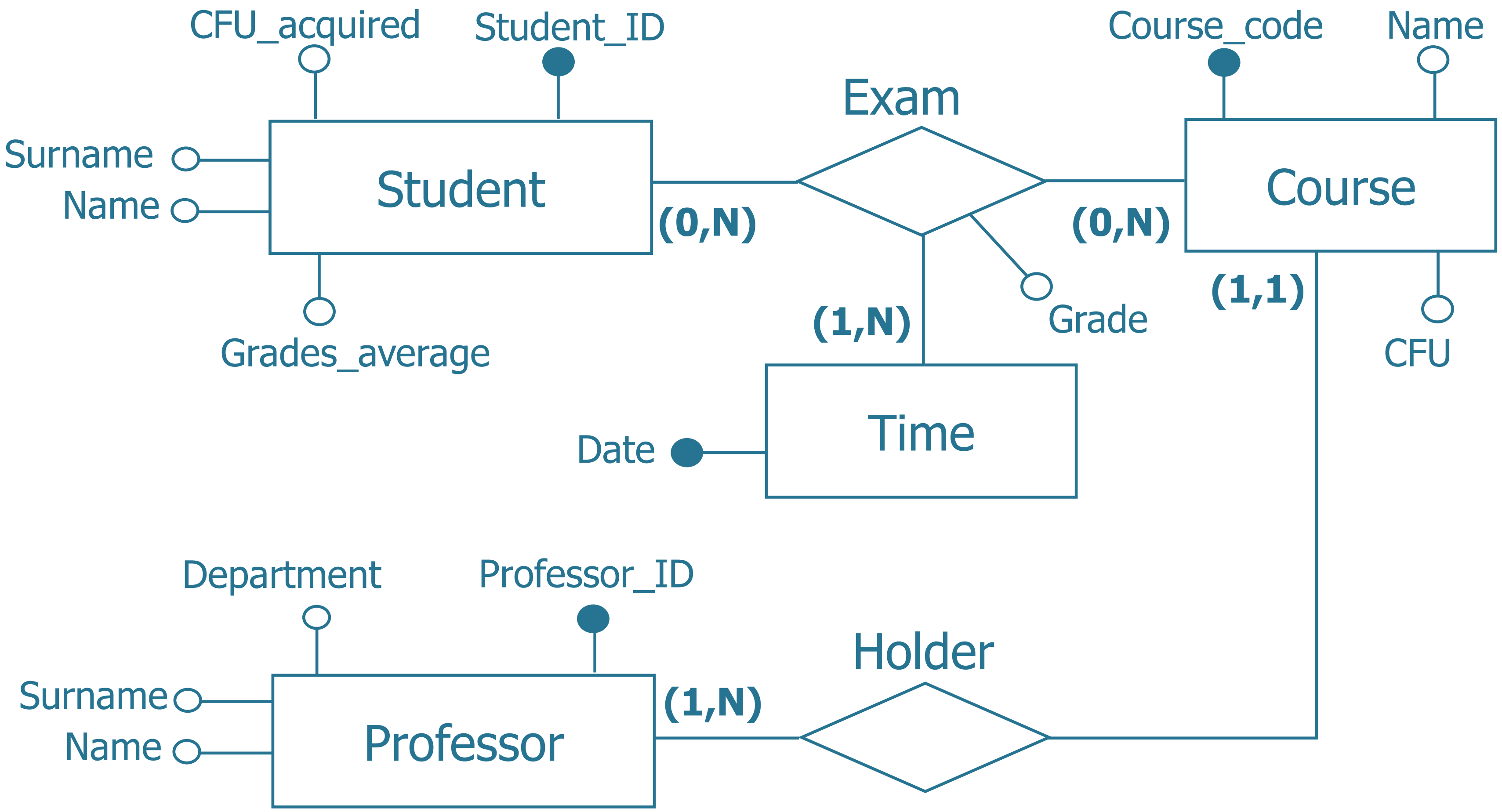
- Particular case of generalization with only one child entity
  - the generalization is always partial and exclusive.



Documentation of E-R schema

Entity-Relationship model

Documentation of E-R schema



Documentation of E-R schema

- Data Dictionary
  - allows to enrich the E-R schema 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 schema

- Data Dictionary
  - It allows to enrich the E-R schema with natural language description of entities, relationships and attributes
- Intregrity constraints on data
  - They may not always be explicitly indicated in an E-R schema
  - They can be described in natural language
- Rules for deriving data
  - To explain that a schema concept can be obtained (by inference or arithmetic calculation) from other schema concepts

Constraints of integrity on data: example

Integrity Constraints	
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

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

UML and E-R

Entity-Relationship model

UML and E-R

- UML (Unified Modeling Language)
  - 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
  - modeling of a database
    - structural aspects of an application
  - useful constructs for the modelling of databases

UML and E-R

- Different formalisms
- The UML class diagram of an application is different from the E-R schema 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

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