



Introduction to databases

Database design

1



Database design

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

2

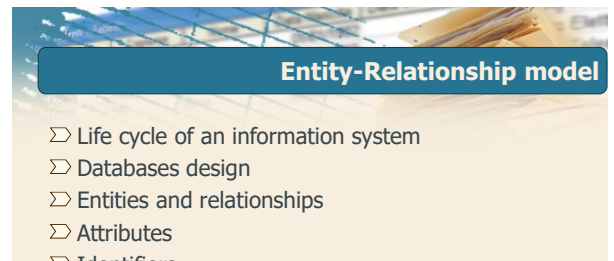
2



Database design

Entity-Relationship Model

3




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

4

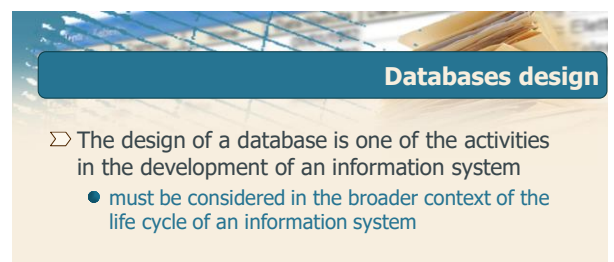
4



Entity-Relationship model

Life cycle of an information system

5



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

6

6

Life cycle of an information system

Feasibility study

7

Life cycle of an information system

Σ Feasibility study

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

8

Life cycle of an information system

Feasibility study

↓

Collection and analysis of the requirements

9

Life cycle of an information system

Σ 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

10

Life cycle of an information system

Feasibility study

↓

Collection and analysis of the requirements

↓

Design

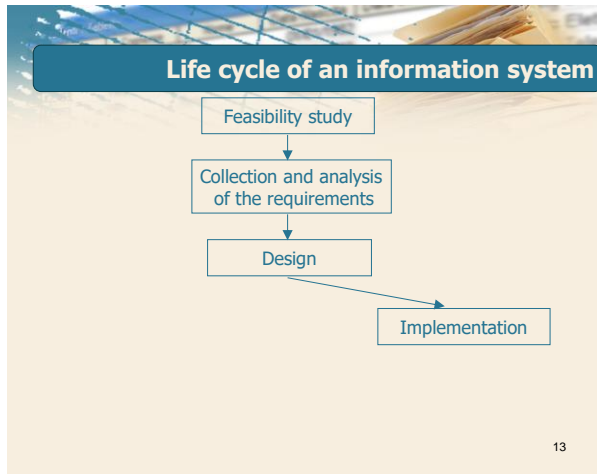
11

Life cycle of an information system

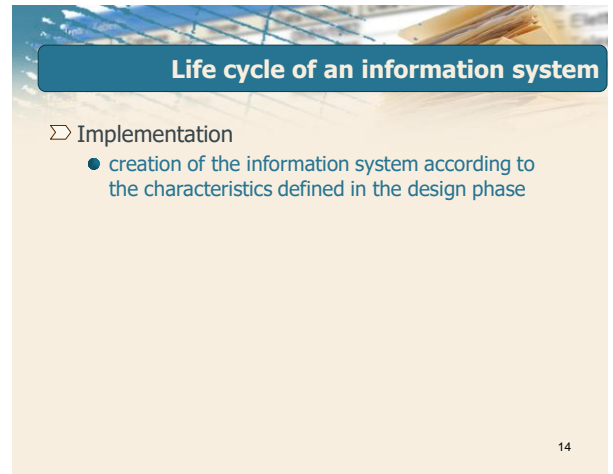
Σ Design

- divided into data and application design
- produces formal descriptions

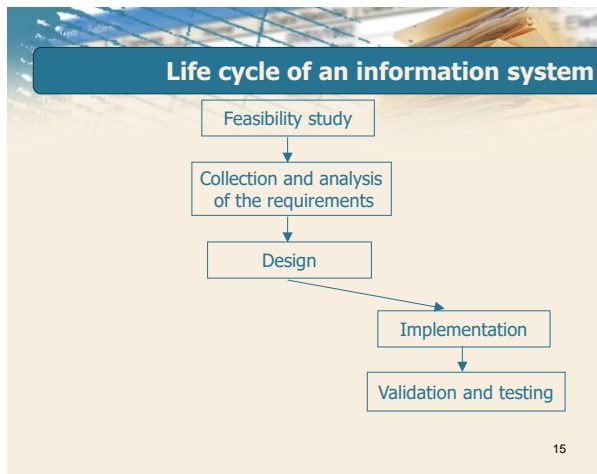
12



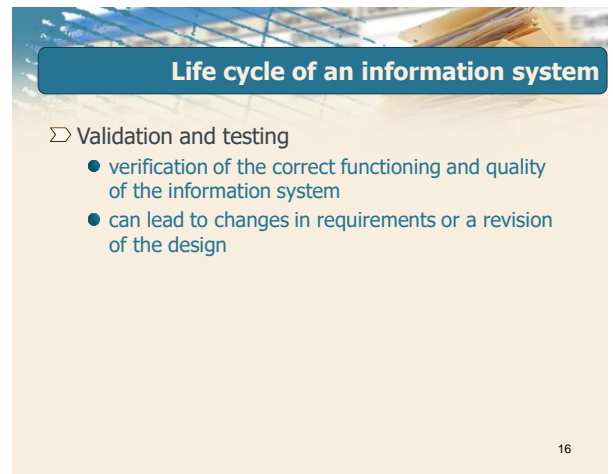
13



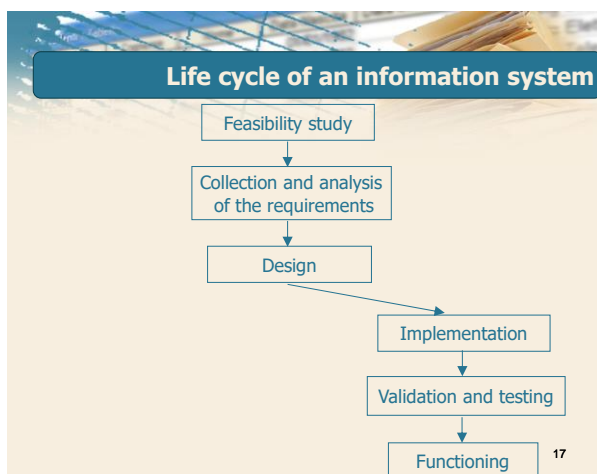
14



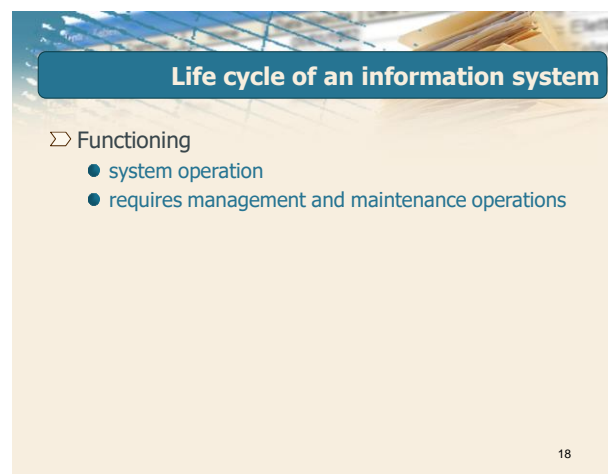
15



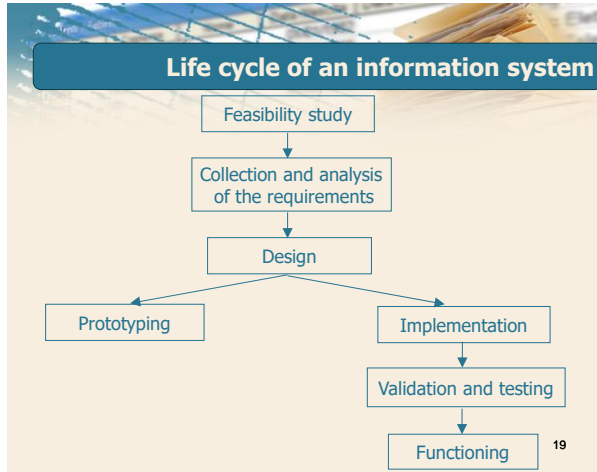
16



17



18



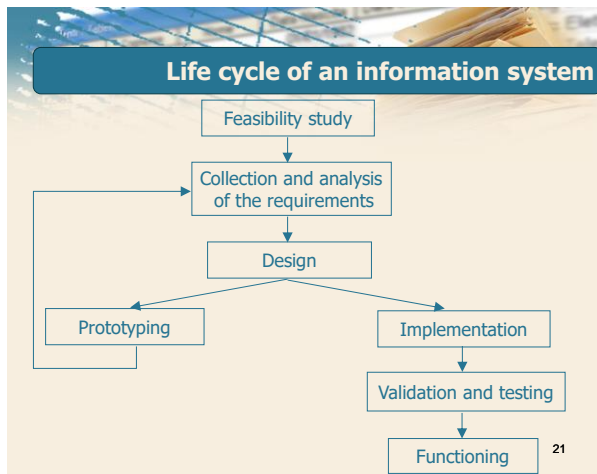
19

Life cycle of an information system

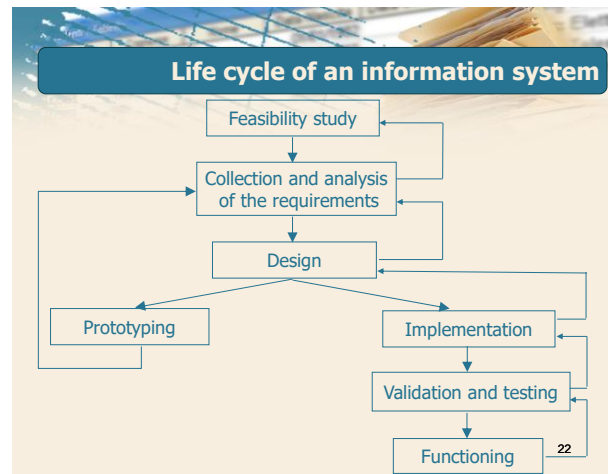
⊃ Prototyping

- 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

20



21



22

Entity-Relationship model

Database design

23

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

24

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

25

25

Design methodology: Example

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

26

26

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

27

27

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

28

28

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

29

29

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

30

30

Stages of database design

31

31

Application requirements

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

32

32

Stages of database design

33

33

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

34

34

Stages of database design

35

35

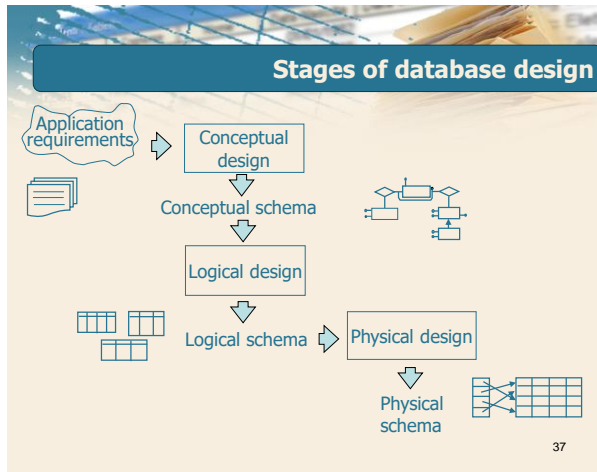
Logical design

▷ 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)

36

36



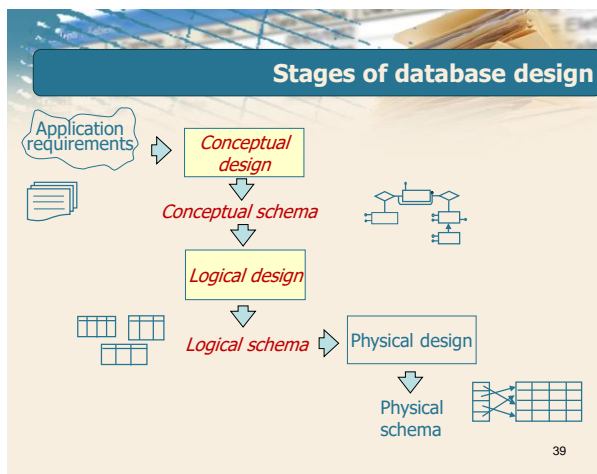
37

Physical design

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

38

38



39

Entity-Relationship model

Entities and relationships

40

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

41

41

Main constructs of the E-R model

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

42

42

Entities

Entity name

- ▷ 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

43

43

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

44

44

Relationships examples

Student

Course

45

45

Relationships examples

46

46

Relationships examples

Students

Passed Exam

Course

Person

Municipality

47

47

Relationships examples

Students

Passed Exam

Course

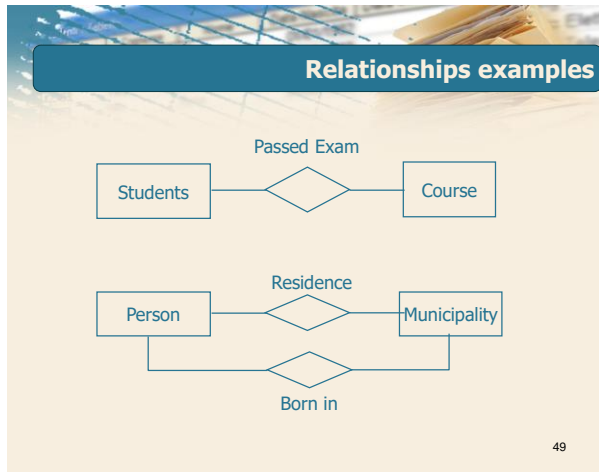
Person

Residence

Municipality

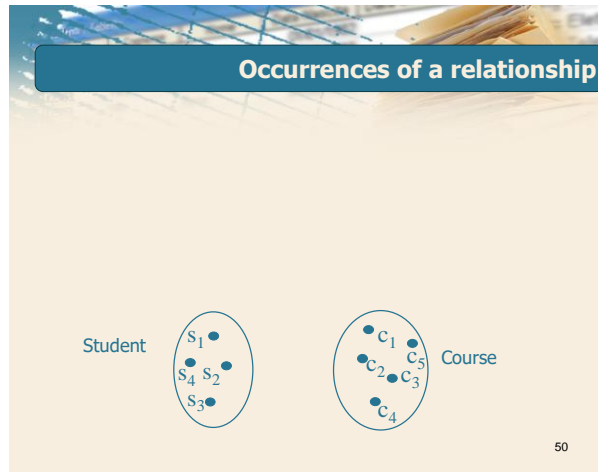
48

48



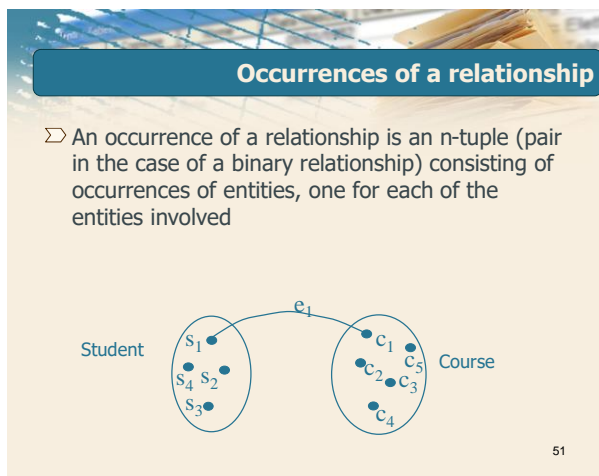
49

49



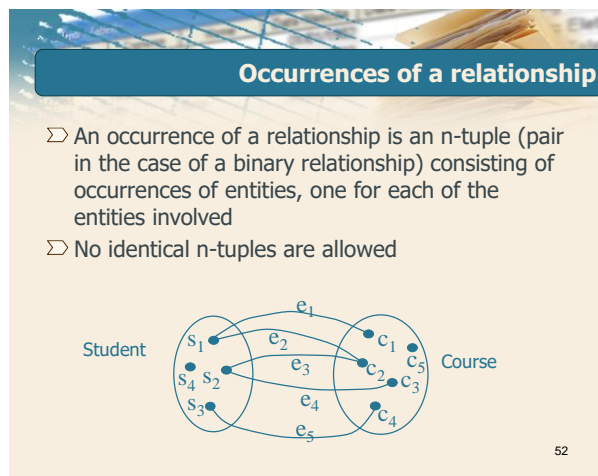
50

50



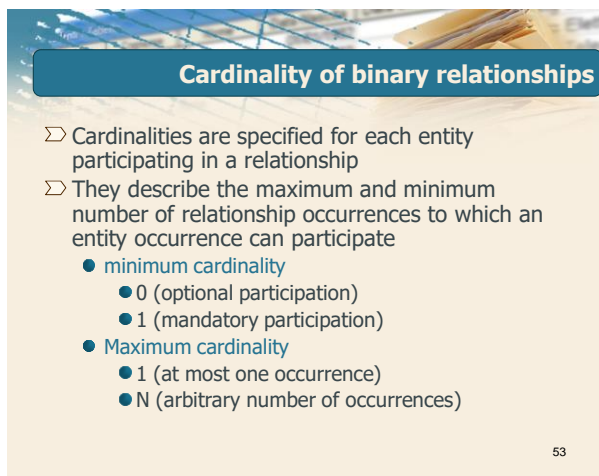
51

51



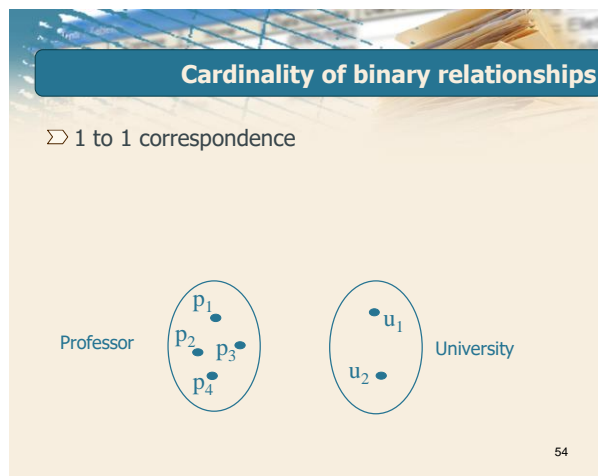
52

52



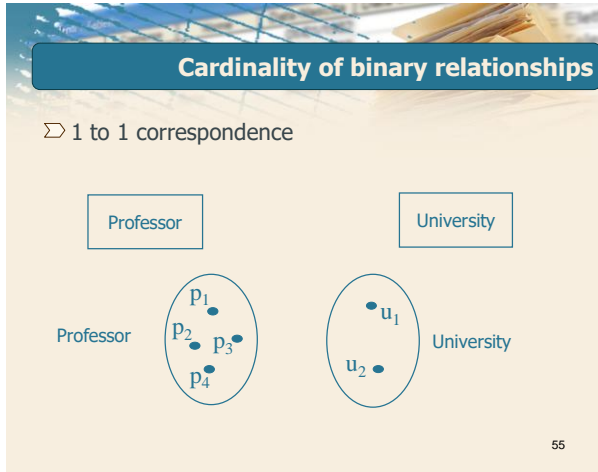
53

53

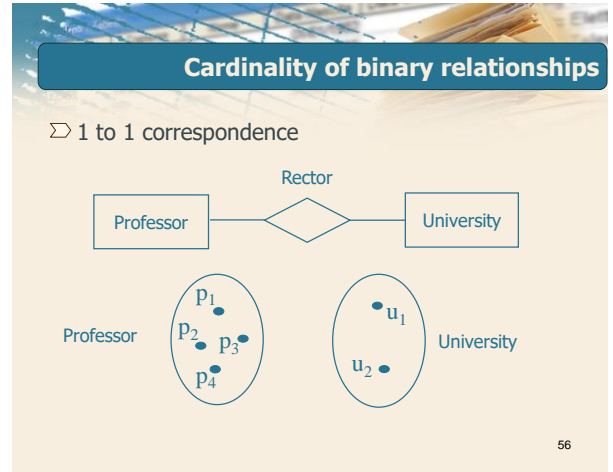


54

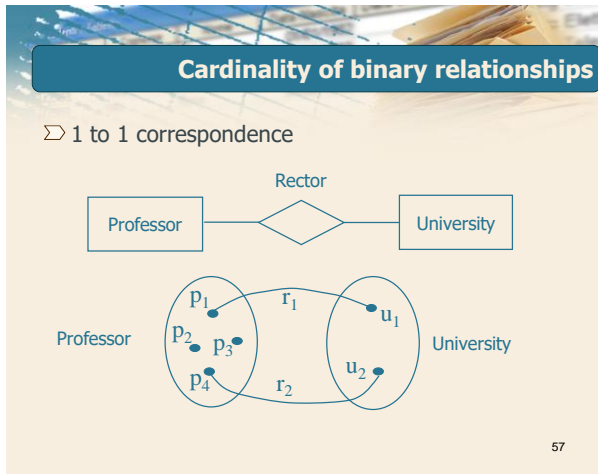
54



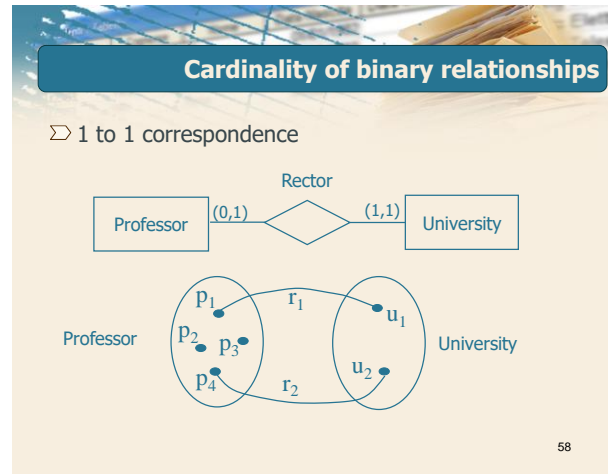
55



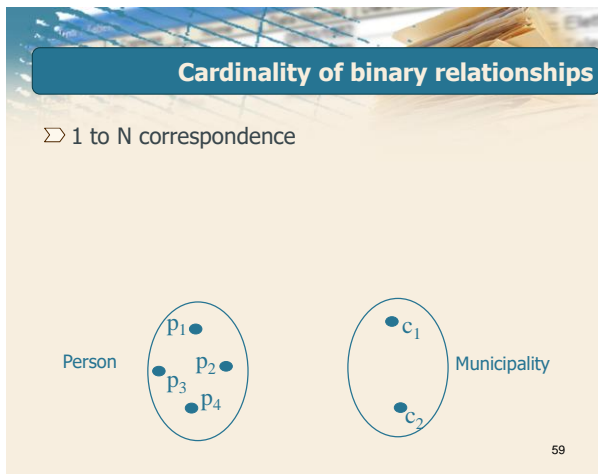
56



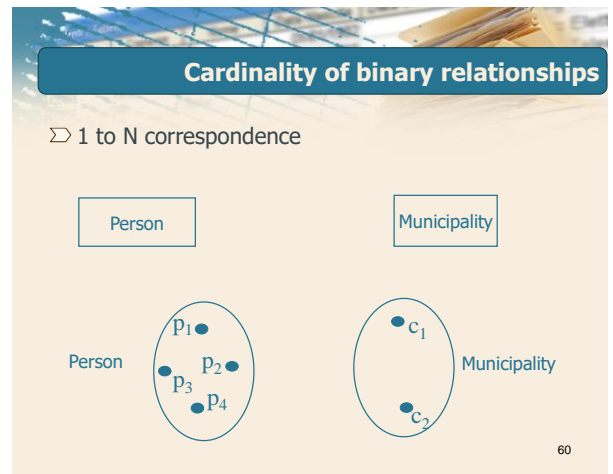
57



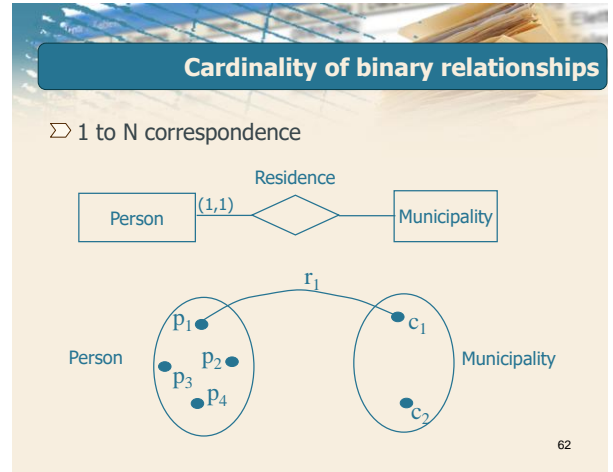
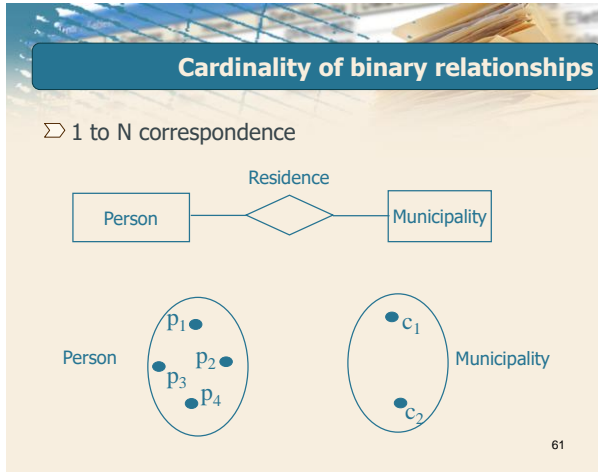
58



59

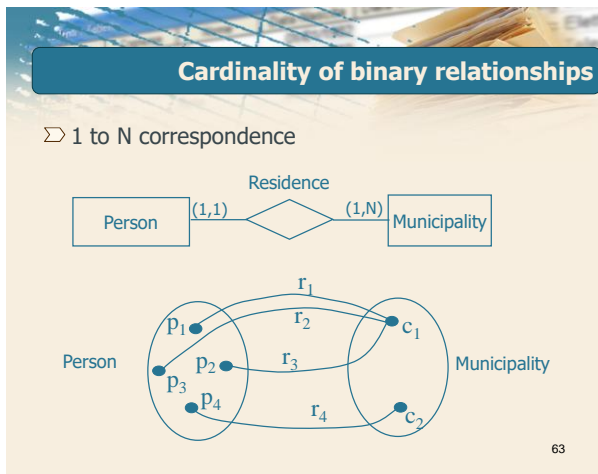


60

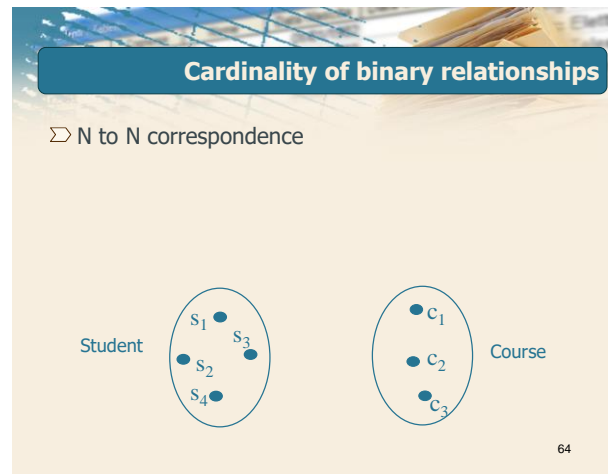


61

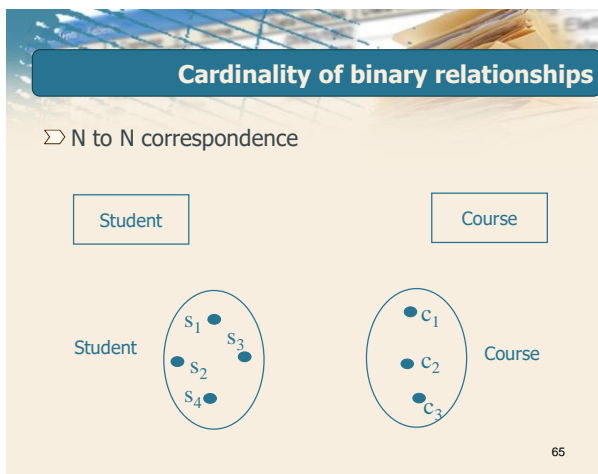
62



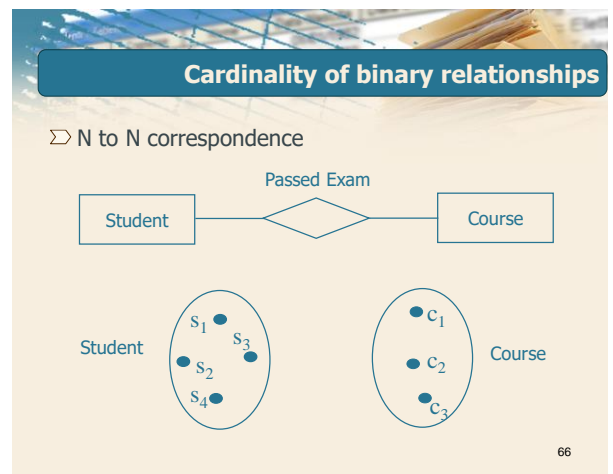
63



64



65



66

Cardinality of binary relationships

▷ N to N correspondence

67

67

Cardinality of binary relationships

▷ N to N correspondence

68

68

Limitations of a binary relationship

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

69

69

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
...		

70

70

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
...		

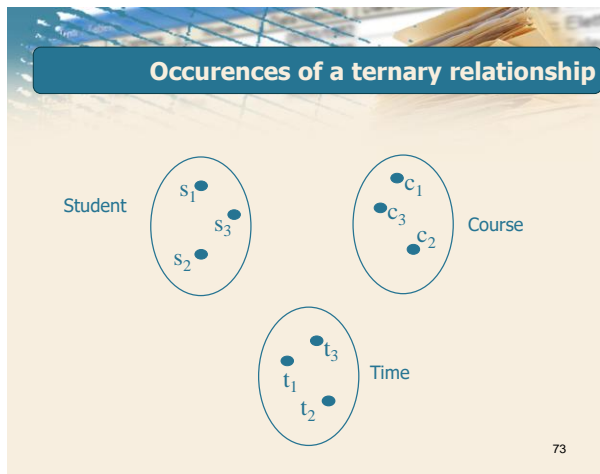
71

71

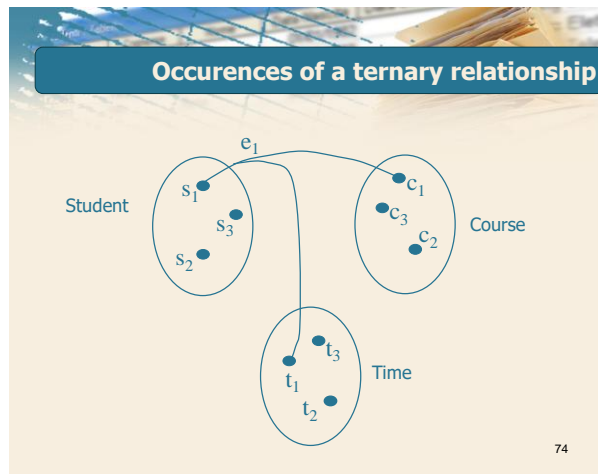
Cardinality of ternary relationships

72

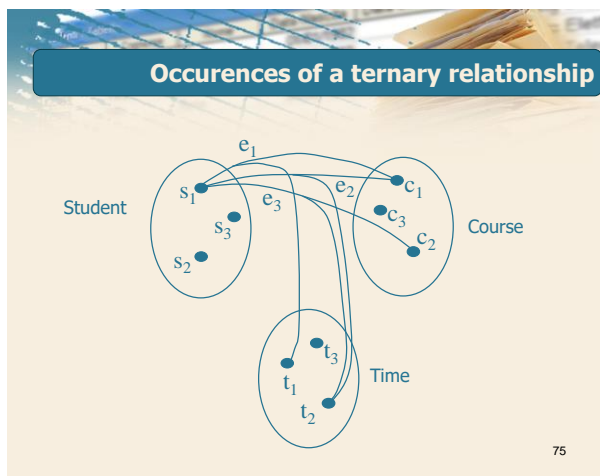
72



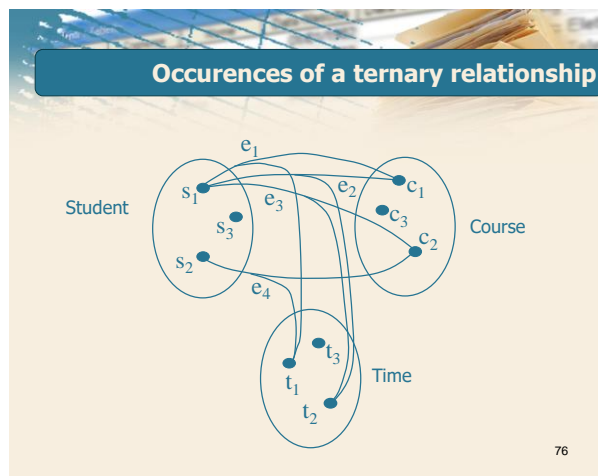
73



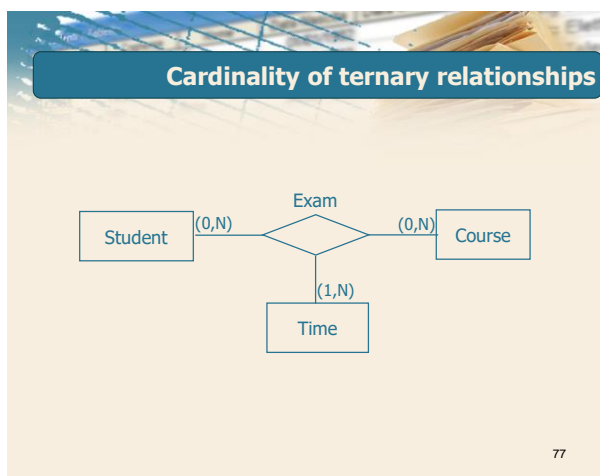
74



75



76



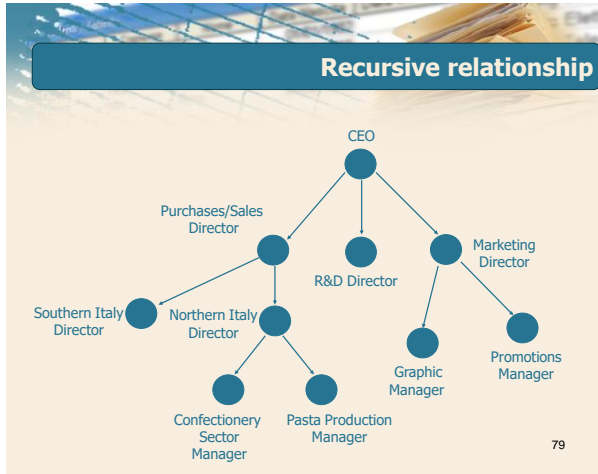
77

Observations

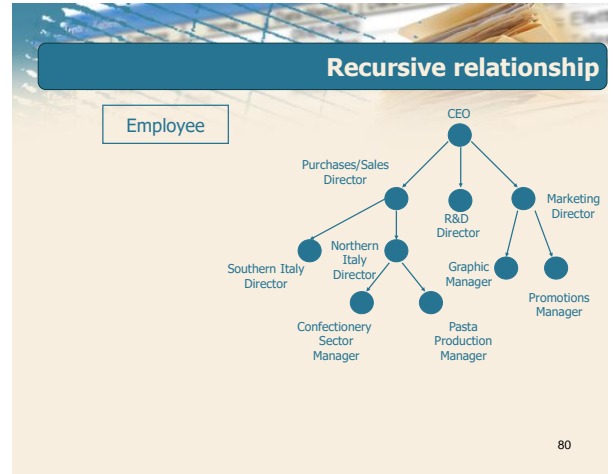
- ▷ Minimum cardinalities are rarely 1 for all entities involved in a relationship
- ▷ The maximum cardinalities of an n-ary relationship are (practically) always N
 - if the participation of an entity E has a maximum cardinality of 1, it is possible to eliminate the n-ary relationship and link entity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

78

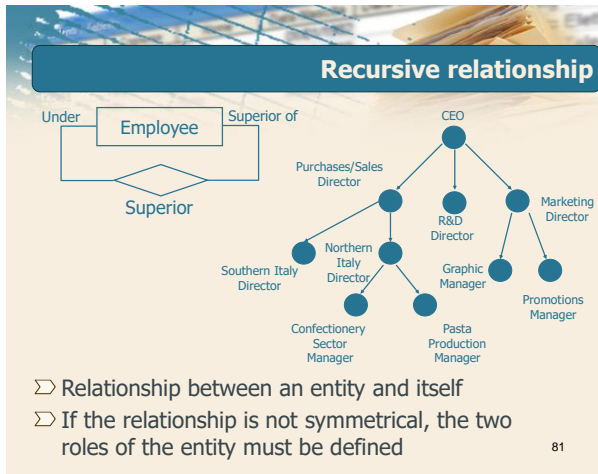
78



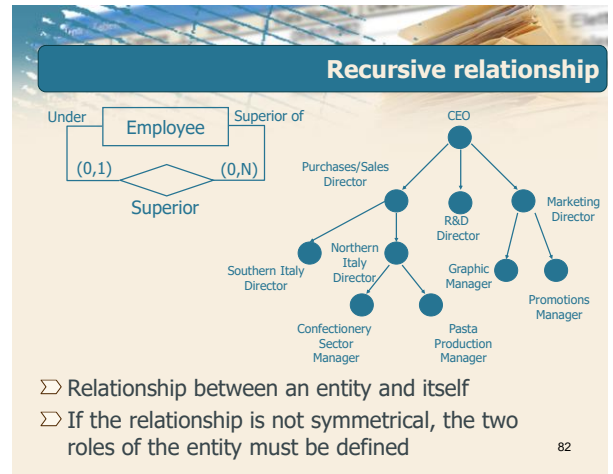
79



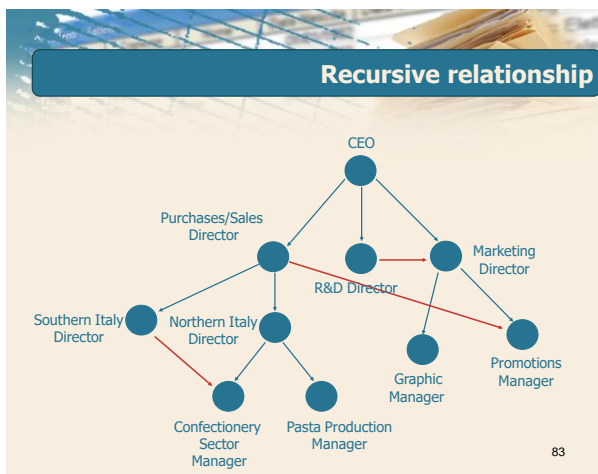
80



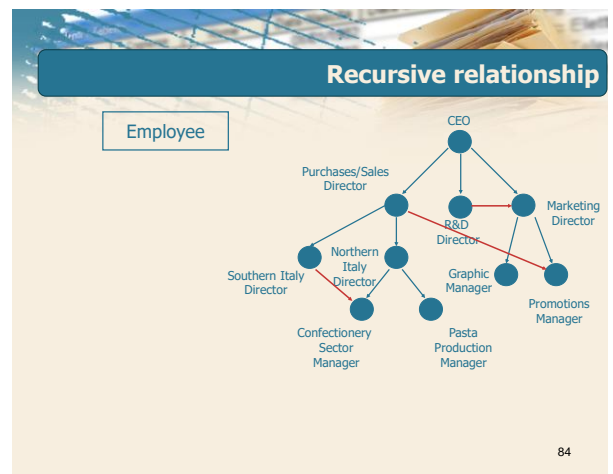
81



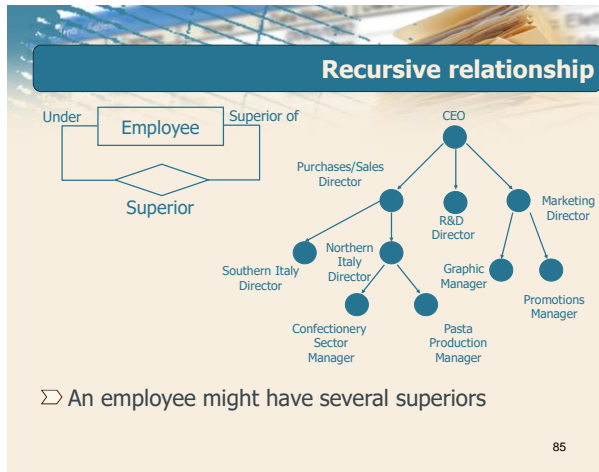
82



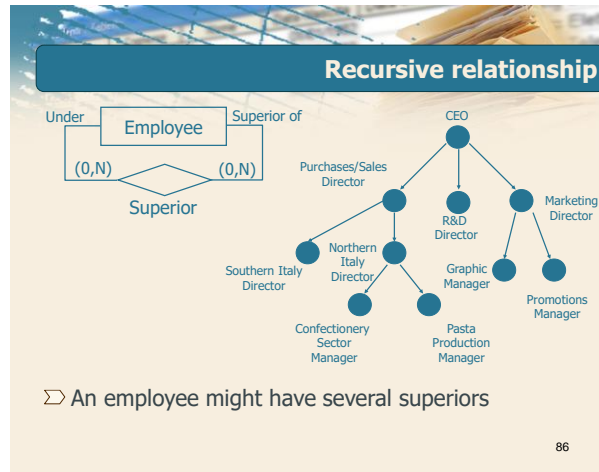
83



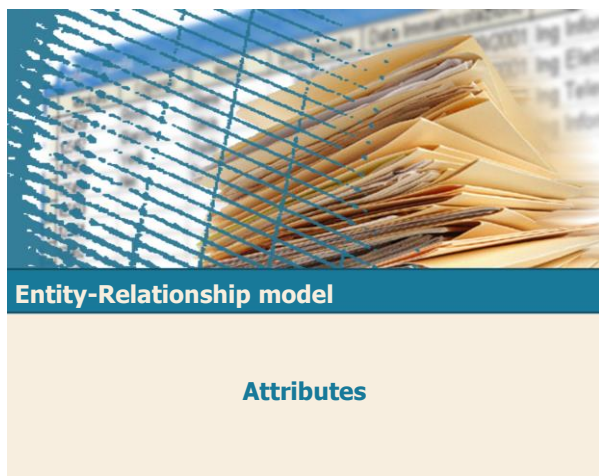
84



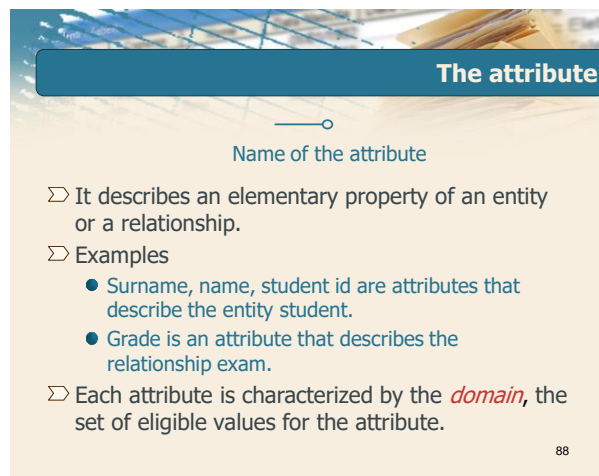
85



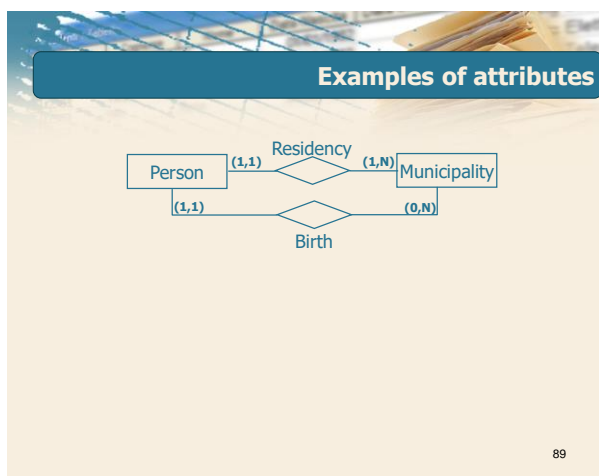
86



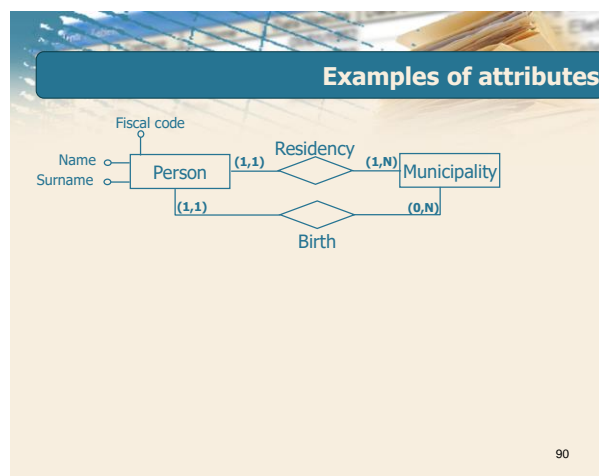
87



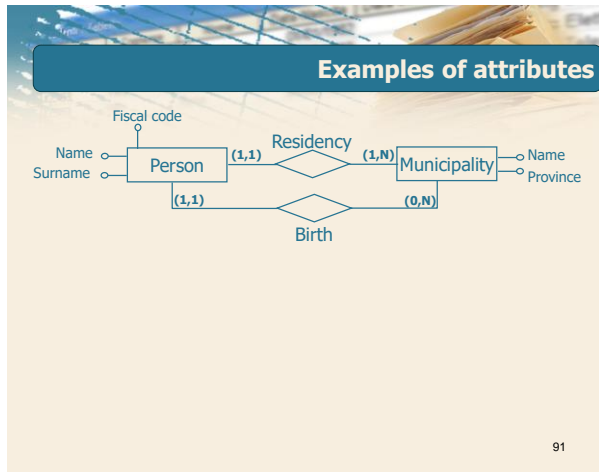
88



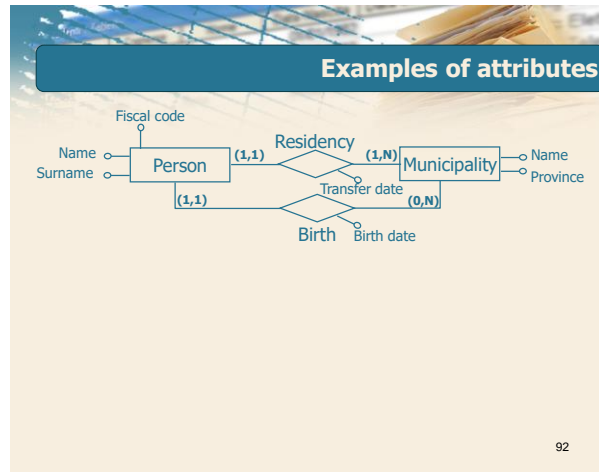
89



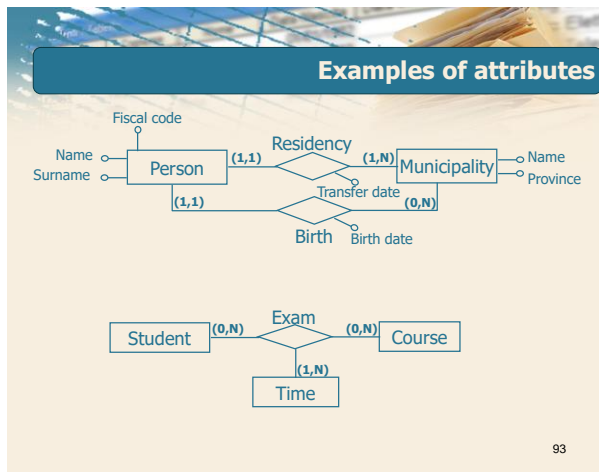
90



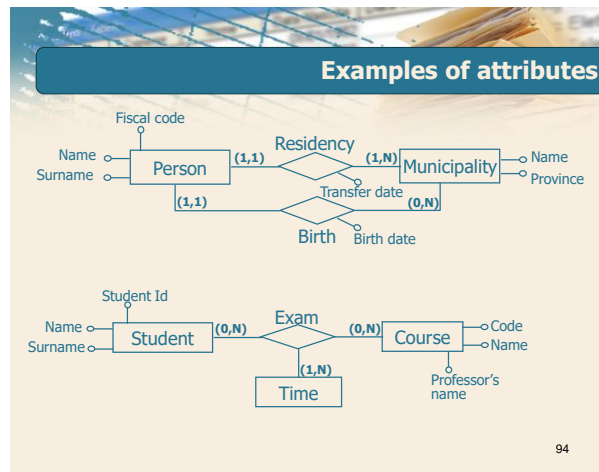
91



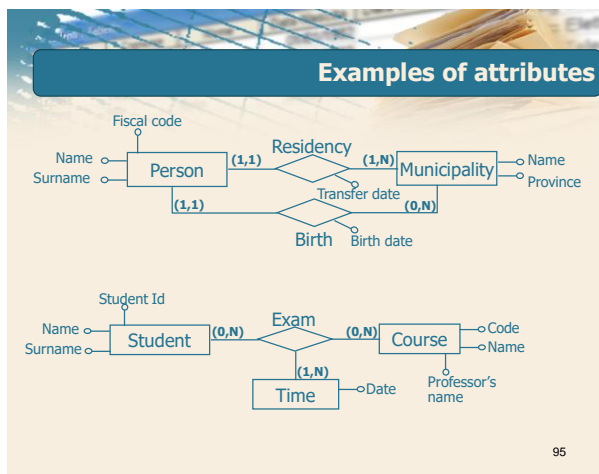
92



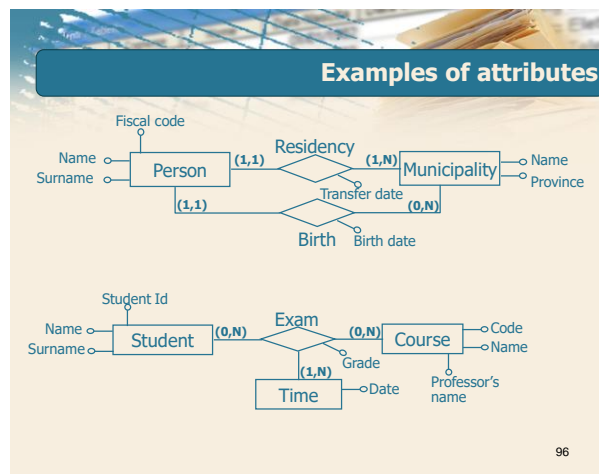
93



94



95



96

Composite attribute

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

97

Composite attribute

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

▷ Example

98

97

98

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)

99

Cardinality of an attribute

100

99

100

Cardinality of an attribute

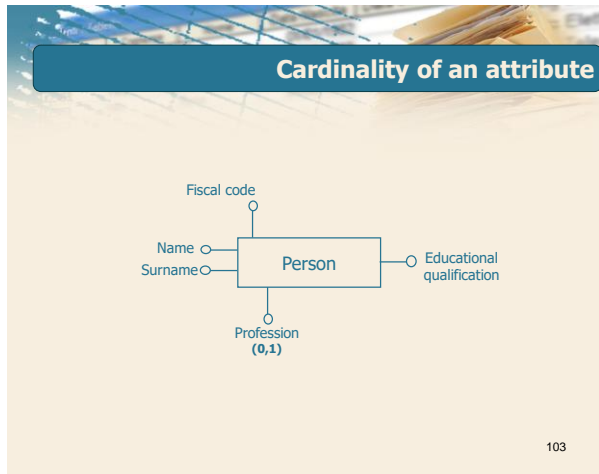
101

Cardinality of an attribute

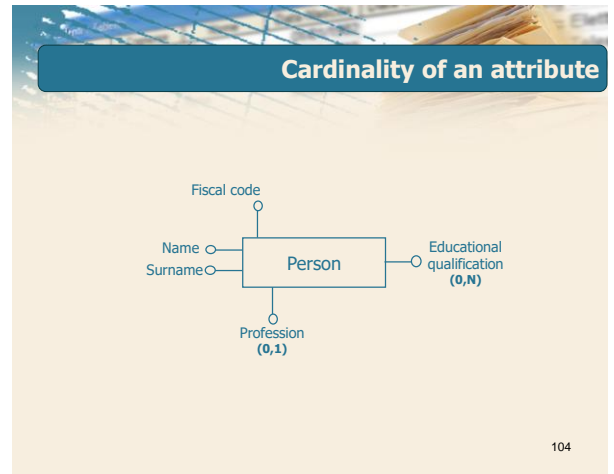
102

101

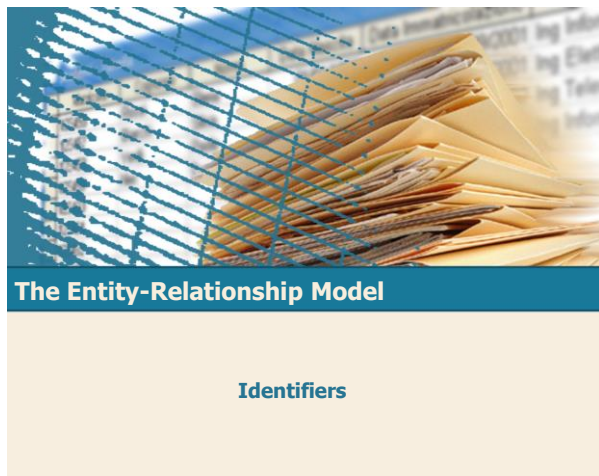
102



103



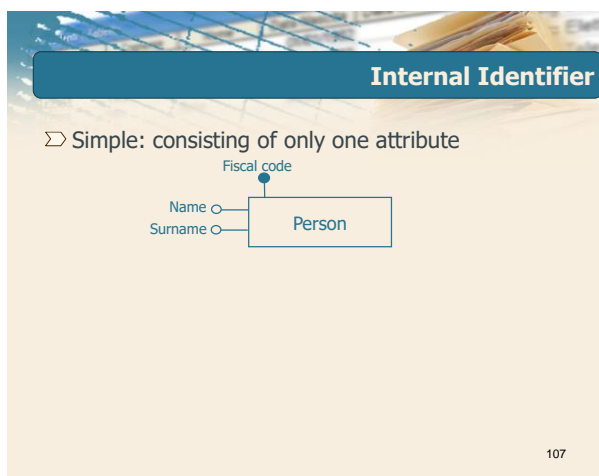
104



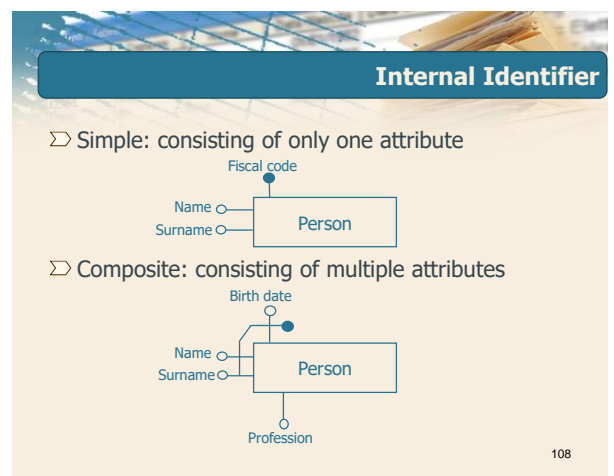
105

- ### 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 more than one appropriated identifier for a given entity.

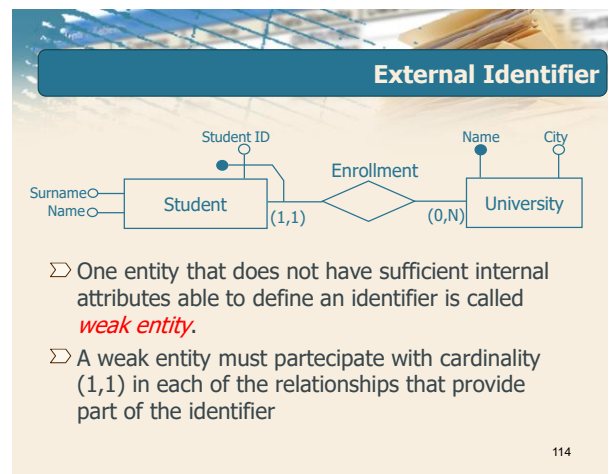
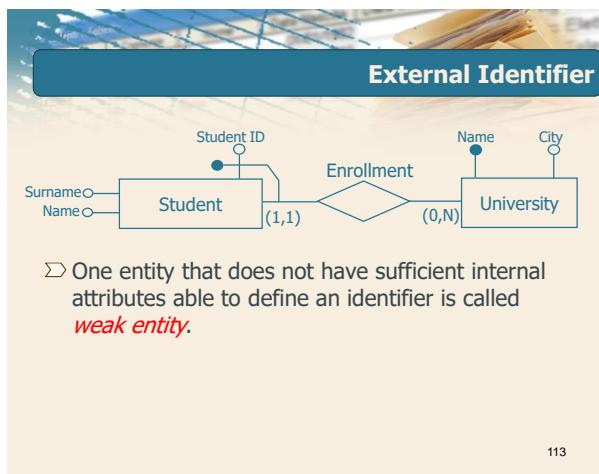
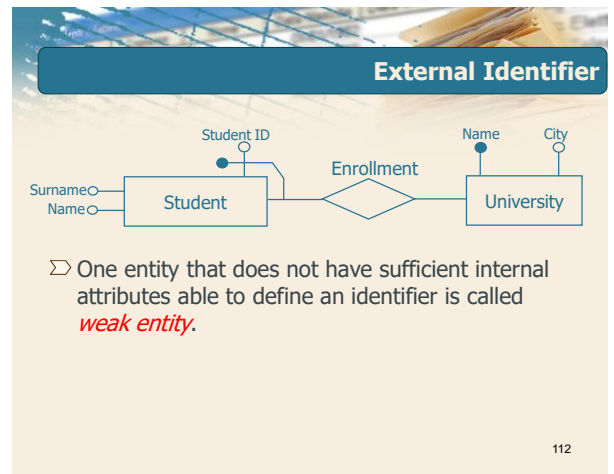
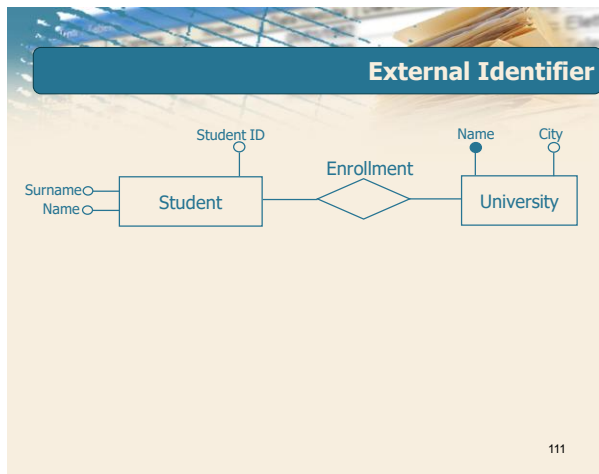
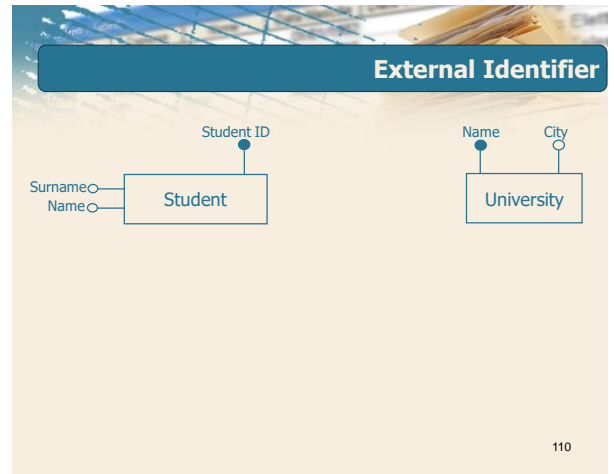
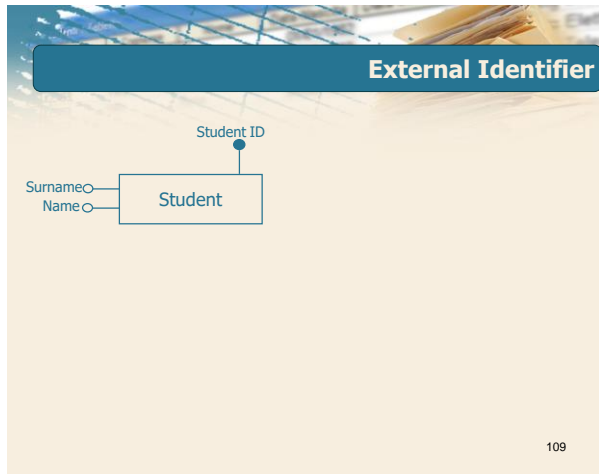
106



107

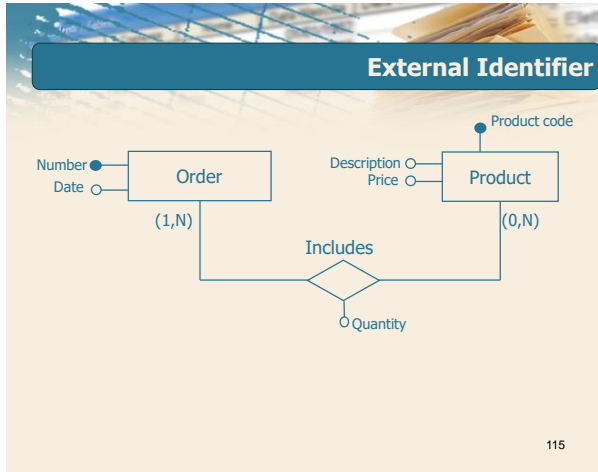


108

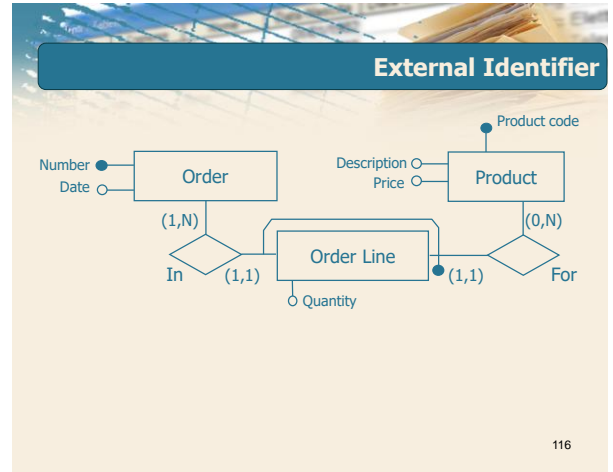


113

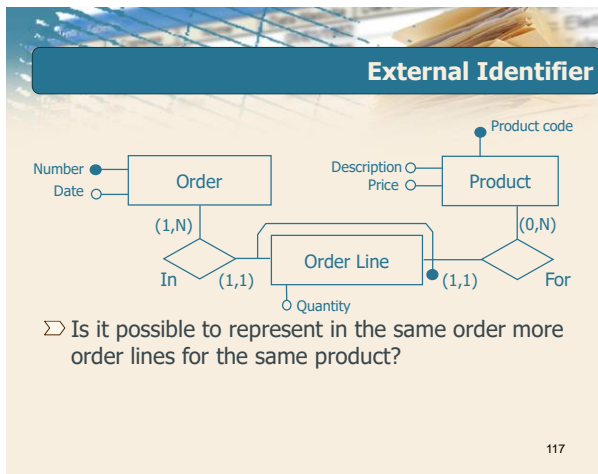
114



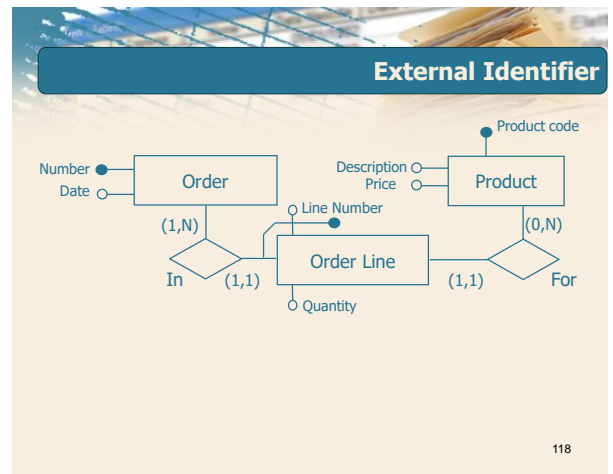
115



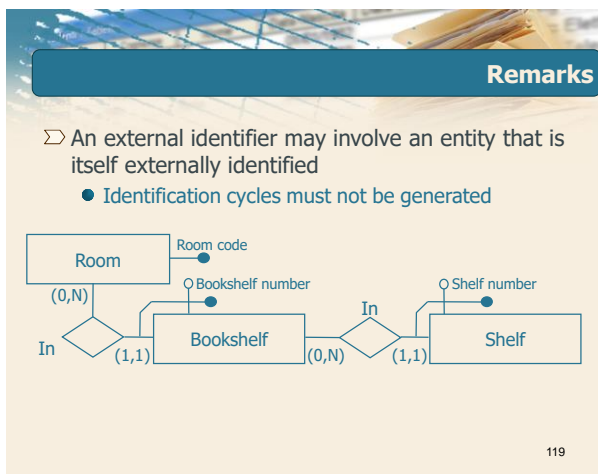
116



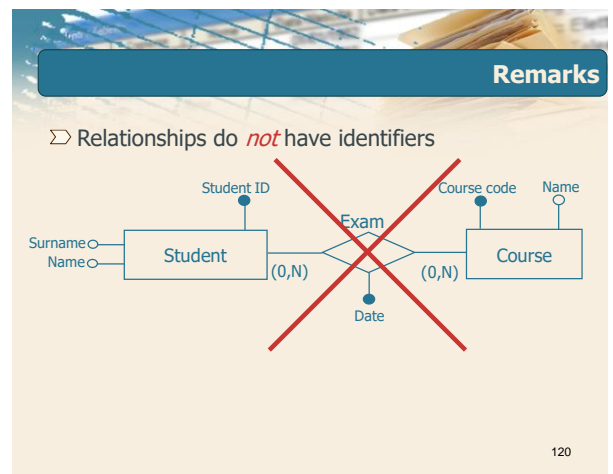
117




118



119




120



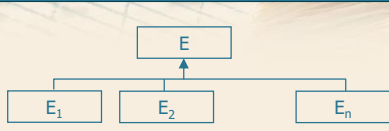
The Entity-Relationship Model

Generalizations

121



Generalization



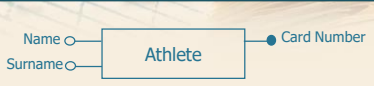
It describes a logical link between an entity E and one or more entities E_1, E_2, \dots, E_n , that are particular cases of E.


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

122

122

Generalization: example

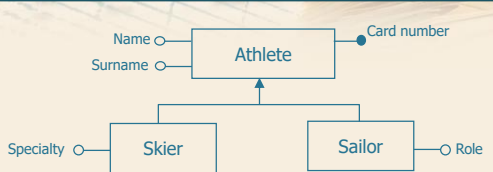


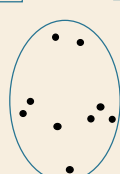


123

123

Generalization: example

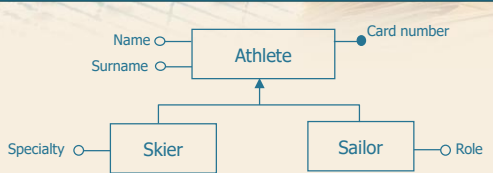





124

124

Generalization: example

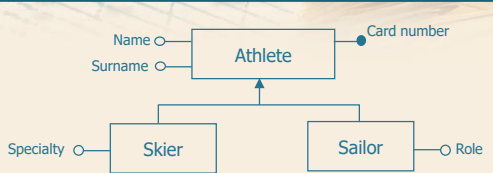


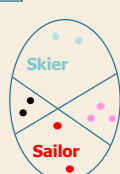


125

125

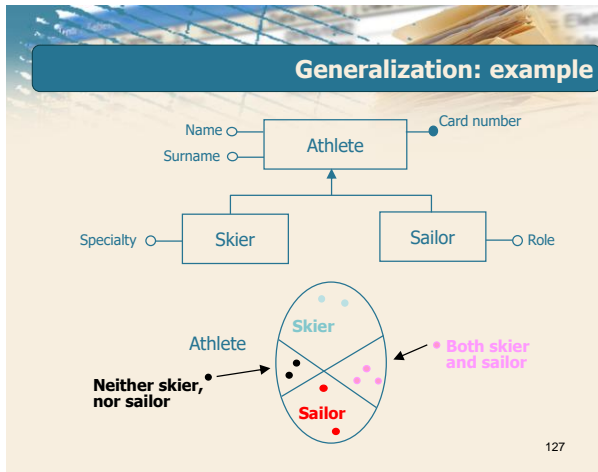
Generalization: example



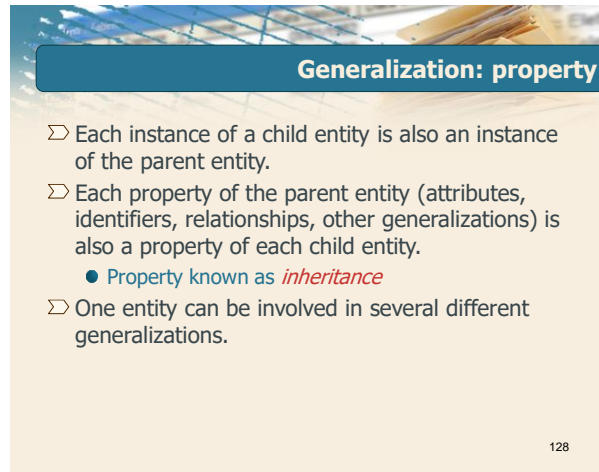


126

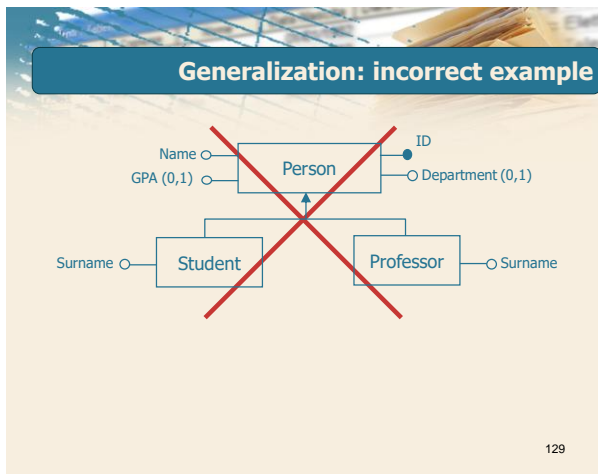
126



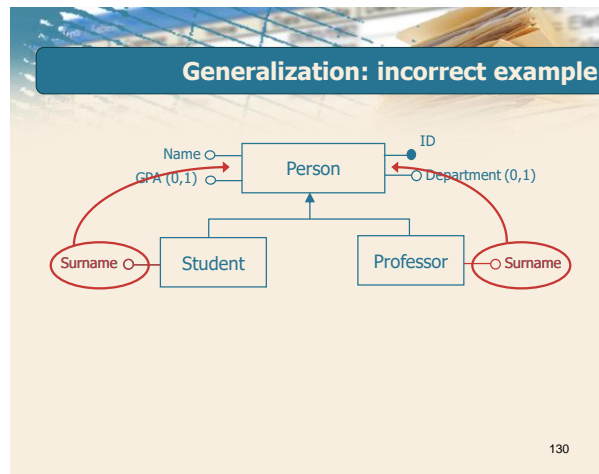
127



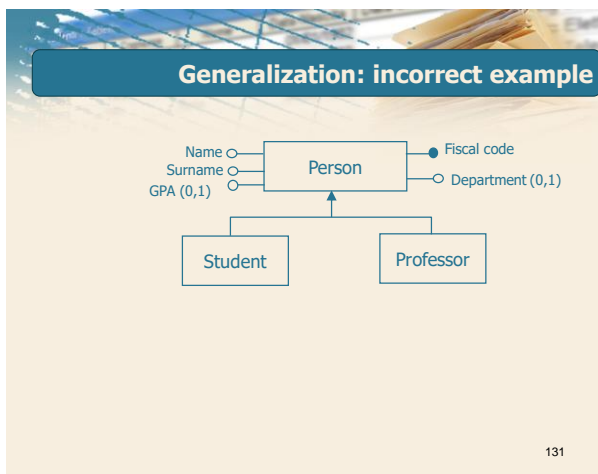
128



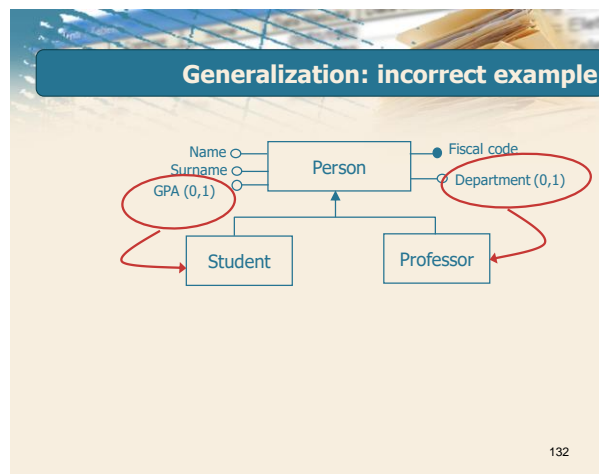
129



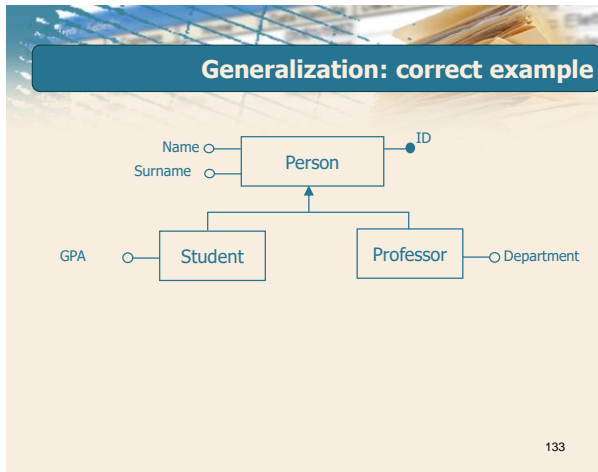
130



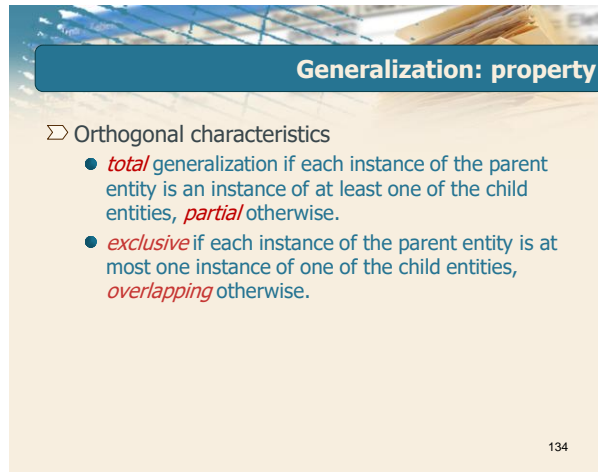
131



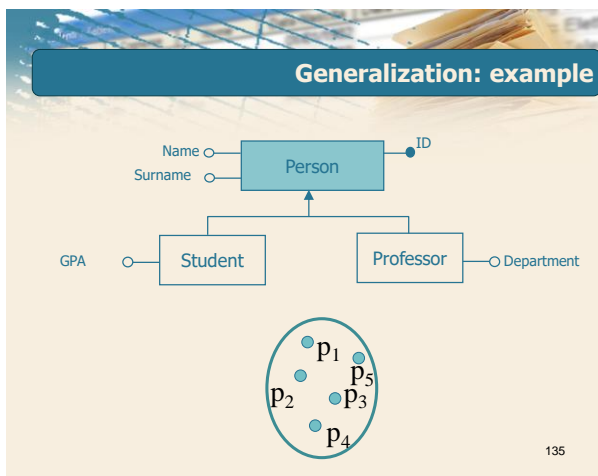
132



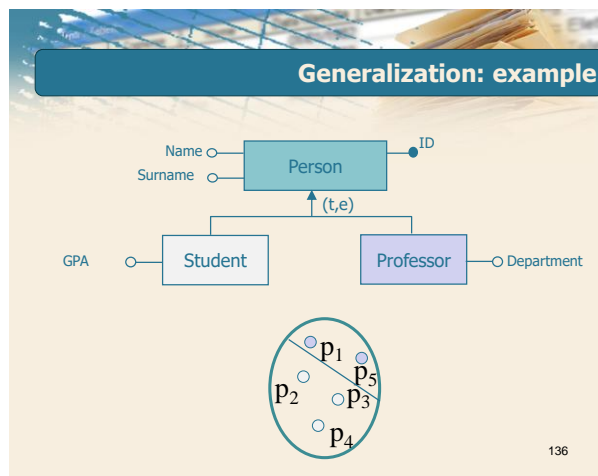
133



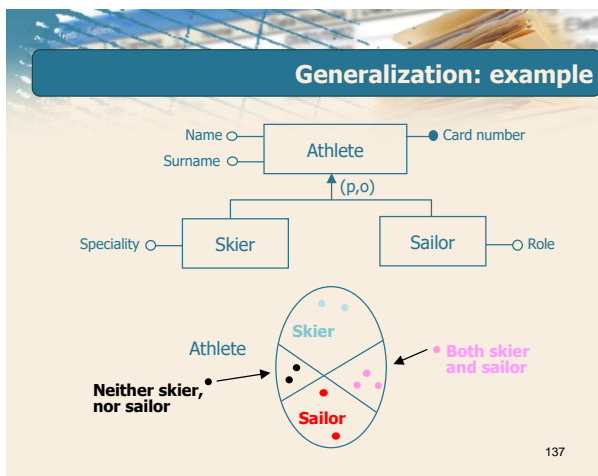
134



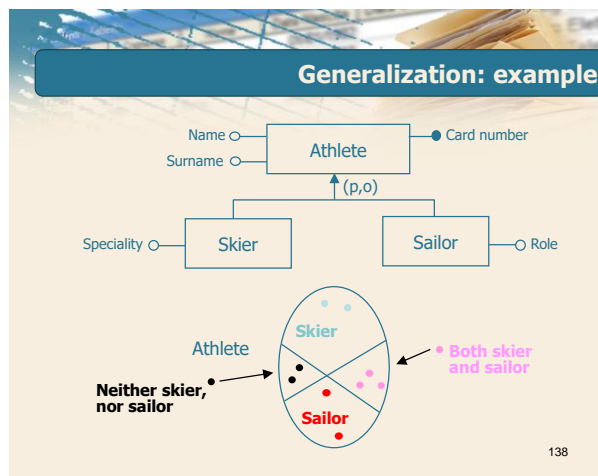
135



136



137



138

Subset

▹ Particular case of generalization with only one child entity

- the generalization is always partial and exclusive.

```

    graph BT
      Employee[Employee]
      FixedTermEmployee[Fixed-term Employee]
      FixedTermEmployee --> Employee
      FixedTermEmployee --- ContractEndDate((Contract end date))
    
```

139

139

The Entity-Relationship Model

Documentation of E-R schemes

140

140

Documentation of E-R schemes

```

    graph TD
      Student[Student] --- Exam{Exam}
      Professor[Professor] --- Exam
      Course[Course] --- Exam
      Course --- Holder{Holder}
      Time[Time] --- Exam
      Exam --- Student
      Exam --- Professor
      Exam --- Course
      Exam --- Time
      Holder --- Course
      Holder --- Professor
    
```

141

141

Documentation of E-R schemes

▹ Data Dictionary

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

142

142

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

143

143

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 teaches the course.	Course (1,1), Professor (0,N)	

144

144

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

145

145

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

146

146

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

147

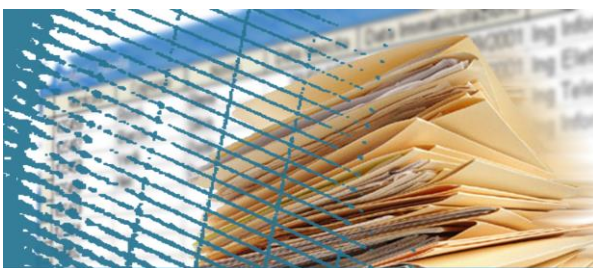
147

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

148

148



The Entity-Relationship Model

UML and E-R

149

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

150

150

UML and E-R

- ▷ Main characteristics 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)

151

151

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

152

152