Chapter 3: Data Modeling Using the Entity-Relationship (ER) Model

CS-6360 Database Design

Chris Irwin Davis, Ph.D.

Email: chrisirwindavis@utdallas.edu
Phone: (972) 883-3574
Office: ECSS 4.705

Chapter 3 Outline

- 3.1 – Using High-Level Conceptual Data Models for Database Design
- 3.2 – A Sample Database Application
- 3.3 – Entity Types, Entity Sets, Attributes, and Keys
- 3.4 – Relationship Types, Relationship Sets, Roles, and Structural Constraints
- 3.5 – Weak Entity Types

Chapter 3 Outline

- 3.6 – Refining the ER Design for the COMPANY Database
- 3.7 – ER Diagrams, Naming Conventions, and Design Issues
- 3.8 – Example of Other Notation: UML Class Diagrams
- 3.9 – Relationship Types of Degree Higher than Two

3.1 – Using High-Level Conceptual Data Models for Database Design

Data Modeling Using the Entity-Relationship (ER) Model

- Entity-Relationship (ER) model
  - Popular high-level conceptual data model
- ER diagrams
  - Diagrammatic notation associated with the ER model
- Unified Modeling Language (UML)

Using High-Level Conceptual Models

- Requirements collection and analysis
  - Database designers interview prospective database users to understand and document data requirements. Result:
  - Data requirements
  - Functional requirements of the application

Using High-Level Conceptual Models

- Conceptual schema
  - Conceptual design
  - Description of data requirements
  - Includes detailed descriptions of the entity types, relationships, and constraints
  - Transformed from high-level data model into implementation data model

Logical and Physical Design

- Logical design or data model mapping
  - Result is a database schema in implementation data model of DBMS
- Physical design phase
  - Internal storage structures, file organizations, indexes, access paths, and physical design parameters for the database files specified

3.2 – A Sample Database Application

A Sample Database Application

- COMPANY
  - Employees, departments, and projects
  - Company is organized into departments
  - Department controls a number of projects
  - Employee: store each employee’s name, Social Security number, address, salary, sex (gender), and birth date
  - Keep track of the dependents of each employee

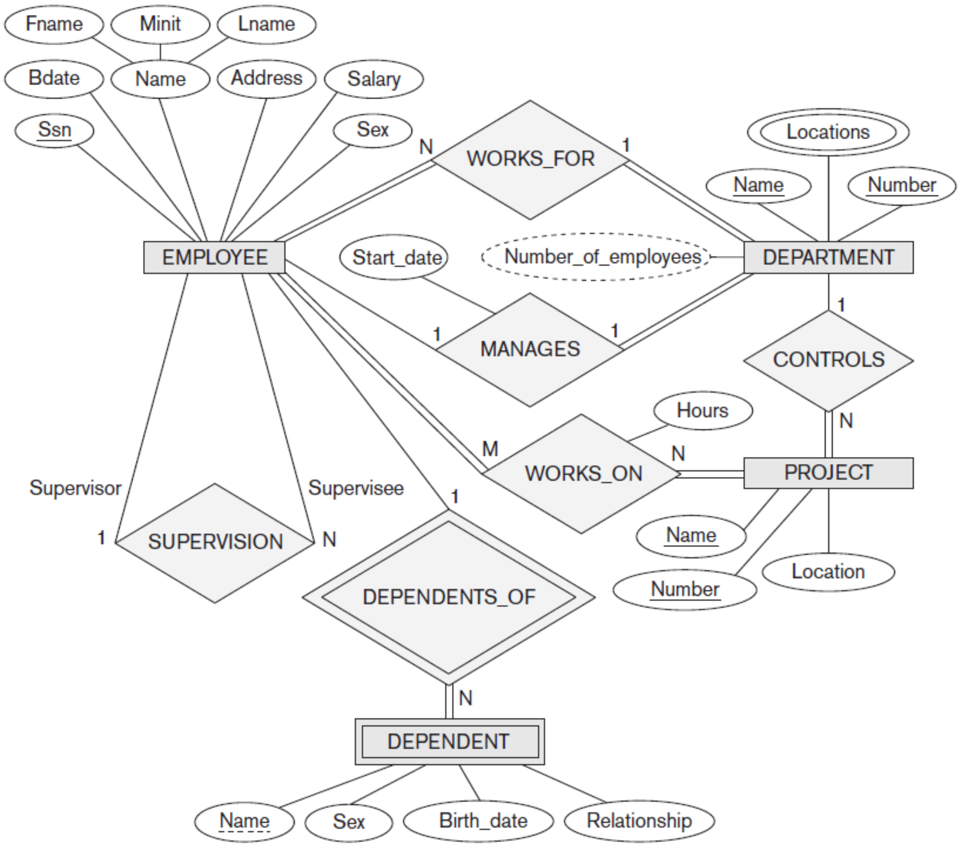


Figure 7.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.

3.3 – Entity Types, Entity Sets, Attributes, and Keys

Entity Types, Entity Sets, Attributes, and Keys

- ER model describes data as:
  - Entities
  - Relationships
  - Attributes

Entities and Attributes

- **Entity**
  - Thing in real world with independent existence
- **Attributes**
  - Particular properties that describe entity

Entities and Attributes

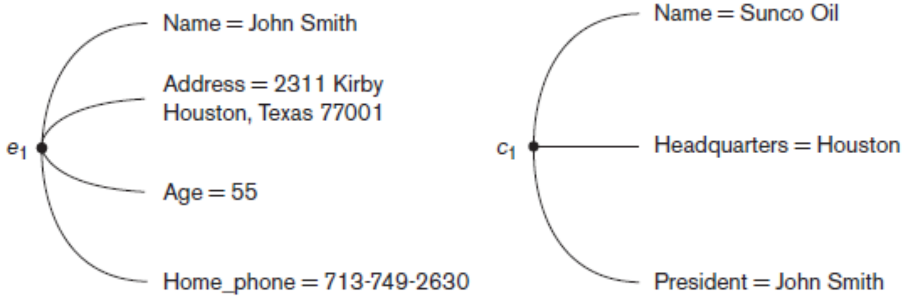


Figure 7.3
Two entities, EMPLOYEE e_1, and COMPANY c_1, and their attributes.

Entity Types, Entity Sets, Keys, and Value Sets

- Entity type
  - Collection (or set) of entities that have the same attributes

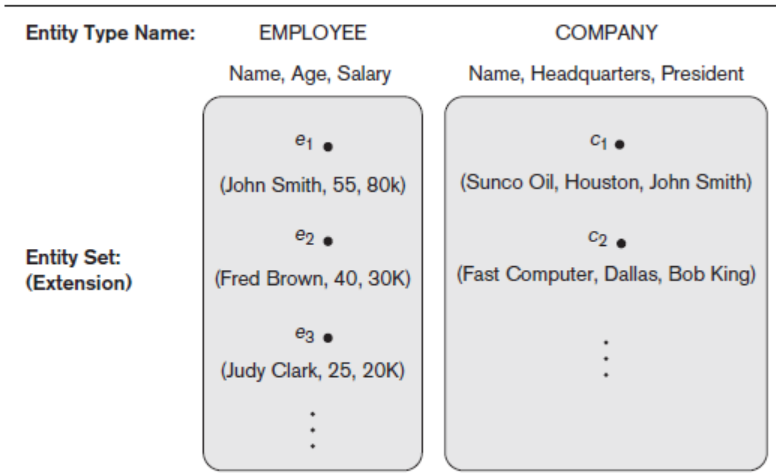


Figure 7.6
Two entity types, EMPLOYEE and COMPANY, and some member entities of each.

Types of Attributes

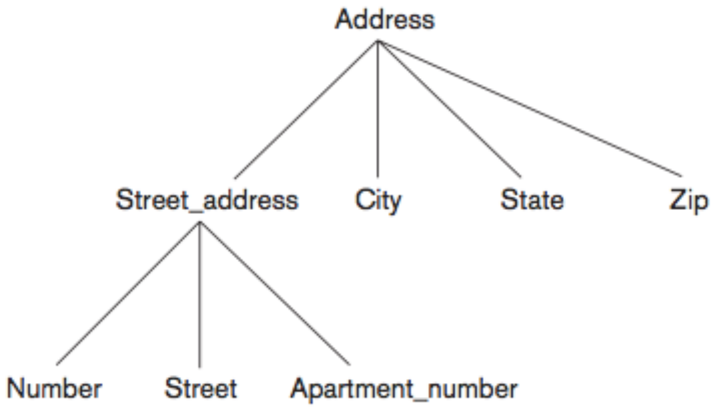
- Several types of attributes occur in the ER model
  - Simple versus composite
  - Single-valued versus multivalued
  - Stored versus derived

Single-Valued versus Multivalued Attributes

- Most attributes have a single value for a particular entity; such attributes are called single-valued
    - For example, Age is a single-valued attribute of a person
- An attribute can have a set of values for the same entity
    - A multivalued attribute may have lower and upper bounds to constrain the number of values allowed for each individual entity

Composite versus Simple (Atomic) Attributes

- Composite attributes can be divided into smaller subparts, which represent more basic attributes with indepen- dent meanings
- Attributes that are not divisible are called simple or atomic attributes.



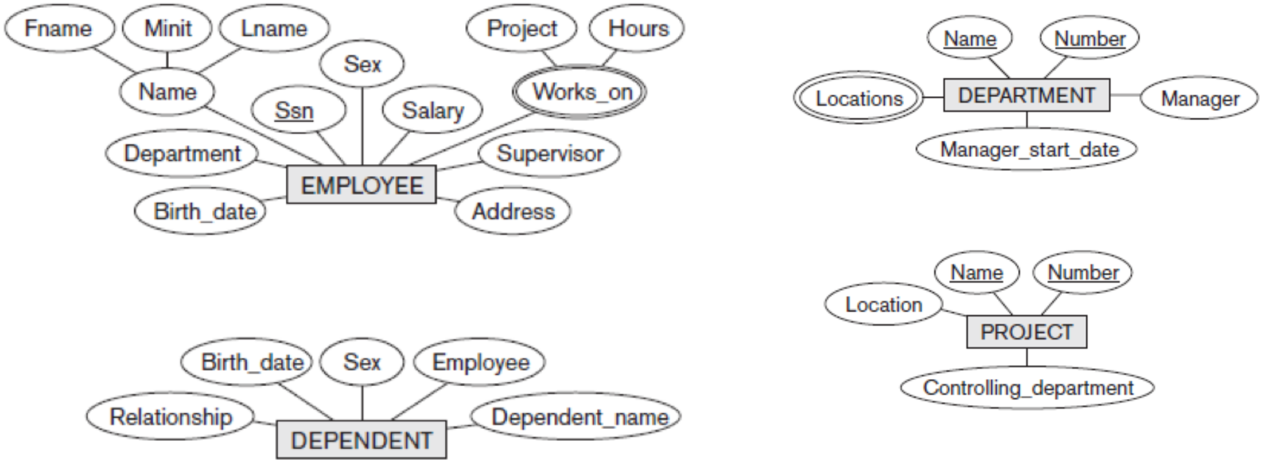
Stored versus Derived Attributes

- Two (or more) attribute values are related
  - e.g. Age and Birth_date
  - Birth_date may be a stored attribute, and
  - Age can be derived from Birth_date
- Can Age be a multi-valued attribute?
- Can Age be a composite attribute?

Attribute Constraints

- Key or uniqueness constraint
  - Attributes whose values are distinct for each individual entity in entity set
  - Key attribute
    - Uniqueness property must hold for every entity set of the entity type
- Value sets (or domain of values)
  - Specifies set of values that may be assigned to that attribute for each individual entity

Initial Conceptual Design of the COMPANY Database



3.4 – Relationship Types, Relationship Sets, Roles, and Structural Constraints

Relationship Types, Relationship Sets, Roles, and

- Relationship
    - When an attribute of one entity type refers to another entity type
    - Represent references as relationships not attributes

Relationship Types, Sets, and Instances

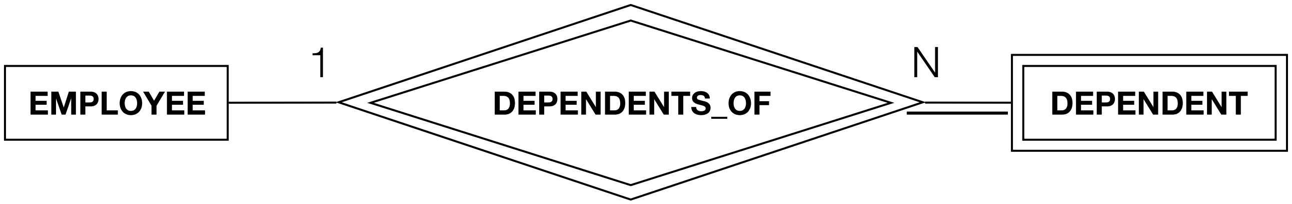
- Relationship type *R* among *n* entity types E_1, E_2, ..., E_n
  - Defines a set of associations among entities from these entity types
- Relationship instances r_i
  - Each r_i associates n individual entities (e_2, e_2, ..., e_n)
  - Each entity e_j in ri is a member of entity set E_j

Relationship Degree

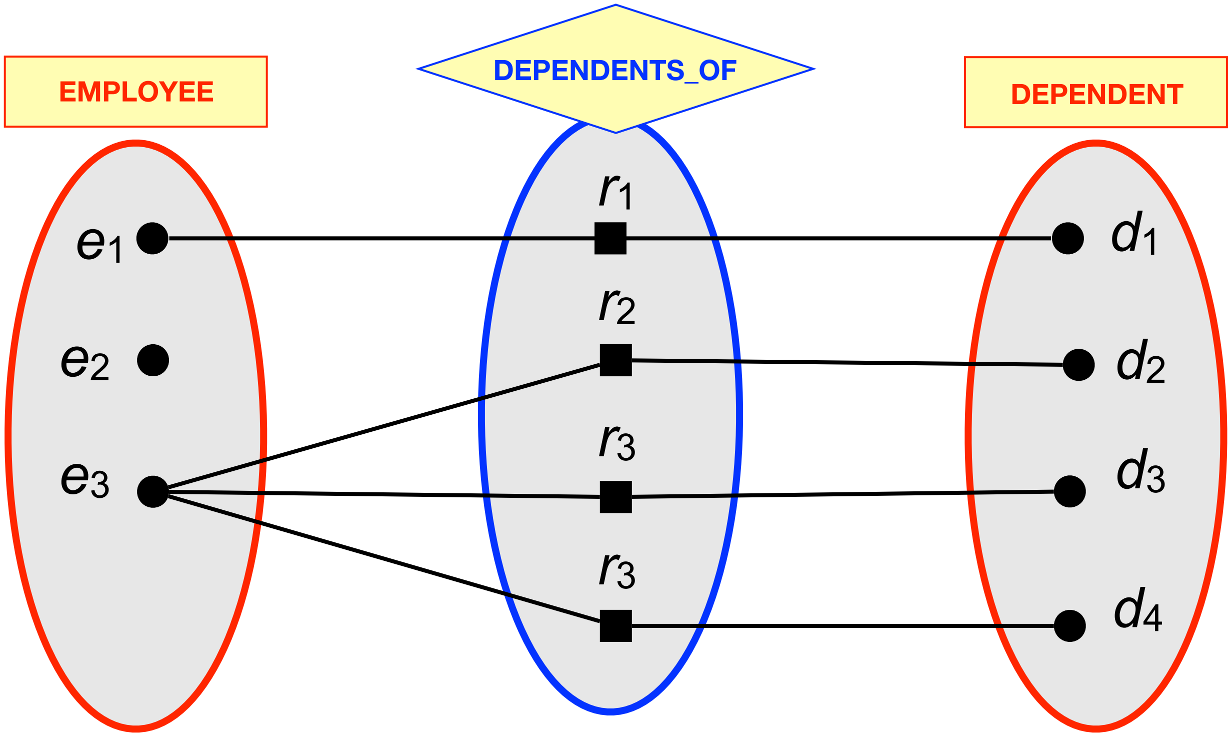
- Degree of a relationship type
  - Number of participating entity types
  - Binary, ternary, *n*-ary
- Relationships as attributes
  - Possible to think of a binary relationship type in terms of attributes (useful in certain scenarios, e.g. works_on)

Example: DEPENDENTS_OF Relationship Type

Consider the following binary Relationship Type: DEPENDENTS_OF



Example: DEPENDENTS_OF Relationship Type



Role Names and Recursive Relationships

- **Role names**
  - Role name signifies the *role* that a participating entity plays in each relationship instance
  - Usually not needed since Entity Type names + Relationship Type names implies the semantic role
- **Recursive relationships**
  - Defined by the scenario where the same Entity Type participates more than once in a relationship type in different roles
  - *Must* specify role name since it is not possible to imply role when both sides of the Relationship has the same Entity Type + Relationship Type

Recursive Relationship SUPERVISION

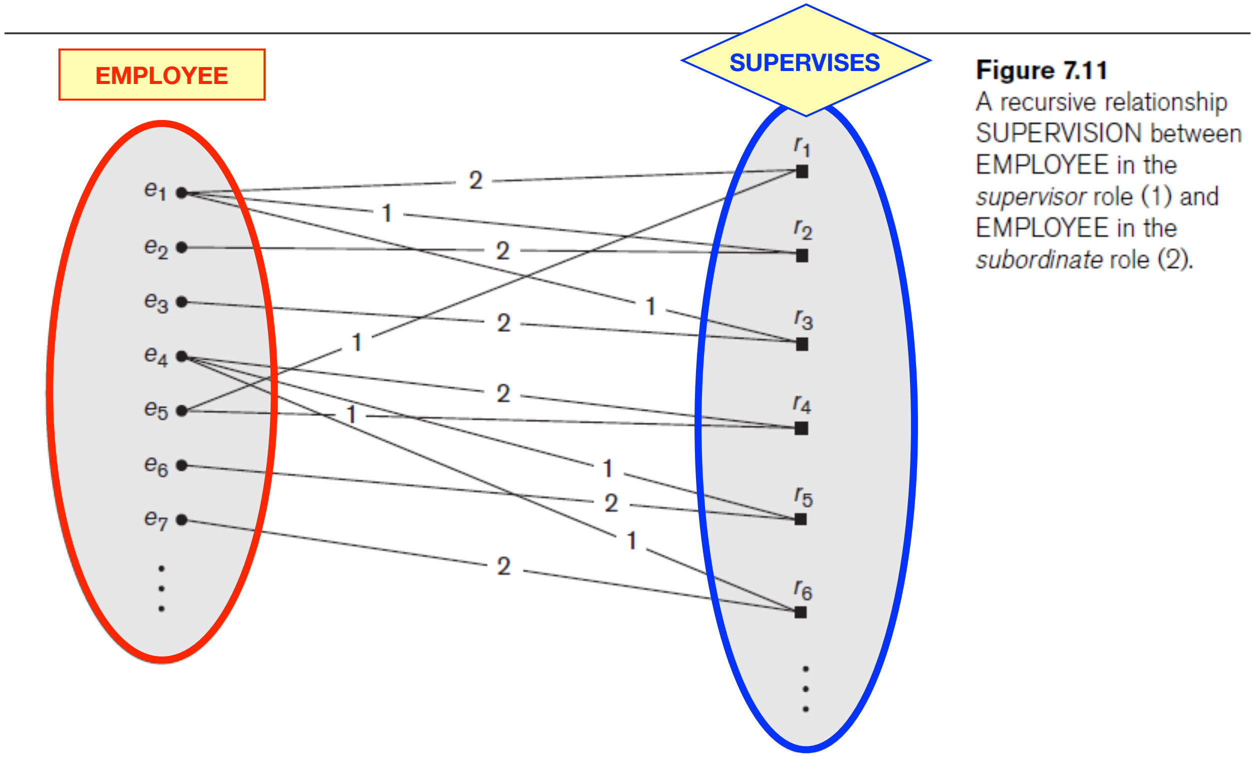
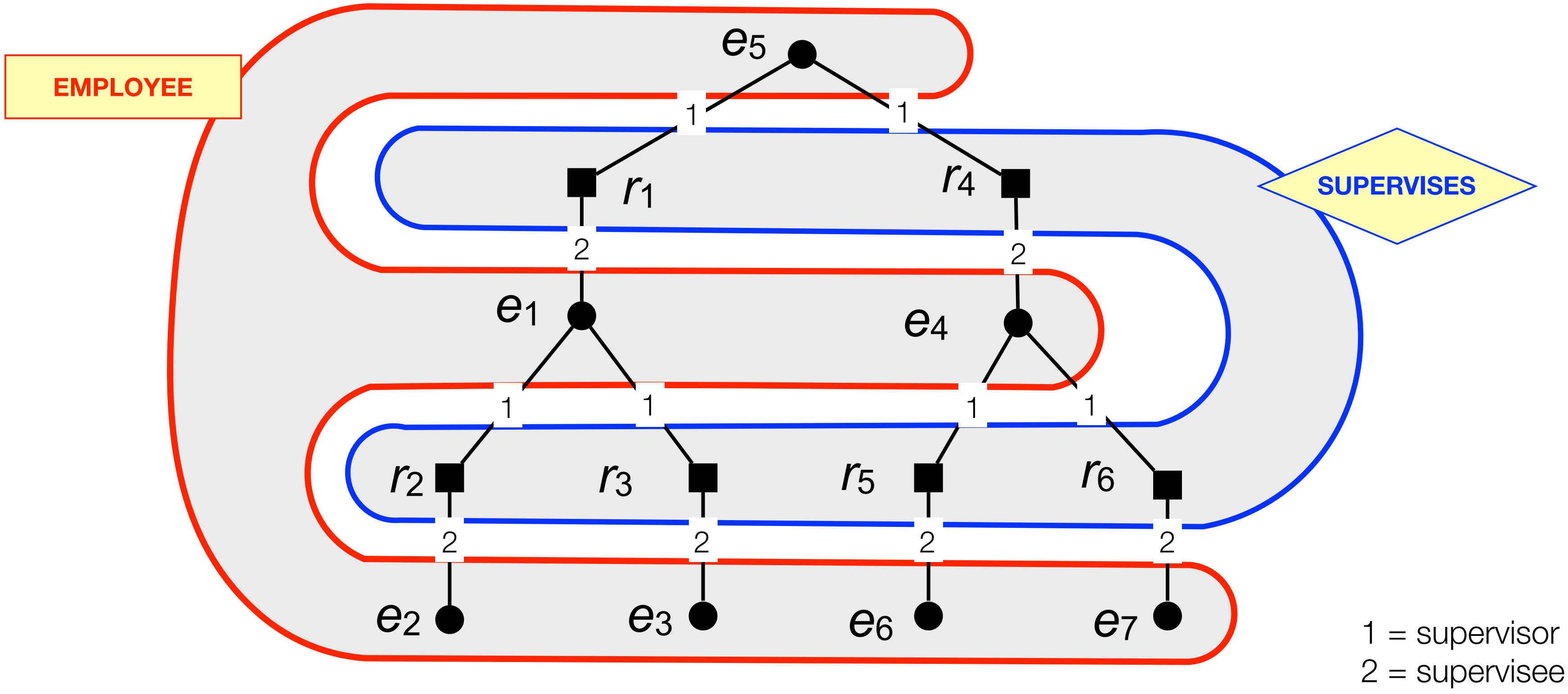


Figure 7.11
A recursive relationship SUPERVISION between EMPLOYEE in the *supervisor* role (1) and EMPLOYEE in the *subordinate* role (2).

Recursive Relationship SUPERVISES



1 = supervisor
2 = supervisee

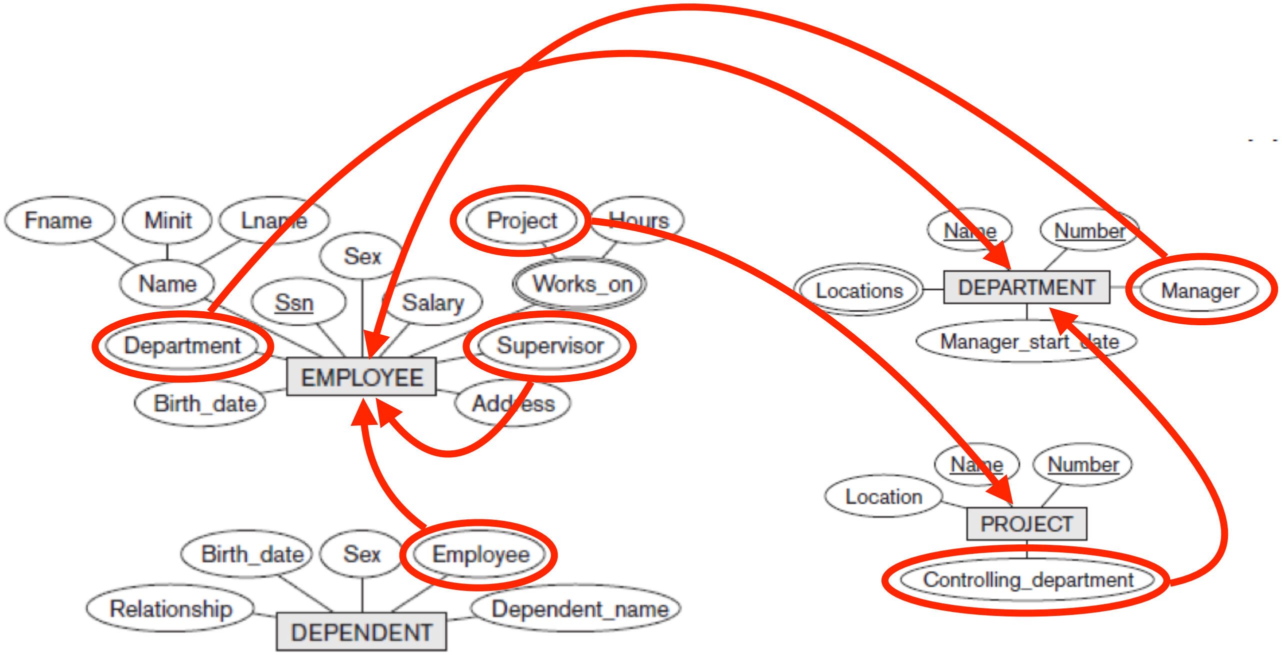
Constraints on Binary Relationship Types

- **Cardinality ratio** for a binary relationship
  - Specifies maximum number of relationship instances in which that entity can participate
- **Participation constraint**
  - Specifies whether existence of entity depends on its being related to another entity
  - Types: **total** and **partial**

Attributes of Relationship Types

- Relationships (like entities) can also have attributes
- Attributes of 1:1 or 1:N relationship types can be migrated to one entity type
- **For 1:1 relationship types**
  - Relationship attribute can be migrated only to either entity type
- **For 1:N relationship types**
  - Relationship attribute can be migrated only to entity type on N-side of relationship
- **For M:N relationship types**
  - Some attributes may be determined by combination of participating entities. Must be specified as relationship attributes

Initial Conceptual Design of the COMPANY Database



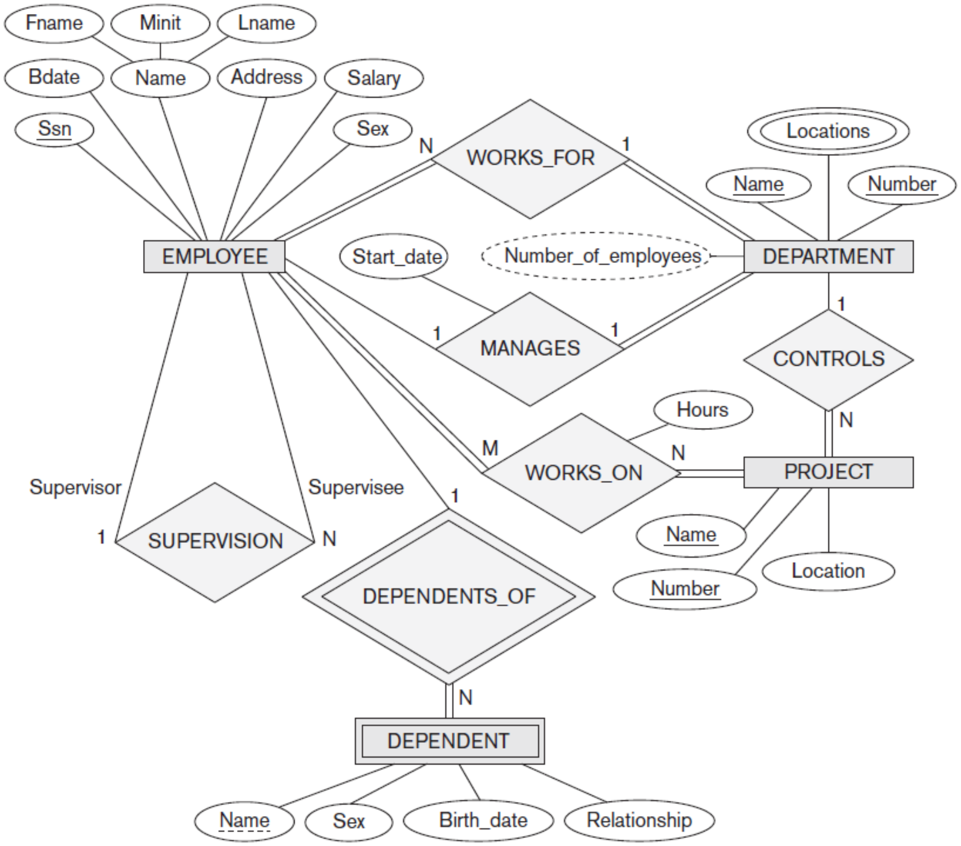


Figure 7.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.

3.5 – Weak Entity Types

Weak Entity Types

- Weak Entities do not have key attributes of their own
  - Identified by being related to specific entities from another entity type. This does not mean they don't have a key, but the "parent" entity's key is part of it.
- Weak Entities have an identifying relationship
  - Relates a weak entity type to its owner
- Always has a total participation constraint
- If the "parent" entity is deleted, all related weak entities are deleted too

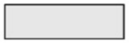
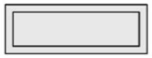
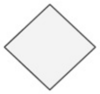
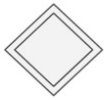
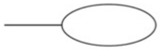
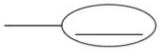
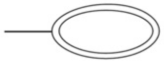
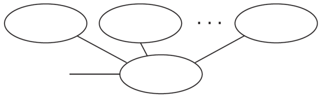
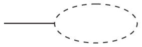
3.6 – Refining the ER Design for the COMPANY Database

Refining the ER Design for the COMPANY Database

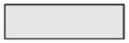
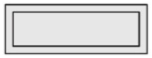
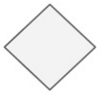
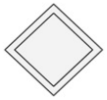
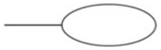
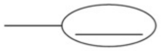
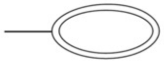
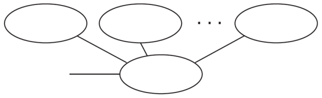
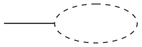
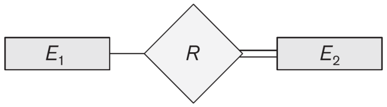
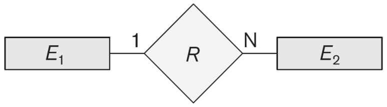
- Change attributes that represent relationships into relationship types
- Determine cardinality ratio and participation constraint of each relationship type

3.7 – ER Diagrams, Naming Conventions, and Design Issues

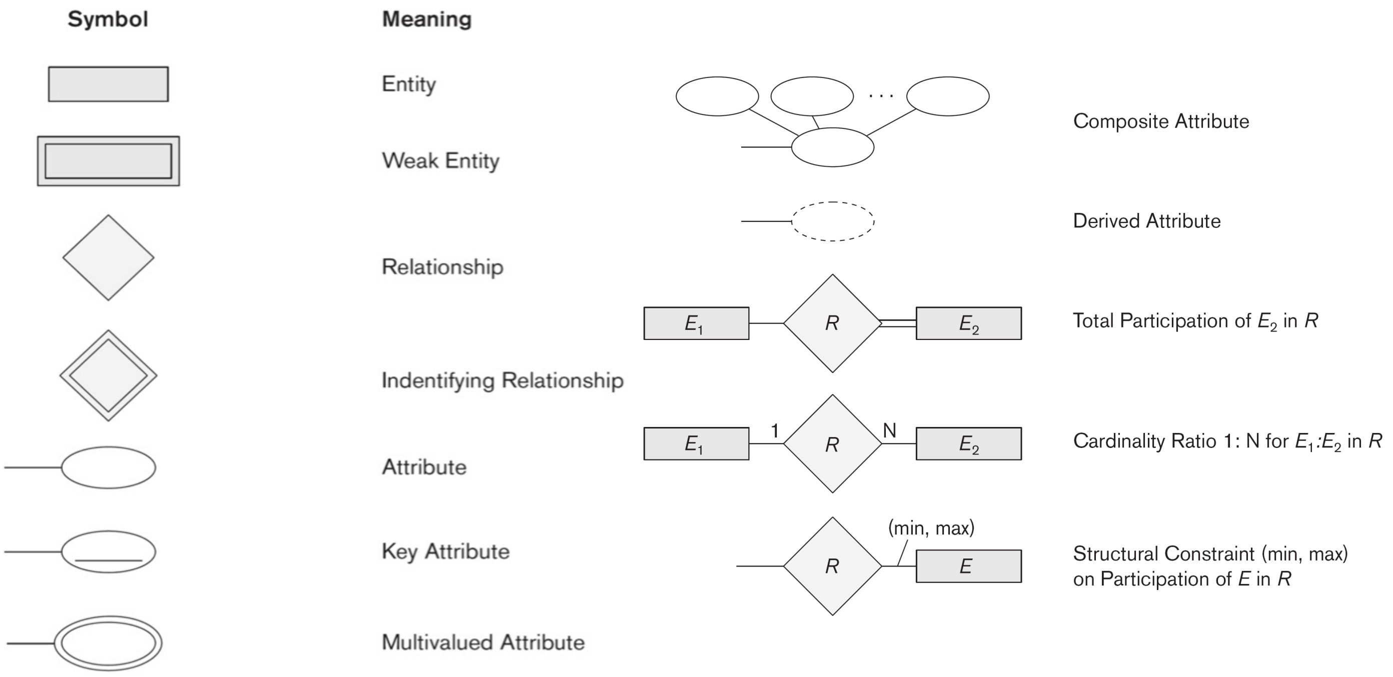
Summary of ER Diagram Notation

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Indentifying Relationship
	Attribute
	Key Attribute
	Multivalued Attribute
	Composite Attribute
	Derived Attribute

Summary of ER Diagram Notation

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Indentifying Relationship
	Attribute
	Key Attribute
	Multivalued Attribute
	Composite Attribute
	Derived Attribute
	Total Participation of E_2 in R
	Cardinality Ratio 1: N for E_1:E_2 in R

Summary of ER Diagram Notation



Symbol **Meaning**

Entity

Weak Entity

Relationship

Indentifying Relationship

Attribute

Key Attribute

Multivalued Attribute

Composite Attribute

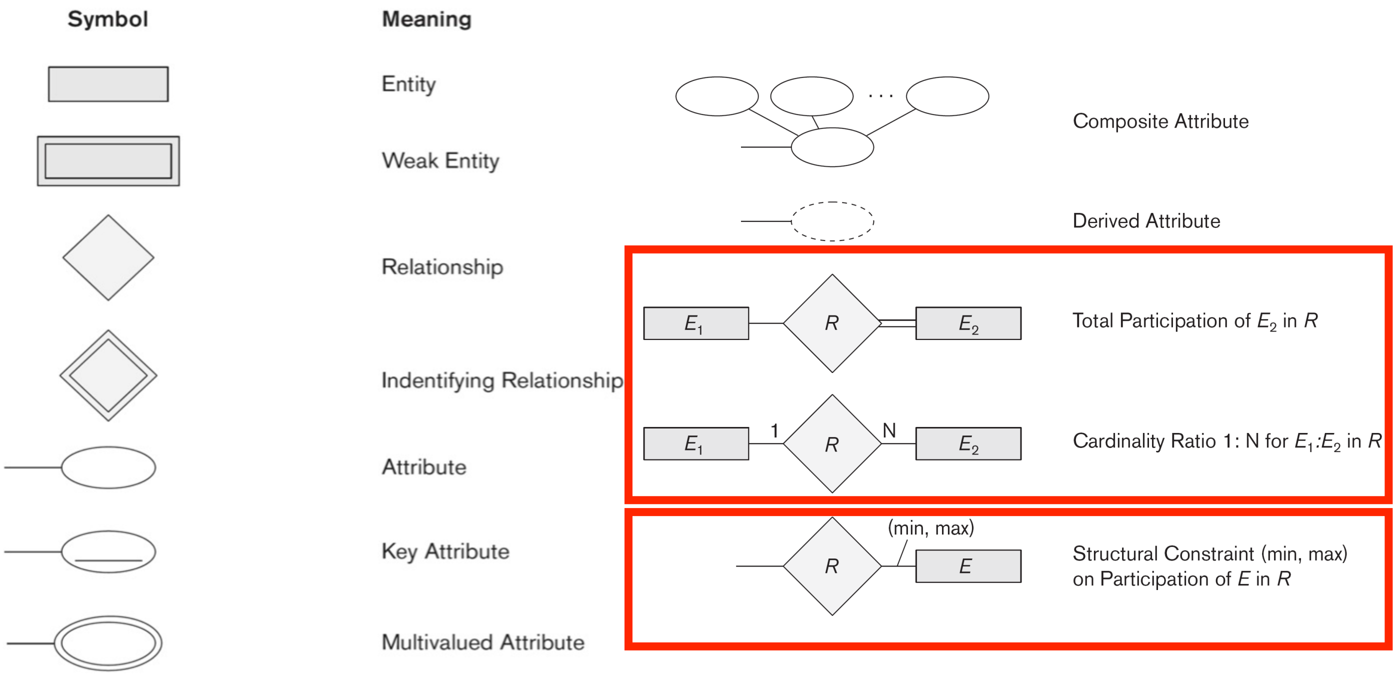
Derived Attribute

Total Participation of E_2 in R

Cardinality Ratio 1: N for E_1:E_2 in R

Structural Constraint (min, max) on Participation of E in R

Summary of ER Diagram Notation



Symbol

Meaning

Entity

Weak Entity

Relationship

Indentifying Relationship

Attribute

Key Attribute

Multivalued Attribute

Composite Attribute

Derived Attribute

Total Participation of E_2 in R

Cardinality Ratio 1: N for E_1:E_2 in R

Structural Constraint (min, max) on Participation of E in R

Proper Naming of Schema Constructs

- Choose names that convey meanings attached to different constructs in schema
- Nouns give rise to entity type names
- Verbs indicate names of relationship types
- Choose binary relationship names to make ER diagram readable from left to right and from top to bottom

Design Choices for ER Conceptual Design

- Model concept first as an attribute
  - Refined into a relationship if attribute is a reference to another entity type
- Attribute that exists in several entity types may be elevated to an independent entity type
  - Can also be applied in the inverse

(min, max) Alternative Notation for

- Specifies structural constraints on relationships
- Replaces cardinality ratio (1:1, 1:N, M:N) and single/double line notation for participation constraint
- Cardinality + Participation and (min,max) are two mutually exclusive options for encoding this info
  - *Mixing or combining these in a homework or exam will be considered incorrect*
- Associate a pair of integer numbers (min, max) with each participation of an entity type E in a relationship type R, where $0 \leq \text{min} \leq \text{max}$ and $\text{max} \geq 1$

ER Diagram for Company Schema

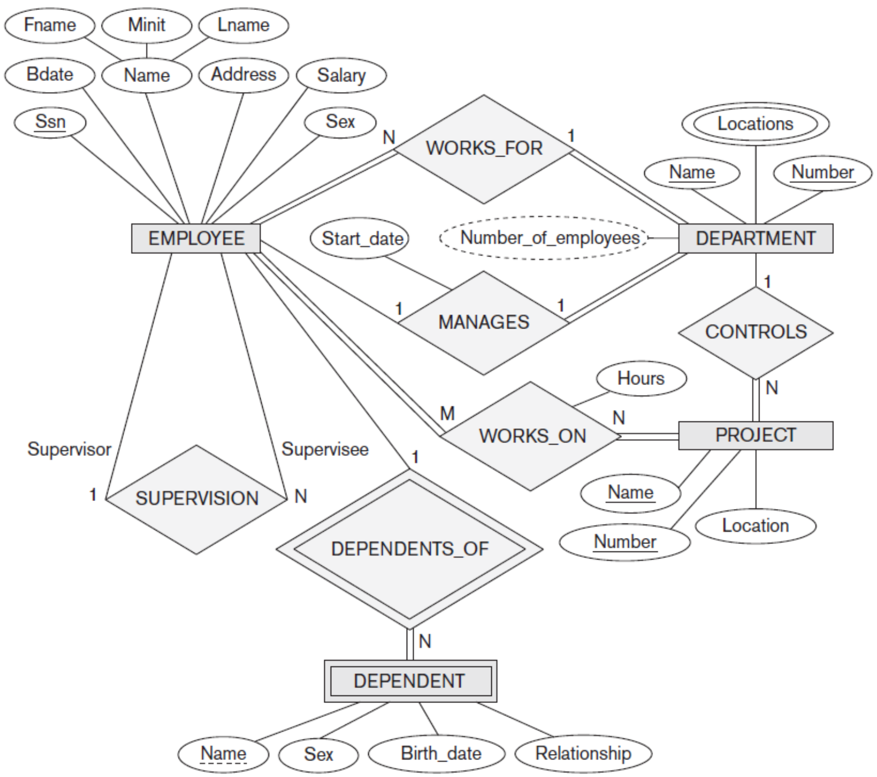


Figure 7.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.

ER Diagram for Company Schema

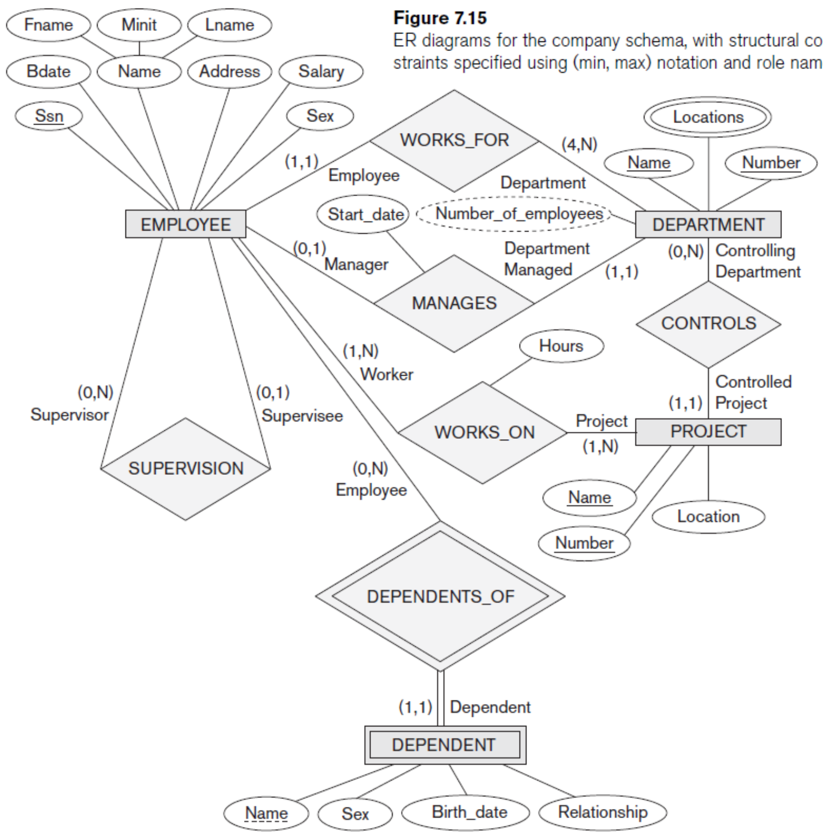


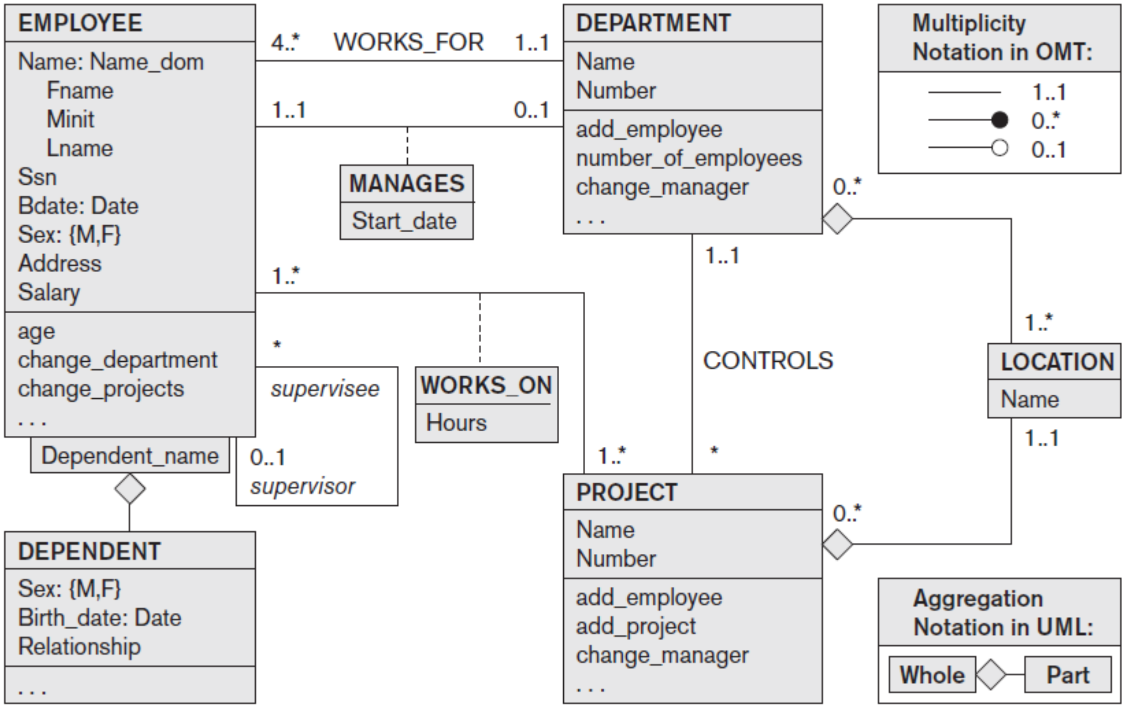
Figure 7.15
ER diagrams for the company schema, with structural constraints specified using (min, max) notation and role names.

Example of Other Notation:

- UML methodology
  - Used extensively in software design
  - Many types of diagrams for various software design purposes
- UML class diagrams
  - Entity in ER corresponds to an object in UML

UML

Figure 7.16
The COMPANY conceptual schema
in UML class diagram notation.



Example of Other Notation:

- Class includes three sections:
    - Top section gives the class name
    - Middle section includes the attributes;
    - Last section includes operations that can be applied to individual objects

Example of Other Notation:

- Associations: relationship types
- Relationship instances: links
- Binary association
  - Represented as a line connecting participating classes
  - May optionally have a name
- Link attribute
  - Placed in a box connected to the association’s line by a dashed line

Example of Other Notation:

- Multiplicities: min..max, asterisk (*) indicates no maximum limit on participation
- Types of relationships: association and aggregation
- Distinguish between unidirectional and bidirectional associations
- Model weak entities using qualified association

3.9 – Relationship Types of Degree Higher than Two

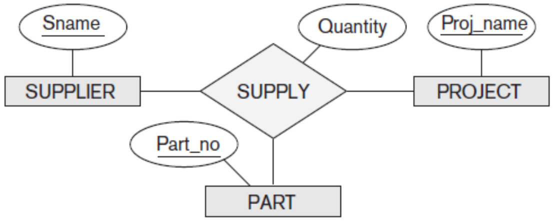
Relationship Types of Degree

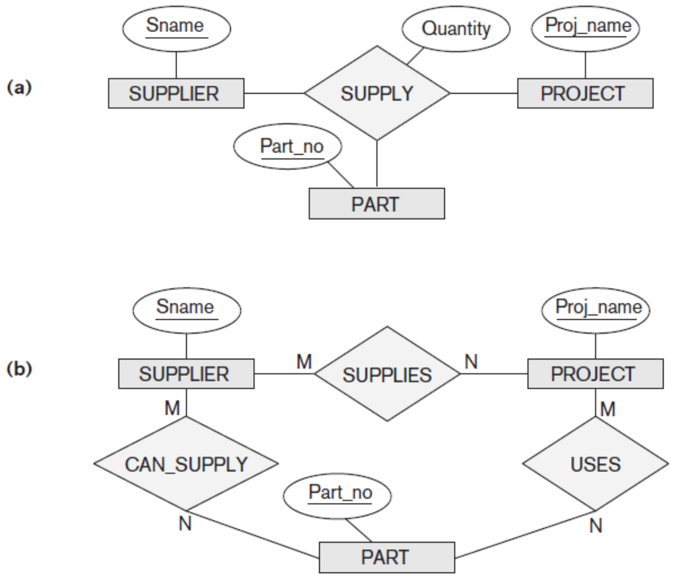
- Degree of a relationship type
  - Number of participating entity types
- Binary
  - Relationship type of degree two
- Ternary
  - Relationship type of degree three

Choosing between Binary and Ternary (or Higher-

- Some database design tools permit only binary relationships
  - Ternary relationship must be represented as a weak entity type
  - No partial key and three identifying relationships
- Represent ternary relationship as a regular entity type
  - By introducing an artificial or surrogate key

(a)


Sname
Quantity
Proj_name
SUPPLIER
SUPPLY
PROJECT
Part_no
PART


(a)
Sname
Quantity
Proj_name
SUPPLIER
SUPPLY
PROJECT
Part_no
PART
(b)
Sname
Proj_name
SUPPLIER
M
SUPPLIES
N
PROJECT
M
M
CAN_SUPPLY
USES
Part_no
N
N
PART

Some toolkits cannot directly represent *n*-ary relationships

(b) is *NOT* equivalent to (a)

(c) *IS* equivalent to (a)

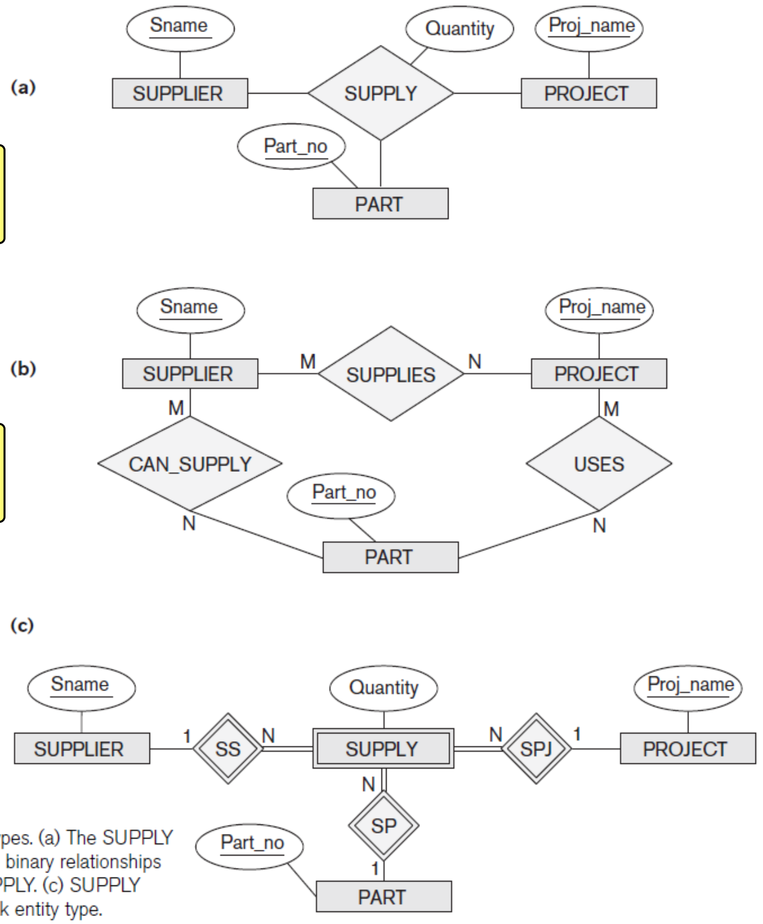
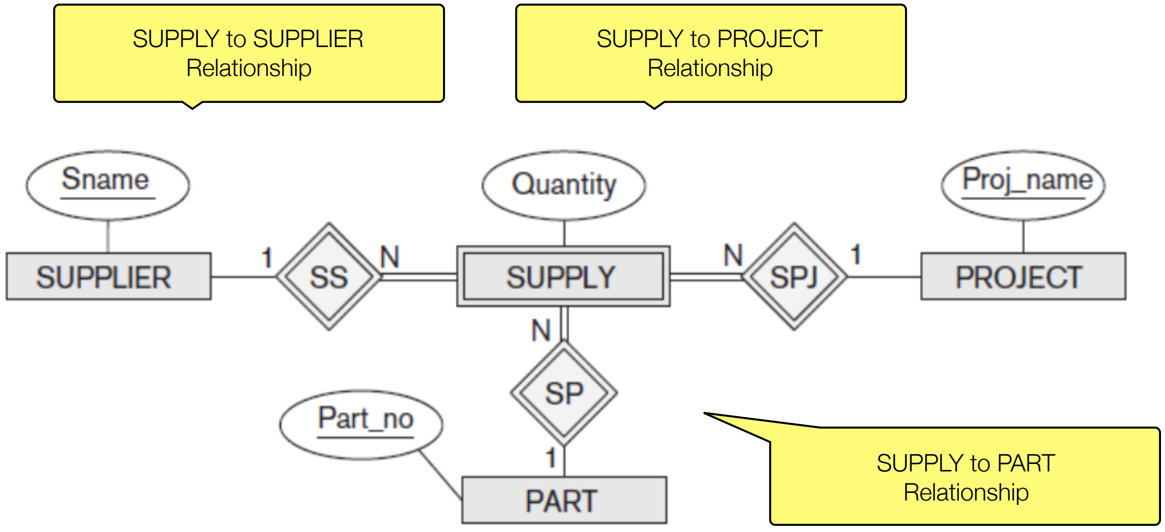


Figure 7.17
Ternary relationship types. (a) The SUPPLY relationship. (b) Three binary relationships not equivalent to SUPPLY. (c) SUPPLY represented as a weak entity type.

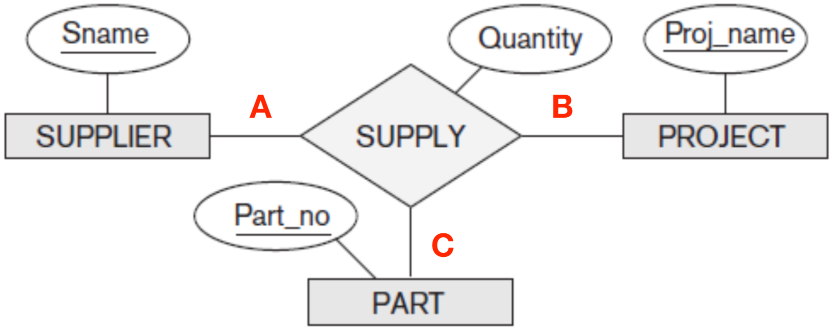
n-ary to Multiple Binary

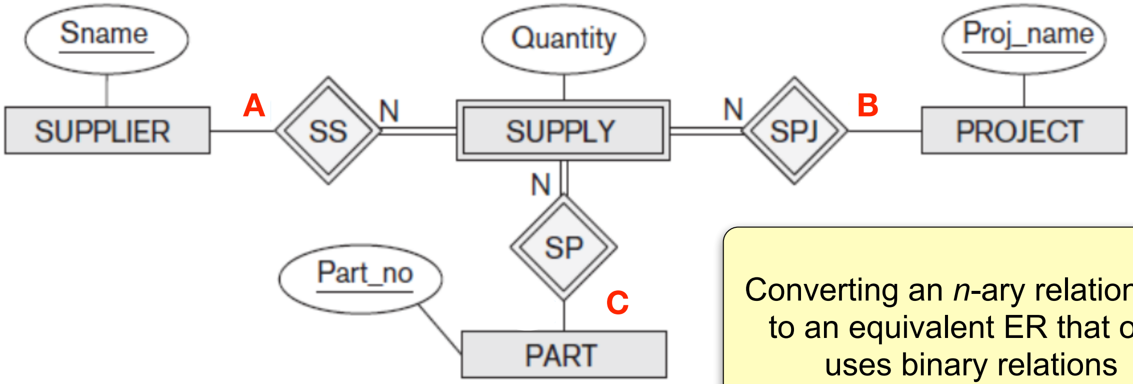

SUPPLY to SUPPLIER Relationship

SUPPLY to PROJECT Relationship

SUPPLY to PART Relationship

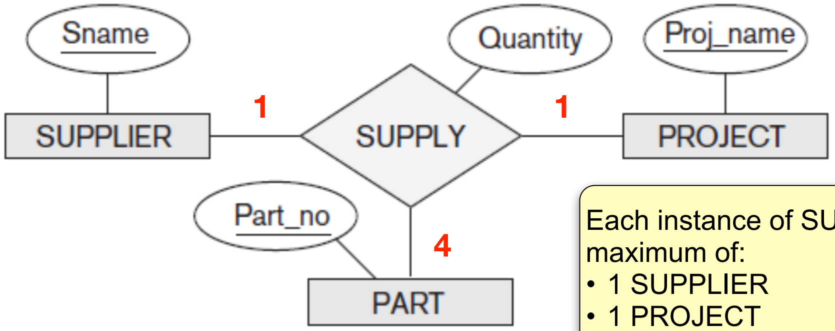
Cardinality Mapping





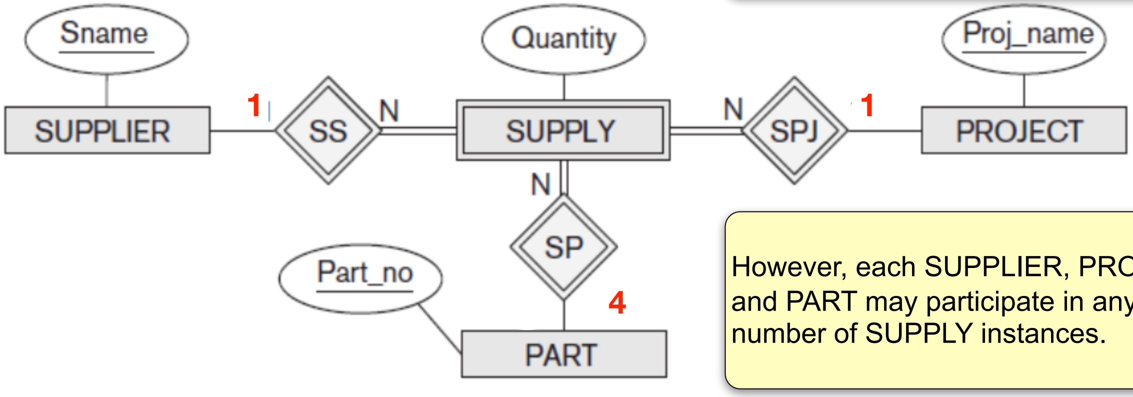
Converting an *n*-ary relationship to an equivalent ER that only uses binary relations

Cardinality Mapping



Each instance of SUPPLY may have a maximum of:

- 1 SUPPLIER
- 1 PROJECT
- 4 PARTS



However, each SUPPLIER, PROJECT, and PART may participate in any number of SUPPLY instances.

Constraints on Ternary (or Higher-Degree)

- Notations for specifying structural constraints on n-ary relationships
  - Should both be used if it is important to fully specify structural constraints