



The Entity-Relationship Model

Attributes

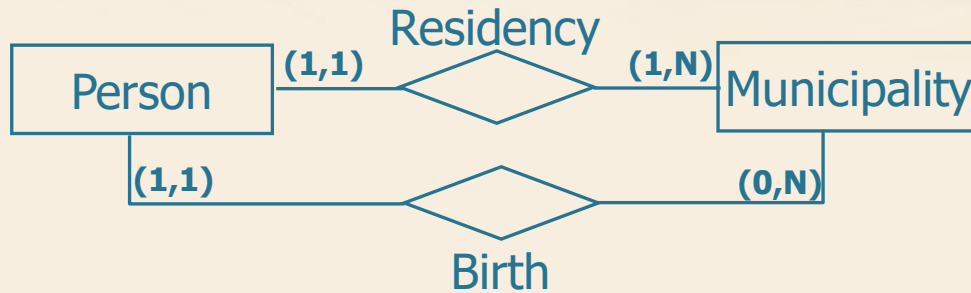
The attribute



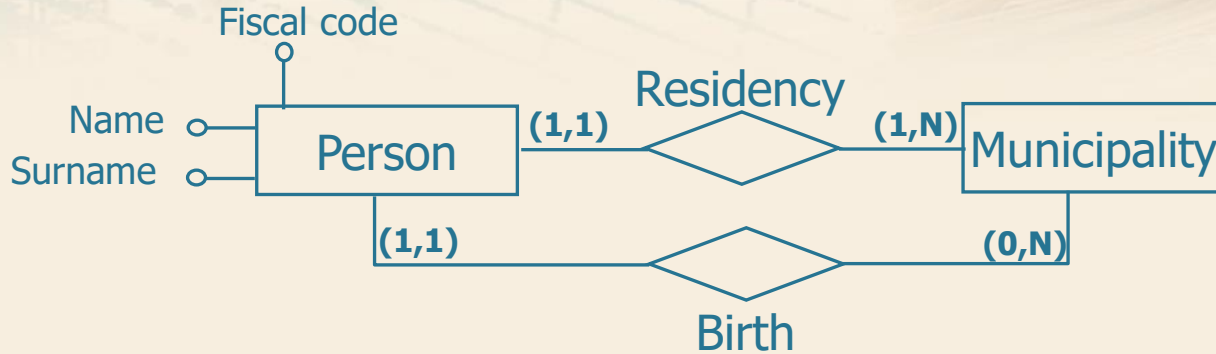
Name of the attribute

- 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.

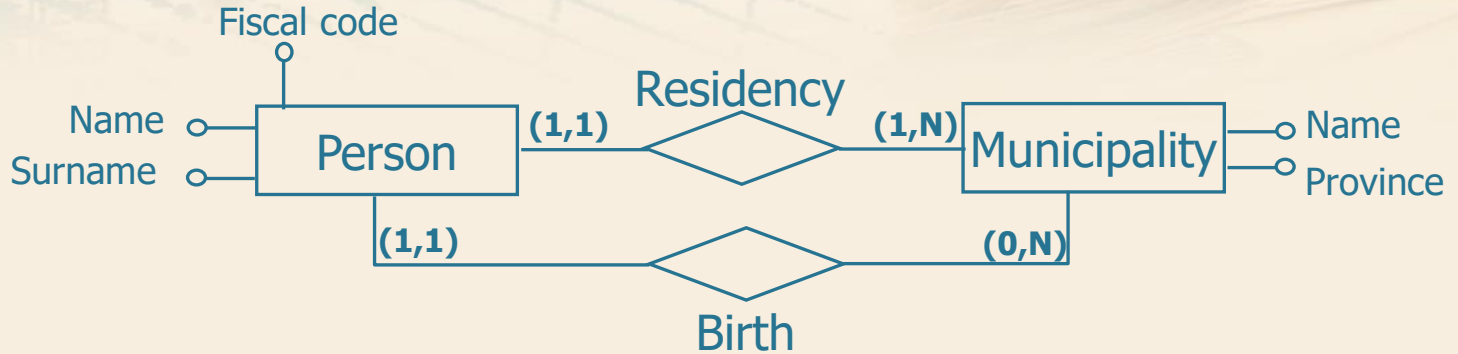
Examples of attributes



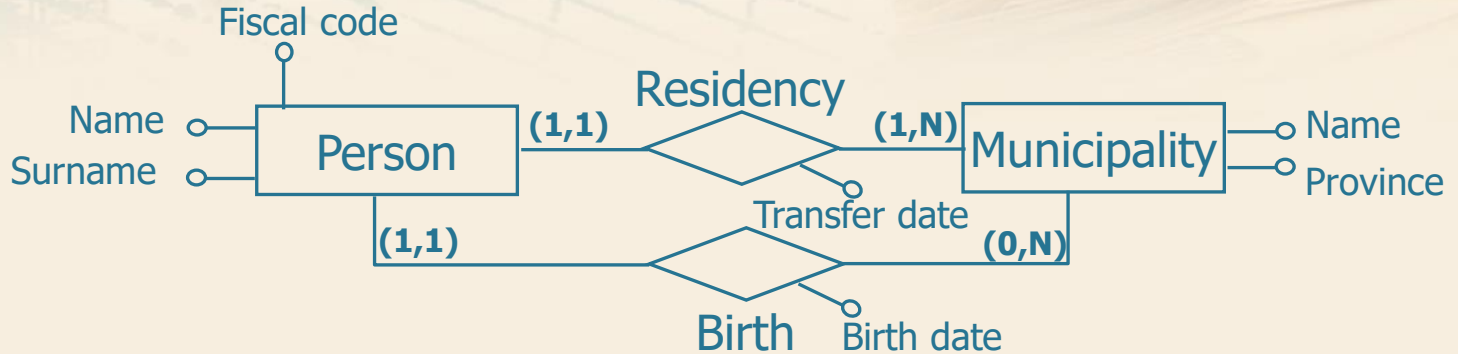
Examples of attributes



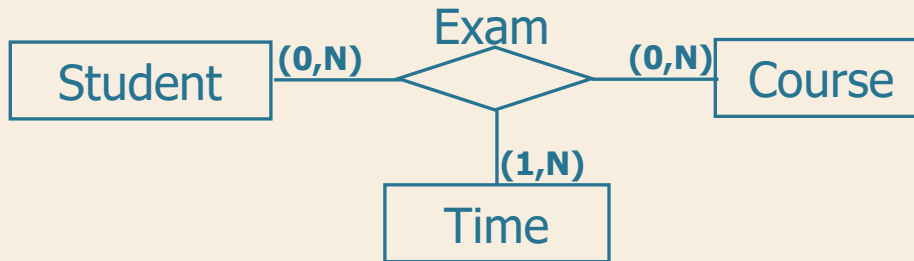
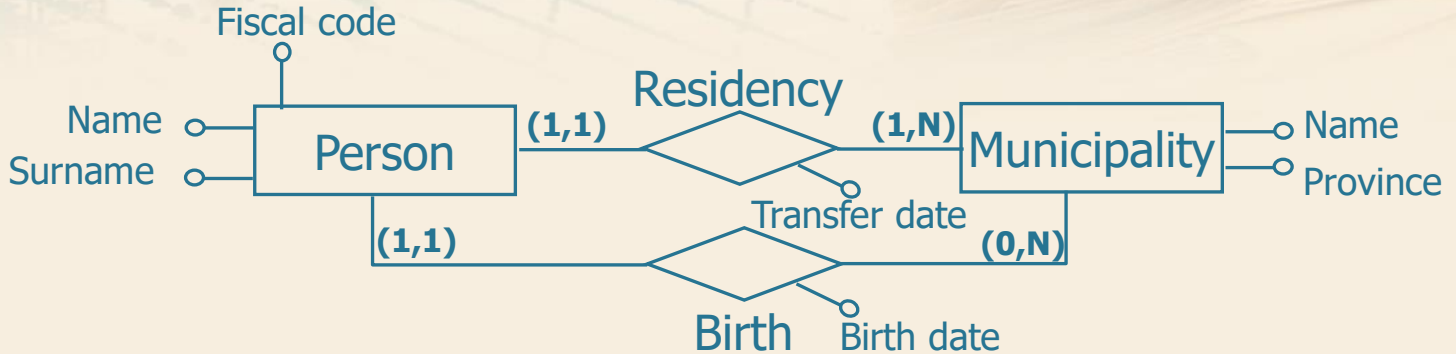
Examples of attributes



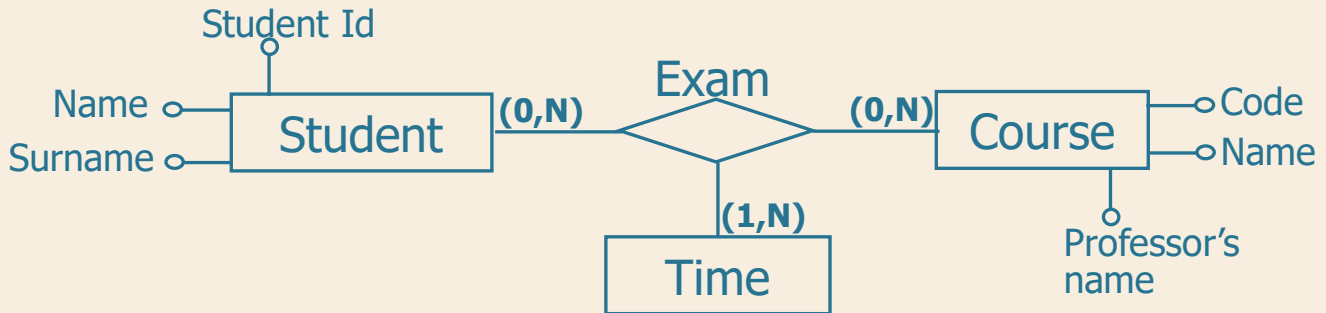
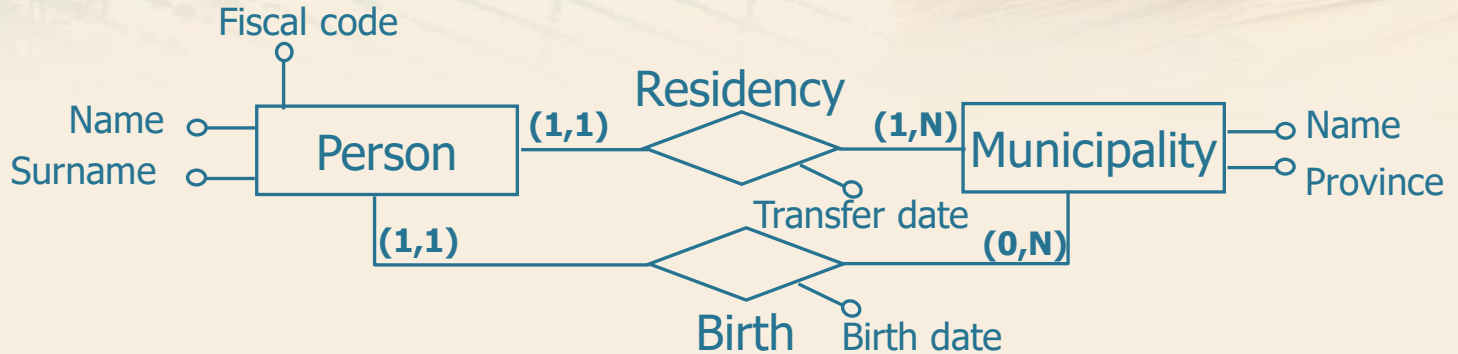
Examples of attributes



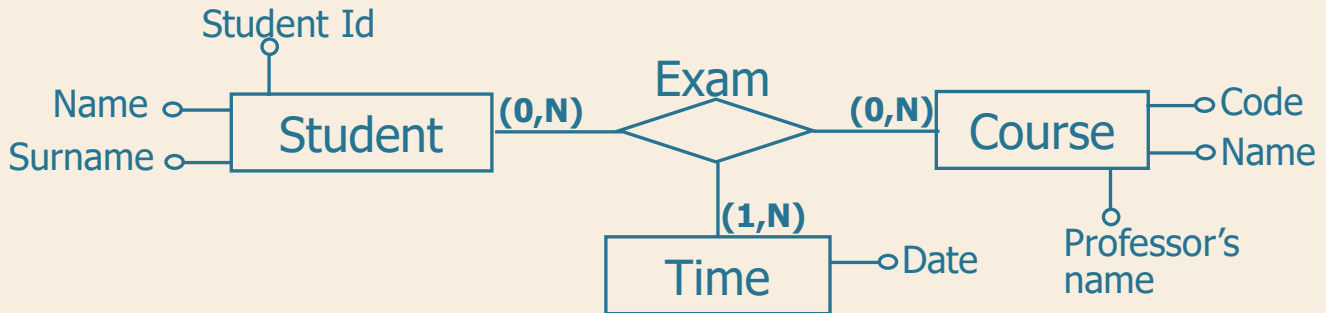
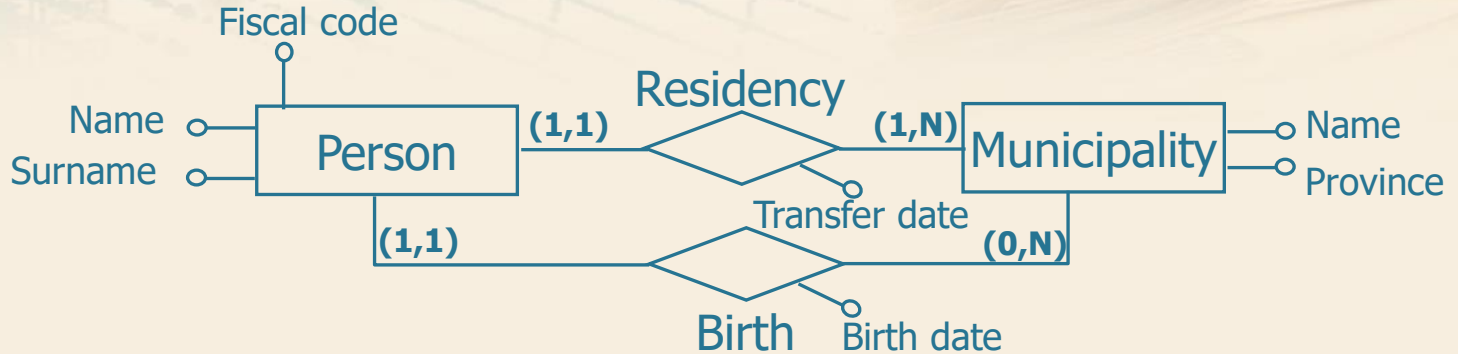
Examples of attributes



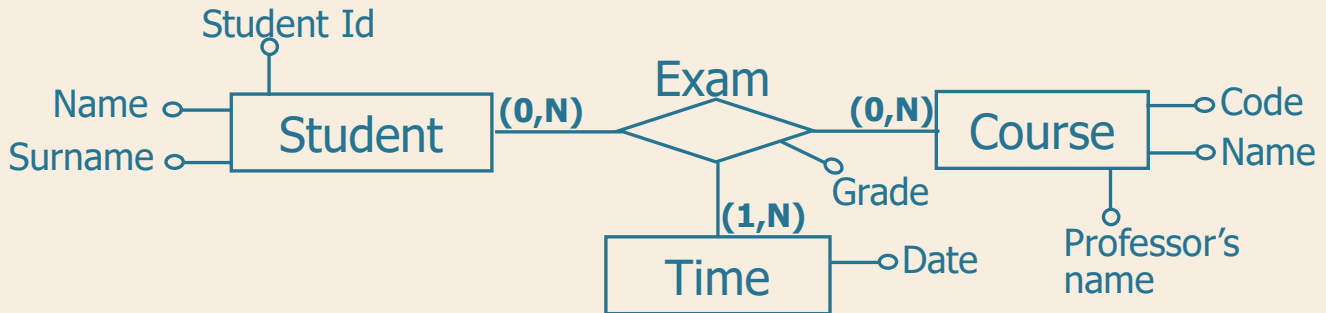
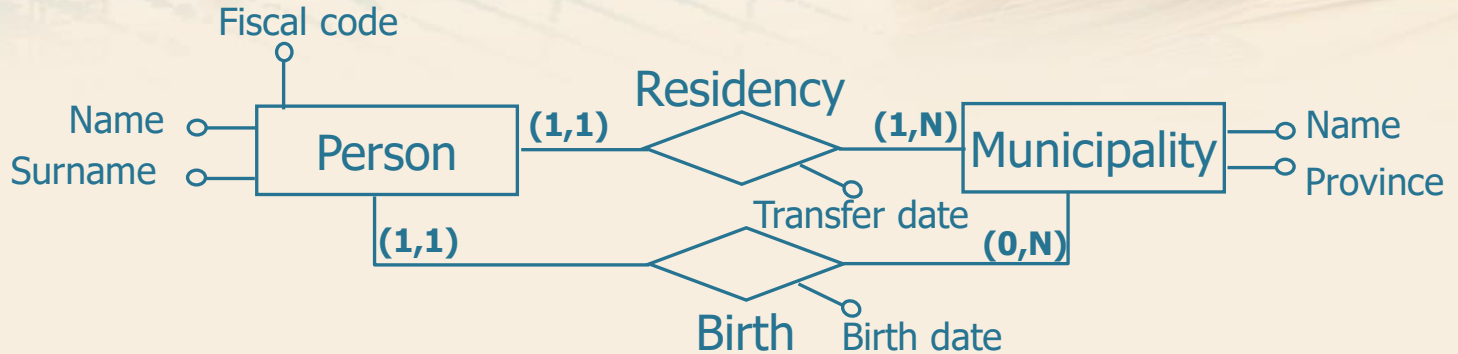
Examples of attributes



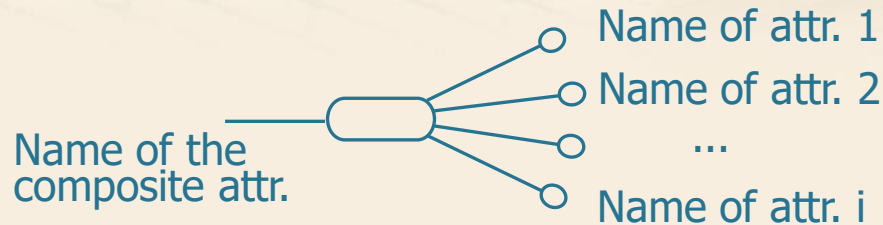
Examples of attributes



Examples of attributes

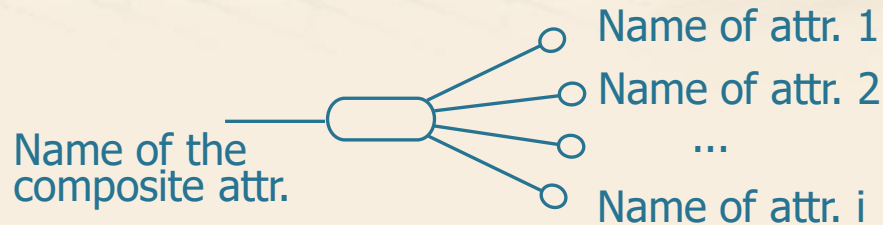


Composite attribute



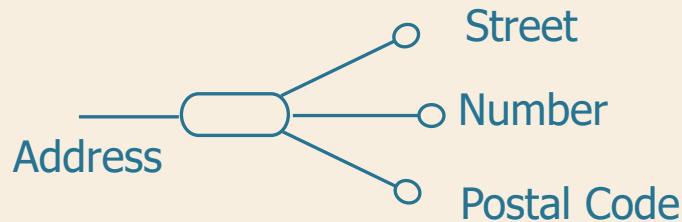
- Group of attributes that have closely connected meanings or uses.

Composite attribute



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

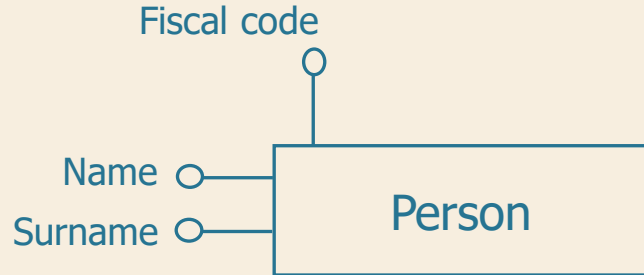
➤ Example



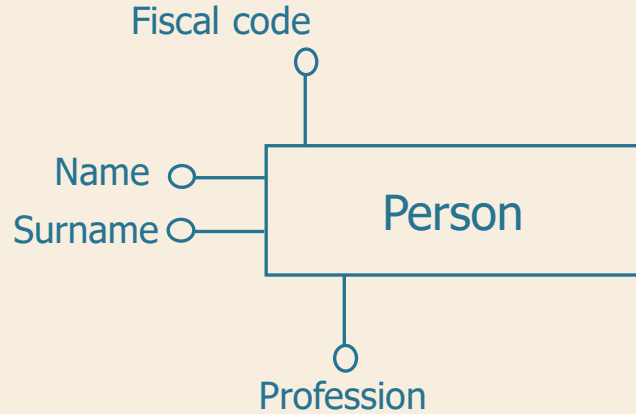
Cardinality of an attribute

- 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)

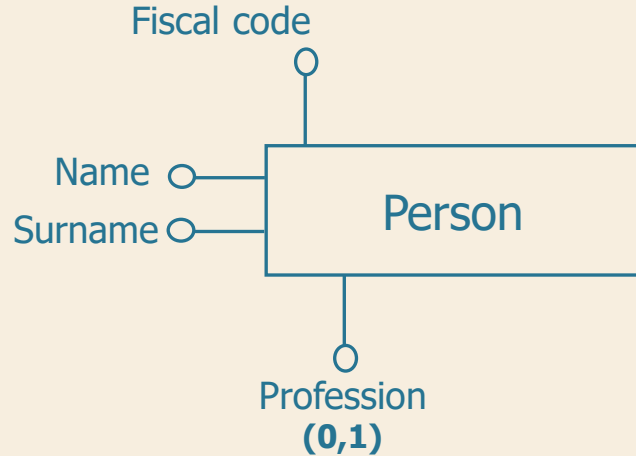
Cardinality of an attribute



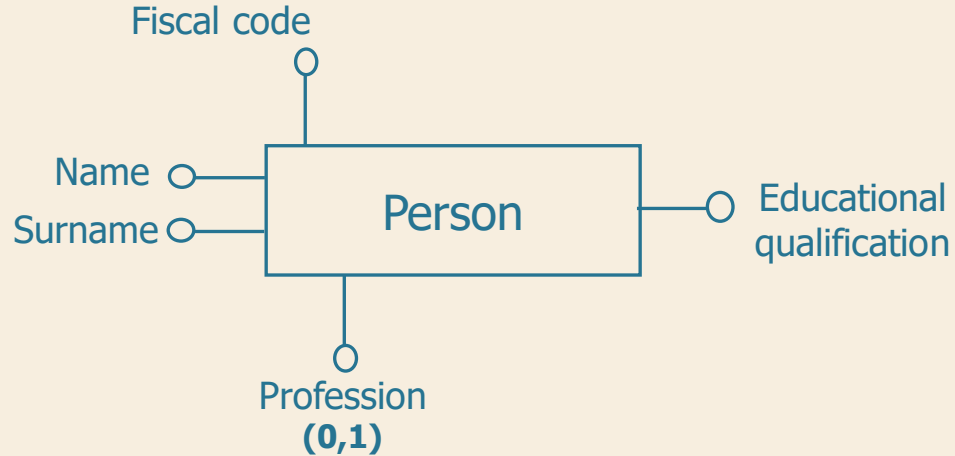
Cardinality of an attribute



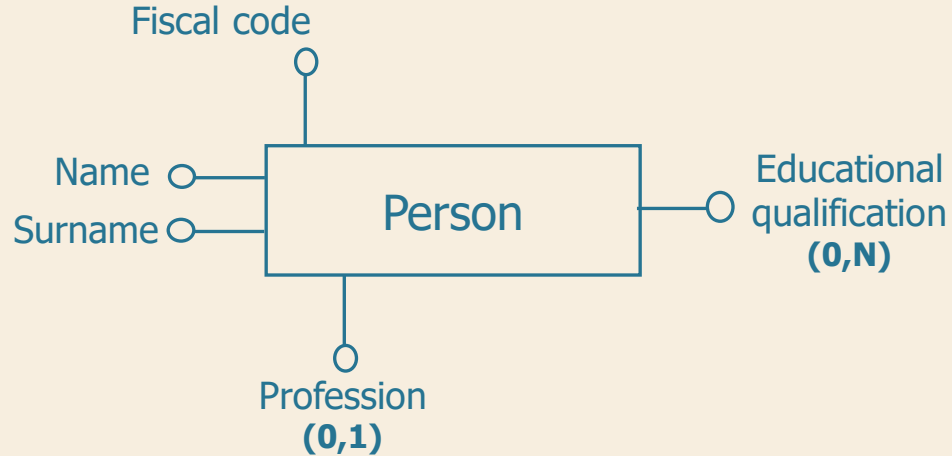
Cardinality of an attribute



Cardinality of an attribute



Cardinality of an attribute





The Entity-Relationship Model

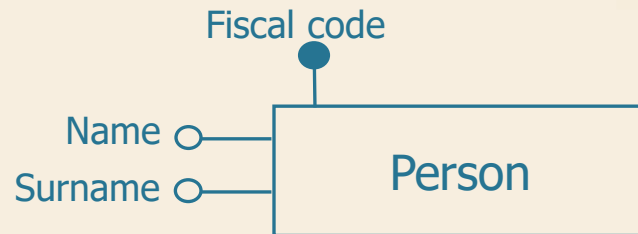
Identifiers

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
 - It can exist more than one appropriated identifier for a given entity.

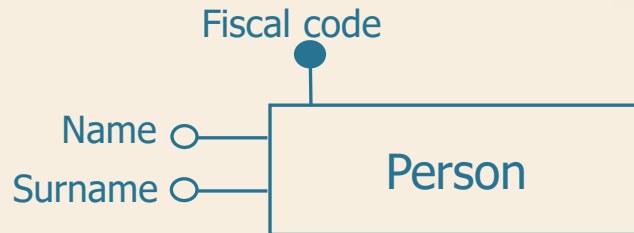
Internal Identifier

➤ Simple: consisting of only one attribute

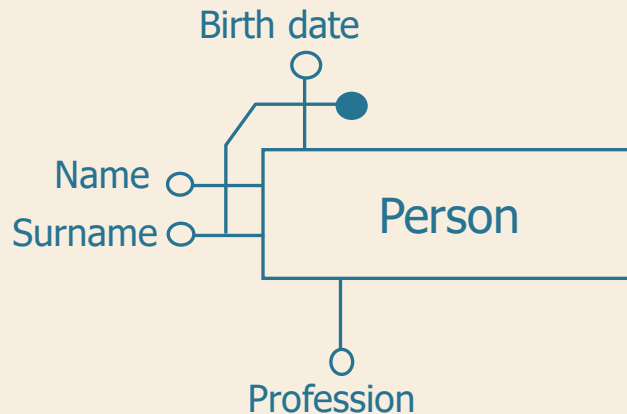


Internal Identifier

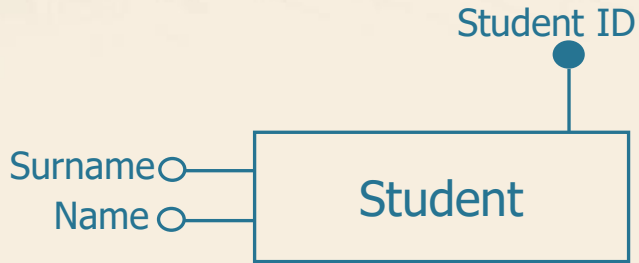
➤ Simple: consisting of only one attribute



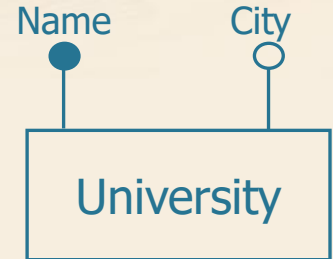
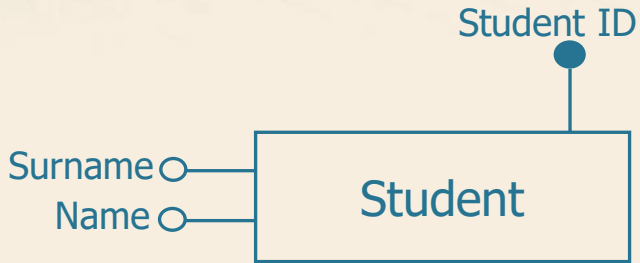
➤ Composite: consisting of multiple attributes



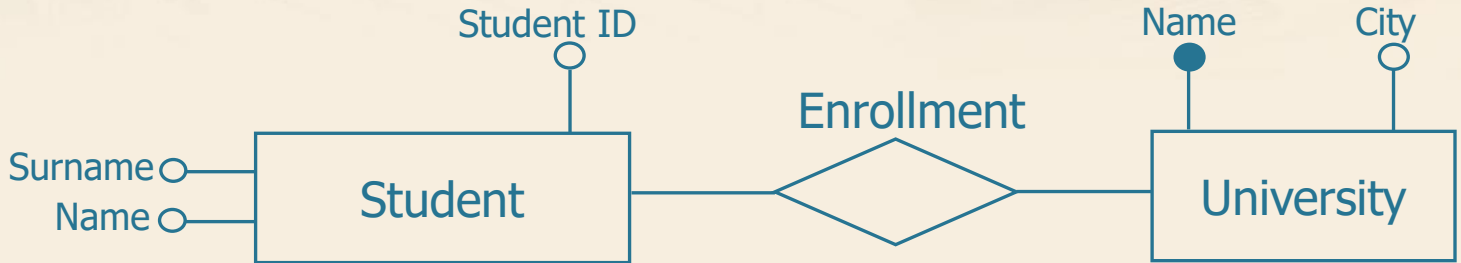
External Identifier



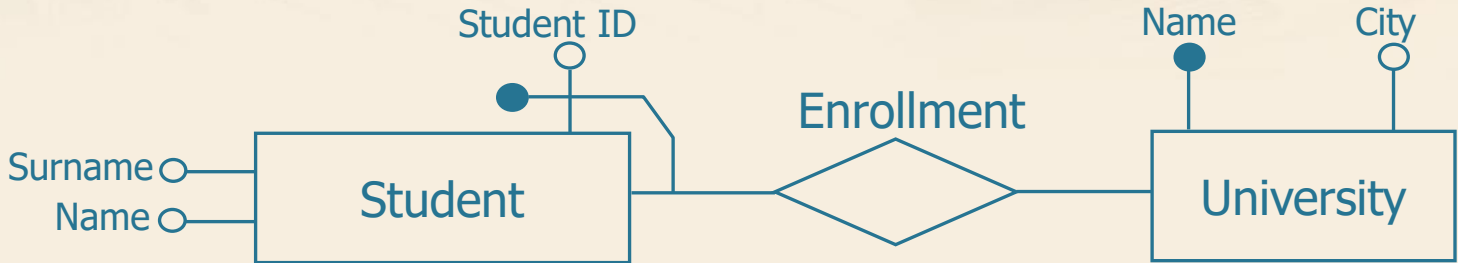
External Identifier



External Identifier

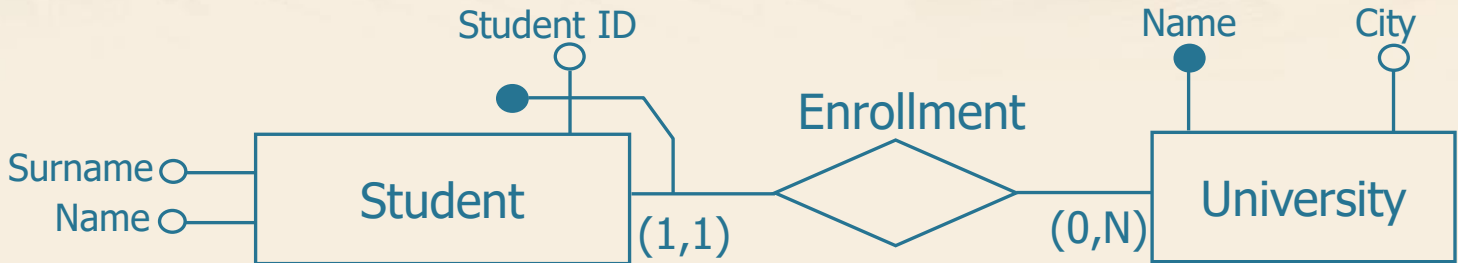


External Identifier



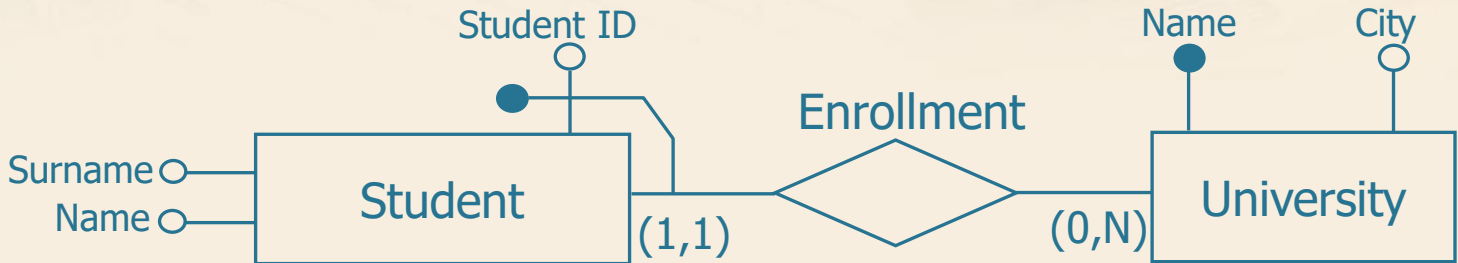
- One entity that does not have sufficient internal attributes able to define an identifier is called *weak entity*.

External Identifier



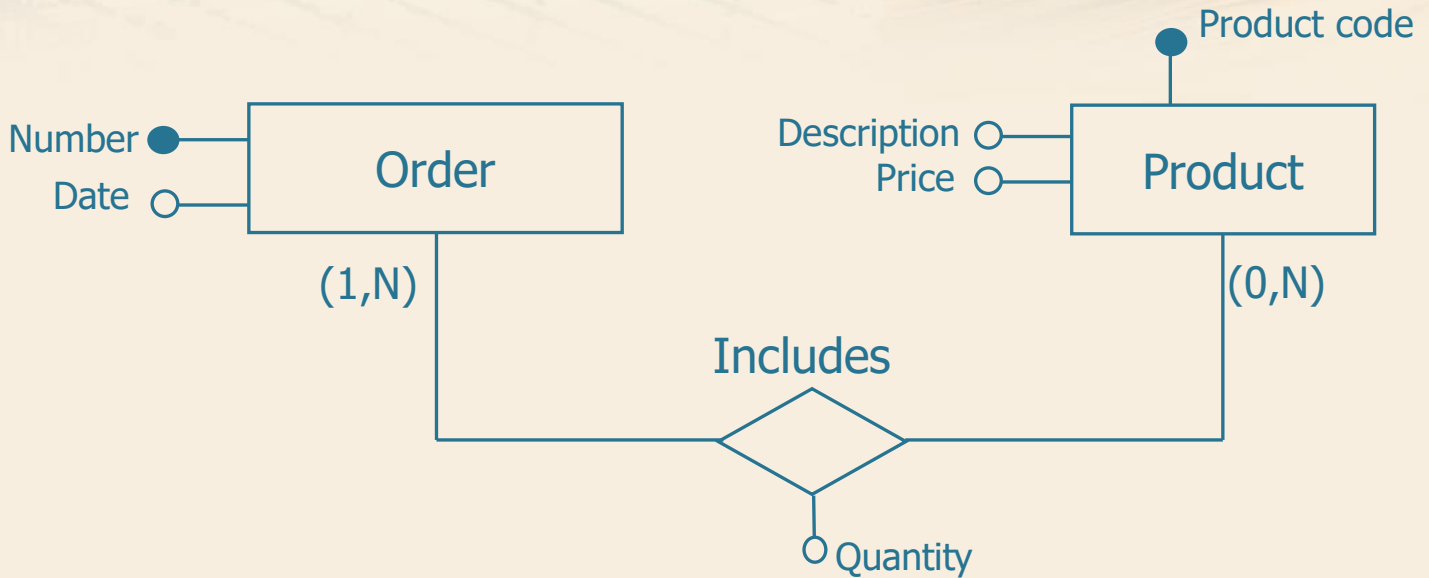
- One entity that does not have sufficient internal attributes able to define an identifier is called *weak entity*.

External Identifier

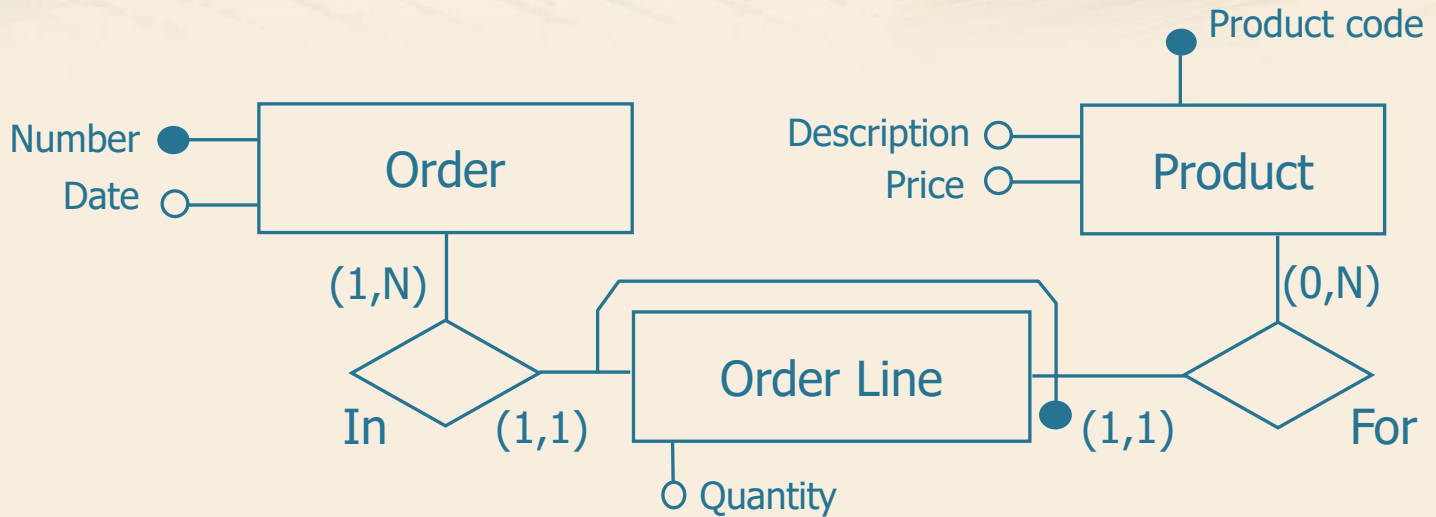


- One entity that does not have sufficient internal attributes able to define an identifier is called *weak entity*.
- A weak entity must participate with cardinality (1,1) in each of the relationships that provide part of the identifier

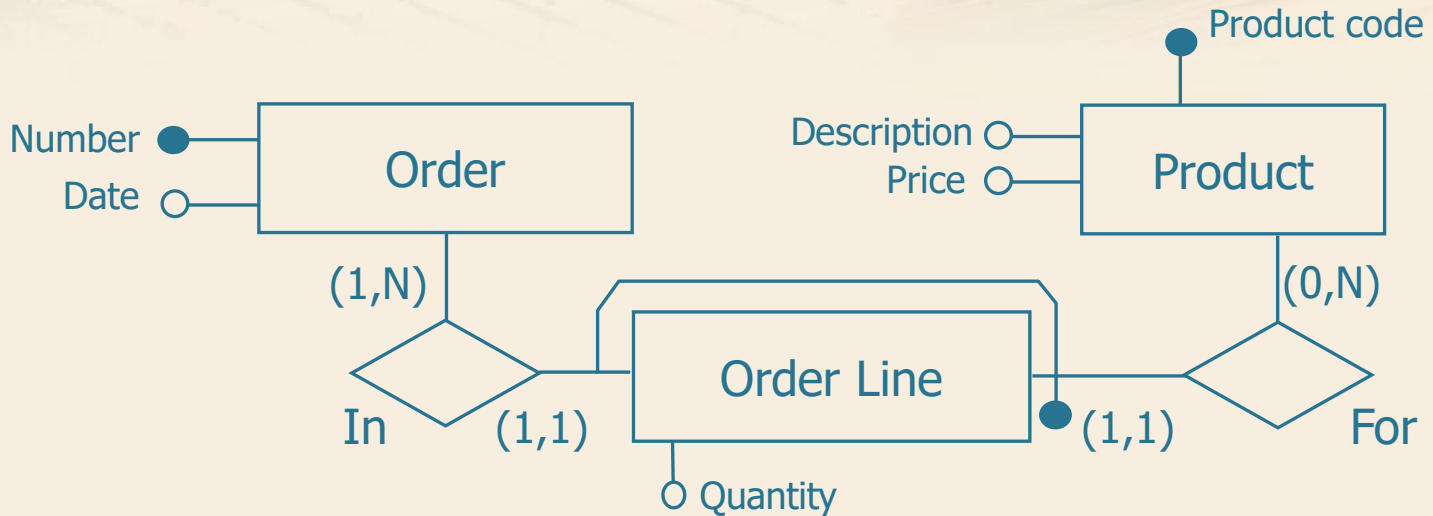
External Identifier



External Identifier

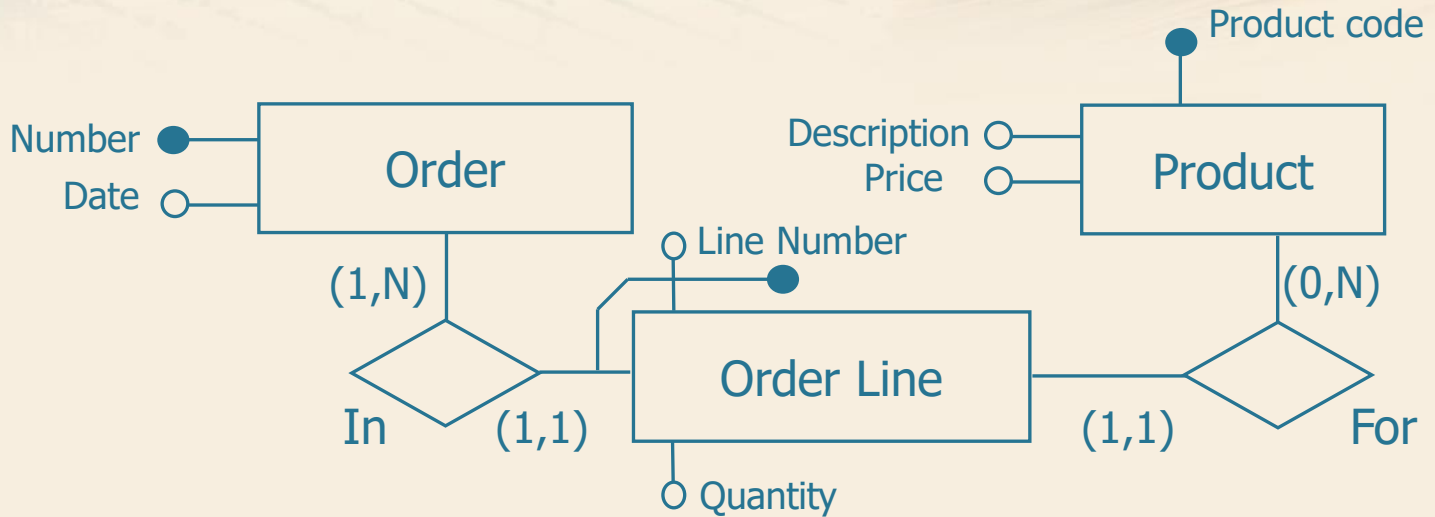


External Identifier

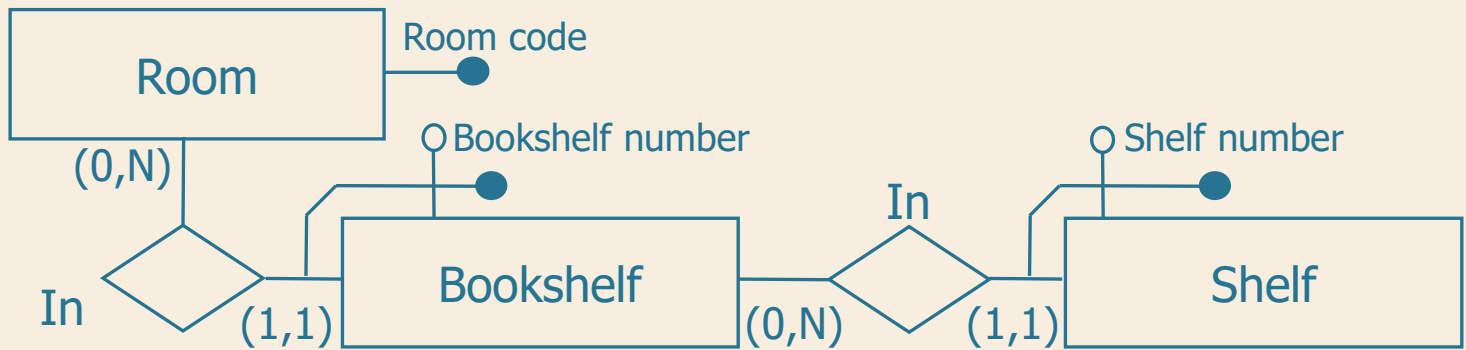


➤ Is it possible to represent in the same order more order lines for the same product?

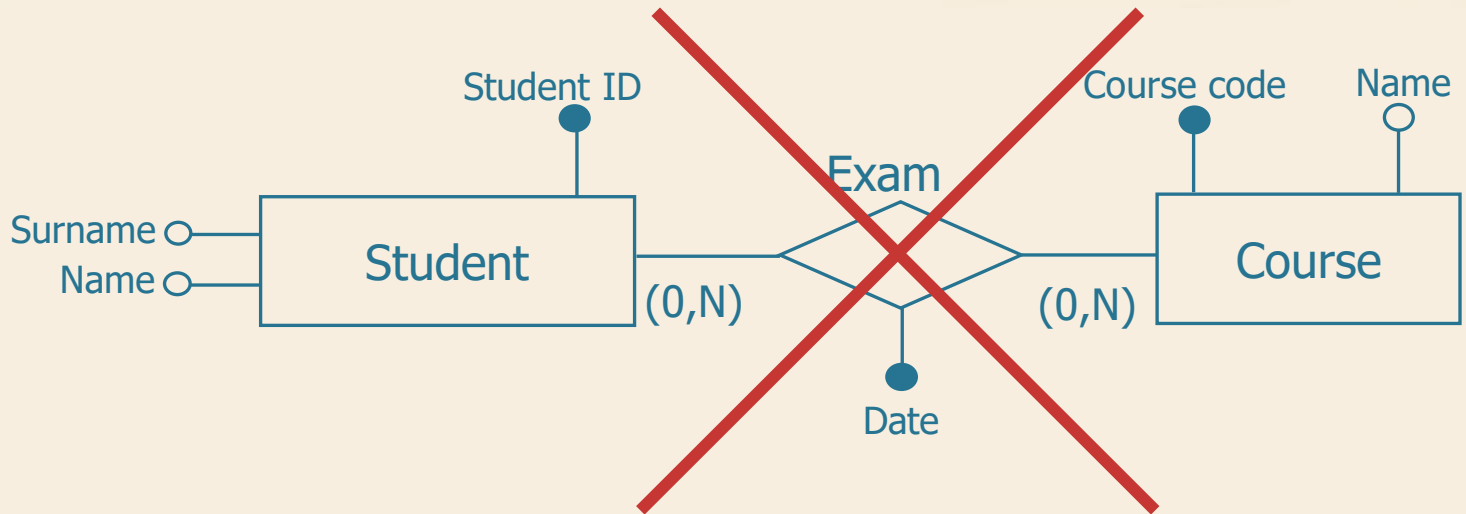
External Identifier



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



➤ Relationships do *not* have identifiers

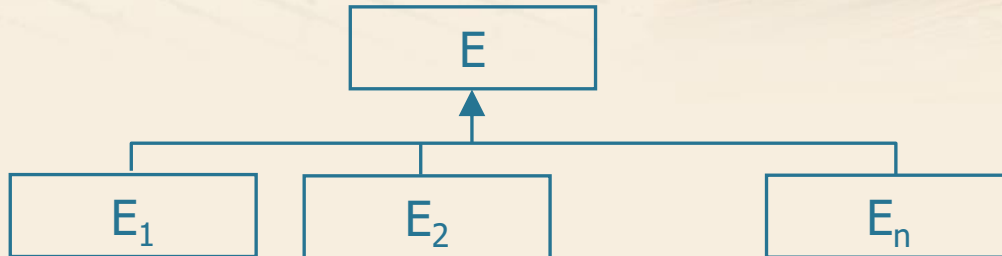




The Entity-Relationship Model

Generalizations

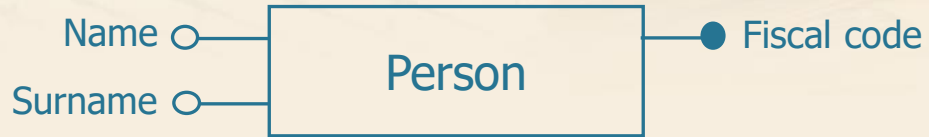
Generalization



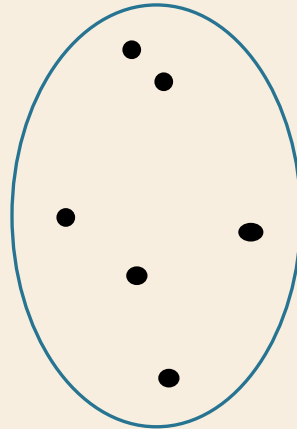
➤ It describes a logical link between an entity E and one or more entities E_1, E_2, \dots, E_n , that are particular cases of E .

- E is called parent entity, is a generalization of E_1, E_2, \dots, E_n
- E_1, E_2, \dots, E_n are called child entities, are specialization of E

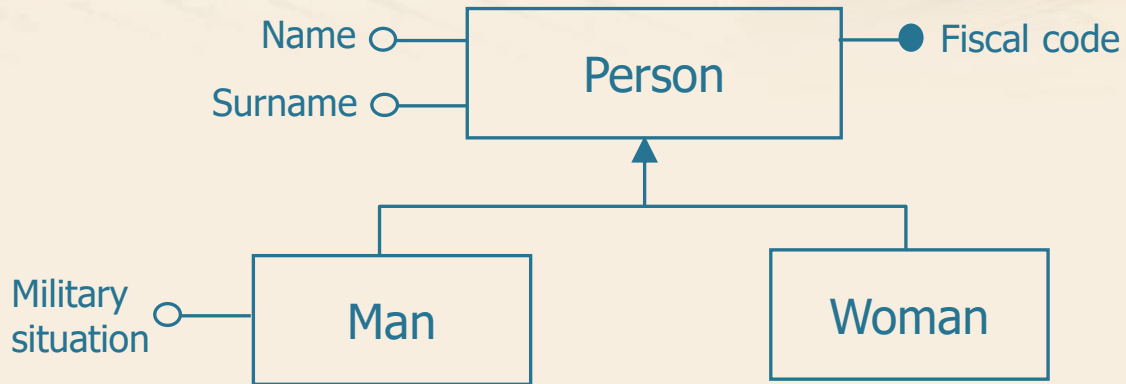
Generalization: example



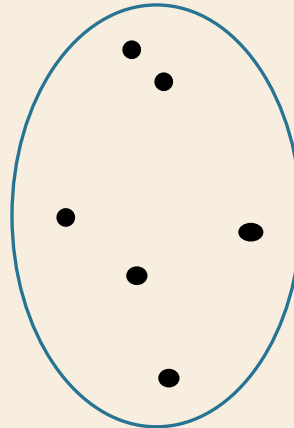
Person



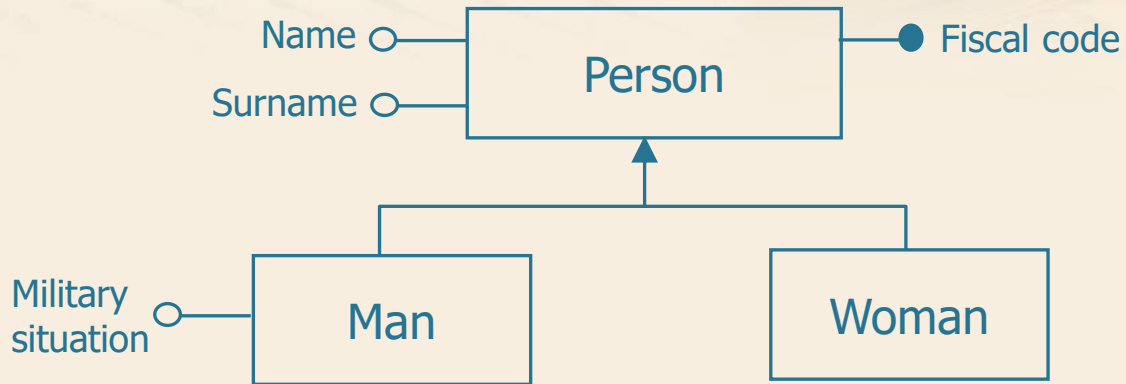
Generalization: example



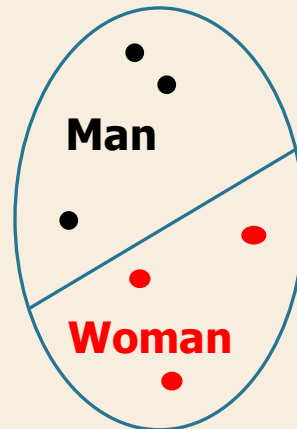
Person



Generalization: example



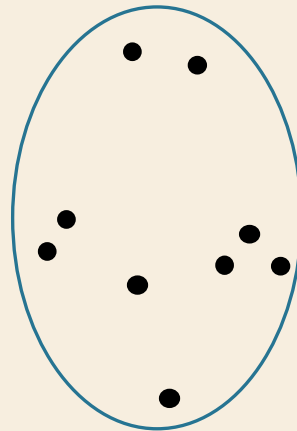
Person



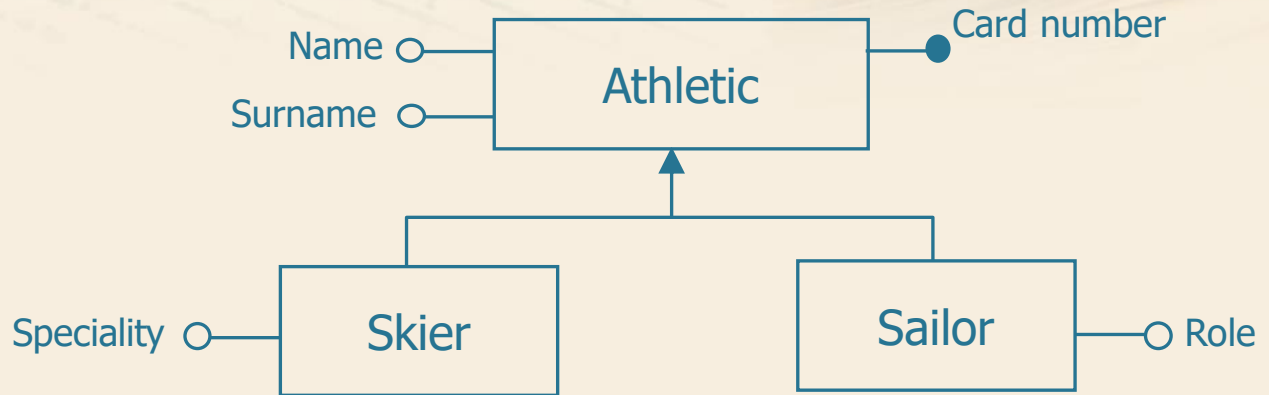
Generalization: example



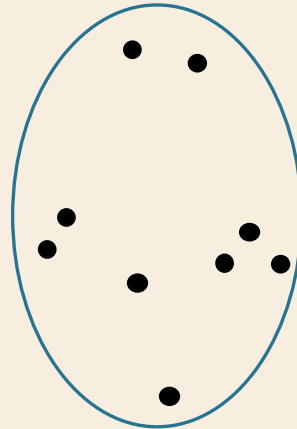
Athletic



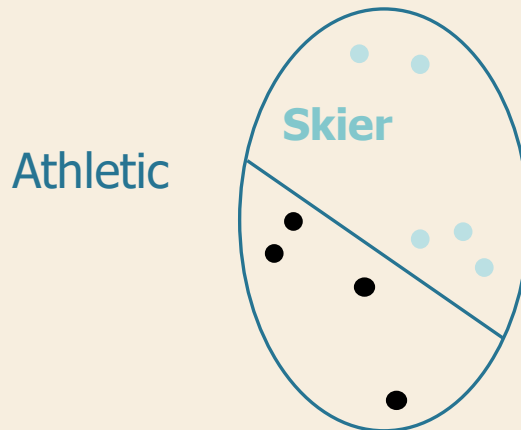
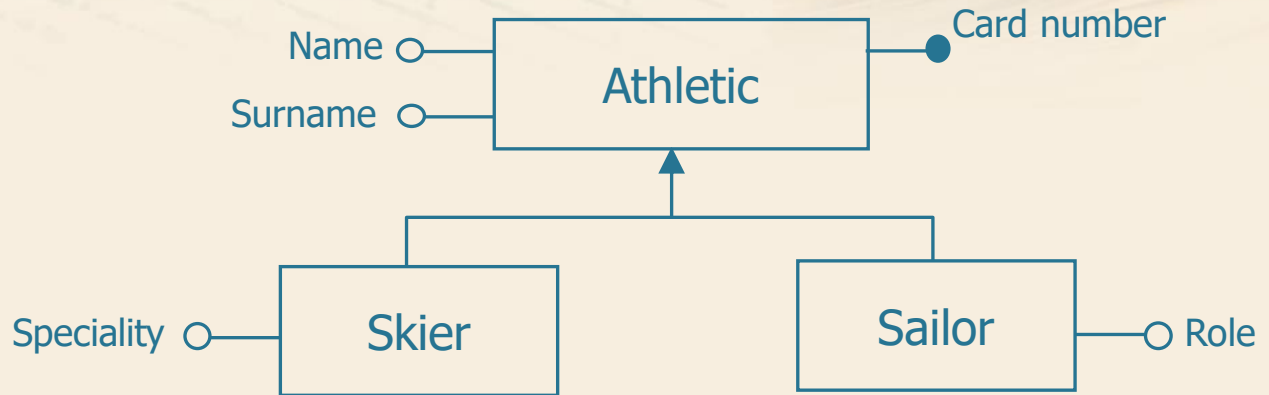
Generalization: example



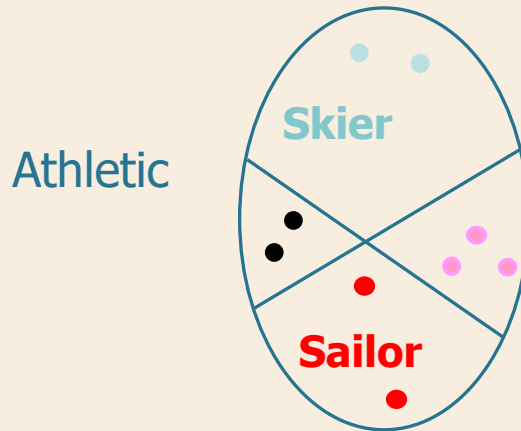
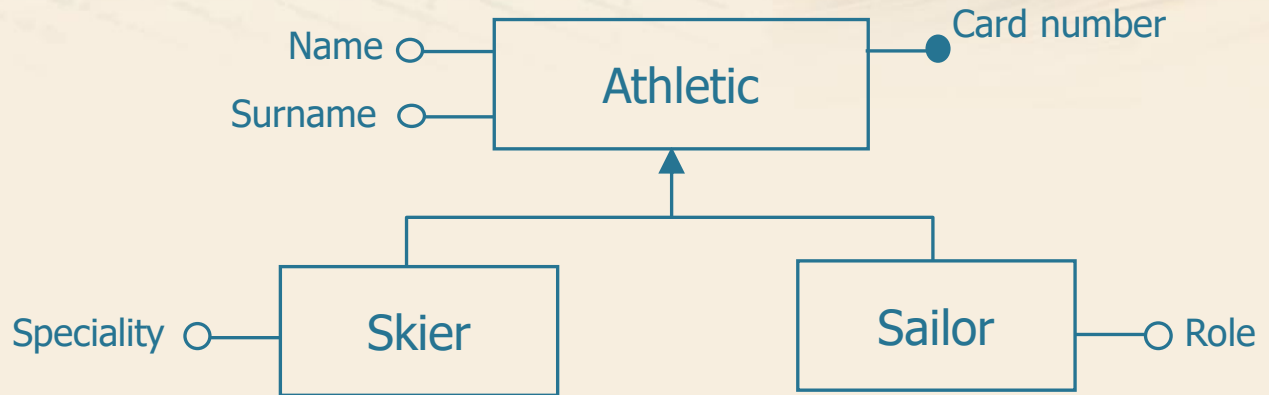
Athletic



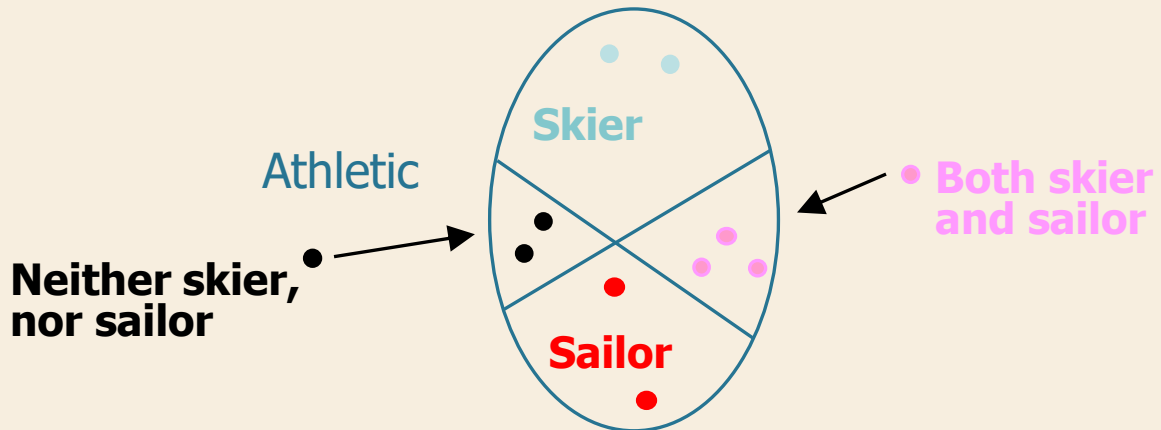
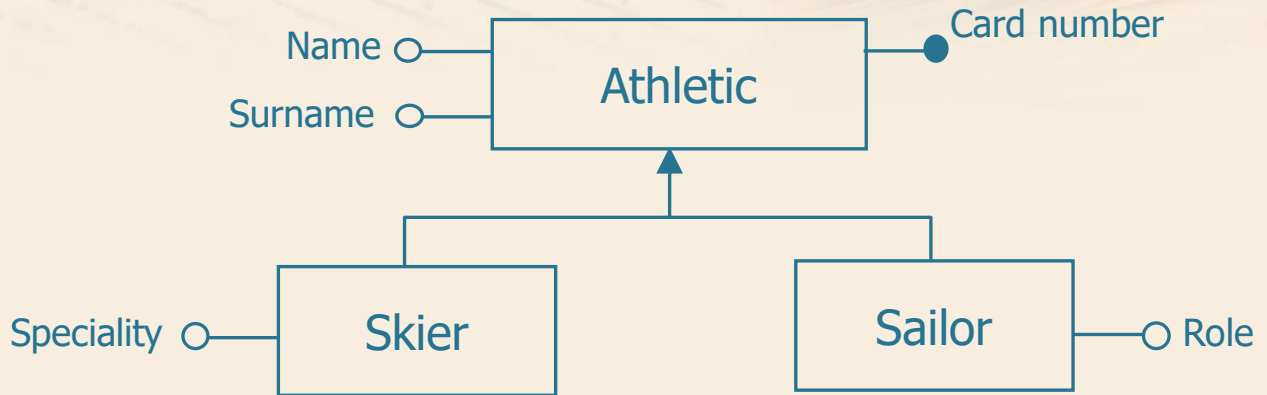
Generalization: example



Generalization: example



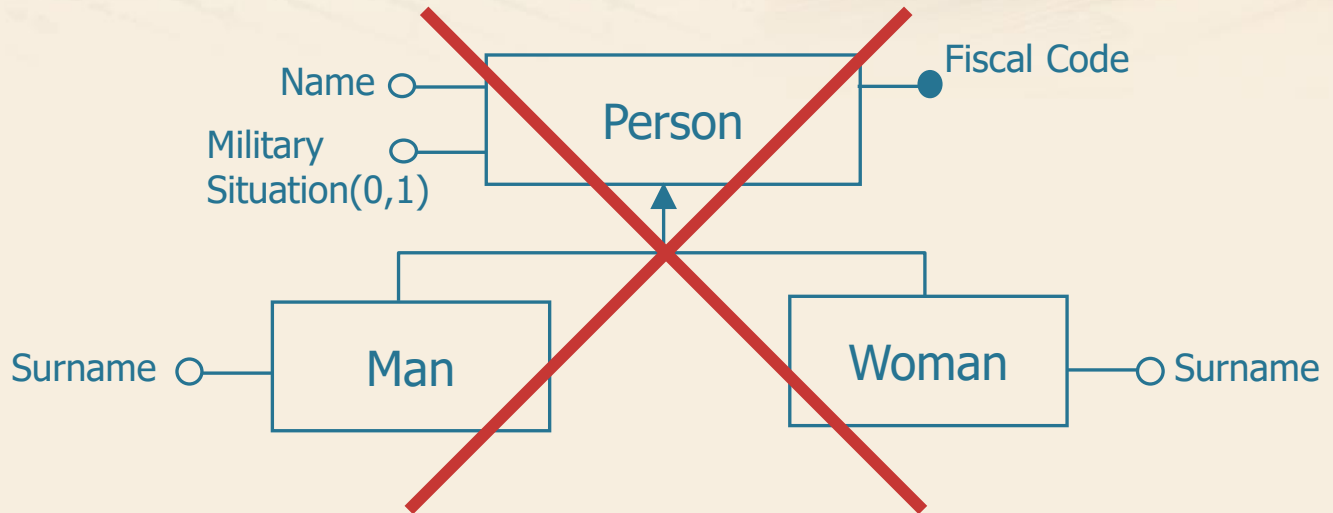
Generalization: example



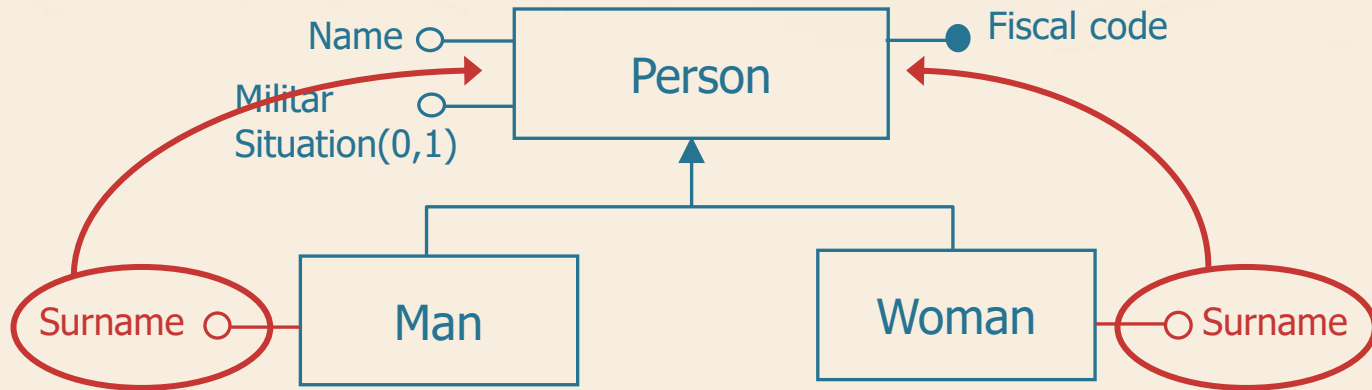
Generalization: property

- 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.

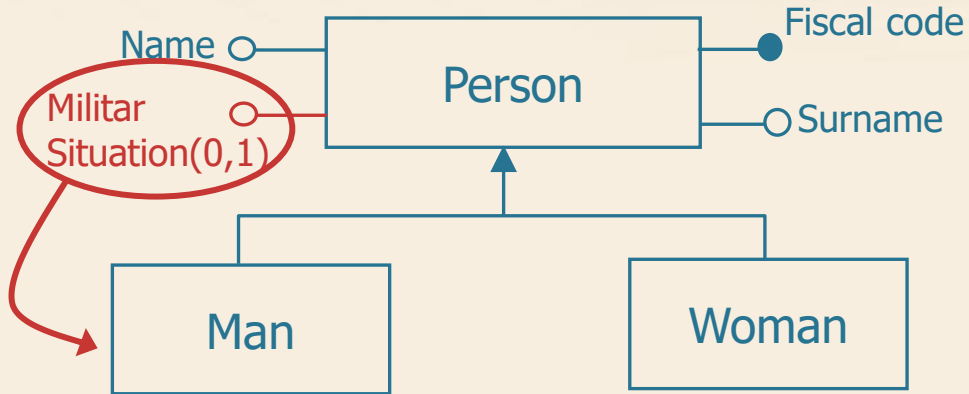
Generalization: incorrect example



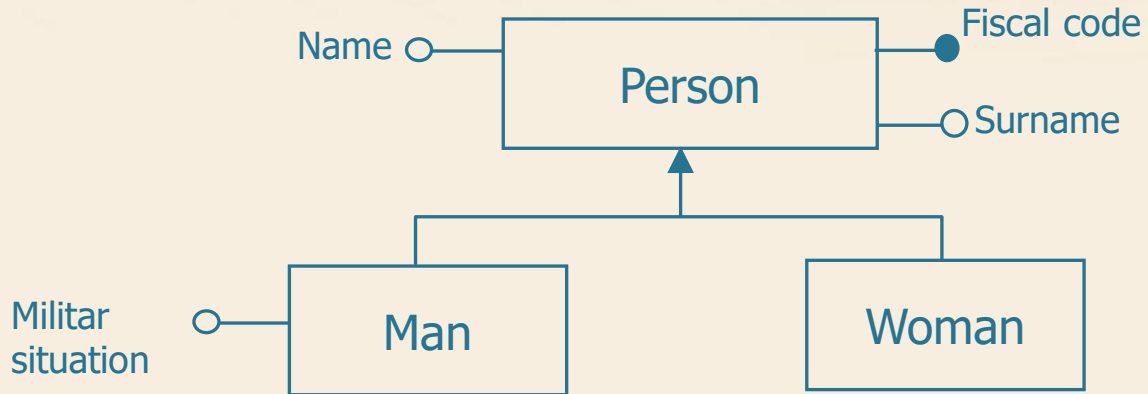
Generalization: incorrect example



Generalization: incorrect example



Generalization: correct example

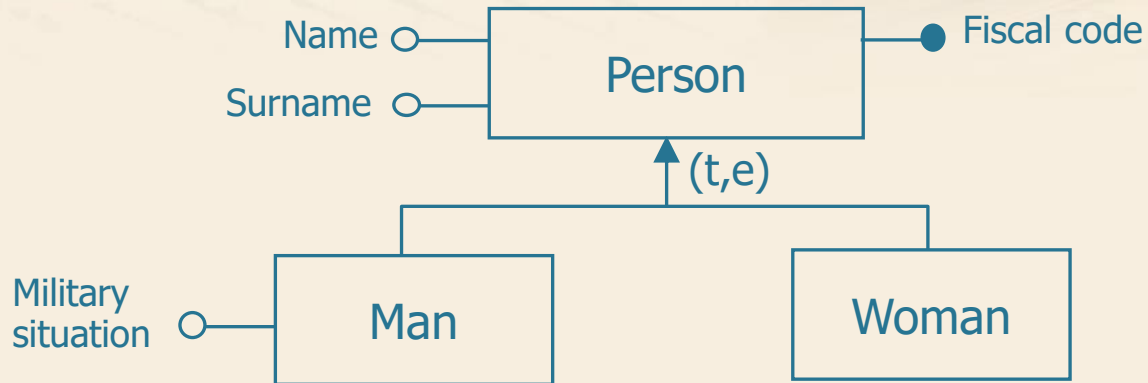


Generalization: property

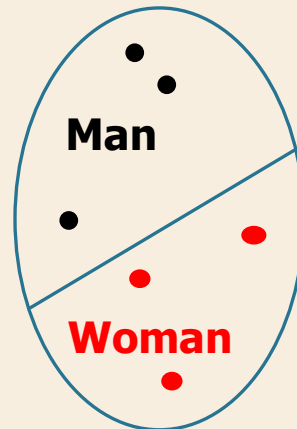
➤ Orthogonal characteristics

- *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.

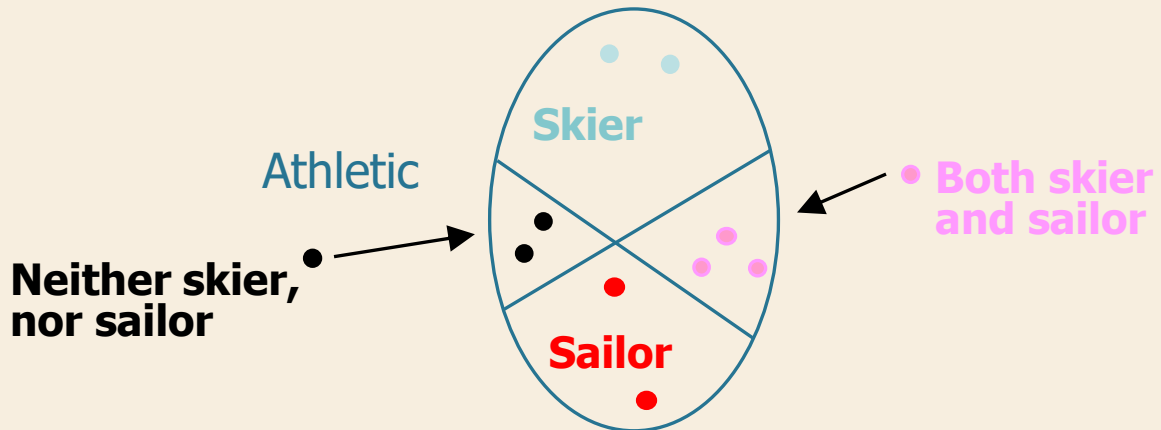
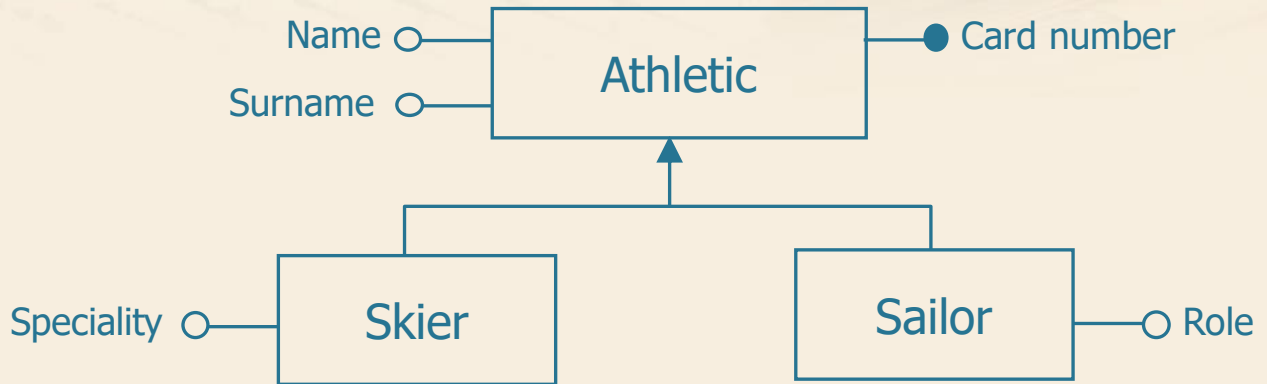
Generalization: example



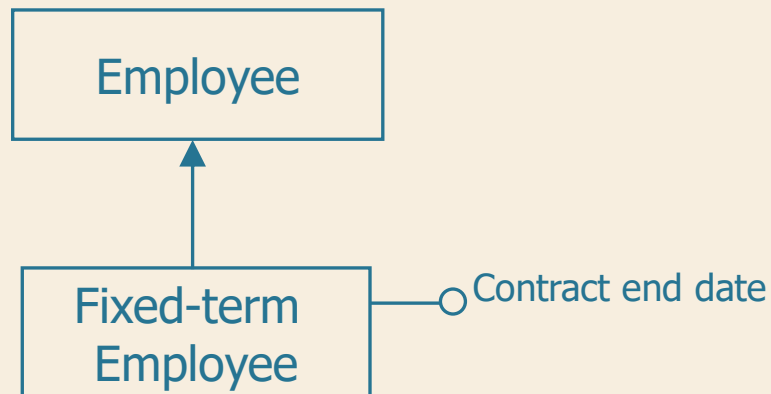
Person



Generalization: example



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

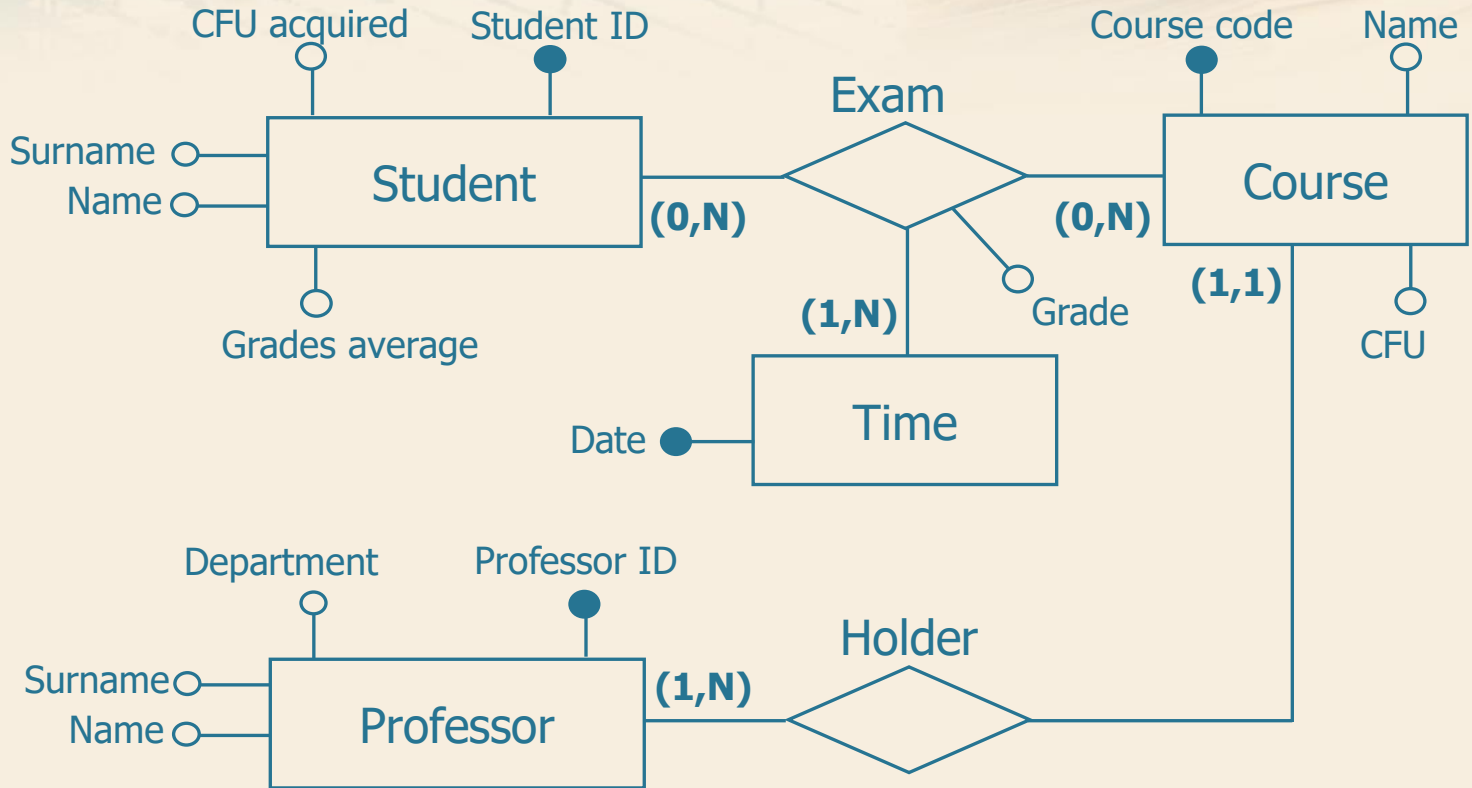




The Entity-Relationship Model

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 (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 communication, of the states,...

➤ E-R

- is a modeling of a data base
 - Structural aspects of an application
- useful constructs for the modelling of databases

- Main characteristics 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)

- 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