Entity Relationship (ER) Models for Conceptual Database Design

Week 4

COS60009: Data Management for the Big Data Age

SWINBURNE UNIVERSITY OF TECHNOLOGY

1

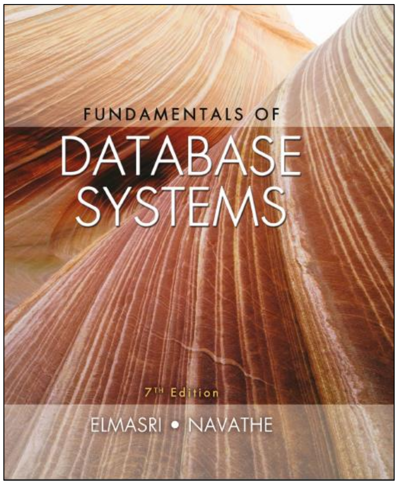
Learning Objectives

- Overview of Database Design Process
- ER Model Concepts
  - Entities and Attributes
  - Entity Types, Value Sets, and Key Attributes
  - Relationships and Relationship Types
  - Weak Entity Types
  - Roles and Attributes in Relationship Types
- ER Diagrams - Notation
- Relationships of Higher Degree
- Relational Database Schema Design by ER-to-Relational Mapping

2

Fundamentals of Database Systems

Seventh Edition



Chapter 3

Data Modeling Using the Entity-Relationship (ER) Model

Chapter 9

Relational Database Design by ER-to-Relational Mapping

3

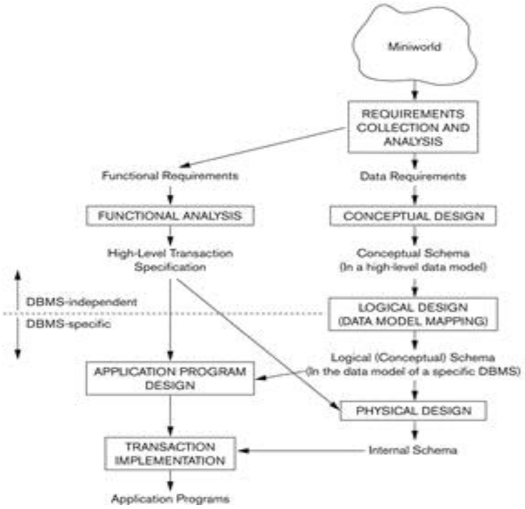
Overview of Database Design Process (1 of 2)

- Two main activities:
  - Database design
  - Applications design
- Focus in this lecture on **conceptual database design**
  - To design the conceptual schema for a database application
- Applications design focuses on the programs and interfaces that access the database
  - Generally considered part of software engineering

4

Overview of Database Design Process (2 of 2)

Figure 3.1 A simplified diagram to illustrate the main phases of database design.



Miniworld

REQUIREMENTS COLLECTION AND ANALYSIS

Functional Requirements

Data Requirements

FUNCTIONAL ANALYSIS

CONCEPTUAL DESIGN

High-Level Transaction Specification

Conceptual Schema (In a high-level data model)

DBMS-independent

DBMS-specific

LOGICAL DESIGN (DATA MODEL MAPPING)

Logical (Conceptual) Schema (In the data model of a specific DBMS)

APPLICATION PROGRAM DESIGN

PHYSICAL DESIGN

TRANSACTION IMPLEMENTATION

Internal Schema

Application Programs

Example COMPANY Database

- We need to create a database schema design based on the following (simplified) **requirements** of the COMPANY Database:
  - The company is organized into DEPARTMENTs. Each department has a name, number and an employee who **manages** the department. We keep track of the start date of the department manager. A department may have several locations.
  - Each department **controls** a number of PROJECTs. Each project has a unique name, unique number and is located at a single location.
  - The database will store each EMPLOYEE's social security number, address, salary, sex, and birthdate.
    - Each employee **works for** one department but may **work on** several projects.
    - The DB will keep track of the number of hours per week that an employee currently works on each project.
    - It is required to keep track of the **direct supervisor** of each employee.
  - Each employee may **have** a number of DEPENDENTs.
    - For each dependent, the DB keeps a record of name, sex, birthdate, and relationship to the employee.

ER Model Concepts

- Entities and Attributes
  - Entity is a basic concept for the ER model. Entities are specific things or objects in the mini-world that are represented in the database.
    - For example the EMPLOYEE John Smith, the Research DEPARTMENT, the ProductX PROJECT
  - Attributes are properties used to describe an entity.
    - For example an EMPLOYEE entity may have the attributes Name, SSN, Address, Sex, BirthDate
  - A specific entity will have a value for each of its attributes.
    - For example a specific employee entity may have Name='John Smith', SSN='123456789', Address ='731, Fondren, Houston, TX', Sex='M', BirthDate='09-JAN-55'
  - Each attribute has a value set (or data type) associated with it – e.g. integer, string, date, enumerated type, …

7

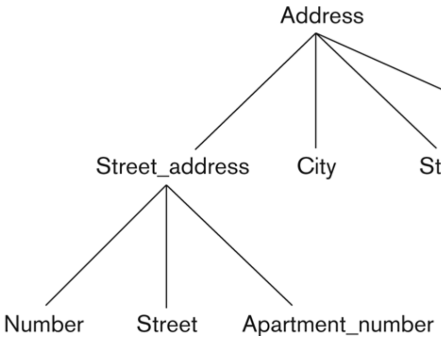
Types of Attributes

- **Simple**: each entity has a single atomic value for the attribute. For example, SSN or Sex.
- **Composite**: the attribute may be composed of several components.
  - For example: Address(Apt#, House#, Street, City, State, ZipCode, Country), or Name(FirstName, MiddleName, LastName).
  - Composition may form a hierarchy where some components are themselves composite.
- **Multi-valued**: an entity may have multiple values for that attribute.
  - For example, Color of a CAR or PreviousDegrees of a STUDENT, denoted as {Color} or {PreviousDegrees}.
- In general, composite and multi-valued attributes may be nested arbitrarily to any number of levels, although this is rare.
  - For example, PreviousDegrees of a STUDENT is a composite multi-valued attribute denoted by {PreviousDegrees (College, Year, Degree, Field)} . Multiple PreviousDegrees values can exist, each has four subcomponent attributes: College, Year, Degree, Field

8

Example of a Composite Attribute

Figure 3.4 A hierarchy of composite attributes.

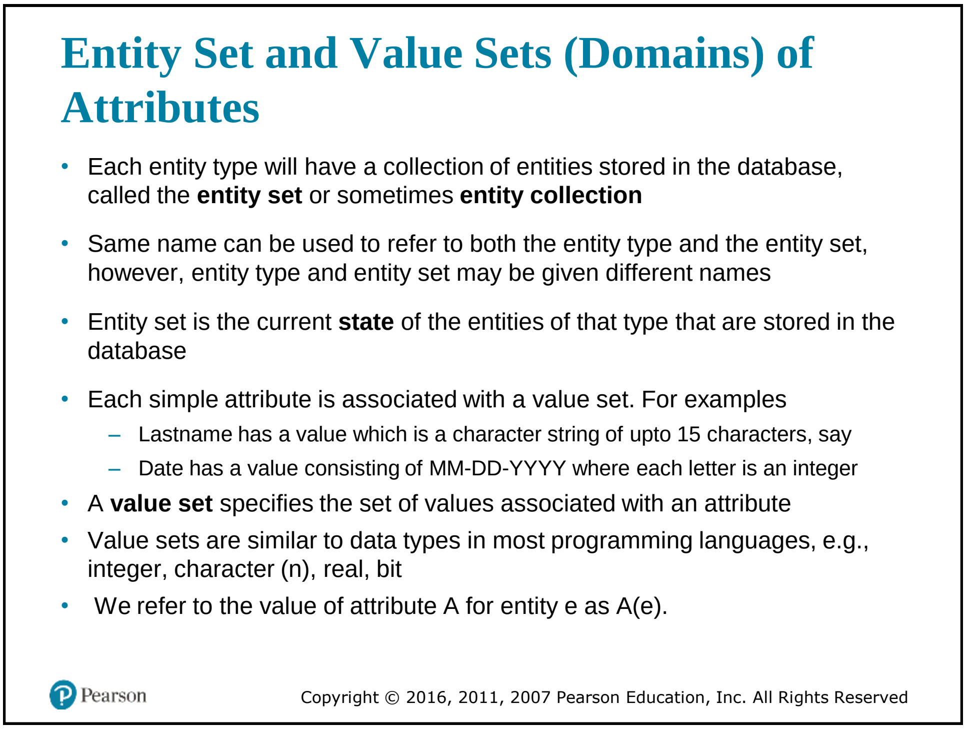


9

Entity Types and Key Attributes

- Entities with the same basic attributes are grouped or typed into an **entity type**.
  - For example, the entity type EMPLOYEE and PROJECT.
- An attribute of an entity type for which each entity must have a unique value is called a **key attribute** of the entity type.
  - For example, SSN of EMPLOYEE.
- A key attribute may be composite.
  - VehicleTagNumber is a key of the CAR entity type with components (Number, State).
- An entity type may have more than one key.
  - The CAR entity type may have two keys:
    - VehicleIdentificationNumber (popularly called VIN)
    - VehicleTagNumber (Number, State), aka license plate number.
- **Each key** is **underlined** (Note: this is different from the relational schema where only one “primary key is underlined).

10

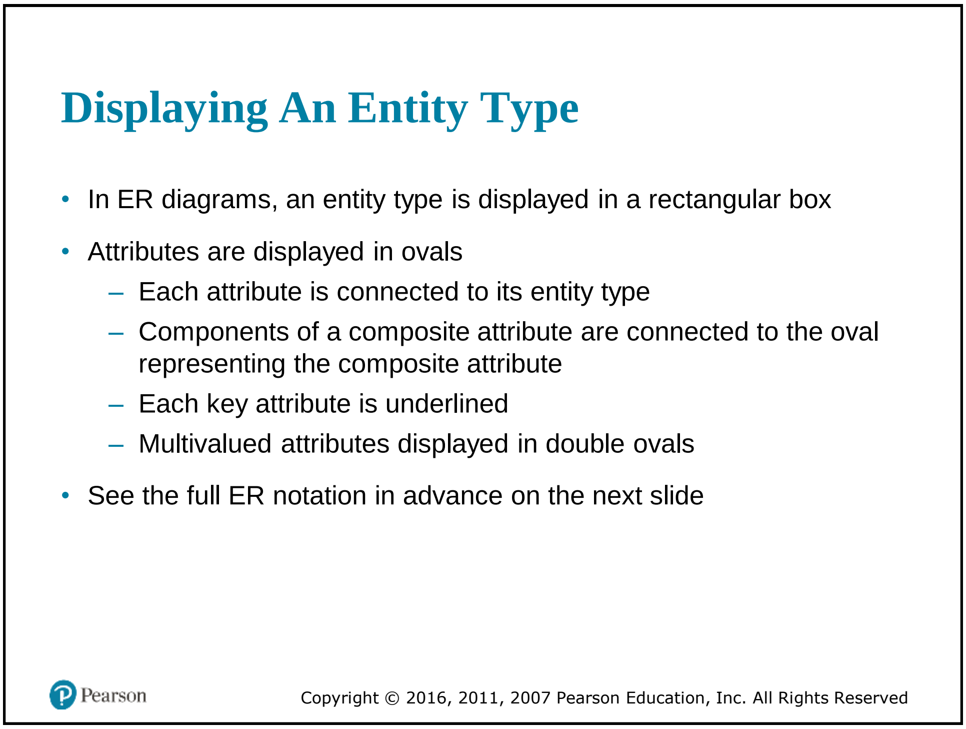


Entity Set and Value Sets (Domains) of Attributes

- Each entity type will have a collection of entities stored in the database, called the **entity set** or sometimes **entity collection**
- Same name can be used to refer to both the entity type and the entity set, however, entity type and entity set may be given different names
- Entity set is the current **state** of the entities of that type that are stored in the database
- Each simple attribute is associated with a value set. For examples
  - Lastname has a value which is a character string of upto 15 characters, say
  - Date has a value consisting of MM-DD-YYYY where each letter is an integer
- A **value set** specifies the set of values associated with an attribute
- Value sets are similar to data types in most programming languages, e.g., integer, character (n), real, bit
- We refer to the value of attribute A for entity e as A(e).

Pearson

11



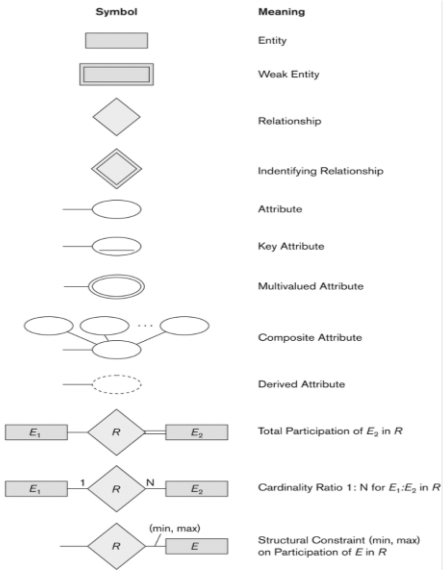
Displaying An Entity Type

- In ER diagrams, an entity type is displayed in a rectangular box
- Attributes are displayed in ovals
  - Each attribute is connected to its entity type
  - Components of a composite attribute are connected to the oval representing the composite attribute
  - Each key attribute is underlined
  - Multivalued attributes displayed in double ovals
- See the full ER notation in advance on the next slide

Pearson

12

Notation for ER Diagrams

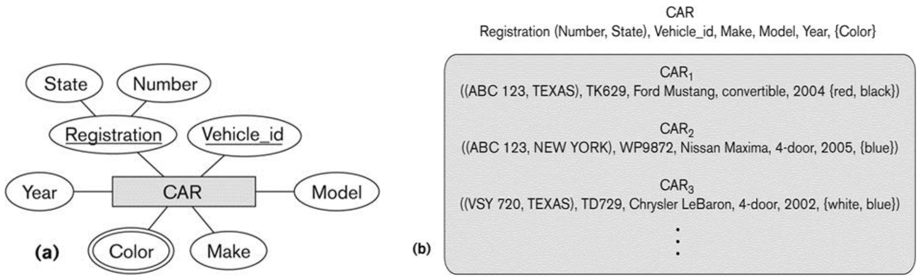


Symbol
Meaning
Entity
Weak Entity
Relationship
Indentifying Relationship
Attribute
Key Attribute
Multivalued Attribute
Composite Attribute
Derived Attribute
E1
R
E2
Total Participation of E2 in R
E1
1
R
N
E2
Cardinality Ratio 1: N for E1:E2 in R
(min, max)
R
E
Structural Constraint (min, max) on Participation of E in R

Figure 3.14 Summary of the notation for E R diagrams.

Entity Type CAR with Two Keys and a Corresponding Entity Set

Figure 3.7 The CAR entity type with two key attributes, Registration and Vehicle_i d. (a) ER diagram notation. (b) Entity set with three entities.



CAR
Registration (Number, State), Vehicle_id, Make, Model, Year, {Color}

CAR_1
((ABC 123, TEXAS), TK629, Ford Mustang, convertible, 2004 {red, black})

CAR_2
((ABC 123, NEW YORK), WP9872, Nissan Maxima, 4-door, 2005, {blue})

CAR_3
((VSY 720, TEXAS), TD729, Chrysler LeBaron, 4-door, 2002, {white, blue})

⋮

Initial Conceptual Design of Entity Types for the COMPANY Database Schema

- Based on the requirements, we can identify four initial entity types in the COMPANY database:
  - DEPARTMENT
  - PROJECT
  - EMPLOYEE
  - DEPENDENT
- Their initial conceptual design of the initial attributes shown are derived from the requirements description

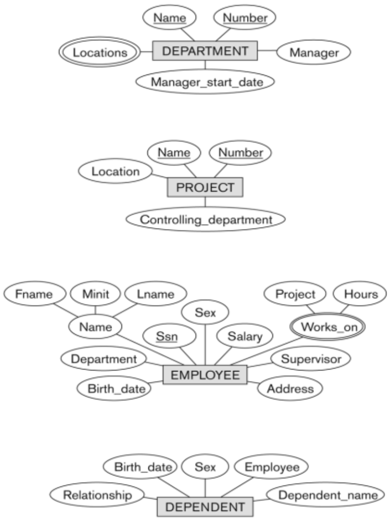


Figure 3.8 Preliminary design of entity types for the COMPANY database. Some of the shown attributes will be refined into relationships.

15

Refining the Initial Design by Introducing Relationships

- The initial design is typically not complete
- Some aspects in the requirements will be represented as **relationships**
- ER model has three main concepts:
  - Entities (and their entity types and entity sets)
  - Attributes (simple, composite, multivalued)
  - Relationships (and their relationship types and relationship sets)
- We introduce relationship concepts next

16

Relationships and Relationship Types

- A **relationship** relates two or more distinct entities with a specific meaning.
  - For example, EMPLOYEE John Smith **works on** the ProductX Project, or EMPLOYEE Franklin Wong **manages** the Research DEPARTMENT.
- Relationships of the same type are grouped or typed into a **relationship type**.
  - For example, the WORKS_ON relationship type in which EMPLOYEEs and PROJECTs participate, or the MANAGES relationship type in which EMPLOYEEs and DEPARTMENTs participate.
- The degree of a relationship type is the number of participating entity types.
  - Both MANAGERS and WORKS_ON are **binary** relationships.

Relationship Instances of the WORKS_FOR N:1 relationship between EMPLOYEE and DEPARTMENT

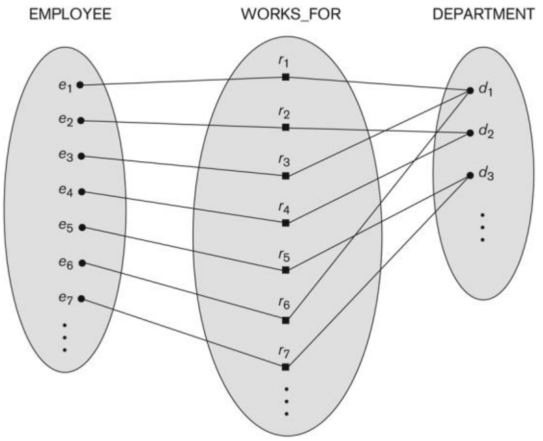


Figure 3.9 Some instances in the WORKS_FOR relationship set, which represents a relationship type WORKS_FOR between EMPLOYEE and DEPARTMENT

Relationship Instances of the M:N WORKS_ON Relationship between EMPLOYEE and PROJECT

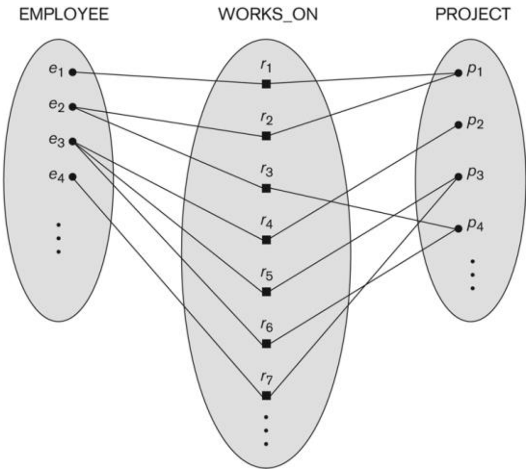


Figure 3.13 An M:N relationship, WORKS_ON.

Relationship Type Vs Relationship Set

- Relationship Type:
  - Is the schema description of a relationship
  - Identifies the relationship name and the participating entity types
  - Also identifies certain relationship constraints
- Relationship Set:
  - The current set of relationship instances represented in the database
  - The current **state** of a relationship type
- Previous figures displayed the relationship sets
- Each instance in the set relates individual participating entities – one from each participating entity type
- In ER diagrams, we represent the **relationship type** as follows:
  - Diamond-shaped box is used to display a relationship type
  - Connected to the participating entity types via straight lines

Refining the Company Database Schema by Introducing Relationships

- By examining the requirements, six relationship types are identified
- All are **binary** relationships(degree 2)
- Listed below with their participating entity types:
  - WORKS_FOR (between EMPLOYEE, DEPARTMENT)
  - MANAGES (also between EMPLOYEE, DEPARTMENT)
  - CONTROLS (between DEPARTMENT, PROJECT)
  - WORKS_ON (between EMPLOYEE, PROJECT)
  - SUPERVISION (between EMPLOYEE (as subordinate), EMPLOYEE (as supervisor))
  - DEPENDENTS_OF (between EMPLOYEE, DEPENDENT)

Pearson

ER Diagram – Relationship Types Are: WORKS_FOR, MANAGES, WORKS_ON, CONTROLS, SUPERVISION, DEPENDENTS_OF

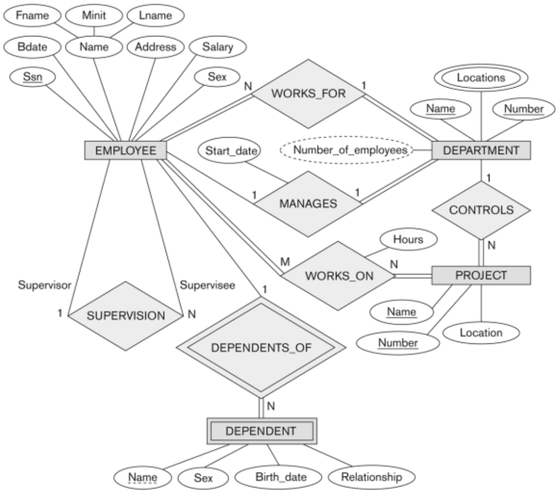
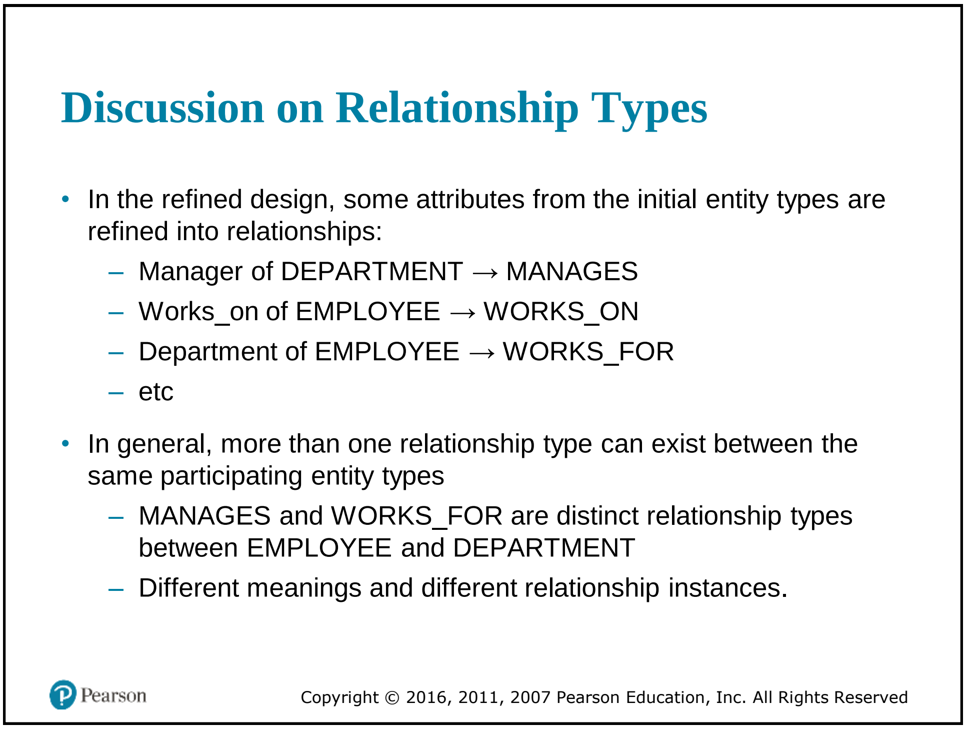


Figure 3.2 An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 3.14.

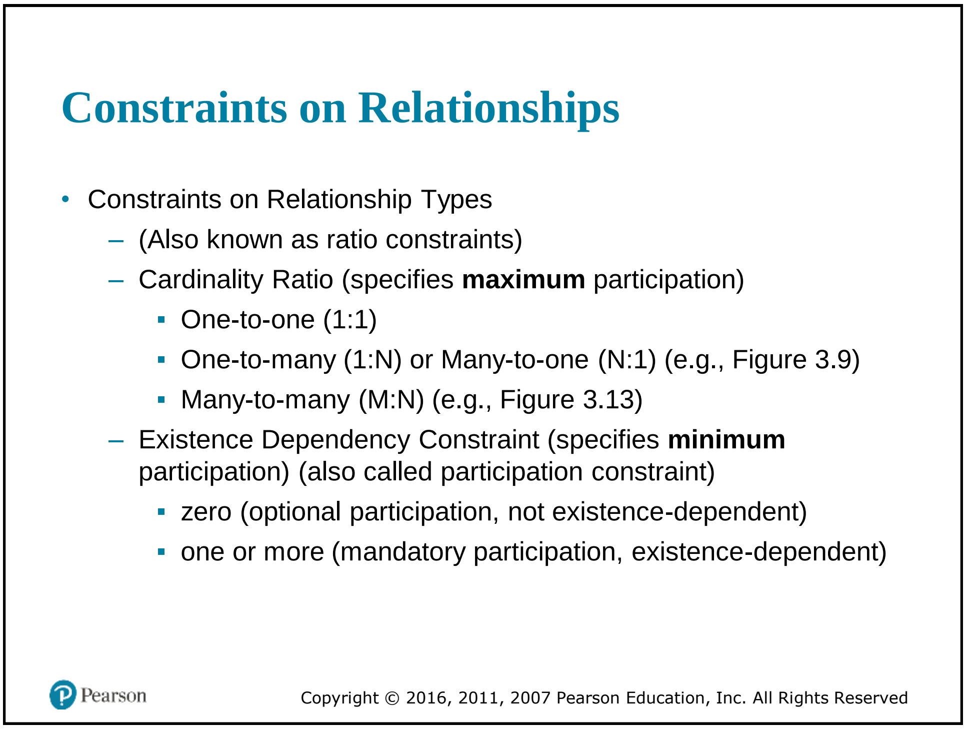
Pearson



Discussion on Relationship Types

- In the refined design, some attributes from the initial entity types are refined into relationships:
  - Manager of DEPARTMENT → MANAGES
  - Works_on of EMPLOYEE → WORKS_ON
  - Department of EMPLOYEE → WORKS_FOR
  - etc
- In general, more than one relationship type can exist between the same participating entity types
  - MANAGES and WORKS_FOR are distinct relationship types between EMPLOYEE and DEPARTMENT
  - Different meanings and different relationship instances.

23



Constraints on Relationships

- Constraints on Relationship Types
  - (Also known as ratio constraints)
  - Cardinality Ratio (specifies **maximum** participation)
    - One-to-one (1:1)
    - One-to-many (1:N) or Many-to-one (N:1) (e.g., Figure 3.9)
    - Many-to-many (M:N) (e.g., Figure 3.13)
  - Existence Dependency Constraint (specifies **minimum** participation) (also called participation constraint)
    - zero (optional participation, not existence-dependent)
    - one or more (mandatory participation, existence-dependent)

24

Recursive Relationship Type

- A relationship type between the same participating entity type in **distinct roles, a**lso called a **self-referencing** relationship type.
- Example: the SUPERVISION relationship
- EMPLOYEE participates twice in two distinct roles:
  - supervisor (or boss) role (labeled with 1 in next slide)
  - supervisee (or subordinate/worker) role (labeled with 2 in next slide)
- Each relationship instance relates two distinct EMPLOYEE entities:
  - One employee in **supervisor** role
  - One employee in **supervisee** role
- In ER diagram, need to display role names to distinguish participations.

A Recursive Relationship Supervision

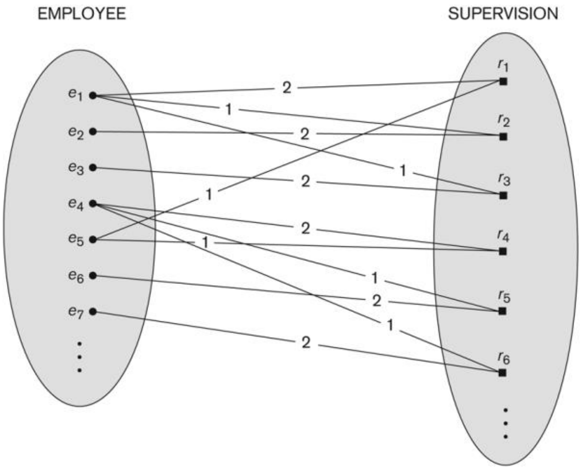


Figure 3.11 A recursive relationship SUPERVISION between EMPLOYEE in the **supervisor** role (1) and EMPLOYEE in the **subordinate** role (2).

Recursive Relationship Type is: Supervision (Participation Role Names Are Shown)

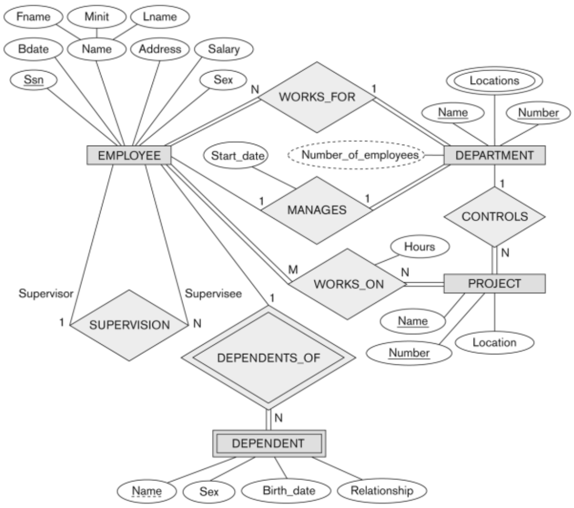


Figure 3.2 An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 3.14.

Pearson

27

Weak Entity Types

- An entity that does not have a key attribute and that is identification-dependent on another entity type.
- A weak entity must participate in an identifying relationship type with an owner or identifying entity type
- Entities are identified by the combination of:
  - A partial key of the weak entity type
  - The particular entity they are related to in the identifying relationship type
- **Example:**
  - A DEPENDENT entity is identified by the dependent’s first name, **and** the specific EMPLOYEE with whom the DEPENDENT is related
  - Name of DEPENDENT is the **partial key**
  - DEPENDENT is a **weak entity type**
  - EMPLOYEE is its identifying entity type via the identifying relationship type DEPENDENT_OF

Pearson

28

Attributes of Relationship Types

- A relationship type can have attributes:
  - For example, HoursPerWeek of WORKS_ON
  - Its value for each relationship instance describes the number of hours per week that an EMPLOYEE works on a PROJECT.
    - A value of HoursPerWeek depends on a particular (employee, project) combination
  - Most relationship attributes are used with M:N relationships
    - In 1:N relationships, they can be transferred to the entity type on the N-side of the relationship

Pearson

29

Example Attribute of a Relationship Type: Hours of WORKS_ON

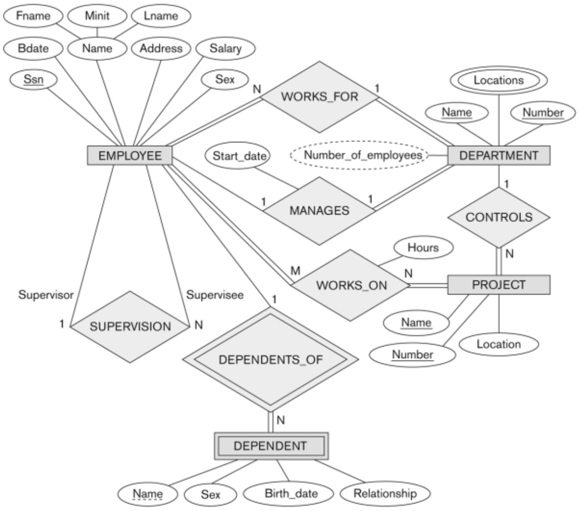


Figure 3.2 An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 3.14.

Pearson

30

Notation for Constraints on Relationships

- Cardinality ratio (of a binary relationship): 1:1, 1:N, N:1, or M:N
  - Shown by placing appropriate numbers on the relationship edges.
- Participation constraint (on each participating entity type): total (called existence dependency) or partial.
  - Total shown by double line, partial by single line.
- **Note**: These are easy to specify for Binary Relationship Types.

31

Summary of Notation for ER Diagrams

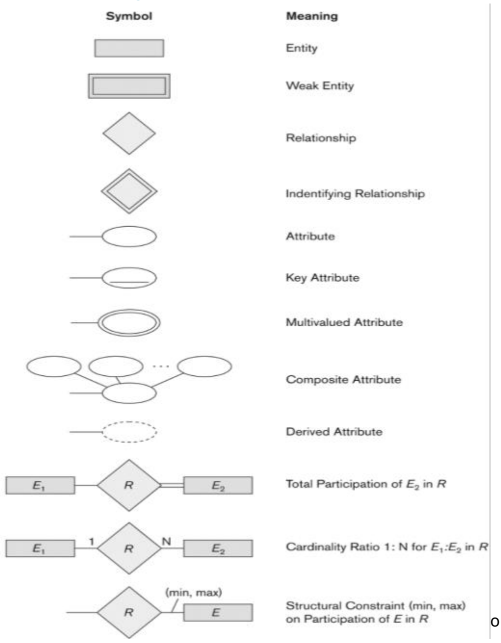

Symbol
Meaning
Entity
Weak Entity
Relationship
Indentifying Relationship
Attribute
Key Attribute
Multivalued Attribute
. . .
Composite Attribute
Derived Attribute
E_1
R
E_2
Total Participation of E_2 in R
E_1
1
R
N
E_2
Cardinality Ratio 1: N for E_1:E_2 in R
(min, max)
R
E
Structural Constraint (min, max) on Participation of E in R

Figure 3.14 Summary of the notation for ER diagrams.

32

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
- In general, 3 binary relationships can represent different information than a single ternary relationship (see Figure 3.17a and b on next slide)
- If needed, the binary and n-ary relationships can all be included in the schema design (see Figure 3.17a and b, where all relationships convey different meanings)
- In some cases, a ternary relationship can be represented as a weak entity if the data model allows a weak entity type to have multiple identifying relationships (and hence multiple owner entity types) (see Figure 3.17c)

33

Example of a Ternary Relationship

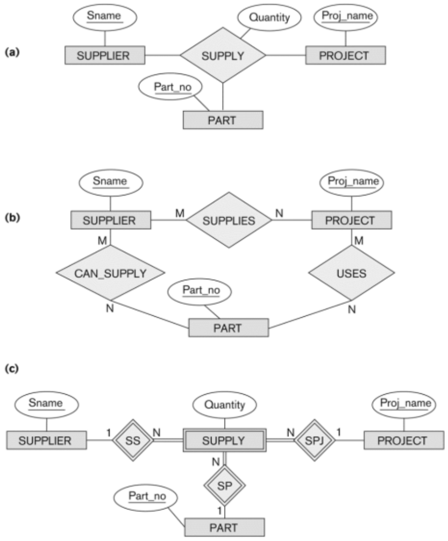
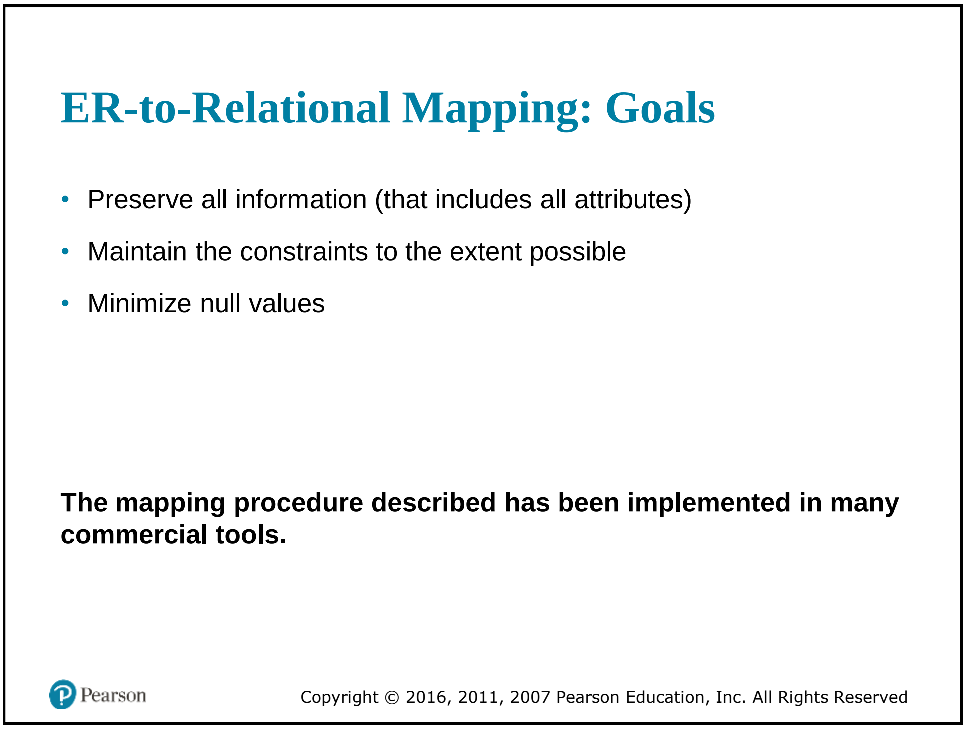


Figure 3.17 Ternary relationship types. (a) The SUPPLY relationship. (b) Three binary relationships not equivalent to SUPPLY. (c) SUPPLY represented as a weak entity type.

34



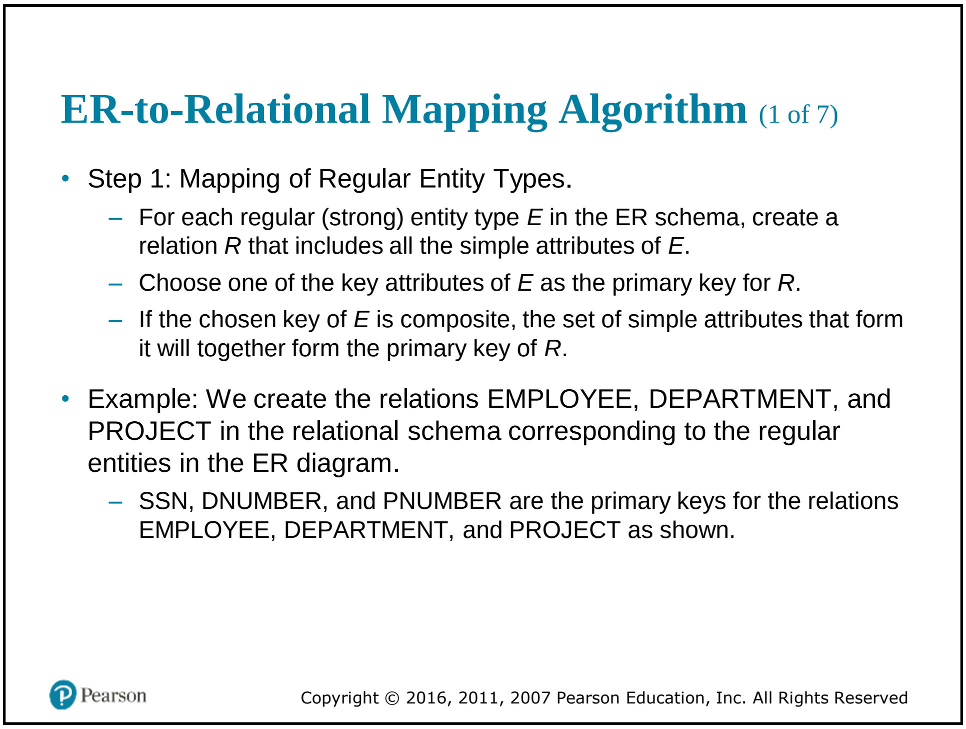
ER-to-Relational Mapping: Goals

- Preserve all information (that includes all attributes)
- Maintain the constraints to the extent possible
- Minimize null values

The mapping procedure described has been implemented in many commercial tools.

Pearson

35



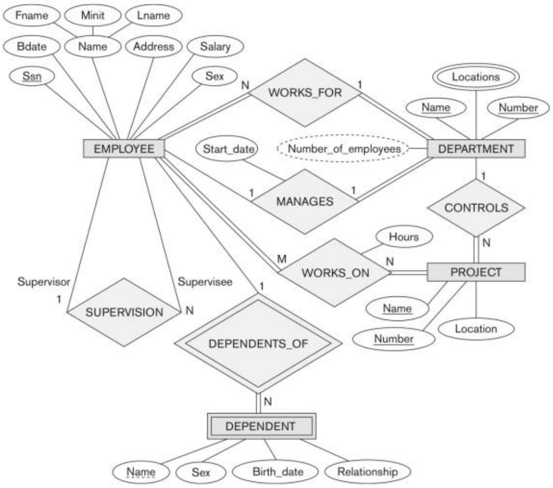
ER-to-Relational Mapping Algorithm (1 of 7)

- Step 1: Mapping of Regular Entity Types.
  - For each regular (strong) entity type *E* in the ER schema, create a relation *R* that includes all the simple attributes of *E*.
  - Choose one of the key attributes of *E* as the primary key for *R*.
  - If the chosen key of *E* is composite, the set of simple attributes that form it will together form the primary key of *R*.
- Example: We create the relations EMPLOYEE, DEPARTMENT, and PROJECT in the relational schema corresponding to the regular entities in the ER diagram.
  - SSN, DNUMBER, and PNUMBER are the primary keys for the relations EMPLOYEE, DEPARTMENT, and PROJECT as shown.

Pearson

36

Figure 9.1 The ER Conceptual Schema Diagram for the COMPANY Database



37

ER-to-Relational Mapping Algorithm (2 of 7)

- **Step 2: Mapping of Weak Entity Types**
  - For each weak entity type *W* in the ER schema with owner entity type *E*, create a relation *R* & include all simple attributes (or simple components of composite attributes) of *W* as attributes of *R*.
  - Also, include as foreign key attributes of *R* the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s).
  - The primary key of *R* is the **combination of** the primary key(s) of the owner(s) and the partial key of the weak entity type *W*, if any.
- **Example:** Create the relation DEPENDENT in this step to correspond to the weak entity type DEPENDENT.
  - Include the primary key SSN of the EMPLOYEE relation as a foreign key attribute of DEPENDENT (renamed to ESSN).
  - The primary key of the DEPENDENT relation is the combination {ESSN, DEPENDENT_NAME} because DEPENDENT_NAME is the partial key of DEPENDENT.

38

ER-to-Relational Mapping Algorithm (3 of 7)

- **Step 3: Mapping of Binary 1:1 Relation Types**
  - For each binary 1:1 relationship type *R* in the ER schema, identify the relations *S* and *T* that correspond to the entity types participating in *R*.
- There are two possible approaches:
  1. **Foreign Key (2 relations) approach:** Choose one of the relations, say *S*, and include a foreign key in *S* the primary key of *T*. It is better to choose an entity type with total participation in *R* in the role of *S*.
     Example: 1:1 relation MANAGES is mapped by choosing the participating entity type DEPARTMENT to serve in the role of S, because its participation in the MANAGES relationship type is total.
  2. **Merged relation (1 relation) option:** An alternate mapping of a 1:1 relationship type is possible by merging the two entity types and the relationship into a single relation. This may be appropriate when both participations are total.

Pearson

39

ER-to-Relational Mapping Algorithm (4 of 7)

- **Step 4: Mapping of Binary 1:N Relationship Types.**
  - For each regular binary 1:N relationship type *R*, identify the relation *S* that represent the participating entity type at the N-side of the relationship type.
  - Include as foreign key in *S* the primary key of the relation *T* that represents the other entity type participating in *R*.
  - Include any simple attributes of the 1:N relation type as attributes of *S*.
- Example: 1:N relationship types WORKS_FOR, CONTROLS, and SUPERVISION in the figure.
  - For WORKS_FOR we include the primary key DNUMBER of the DEPARTMENT relation as foreign key in the EMPLOYEE relation and call it DNO.

Pearson

40

ER-to-Relational Mapping Algorithm (5 of 7)

- **Step 5: Mapping of Binary M:N Relationship Types.**
  - For each regular binary M:N relationship type *R*, **create a new relation** *S* to represent *R*. This is a **relationship relation**.
  - Include as foreign key attributes in *S* the primary keys of the relations that represent the participating entity types; **their combination will form the primary key** of *S*.
  - Also include any simple attributes of the M:N relationship type (or simple components of composite attributes) as attributes of *S*.
- Example: The M:N relationship type WORKS_ON from the ER diagram is mapped by creating a relation WORKS_ON in the relational database schema.
  - The primary keys of the PROJECT and EMPLOYEE relations are included as foreign keys in WORKS_ON and renamed PNO and ESS N, respectively.
  - Attribute HOURS in WORKS_ON represents the HOURS attribute of the relation type. The primary key of the WORKS_ON relation is the combination of the foreign key attributes {ESSN, PND}.

ER-to-Relational Mapping Algorithm (6 of 7)

- **Step 6: Mapping of Multivalued attributes.**
  - For each multivalued attribute *A*, create a new relation *R*.
  - This relation *R* will include an attribute corresponding to *A*, plus the primary key attribute *K*-as a foreign key in *R*-of the relation that represents the entity type of relationship type that has *A* as an attribute.
  - The primary key of *R* is the combination of *A* and *K*. If the multivalued attribute is composite, we include its simple components.
- **Example:** The relation DEPT_LOCATIONS is created.
  - The attribute DLOCATION represents the multivalued attribute LOCATIONS of DEPARTMENT, while DNUMBER-as foreign key-represents the primary key of the DEPARTMENT relation.
  - The primary key of *R* is the combination of {DNUMBER, DLOCATION}.

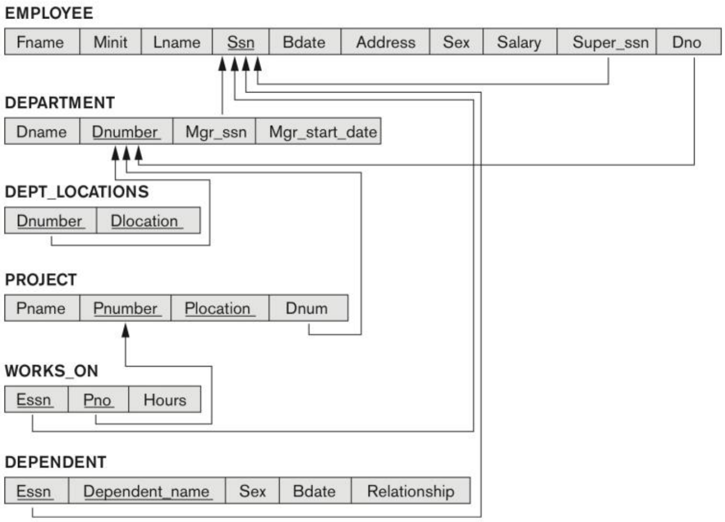
ER-to-Relational Mapping Algorithm (7 of 7)

- **Step 7: Mapping of *N*-ary Relationship Types.**
  - For each *n*-ary relationship type *R*, where *n*>2, create a new relationship *S* to represent *R*.
  - Include as foreign key attributes in *S* the primary keys of the relations that represent the participating entity types.
  - Also include any simple attributes of the *n*-ary relationship type (or simple components of composite attributes) as attributes of *S*.
- **Example:** The relationship type SUPPY in the E R on the next slide.
  - This can be mapped to the relation SUPPLY shown in the relational schema, whose primary key is the combination of the three foreign keys {SNAME, PARTNO, PROJNAME}

Pearson

43

Figure 9.2 Result of Mapping the COMPANY ER Schema into a Relational Database Schema



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

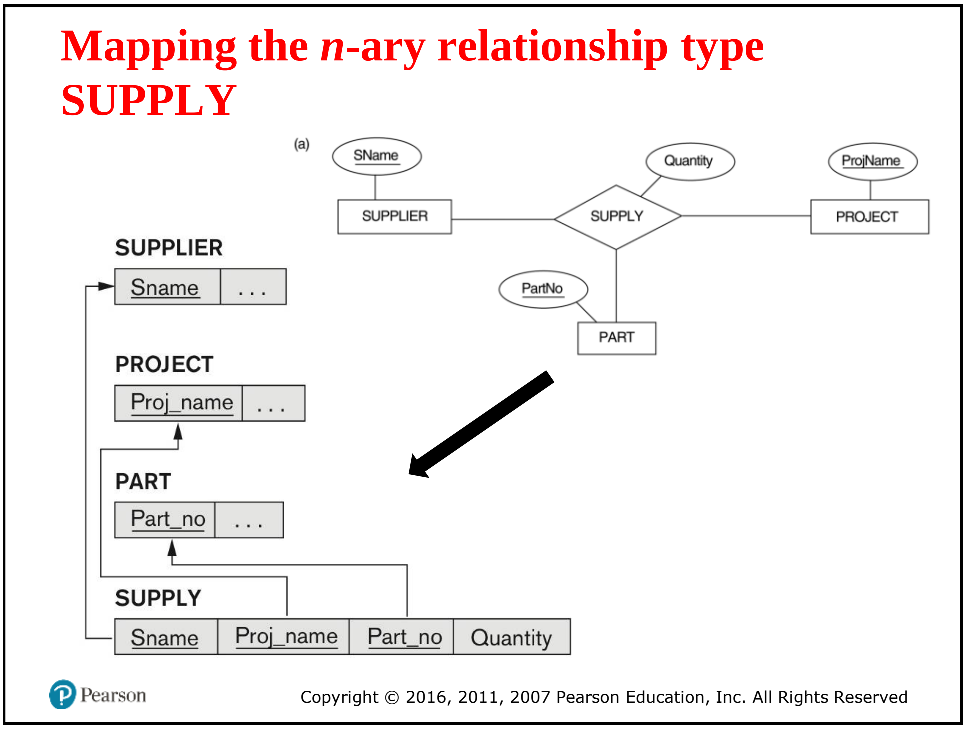
Essn	Pno	Hours

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Pearson

44



Mapping the *n*-ary relationship type SUPPLY

SUPPLIER

Sname	. . .

PROJECT

Proj_name	. . .

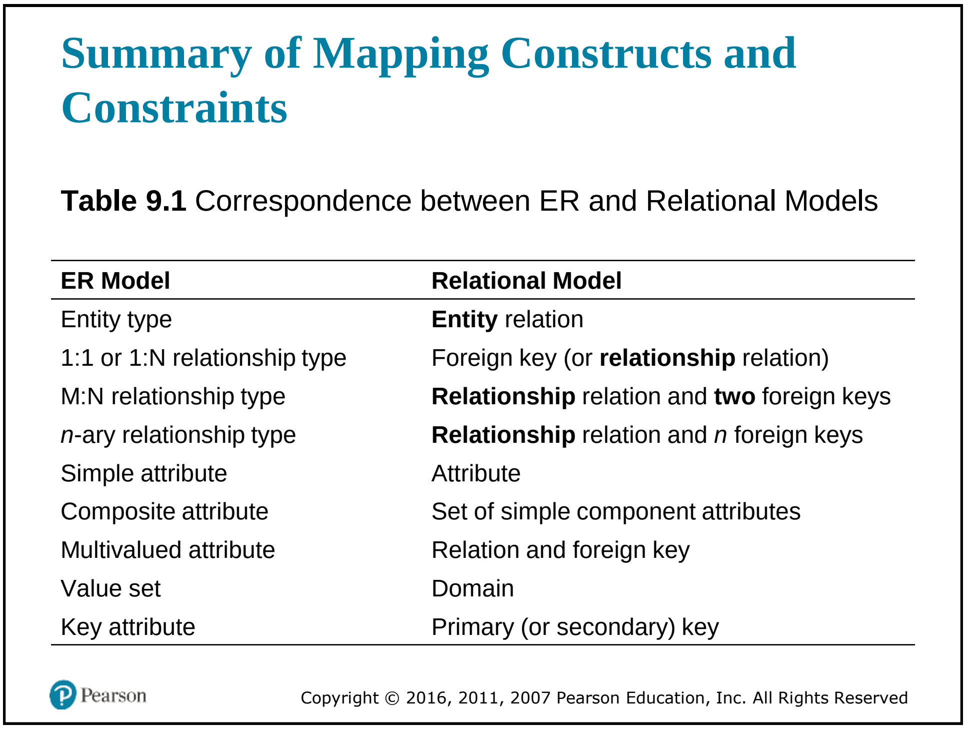
PART

Part_no	. . .

SUPPLY

Sname	Proj_name	Part_no	Quantity

45

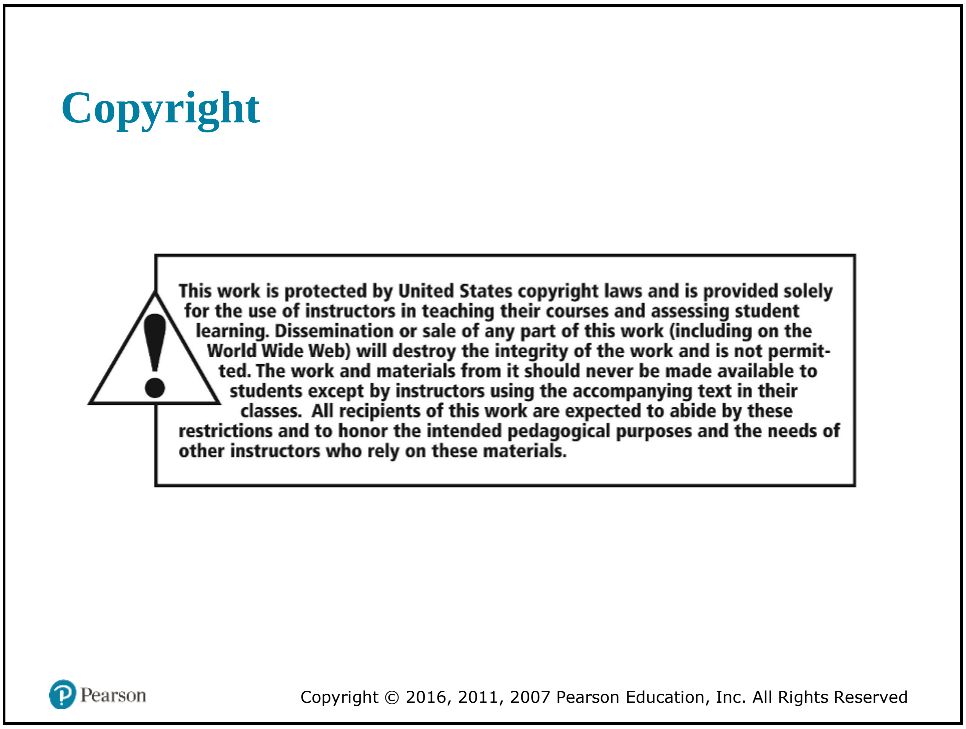


Summary of Mapping Constructs and Constraints

Table 9.1 Correspondence between ER and Relational Models

ER Model	Relational Model
Entity type	**Entity** relation
1:1 or 1:N relationship type	Foreign key (or **relationship** relation)
M:N relationship type	**Relationship** relation and **two** foreign keys
n-ary relationship type	**Relationship** relation and *n* foreign keys
Simple attribute	Attribute
Composite attribute	Set of simple component attributes
Multivalued attribute	Relation and foreign key
Value set	Domain
Key attribute	Primary (or secondary) key

46



Copyright

Pearson

47