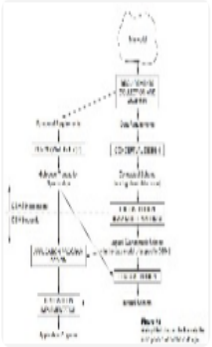
Entity-Relationship Data Model

- By

Jyoti Tryambake

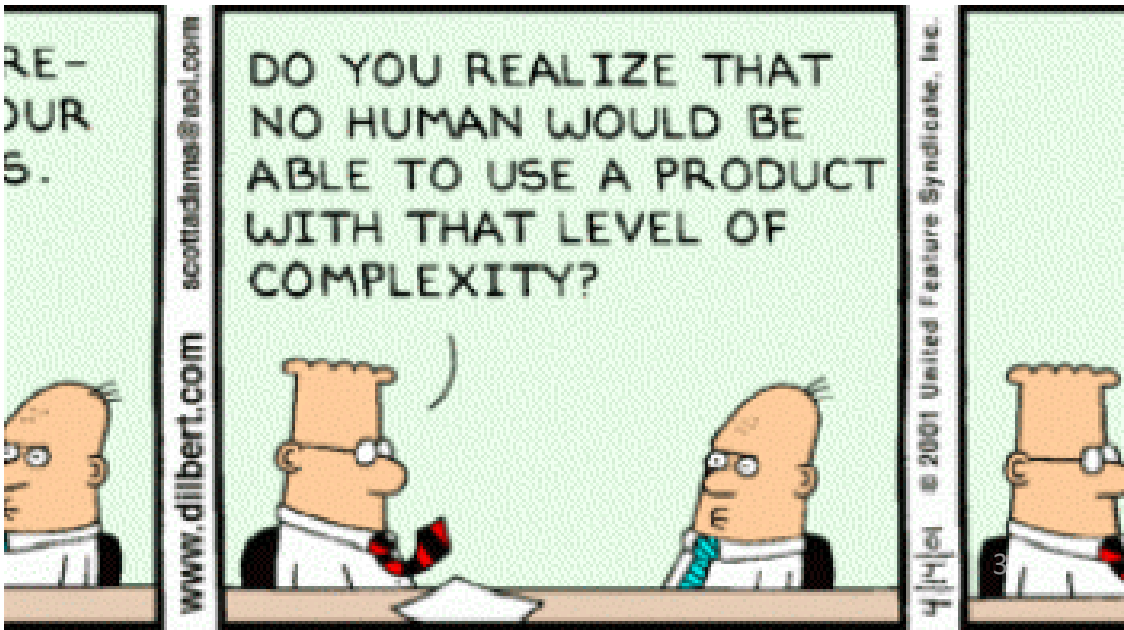
Entity Relationship Data Model

- 1. Introduction
  - Why?
- 2. Benefits of data modeling
  - Focusing on Essentials
  - Ease of communication and understanding
  - Product or process improvement
  - Exploring alternatives
- 3. Types of Models
  - 1. Descriptive
  - 2. Prescriptive
  - 3. Representative
- 4. Phases of Database Modelling
  - 
  - Requirements analysis
  - Conceptual Design
  - Logical Design
  - Physical Design
- 5. The ER Model
- 6. EER
  - Generalization, Specialization, Aggregation
- 7. Mapping of ER and EER to Relational Model

Data Modeling – An Introduction

User's requirements

Analysis



Data Modeling – An Introduction (cont..)

Design

Data Modeling - An Introduction (cont..)

- Is the process of creating a data model for the data to be stored in a Database
- Is a conceptual representation of - data objects, association between different data objects, the rules
- Emphasizes on what data is needed and how it should be organized instead of what operations need to be performed on the data

Benefits of Data Modeling

- Focusing on essentials
- Ease of communication and understanding
- Product or process improvement
- Exploring alternatives

Types of Data Models

Conceptual

WHAT the system contains

- This model is typically created by Business stakeholders and Data Architects.
- The purpose is to organize, scope and define business concepts and rules.
- Example- ERD

Logical

HOW the system should be implemented regardless of the DBMS

- This model is typically created by Data Architects and Business Analysts.
- The purpose is to developed technical map of rules and data structures.

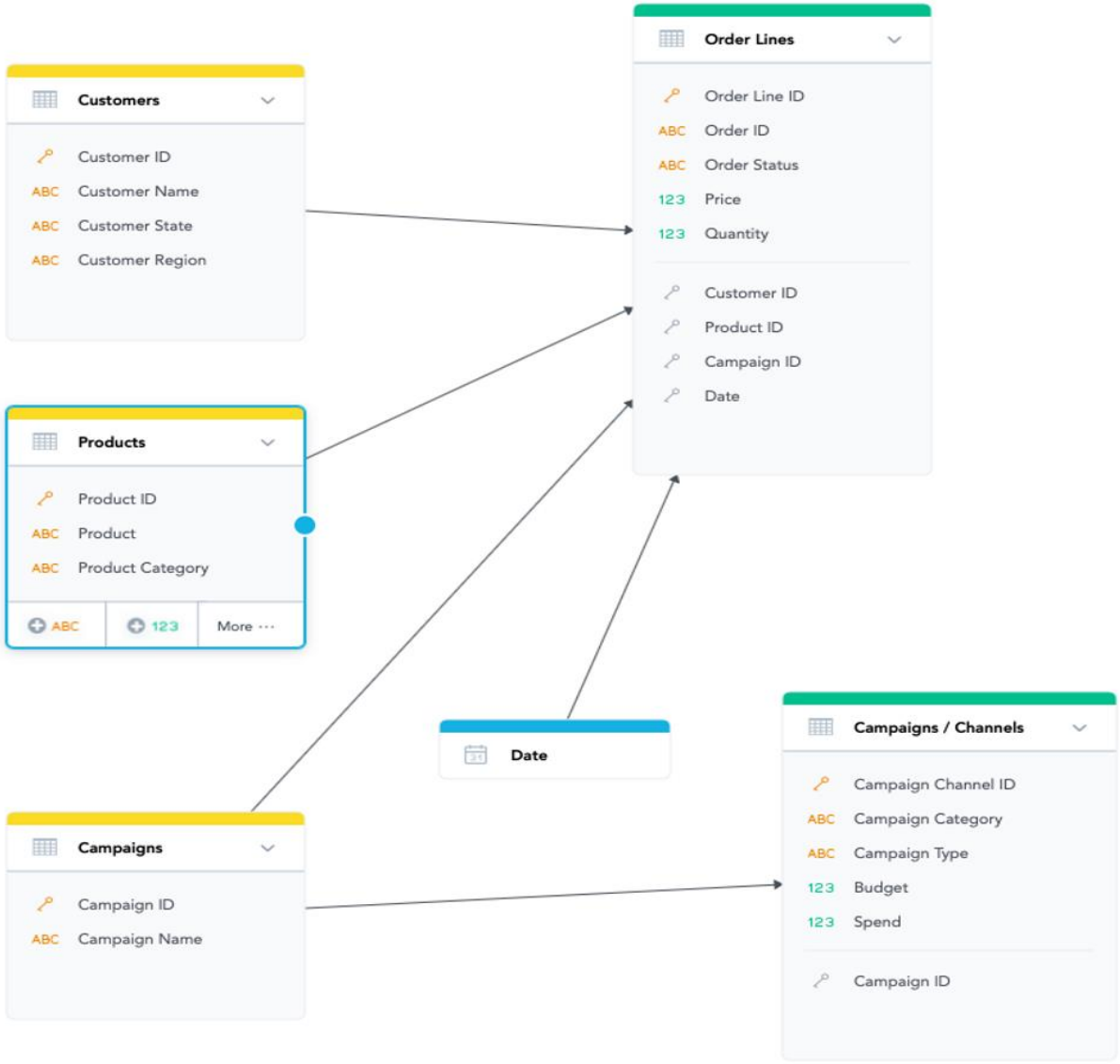
Physical

HOW the system will be implemented using a specific DBMS system.

- This model is typically created by DBA and developers.
- The purpose is actual implementation of the database.

Types of Data Models (cont.)

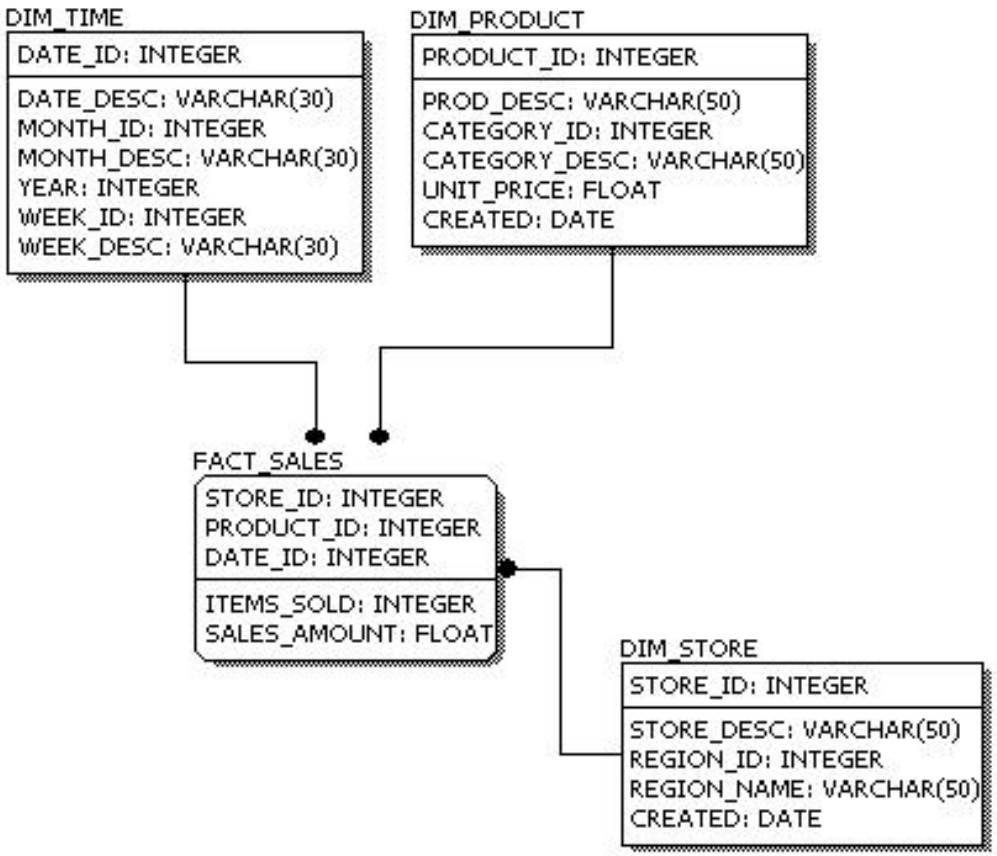
- Logical Data Model Structure:



Types of Data Models (cont.)

- Physical Data Model :

1. Convert entities into tables.
2. Convert relationships into foreign keys.
3. Convert attributes into columns.
4. Modify the physical data model based on physical constraints / requirements.



Phases of Database Modeling

- The first phase - Conceptual Modeling
  - Building an overall view
  - The objective is to represent the information structure of enterprise as accurate as possible
  - Also known as, *enterprise conceptual schema*
  - Known as requirements analysis phase of database development life cycle
  - Assist in communication between users and designers of db system

Phases of Database Modeling (cont.)

- The Second phase - Logical Design
  - Abstraction process - captures details of individual data values or DBMS to be used

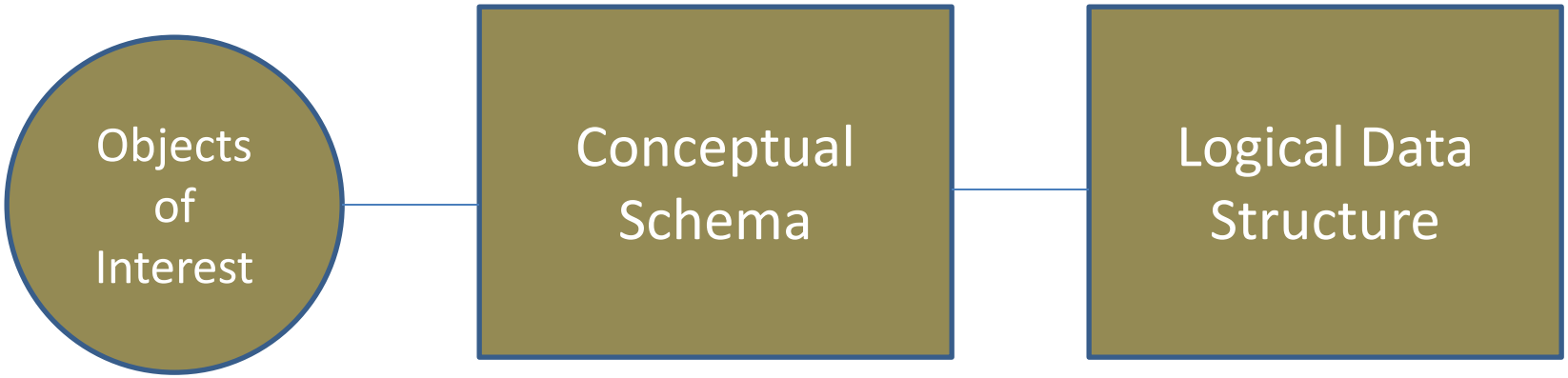
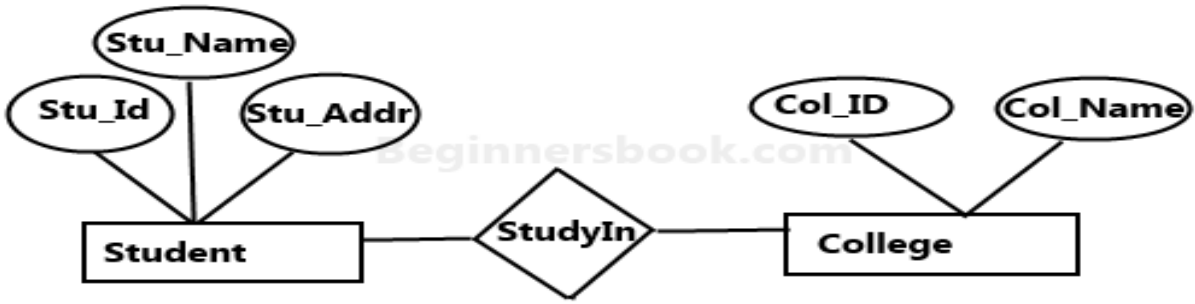


Fig. Two phases of database modeling

- The two types of Data Models techniques are
  - Entity Relationship (E-R) Model
  - UML (Unified Modeling Language)

The Entity-Relationship Model

- Widely used database modeling technique
- Simple and easy to understand
- Based on the concept that organization consist of people, facilities and objects
- For ex. Book publishing business involves books, authors, printers, distributors, editors, sales data etc.



Sample E-R Diagram

The Entity-Relationship Model (cont.)

- Entity
  - Real or abstract object that can be distinctly identified and it must be something about which information is important
  - Ex. Person, a place, a building or an event or an order
- Entity sets
  - Is a set of objects called entity instances or entity occurrences
  - Entity instance is an individual object of a given entity type
  - Ex. All cricket players may constitute an entity set Player

The Entity-Relationship Model (cont.)

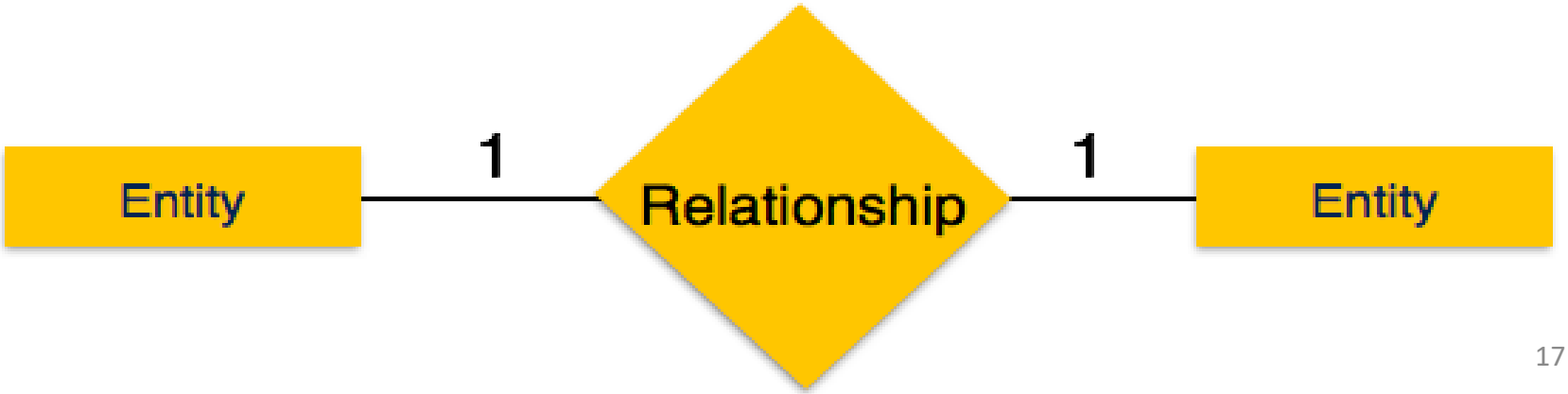
- Entity
  - Entities are represented by means of rectangles. Rectangles are named with the entity set they represent.
  - For ex. Shown below;

The Entity-Relationship Model (cont.)

- Relationships
  - Is an association among entities
  - A subset of all possible combinations of entity instances from each entity type participating in the relationship
  - The entities participating in relationship may be of same type or of different type
  - Often relationship names are verbs and entity names are noun. For ex. Student borrows a book

The Entity-Relationship Model (cont.)

- Relationships
  - Relationships are represented by diamond-shaped box. Name of the relationship is written inside the diamond-box.
  - All the entities (rectangles) participating in a relationship, are connected to it by a line.
  - Consider binary relationship R between two entity types E1 and E2

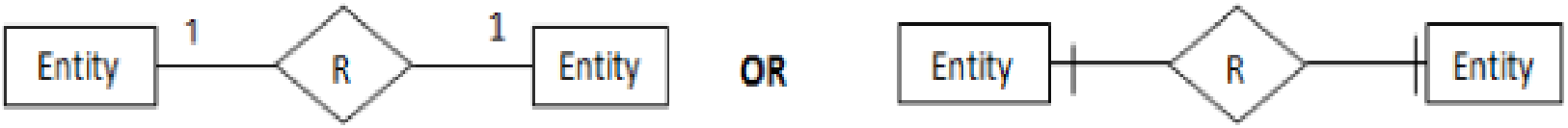


The Entity-Relationship Model (cont.)

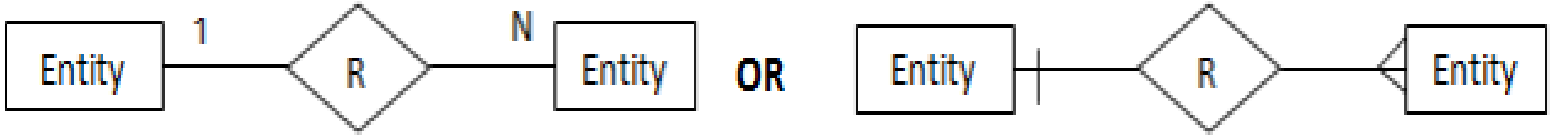
- Relationships
  - Most common relationships are binary relationships between two entity sets
  - Degree of relationship – number of entity sets associated in the relationship
    - Types – Unary -1, Binary – 2, Ternary – 3
  - Cardinality - is the number of instance of an entity from a relation that can be associated with the relation, also called as constraints on two relationships or mappings

The Entity-Relationship Model (cont.)

- Relationships
  - – Cardinality –
    - **One-to-one** – When only one instance of an entity is associated with the relationship, it is marked as '1:1'.

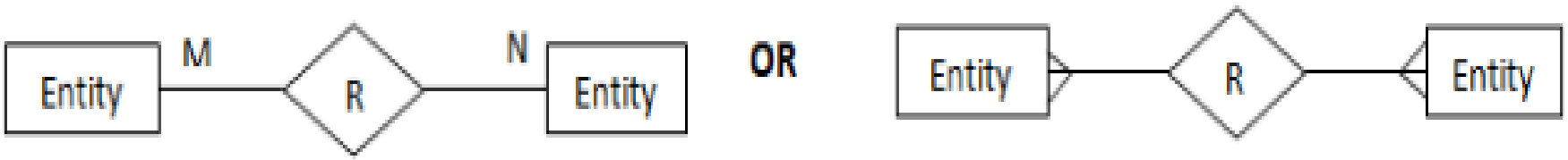


    - **One-to-many** – When more than one instance of an entity is associated with a relationship, it is marked as '1:N'.



The Entity-Relationship Model (cont.)

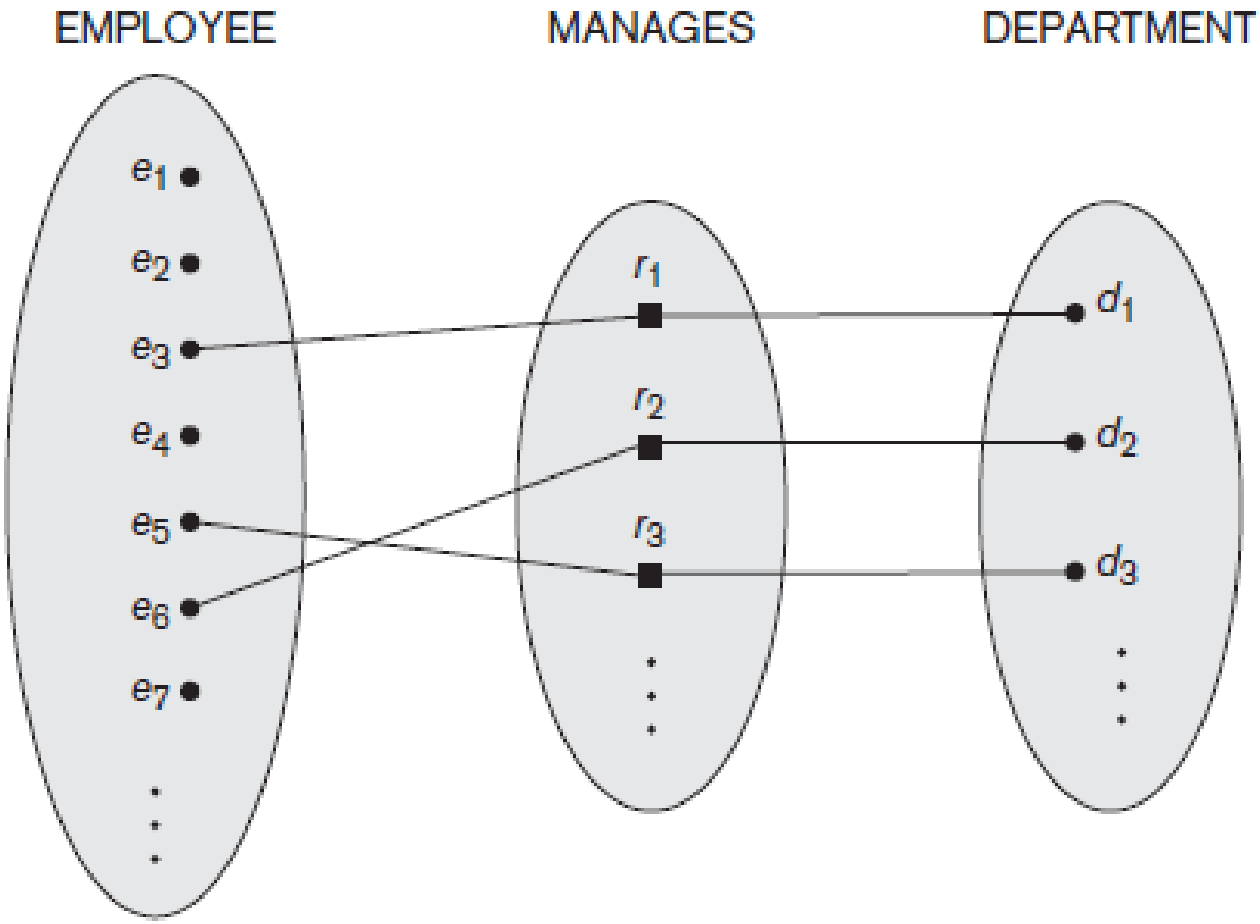
- Relationships
  - – Cardinality –
    - **Many-to-one** – When more than one instance of entity is associated with the relationship, it is marked as 'N:1'.
    - **Many-to-many** – The following image reflects that more than one instance of an entity on the left and more than one instance of an entity on the right can be associated with the relationship.



The Entity-Relationship Model (cont.)

- Cardinality Examples: - 1:1

Figure 7.12
A 1:1 relationship, MANAGES.



The Entity-Relationship Model (cont.)

- Cardinality Examples: M:N

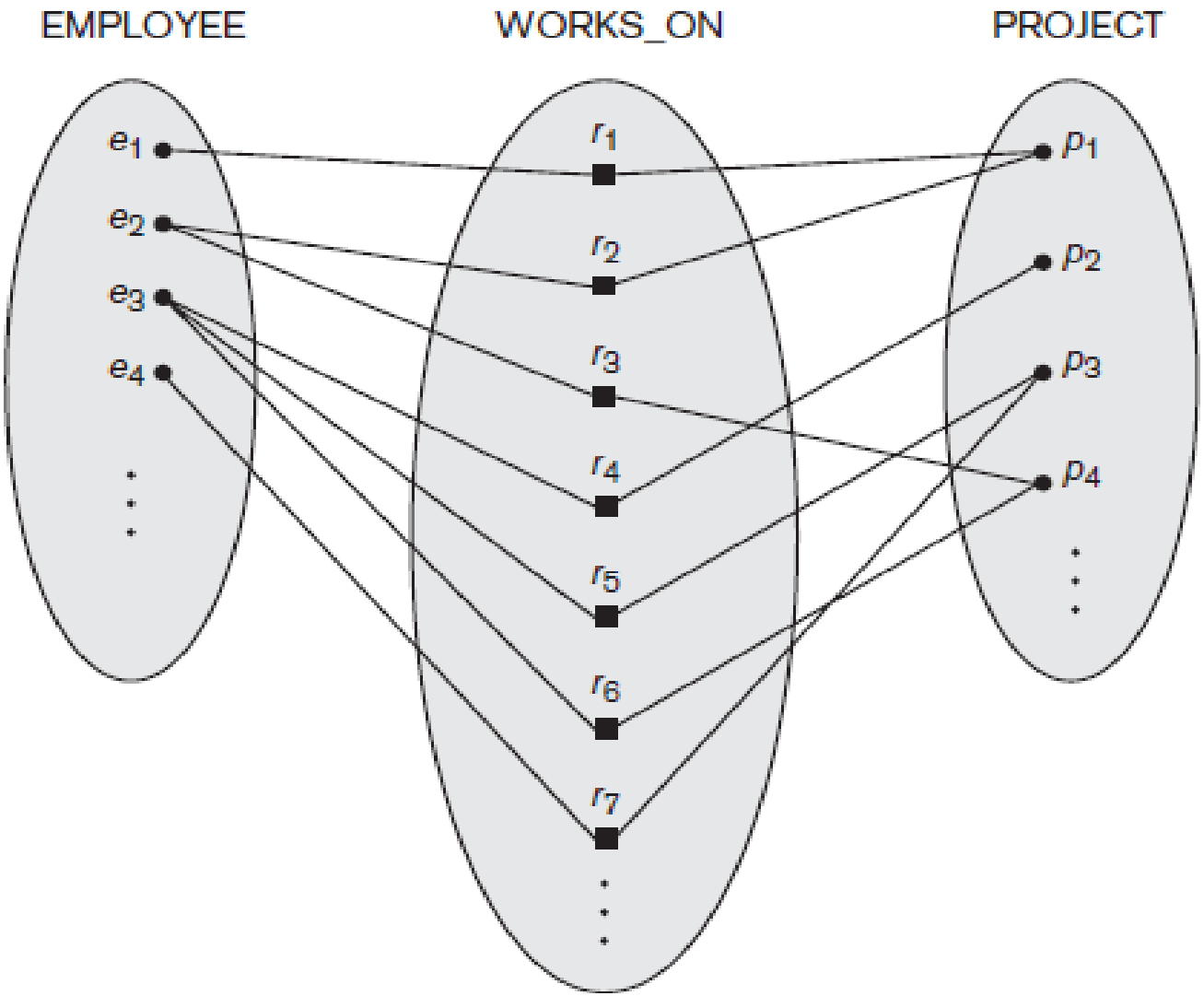


Figure 7.13
An M:N relationship, WORKS_ON.

The Entity-Relationship Model (cont.)

- Cardinality Examples: M:1

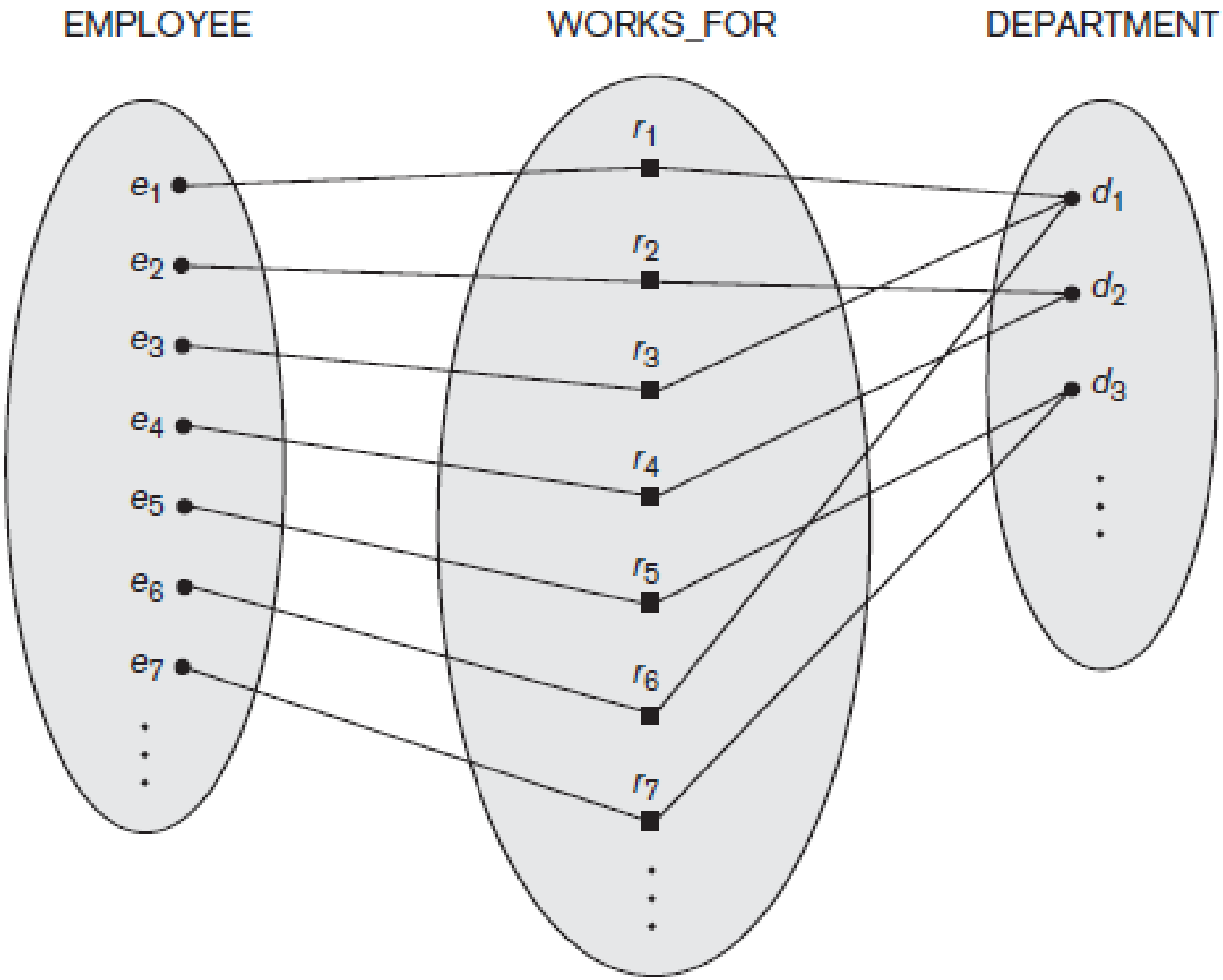


Figure 7.9
Some instances in the WORKS_FOR relationship set, which represents a relationship type WORKS_FOR between EMPLOYEE and DEPARTMENT.

The Entity-Relationship Model (cont.)

- As per korth,

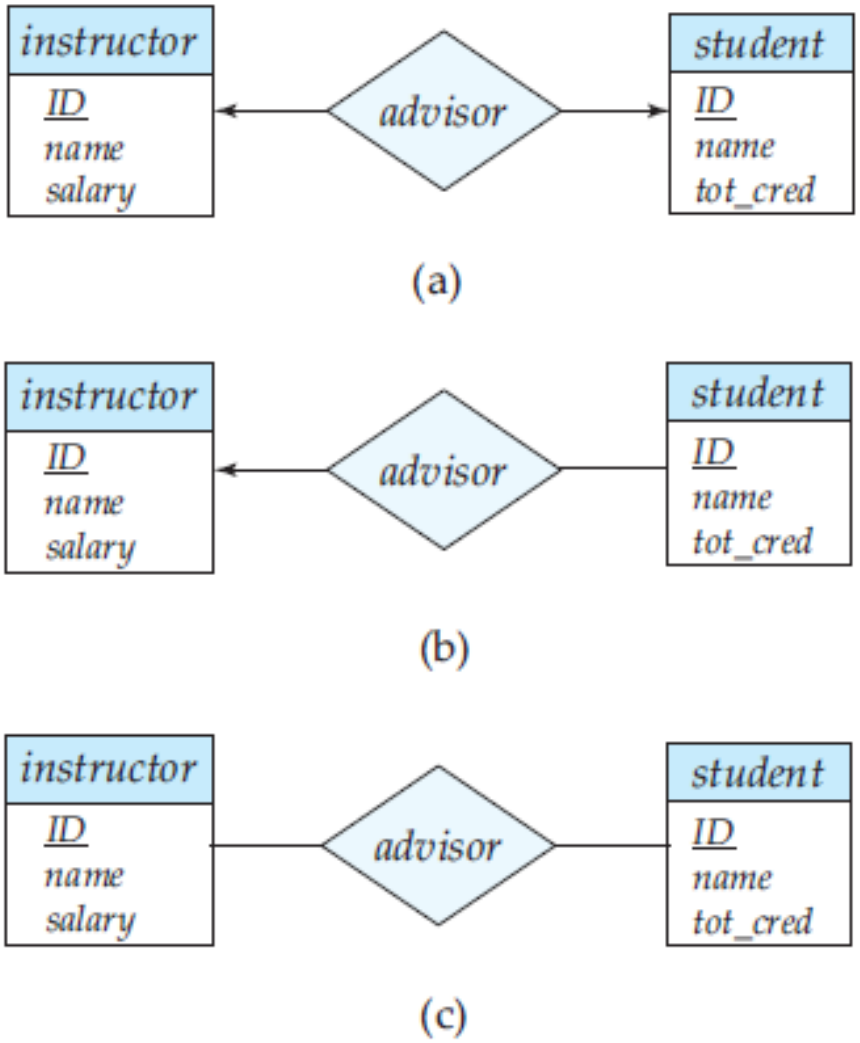


Figure 7.9 Relationships. (a) One-to-one. (b) One-to-many. (c) Many-to-many.

The Entity-Relationship Model (cont.)

- Cardinality:
  - E-R diagrams also provide a way to indicate more complex constraints on the number of times each entity participates in relationships in a relationship set.
  - A line may have an associated minimum and maximum cardinality, shown in the form *l..h*, where *l* is the minimum and *h* the maximum cardinality.
  - A minimum value of 1 indicates total participation of the entity set in the relationship set; -
    - each entity in the entity set occurs in at least one relationship in that relationship set.
  - A maximum value of 1 indicates that the entity participates in at most one relationship
  - a maximum value ∗ indicates no limit

The Entity-Relationship Model (cont.)

- Cardinality:

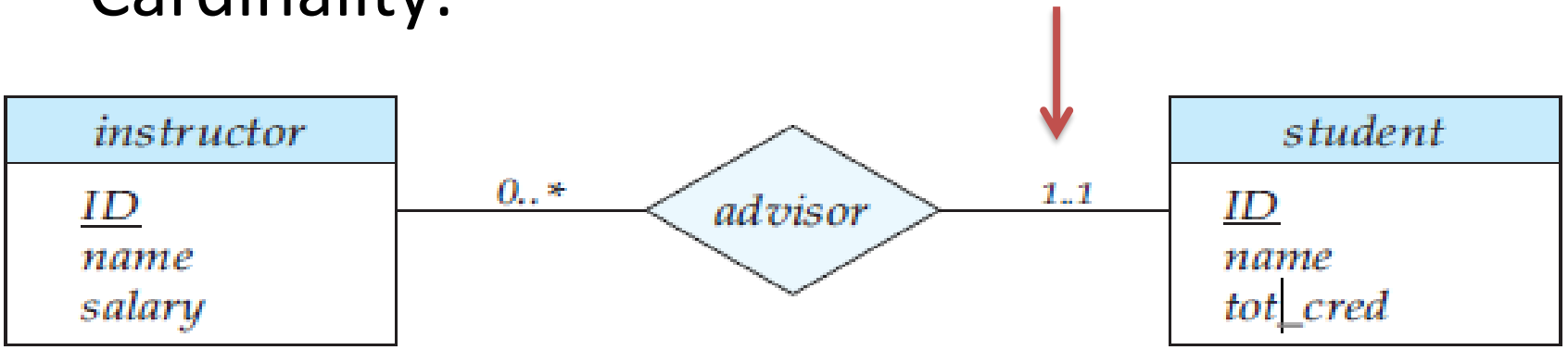


Figure 7.10 Cardinality limits on relationship sets.

- The line between *advisor* and *student* has a cardinality constraint of 1..1
  - meaning the minimum and the maximum cardinality are both 1
  - That is, each student must have exactly one advisor

The Entity-Relationship Model (cont.)

- Cardinality:

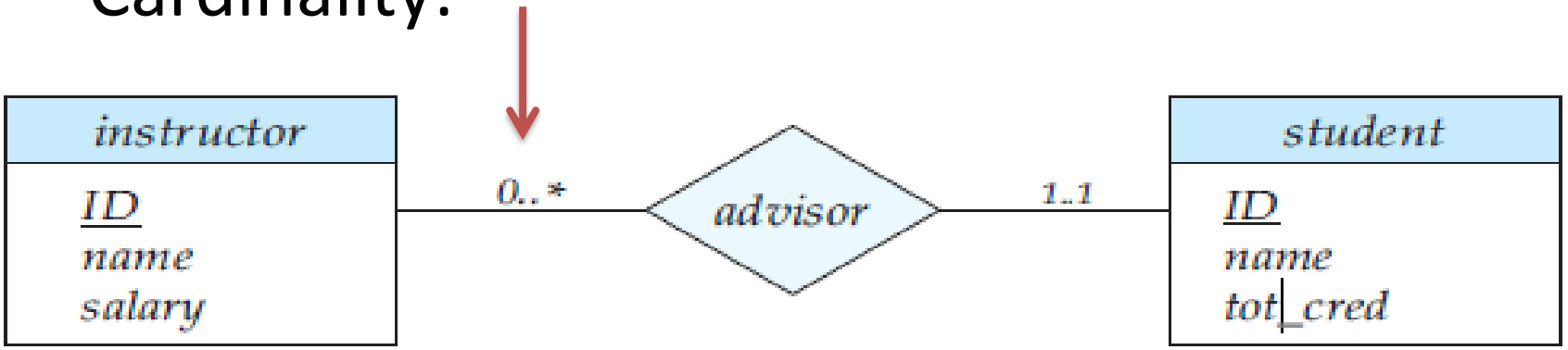


Figure 7.10 Cardinality limits on relationship sets.

- The limit 0..∗ on the line between *advisor* and *instructor* indicates that
  - an instructor can have zero or more students
- the relationship *advisor* is one-to-many from *instructor* to *student*
- the participation of *student* in *advisor* is total, implying that a student must have an advisor

The Entity-Relationship Model (cont.)

- Cardinality:
  - 2nd Example:
  - A department has exactly one manager and an employee can manage at most one department.
    - Specify (0,1) for participation of EMPLOYEE in MANAGES
    - Specify (1,1) for participation of DEPARTMENT in MANAGES

The Entity-Relationship Model (cont.)

- Cardinality:
  - 2nd Example:
  - An employee can work for exactly one department but a department can have any number of employees.
    - Specify (1,1) for participation of EMPLOYEE in WORKS_FOR
    - Specify (1,N) for participation of DEPARTMENT in WORKS_FOR

The Entity-Relationship Model (cont.)

- Role Names and Recursive Relationships:
  - Signifies the role that a participating entity from the entity type plays in each relationship instance and helps to explain what the relationship means
  - For ex. WORKS_FOR relationship type, EMPLOYEE plays the role of *employee* or *worker* and DEPARTMENT plays the role of *department* or *employer*

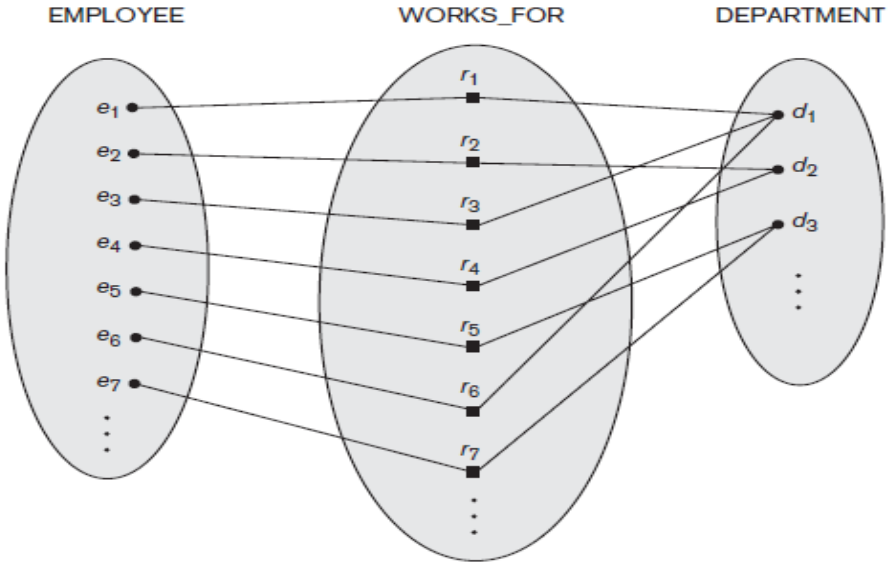


Figure 7.9
Some instances in the WORKS_FOR relationship set, which represents a relationship type WORKS_FOR between EMPLOYEE and DEPARTMENT.

The Entity-Relationship Model (cont.)

- Role Names and Recursive Relationships:
  - One entity plays different roles with one entity
  - One entity plays different roles with different entities
  - One role plays different roles with itself (recursive)

The Entity-Relationship Model (cont.)

- Role Names and Recursive Relationships (cont.):
  - the role name becomes essential for distinguishing the meaning of the role when the *same* entity type participates more than once in a relationship type in *different roles*.
  - Such relationship types are called **recursive relationships**

The Entity-Relationship Model (cont.)

- Role Names and Recursive Relationships (cont.):
  - **Recursive relationships**

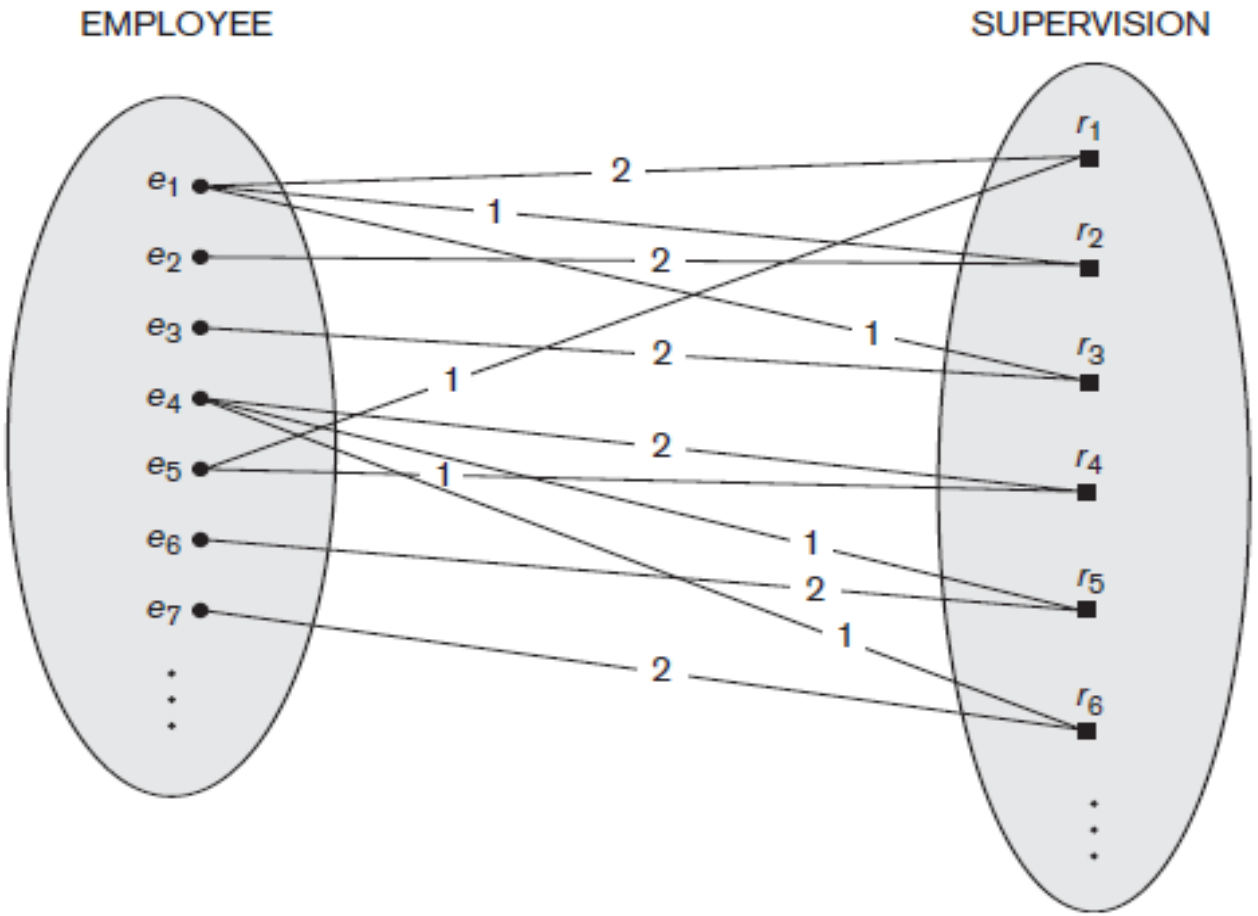


Figure 7.11
A recursive relationship SUPERVISION between EMPLOYEE in the *supervisor* role (1) and EMPLOYEE in the *subordinate* role (2).

1 – Supervisor
2- Supervisee

The Entity-Relationship Model (cont.)

- Role Names and Recursive Relationships (cont.):
- **Recursive relationships (cont.)**
  - **As per diagram,** The SUPERVISION relationship type relates an employee to a supervisor, where both employee and supervisor entities are members of the same EMPLOYEE entity set.
  - the EMPLOYEE entity type *participates twice* in SUPERVISION: once in the role of *supervisor* (or *boss*), and once in the role of *supervisee* (or *subordinate*).
  - Each relationship instance r_i in SUPERVISION associates two employee entities e_j and e_k, one of which plays the role of supervisor and the other the role of supervisee

The Entity-Relationship Model (cont.)

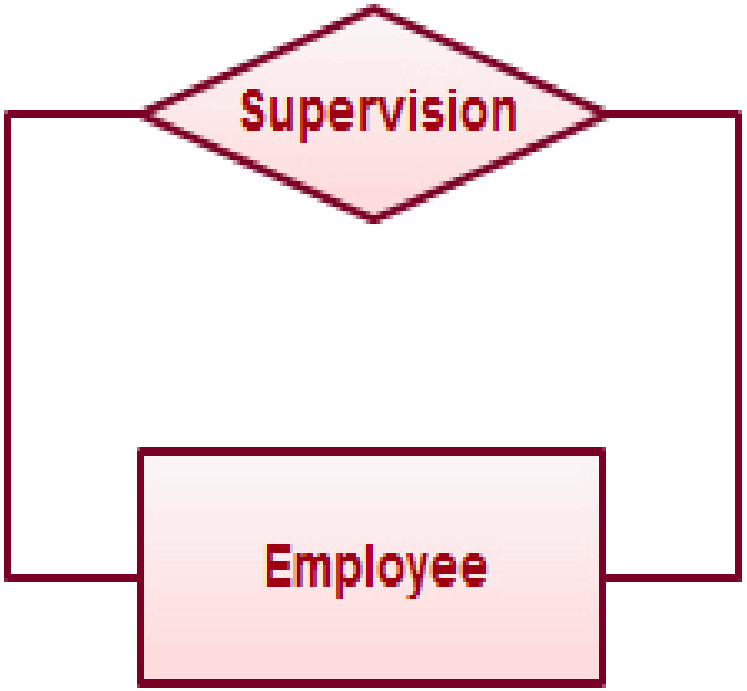
- Role Names and Recursive Relationships (cont.):
- **Recursive relationships (cont.)**
  - **As per diagram,** '1' represent the supervisor role, and those marked '2' represent the supervisee role
  - hence, e_1 supervises e_2

    e_3, e_4 supervises e_6 and e_7,

    and e_5 supervises e_1 and e_4.

The Entity-Relationship Model (cont.)

- Role Names and Recursive Relationships (cont.):
- **Recursive relationships (cont.)**
  - ERD Notation:



The Entity-Relationship Model (cont.)

- Participation Constraints and Existence Dependencies
  - specifies whether the existence of an entity depends on its being related to another entity via the relationship type
  - This constraint specifies the *minimum* number of relationship instances that each entity can participate in, and called the minimum cardinality constraint.
  - There are two types of participation constraints—
    - Total Participation
    - Partial Participation

The Entity-Relationship Model (cont.)

- Participation Constraints and Existence Dependencies
  - Total Participation : known as **existence dependency**

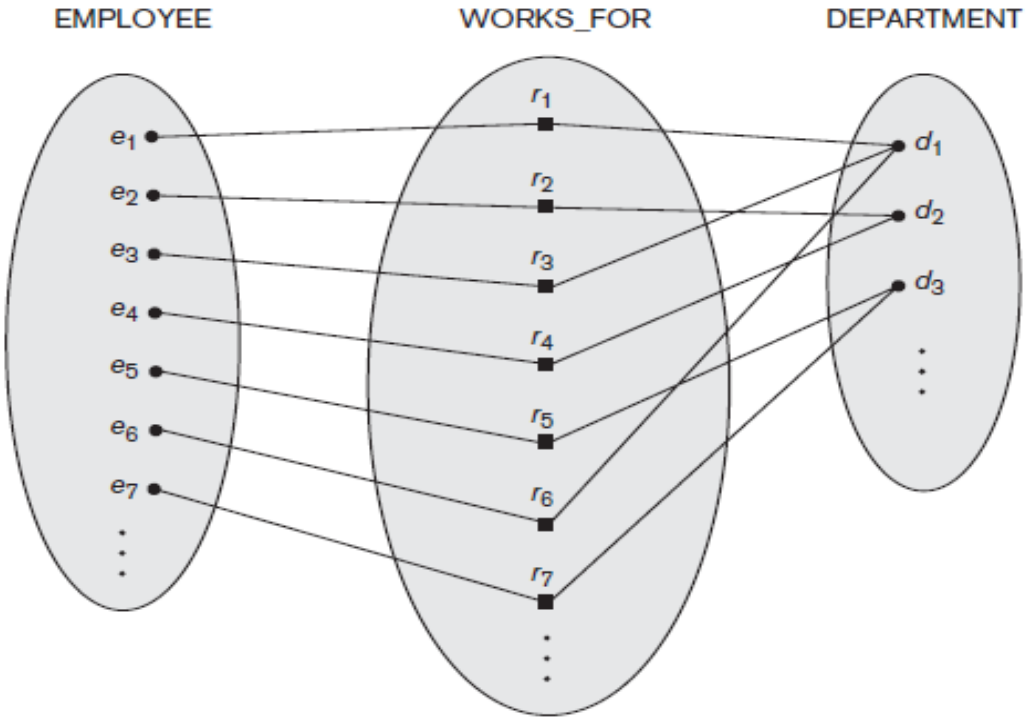
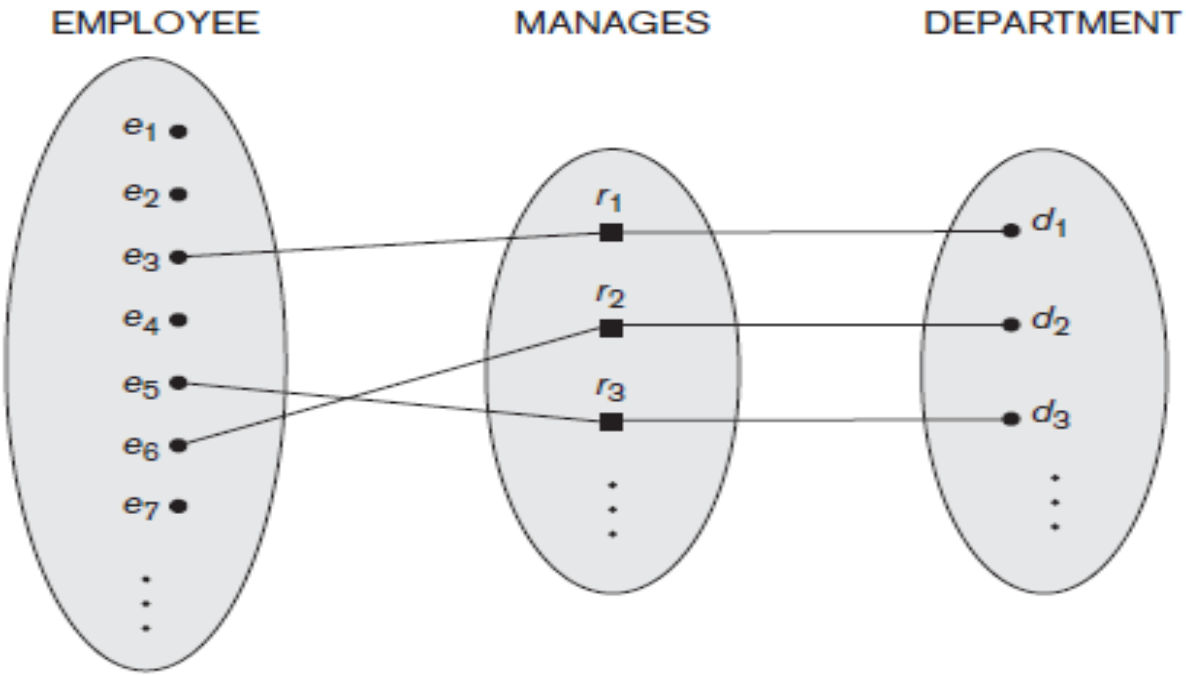


Figure 7.9
Some instances in the WORKS_FOR relationship set, which represents a relationship type WORKS_FOR between EMPLOYEE and DEPARTMENT.

- every entity in *the total set* of employee entities must be related to a department entity via WORKS_FOR

The Entity-Relationship Model (cont.)

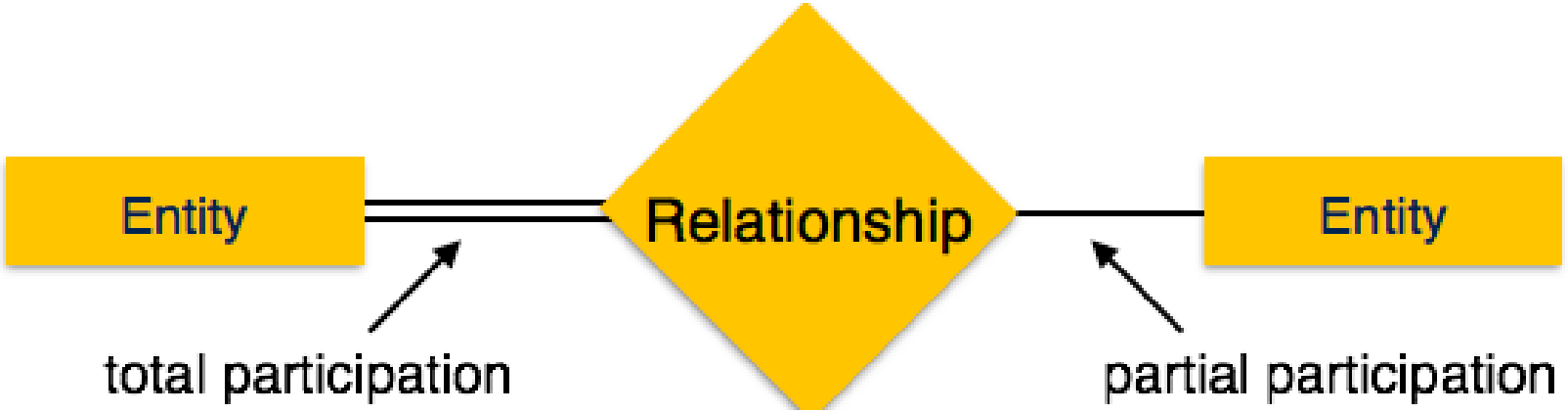
- Participation Constraints and Existence Dependencies
  - Partial Participation :



  - *some* or *part of the set of* employee entities are related to some department entity via MANAGES, but not necessarily all

The Entity-Relationship Model (cont.)

- Participation Constraints and Existence Dependencies
  - **Total Participation** – Each entity is involved in the relationship. Total participation is represented by double lines
  - **Partial participation** – Not all entities are involved in the relationship. Partial participation is represented by single lines

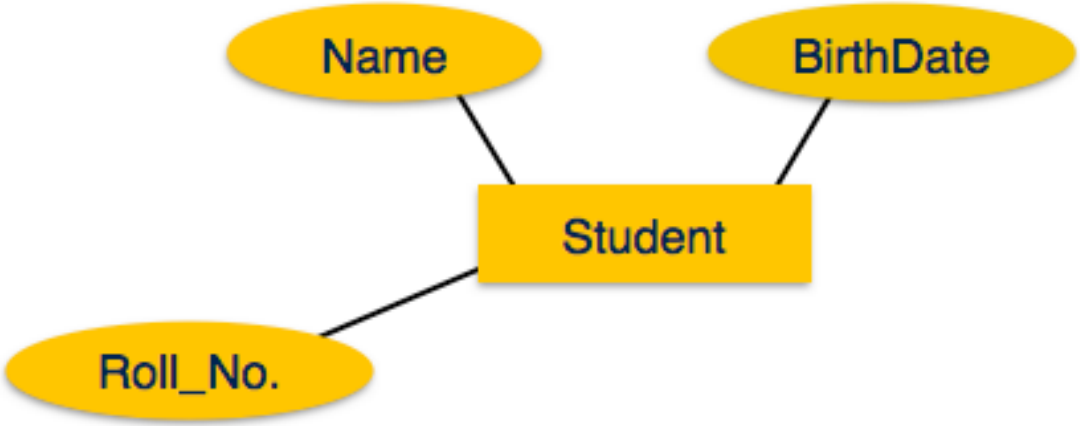


The Entity-Relationship Model (cont.)

- Participation Constraints and Existence Dependencies
  - Structural Constraints = Cardinality ratio + Participation Constraints

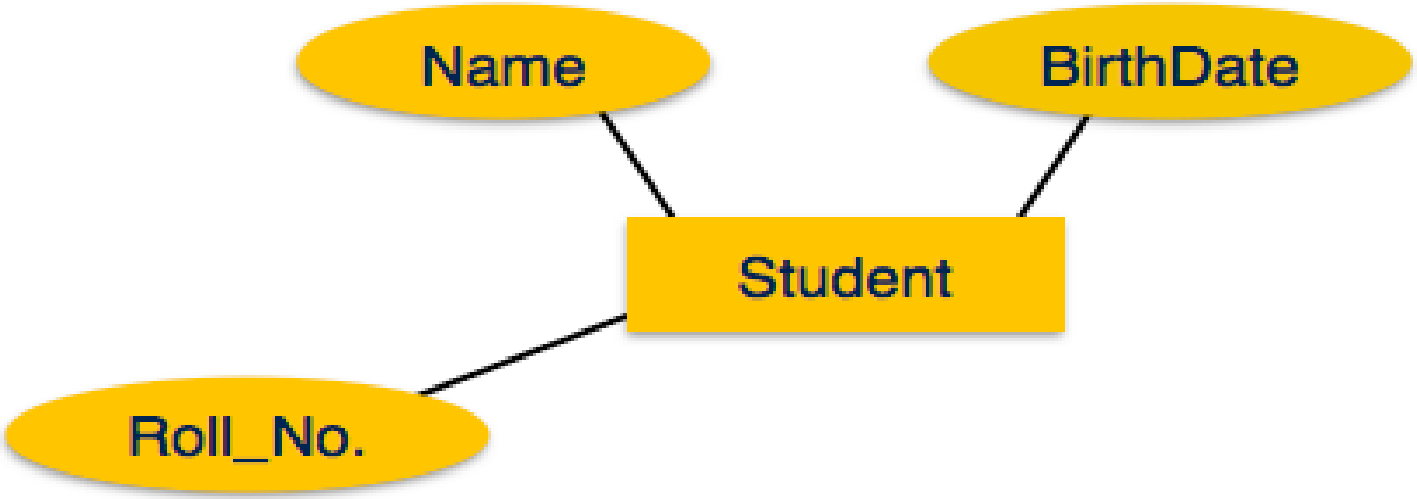
The Entity-Relationship Model (cont.)

- Attributes:
  - Attributes are the properties of entities. Attributes are represented by means of ellipses.
  - Every ellipse represents one attribute and is directly connected to its entity (rectangle).



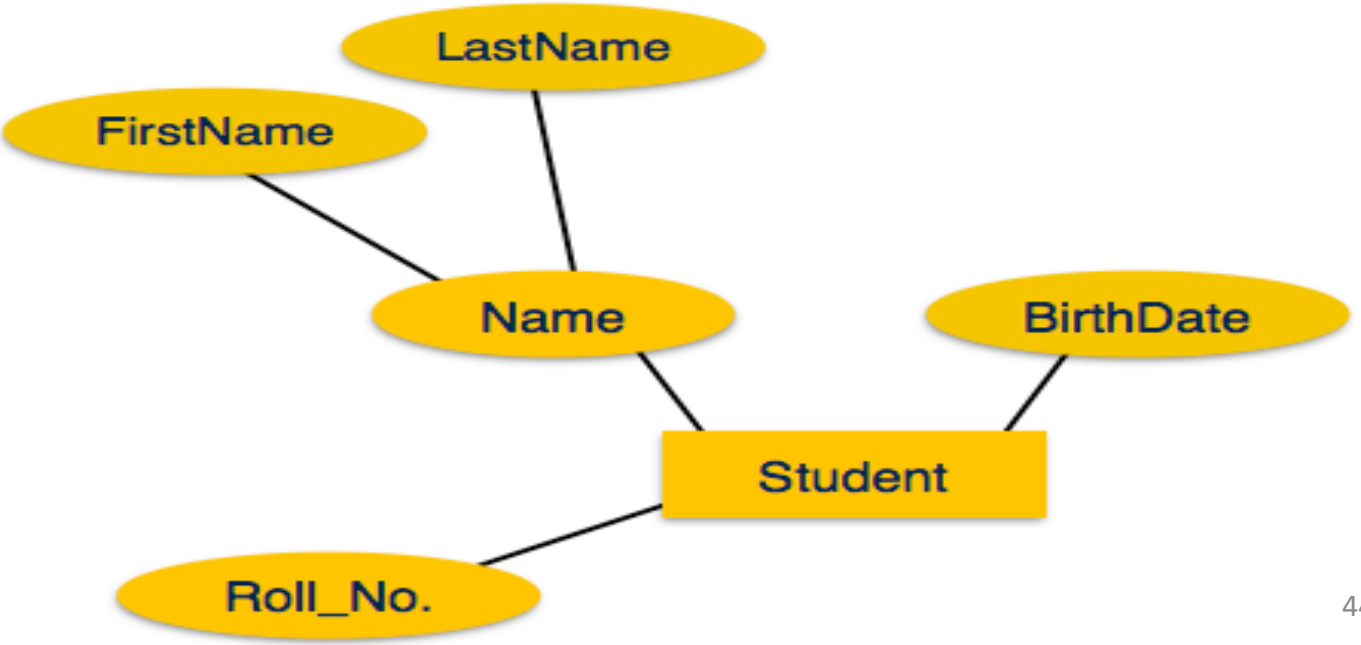
The Entity-Relationship Model (cont.)

- Attributes (Types) –
  - Simple or Composite Attributes:
    - Simple/Atomic – can not be divided into smaller component
      - Ex. Country, place of birth, ID etc.



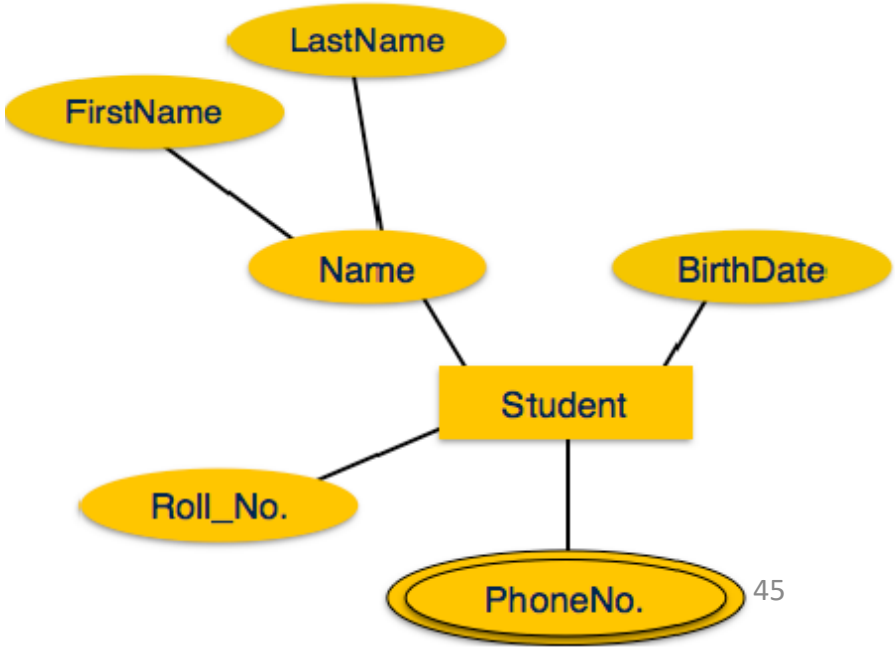
The Entity-Relationship Model (cont.)

- Attributes (Types) –
  - Simple or Composite Attributes:
    - Composite – divided into smaller components
      - Ex. Name (First name, last name), date of birth (day, month and year), address (street number, street name, city, PIN)



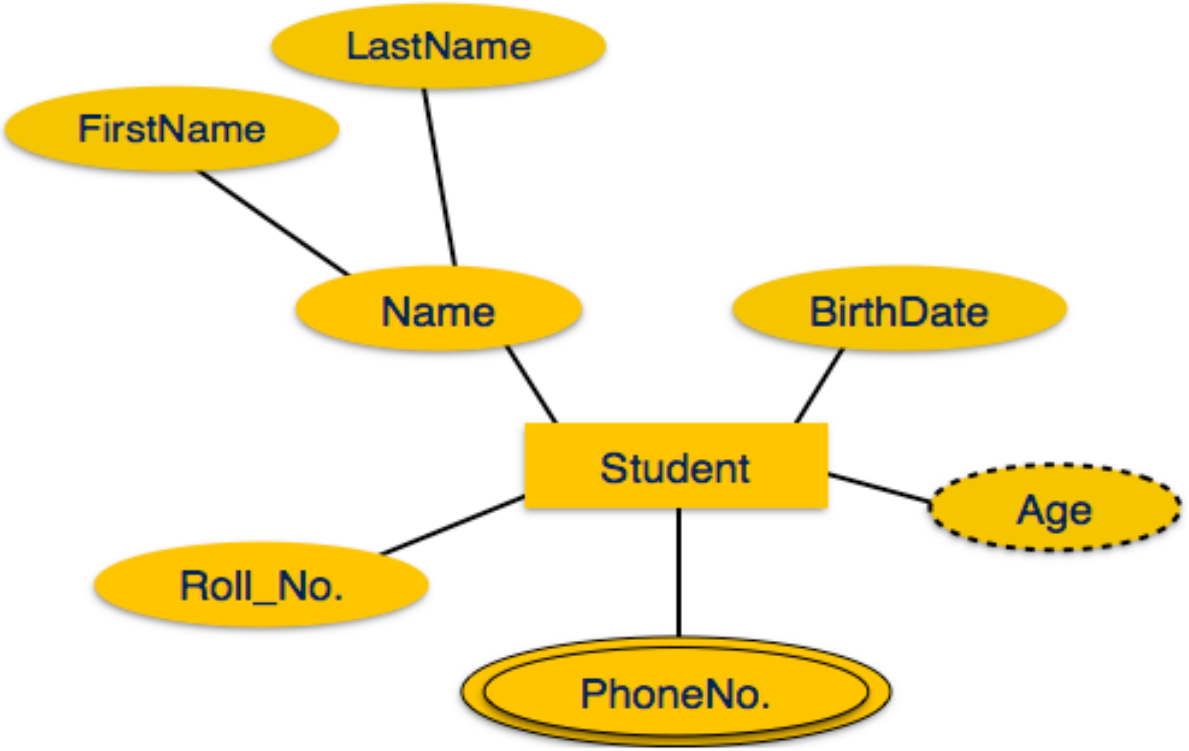
The Entity-Relationship Model (cont.)

- Attributes (Types) –
  - Single valued or Multivalued:
    - Single valued –
      - Ex. Age, ID etc.
    - Multivalued – depicted by double ellipse
      - Ex. Address, phone number, no_of_courses (student enrolled in)



The Entity-Relationship Model (cont.)

- Attributes (Types) –
  - Stored or Derived:
    - **Derived** attributes are depicted by dashed ellipse.
    - Ex. Age is a derived attribute if database has stored attribute – date_of_birth



The Entity-Relationship Model (cont.)

- Attributes of Relationship Types–
- A relationship type can have attributes (descriptive attribute):
  - For example, include the date on which a manager started managing a department via an attribute Start_date for the MANAGES relationship type
  - attributes of 1:1 or 1:N relationship types can be migrated to one of the participating entity types

    - 1:1 – either side
    - 1:N – many side
    - M:N – a separate relation should be created for relationship along with attribute

The Entity-Relationship Model (cont.)

- Strong Entity and Weak Entity:

- Strong Entity:
  - is the one whose existence does not depend on the existence of any other entity in a schema.
  - denoted by a **single rectangle**
  - always has the **primary key** in the set of attributes that describes the strong entity
  - It indicates that each entity in a strong entity set can be uniquely identified.

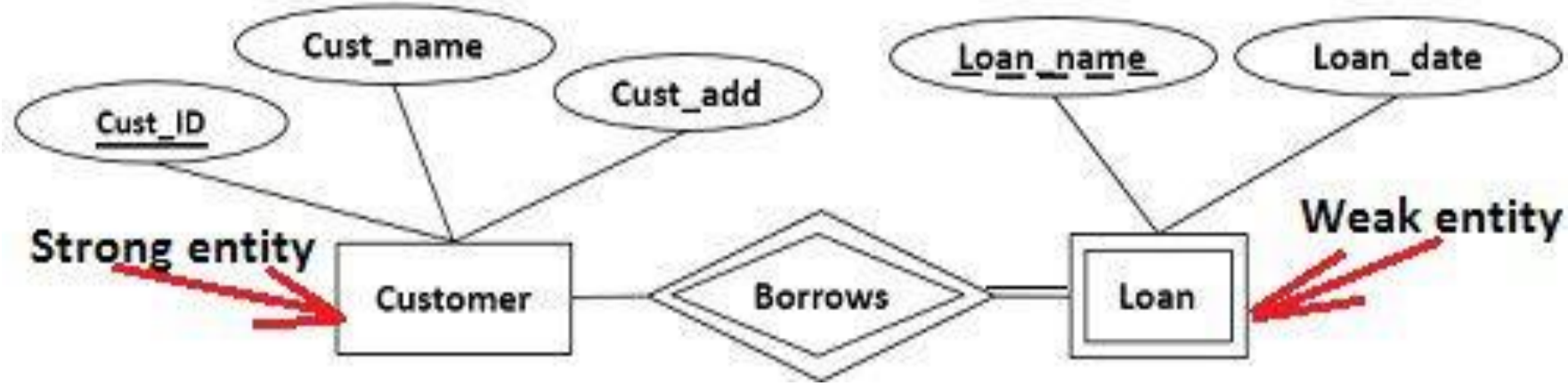
The Entity-Relationship Model (cont.)

- Strong Entity and Weak Entity:

- Weak Entity:
  - the one that depends on its owner entity i.e. a strong entity for its existence.
  - denoted by the **double rectangle**
  - do **not** have the **primary key** instead it has a **partial key** that uniquely discriminates the weak entities
  - The **primary key of a weak entity** is a composite key formed from the **primary key of the strong entity** and **partial key of the weak entity**.

The Entity-Relationship Model (cont.)

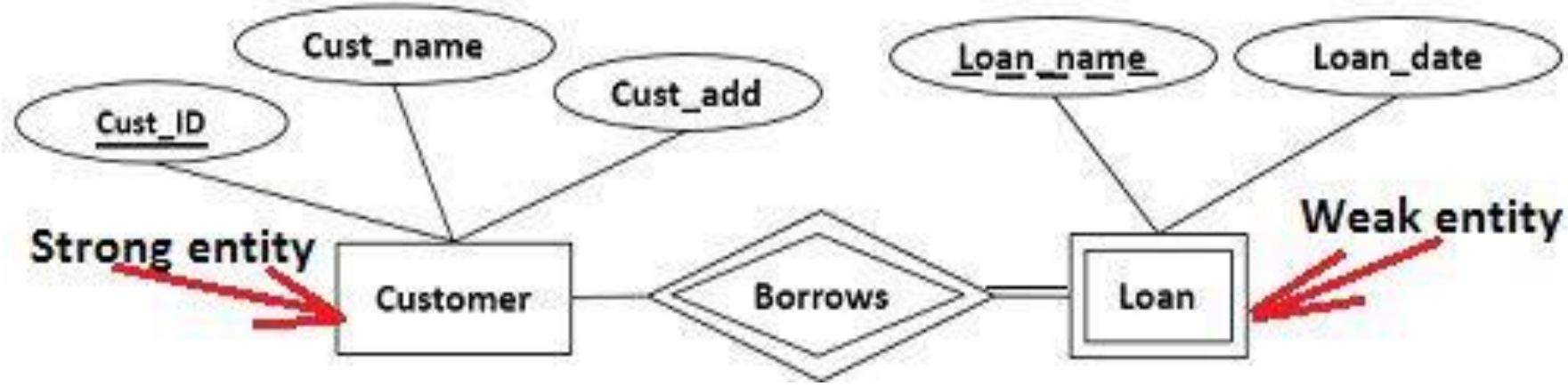
- Strong Entity and Weak Entity:



- A strong entity holds the relationship with the weak entity via an **Identifying Relationship**, which is denoted by double diamond in the ER diagram

The Entity-Relationship Model (cont.)

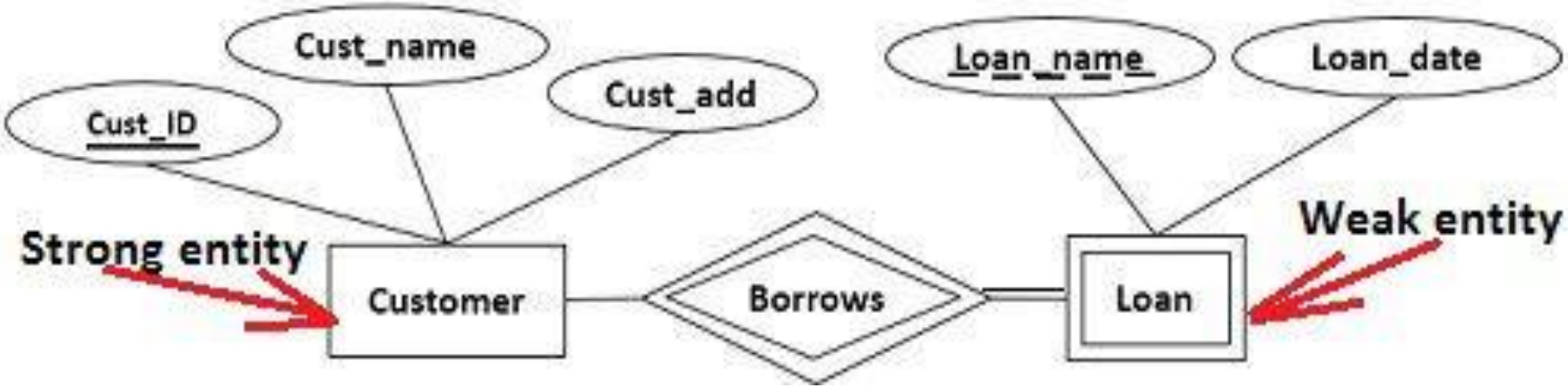
- Strong Entity and Weak Entity:



  - for each loan, there should be at least one borrower otherwise that loan would not be listed in Loan entity set
  - even if a customer does not borrow any loan it would be listed in Customer entity set
  - Conclusion: a customer entity does not depend on a loan entity

The Entity-Relationship Model (cont.)

- Strong Entity and Weak Entity:



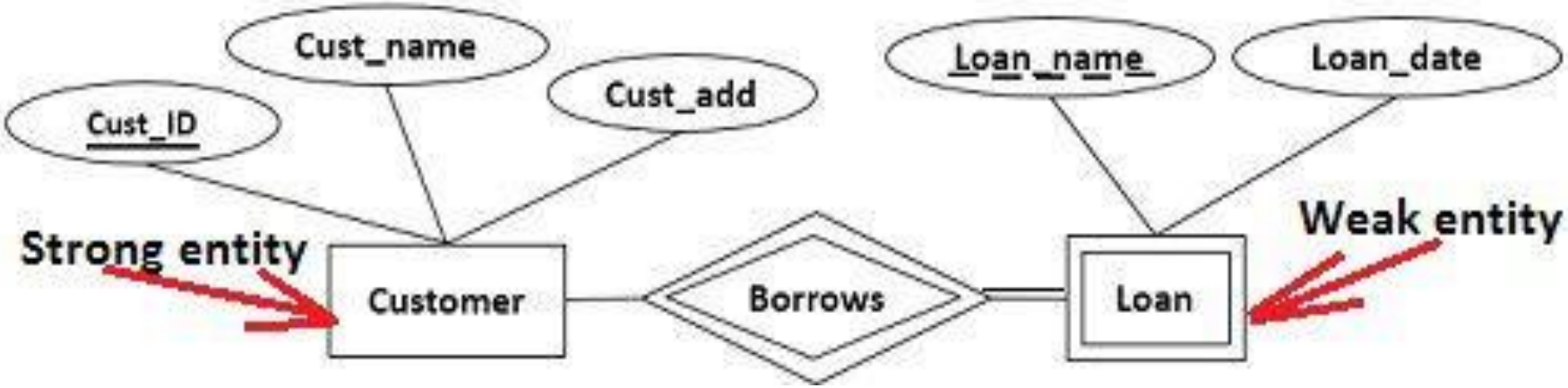
Cust_ID (primary key) which uniquely identify each entity in Customer Entity set

Cust_ID	Cust_name	Cust_add
101	John	Xyzzy
103	Ruby	Pqr
109	John	Uvfw

Customer Entity Set

The Entity-Relationship Model (cont.)

- Strong Entity and Weak Entity:



Loan_name, the partial key of the weak entity and Cust_ID primary key of customer entity makes primary key of loan entity

Loan_name	Loan_date	Amount
Home	20/11/2015	20000
Education	5/10/2015	10000
Home	20/11/2015	20000

Loan Entity Set

The Entity-Relationship Model (cont.)

- Relationships of Higher Degree
  - Relationship types of degree 2 are called binary
  - Relationship types of degree 3 are called ternary and of degree n are called n-ary
  - In general, an n-ary relationship is not equivalent to n binary relationships
  - Constraints are harder to specify for higher-degree relationships (n > 2) than for binary relationships

The Entity-Relationship Model (cont.)

- Discussion of n-ary relationships (n > 2)

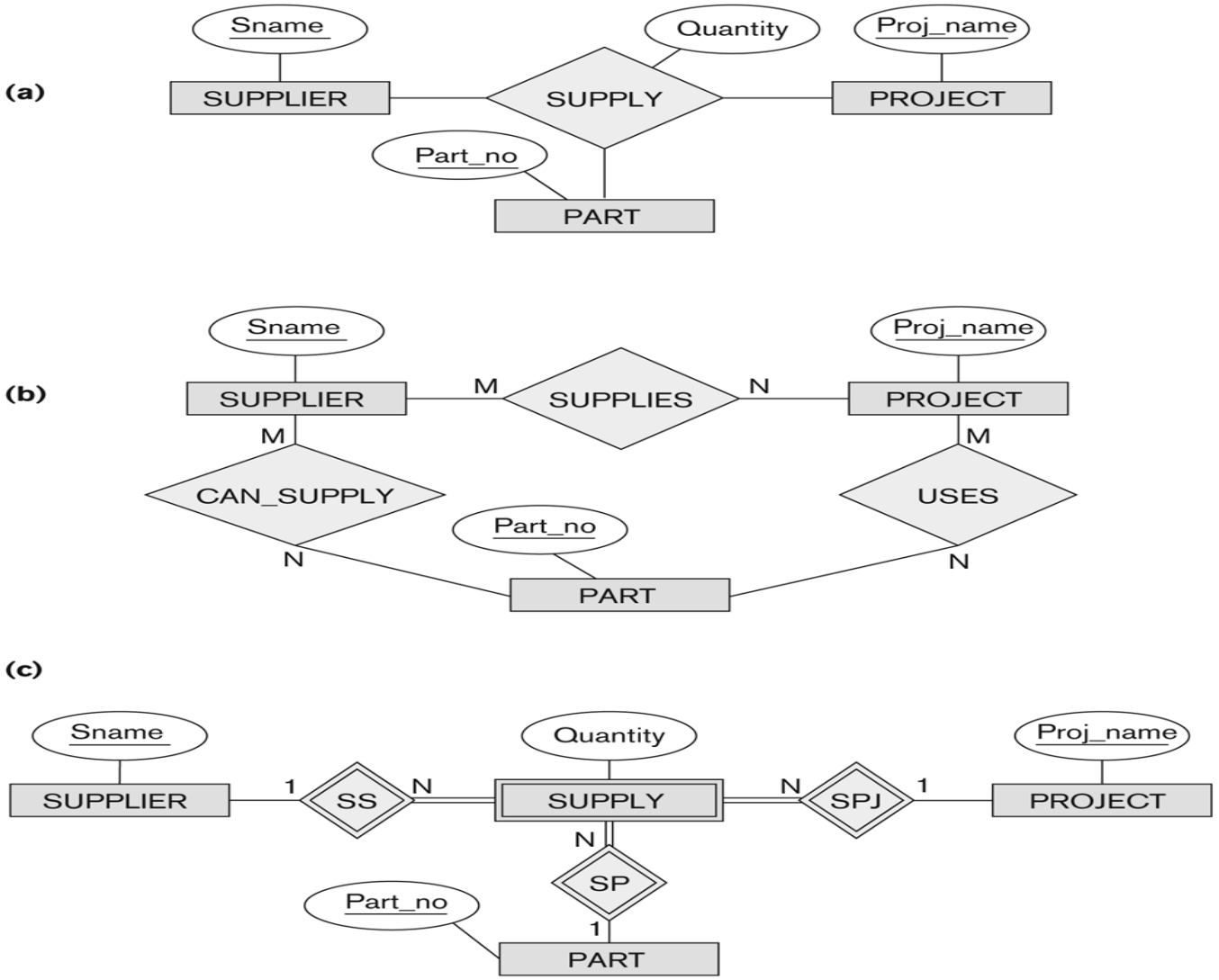
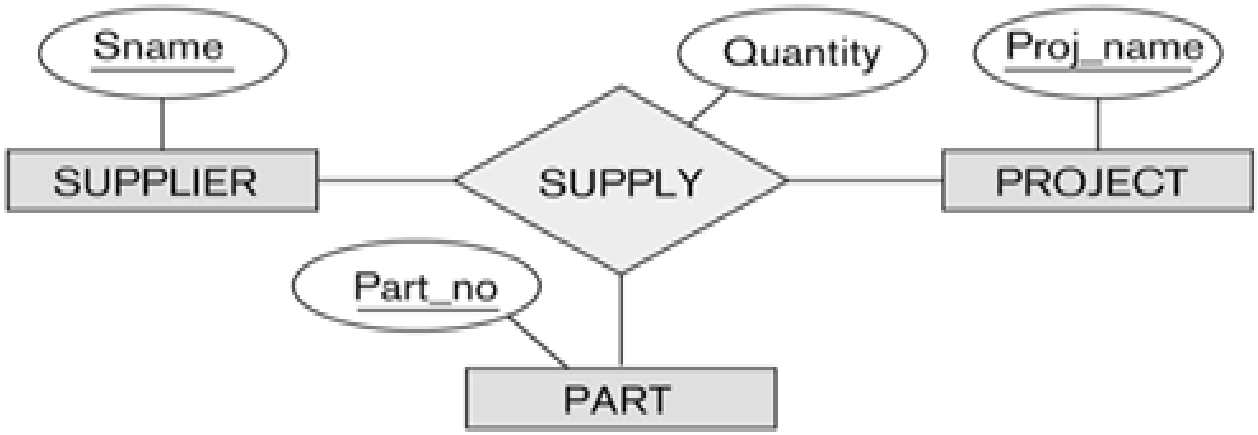


Figure 3.17
Ternary relationship types. (a) The SUPPLY relationship. (b) Three binary relationships not equivalent to SUPPLY. (c) SUPPLY represented as a weak entity type.

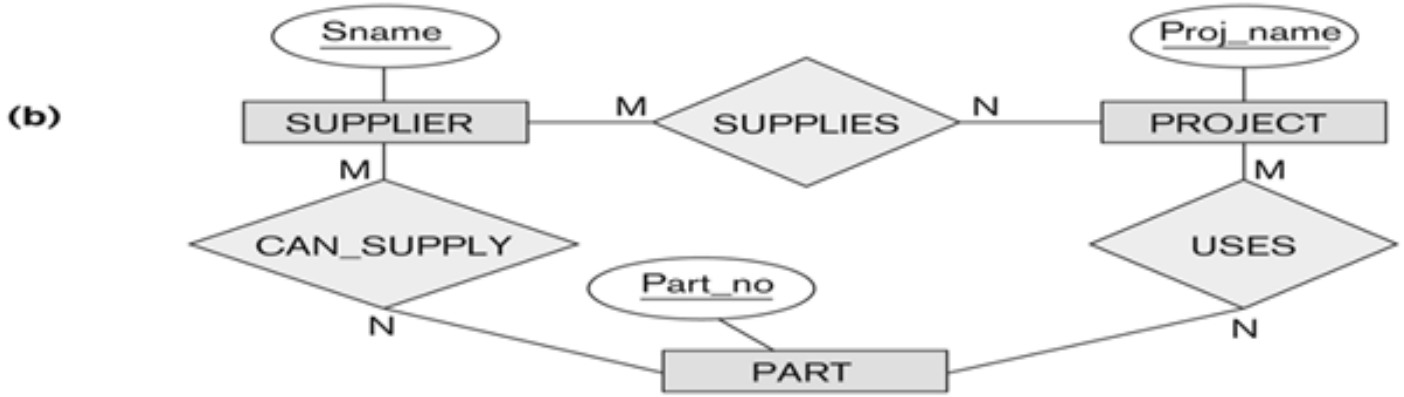
Ternary Relationship

(a)



Supply relation table		
supplier	**part**	**Project**
abc	p1	Proj1
Abc	P2	Proj2
Xyz	P3	Proj3

3 binary relationship



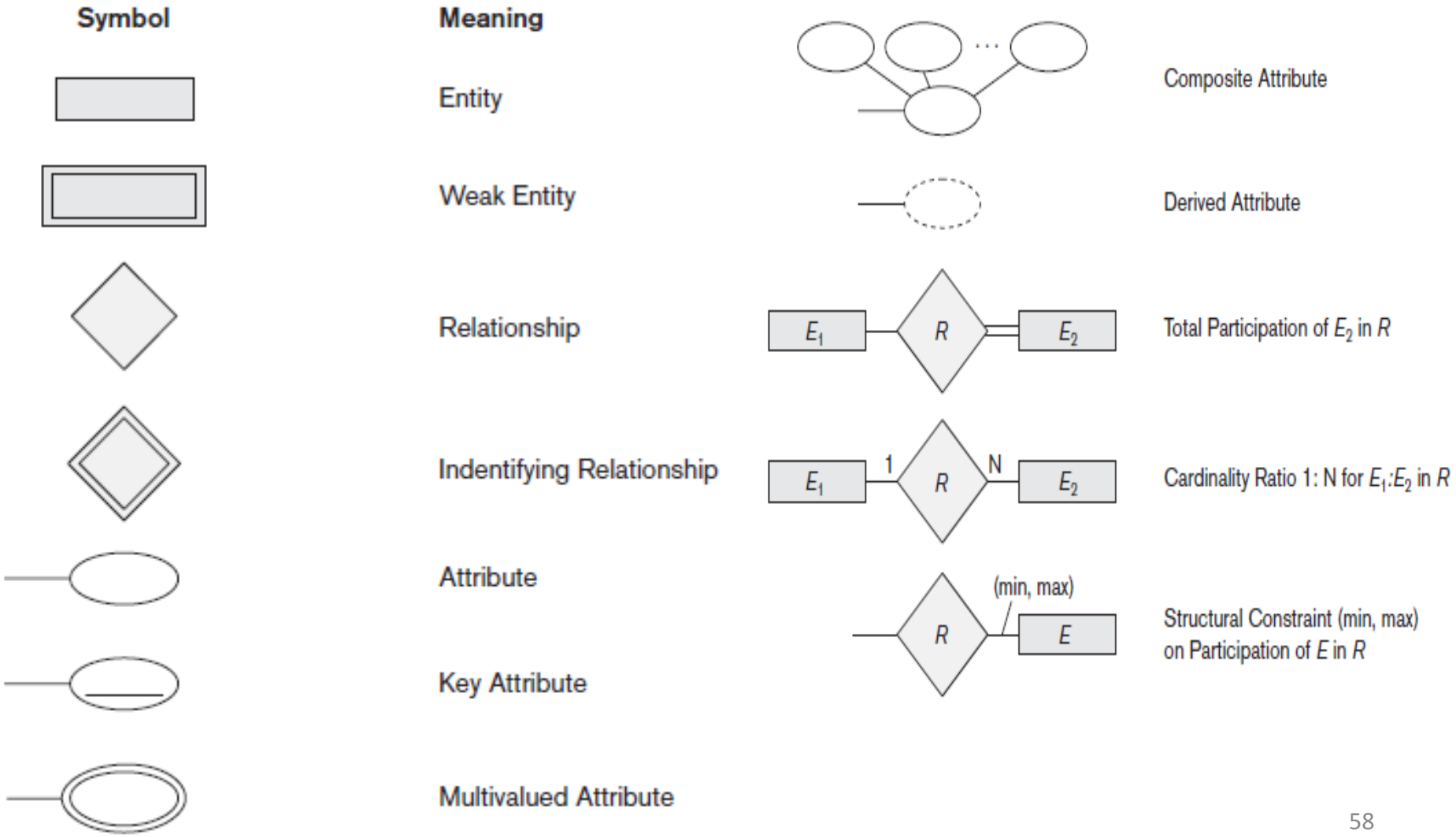
supplier	part
abc	p2
xyz	P1
Abc	P3
Xyz	P3

Project	part
Proj1	p1
Proj1	P2
Proj3	P3

supplier	Project
abc	Proj1
xyz	Proj1
Abc	Proj2
Xyz	Proj3

The Entity-Relationship Model (cont.)

- ERD Notations

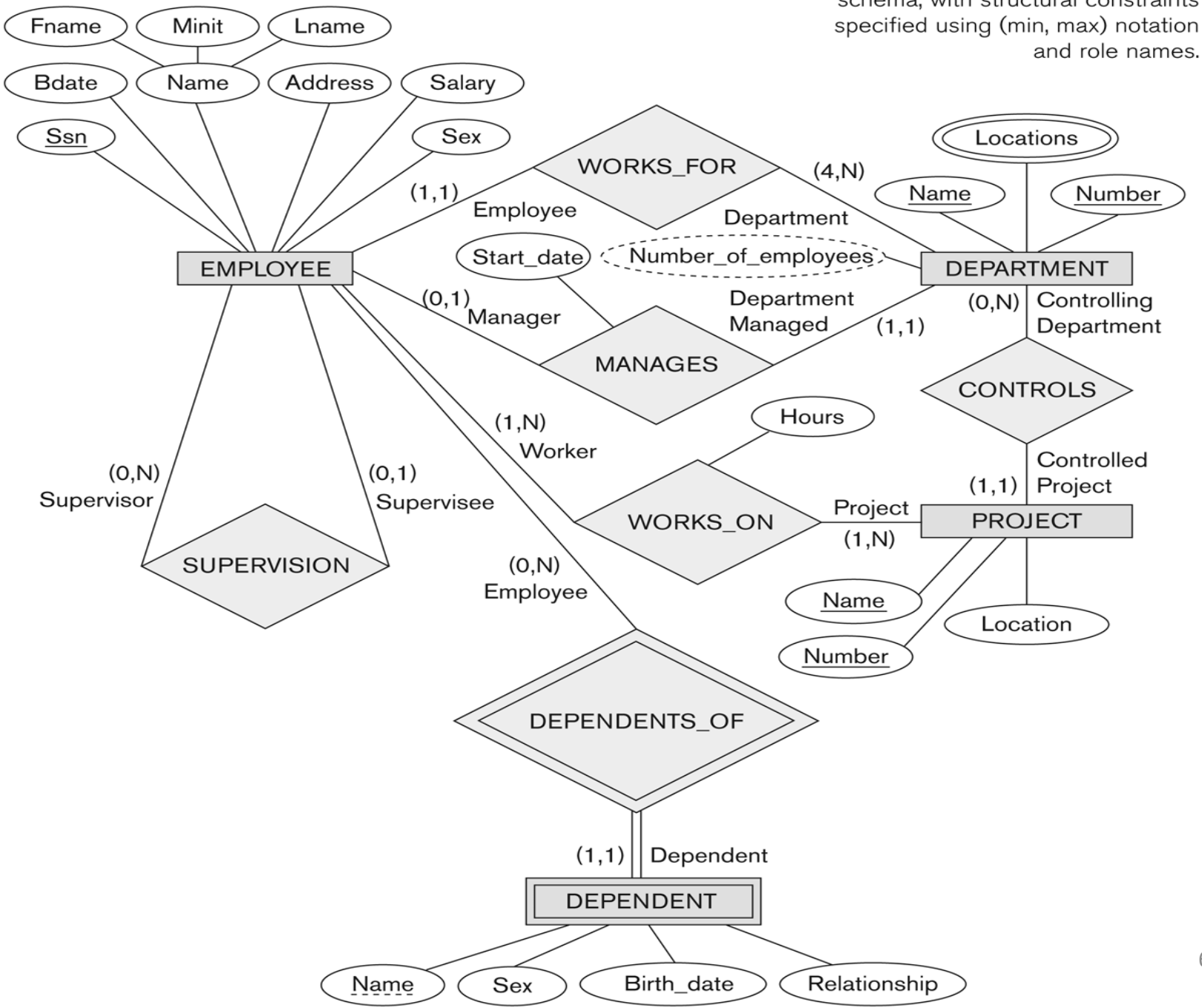


The Entity-Relationship Model (cont.)

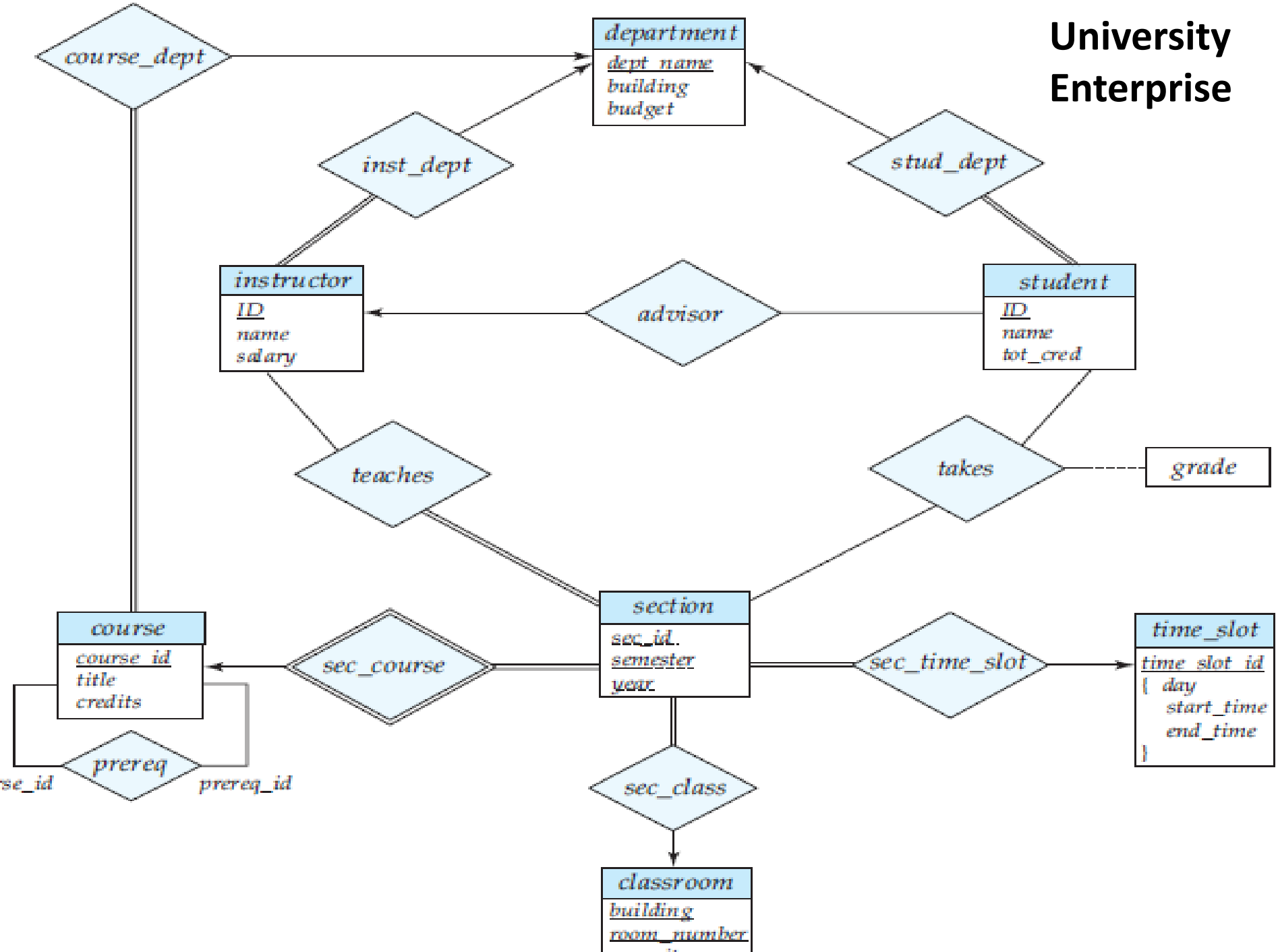
- **Proper Naming of Schema Constructs**

Self - Study from Navathe. Chapter 7, page number 222, 6th edition

Figure 3.15
ER diagrams for the company schema, with structural constraints specified using (min, max) notation and role names.



University Enterprise



References

- G K Gupta, Database Management Systems
- Elmasri Navathe, Fundamentals of Datbase Systems