CS430: Entity-Relationship Model (E-R model)

Dr. Chetan Jaiswal,
Department of Computer Science,
Truman State University
cjaiswal@truman.edu

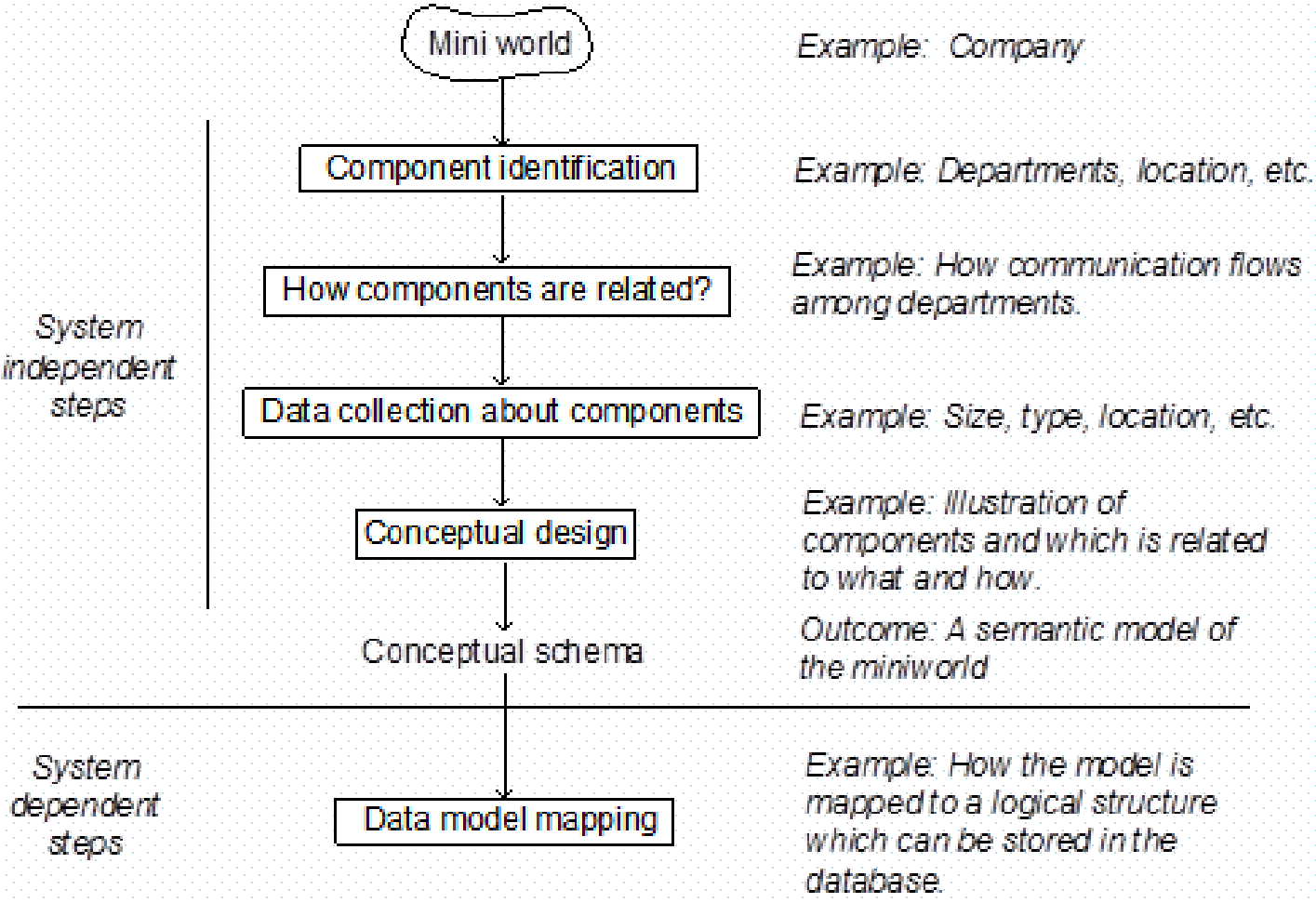
Chapter 3

- Introduction to E-R model
- A Sample Database Application
- Entity Types, Entity Sets, Attributes, and Keys
- Relationship Types, Relationship Sets, Roles, and Structural Constraints
- Weak Entity Types
- Refining the ER Design for the COMPANY Database
- ER Diagrams, Naming Conventions, and Design Issues
- Relationship Types of Degree Higher than Two

Conceptual Database Design by E-R Model

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

Database Modeling Process



Conceptual Database Design by E-R Model

- **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

Conceptual Database Design by E-R Model

- **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

A Sample Database by E-R Model

- **COMPANY**
  - Result is a database schema in implementation data model of DBMS
  - 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

A Sample Database by E-R Model

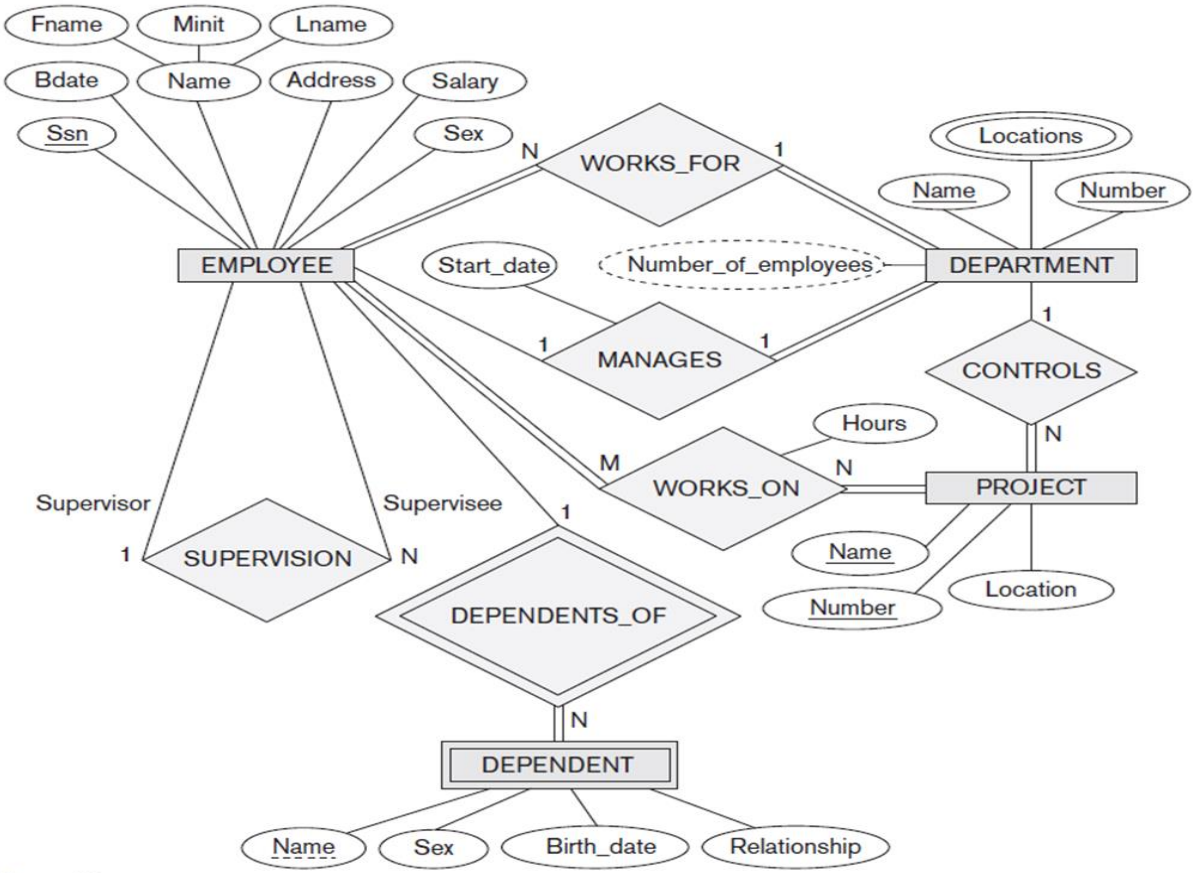


Figure 7.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.

Entity Types, Entity Sets, Attributes, and Keys

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

Entity

- **Entities**
  - Thing in real world with independent existence. It is also referred to as object. Example: Chair, Computer, Employee, etc.
- **Entity or Strong Entity**
  - Exists independently. Ex: Employee
- **Weak entity**
  - Its existence depends on a strong entity to uniquely identify itself. Example: Dependent. Employee has dependent. If employee disappears then the dependent also disappears.

Attributes

- **Attributes**
  - **Particular properties that describe entity**
  - **Types of attributes**
    - ***Composite* versus *simple* (atomic) *attributes***
    - **Single-valued versus multivalued attributes**
    - **Stored versus derived attributes**
    - **NULL values**
    - **Complex attributes**

Attributes Types

- **Atomic**
  - Cannot be split into meaningful parts. Example: course grade, SSN, etc. SSN can be split into parts but a part will have no meaning.
- **Composite**
  - Can be split into more than one part and each part have some meaning. Example: Phone number. Can be split into three parts: area code, location code and the number.

Attributes Types

- **Single value**
  - Can have only one value. Example: SSN.
- **Multivalued**
  - Can have more than one valid values for the same entity
  - Example: Phone number. An employee John can have more than one phone number.

    - Stored versus derived attributes
    - NULL values
    - Complex attributes

Attributes Types

- **Derived**
  - **Its value can be derived (compute) from another attribute value. Example: Age (derived) value can be computed from the date of birth.**

Attributes Types

- **Complex:**

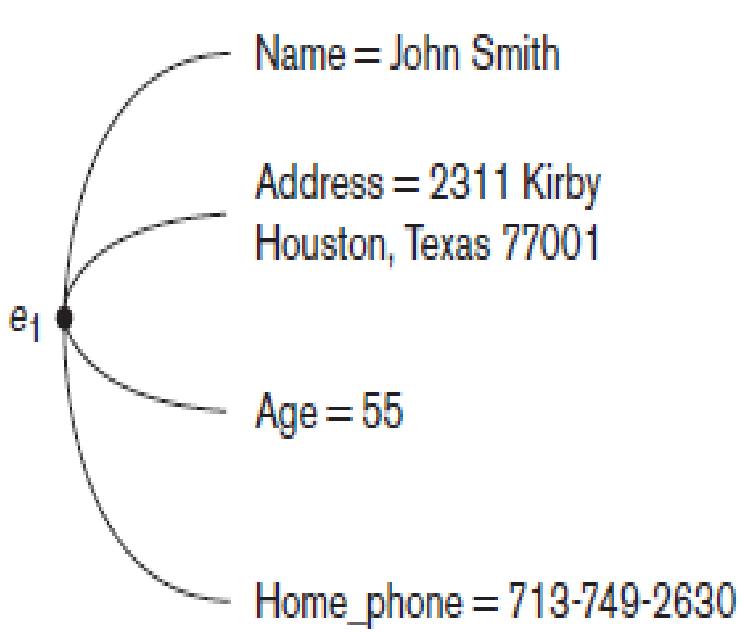
Multivalued = {}, Composite = (). One attribute may have both features.

Ex. A person can have more than one residence and each residence can have a single address and multiple phones.

:: {Address_phone({Phone(Area_code,Phone_number)}, Address(Street_address(Number, Street, Apartment_number), City, State, Zip)) }

- **Key**
  - **An attribute that uniquely identifies an entity. Example: SSN value uniquely identifies an employee or a person.**
  - **No primary key concept like relational algebra in ER model**
  - **Primary key is chosen during mapping from ER Model to Relational Schema**

Entity Types, Entity Sets, Attributes, and Keys



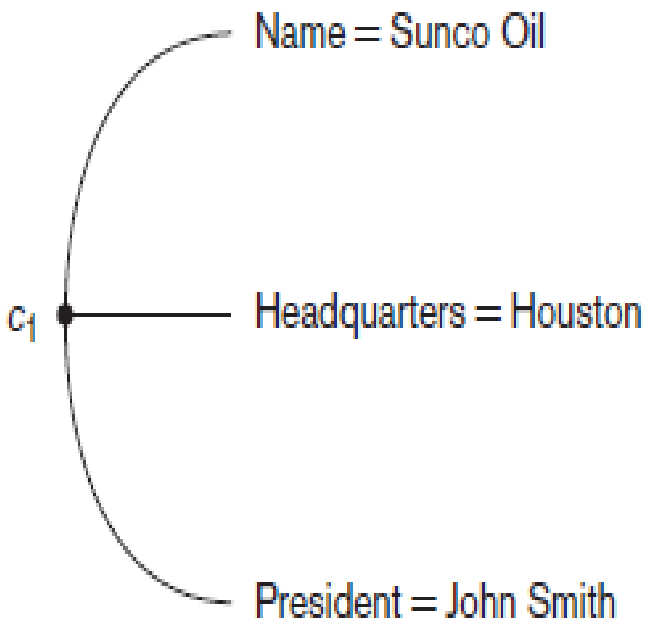


Figure 7.3
Two entities, EMPLOYEE e_1, and COMPANY c_1, and their attributes.

Entity Types, Entity Sets, Attributes, and Keys

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

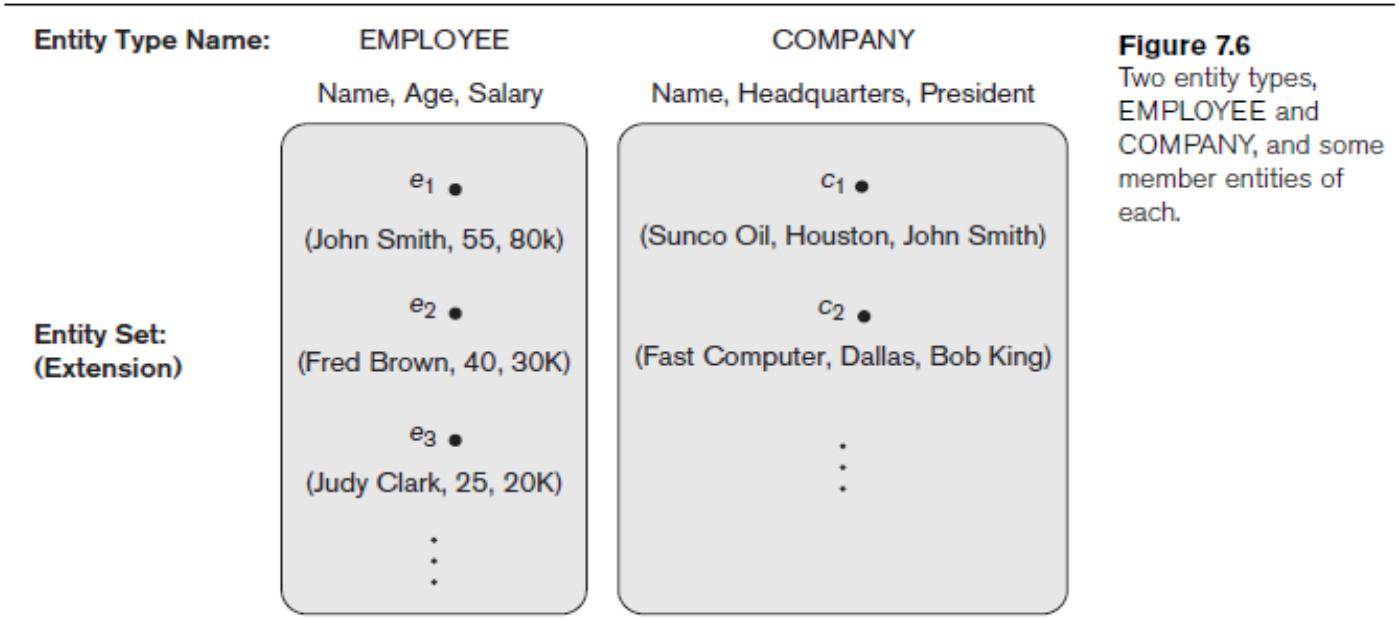


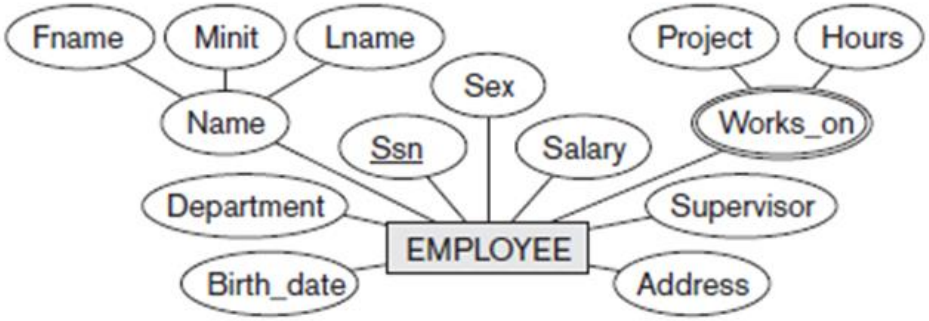
Figure 7.6
Two entity types, EMPLOYEE and COMPANY, and some member entities of each.

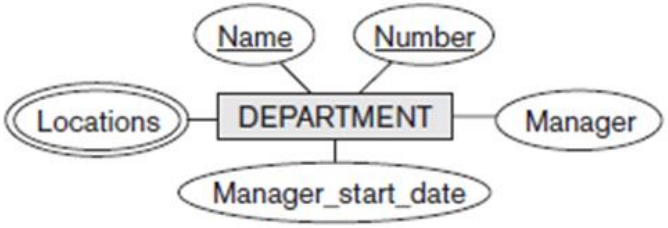
Value Set

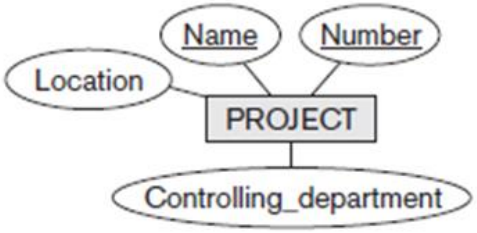
- **Value sets** (or **domain of values**)
  - **Specifies set of values that may be assigned to that attribute for each individual entity**

Initial Conceptual Design of the COMPANY

ER Diagram
1. How attributes are represented?
2. How multivalued or composite are represented?
3. How entity type name is represented?







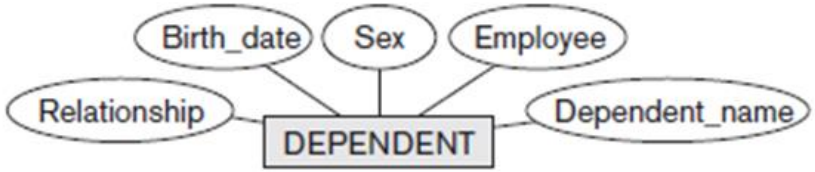


Figure 7.8
Preliminary design of entity types for the COMPANY database. Some of the shown attributes will be refined into relationships.

How normal people see relationships

How database engineers see relationships

Relationship Types, Relationship Sets, Roles, and Structural Constraints

- **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_1, e_2, ..., e_n)
  - Each entity e_j in r_i is a member of entity set E_j

Relationship Degree

- Degree of a relationship type
  - Number of participating entity types
  - Binary, ternary
- Relationships as attributes
  - Think of a binary relationship type in terms of attributes

Relationship Degree

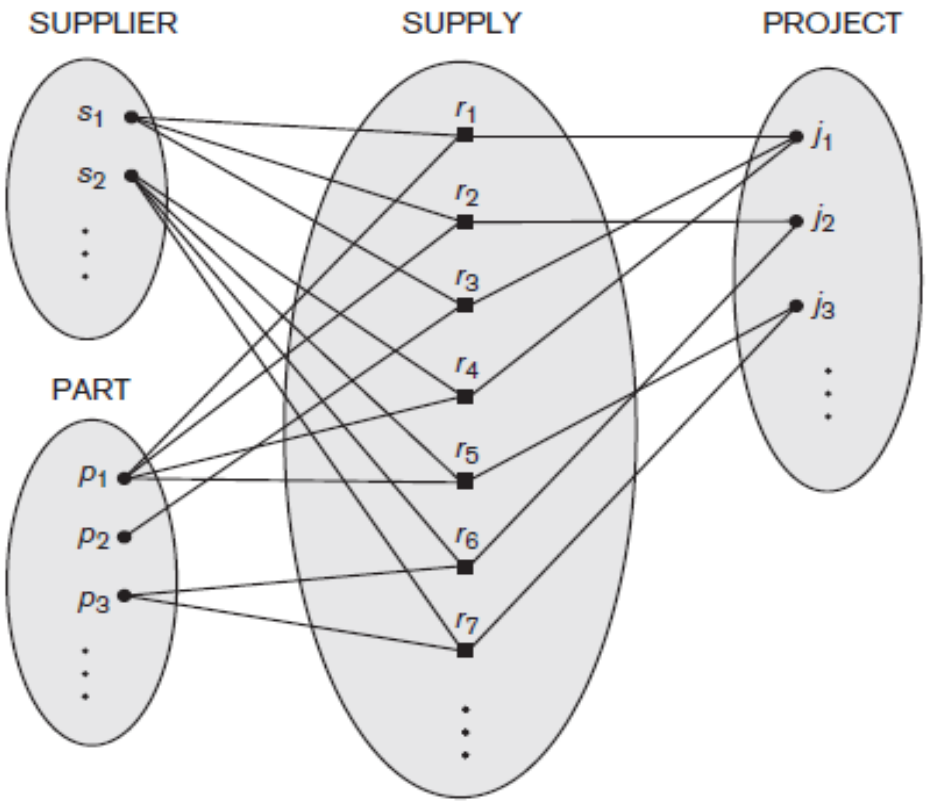


Figure 7.10
Some relationship instances in the SUPPLY ternary relationship set.

Role Names and Recursive Relationships

- **Role names**
  - Role name signifies role that a participating entity plays in each relationship instance
- **Recursive relationships**
  - Same entity type participates more than once in a relationship type in different roles
  - Must specify role name

Role Names and Recursive Relationships

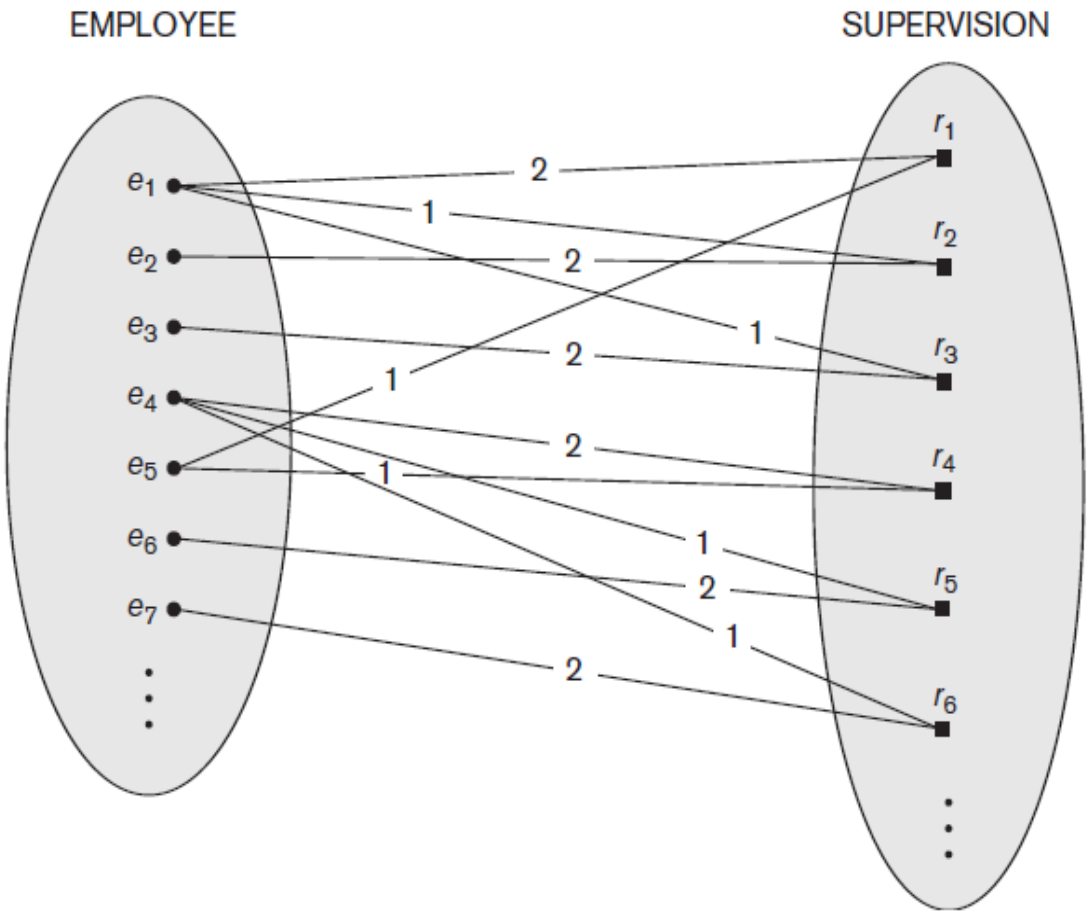
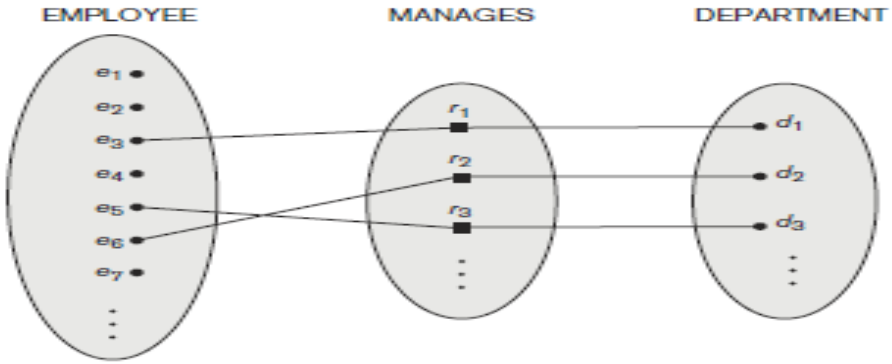


Figure 7.11
A recursive relationship SUPERVISION between EMPLOYEE in the *supervisor* role (1) and EMPLOYEE in the *subordinate* role (2).

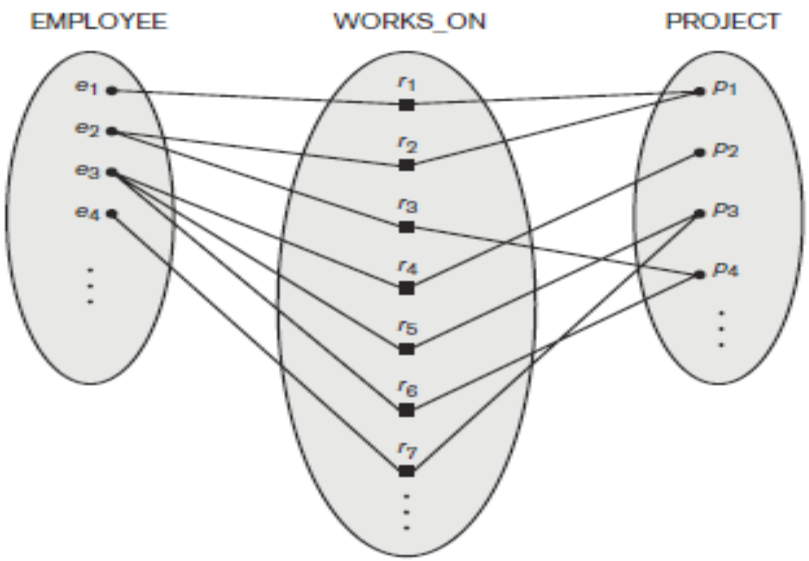
Constraints on Binary Relationship Types

- **Cardinality ratio for a binary relationship**
  - **Specifies maximum number of relationship instances that entity can participate in (for ex. DEPARTMENT:EMPLOYEE, works for binary relationship has cardinality ratio 1:N)**
  - **Can be 1:1, 1:N, N:1, M:N. Is below 1:1 or 1:N?**



Constraints on Binary Relationship Types

- How about this?



Constraints on Binary Relationship Types

- **Participation constraint**
  - Specifies whether existence of entity depends on its being related to another entity
  - Types: *total* and *partial*

Constraints on Binary Relationship Types

- **Participation constraint**
  - Specifies whether existence of entity depends on its being related to another entity
  - Types: *total* and *partial*
  - If company policy says: every employee must work for a department, in other words employee entity can exist only if it participates in WORKS_FOR relationship instance, also called total participation.
  - Every entity in total set of employee entities must be related to department entity via WORKS_FOR, also called existence dependency.

Constraints on Binary Relationship Types

- **Participation constraint**
  - Lets say, EMPLOYEE and DEPARTMENT related through MANAGES relationship.
  - We do not expect every employee to manage a department, thus the participation of EMPLOYEE in the MANAGES relationship type is partial.
  - Some or part of the set EMPLOYEE entities are related to some department entity via MANAGES

Constraints on Binary Relationship Types

- Cardinality ratio and participation constraint together called structural constraint of a relationship type.

In ER,

total participation = double line

partial participation = single line

Entity Types

- **Weak Entity types: Do not have key attributes of their own**
  - **Identified by being related to specific entities from another entity type**
  - **Identifying relationship: Relates a weak entity type to its owner**
  - **Always has a total participation constraint**
- **Strong entity types: Do have a key attribute**

Entity Types

- **Weak Entity types:**

-Not every total participation/existence dependency results in a weak entity type. Ex. DRIVER_LICENSE cannot exist without PERSON, but it has its own key, not a weak entity type

-DEPENDENT:EMPLOYEE a N:1 relationship. Two dependents can have same credentials, they are identified only after determining the particular employee entity

Refining the ER Schema of the COMPANY Database

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

ER Model, Naming Conventions, and Design Issues

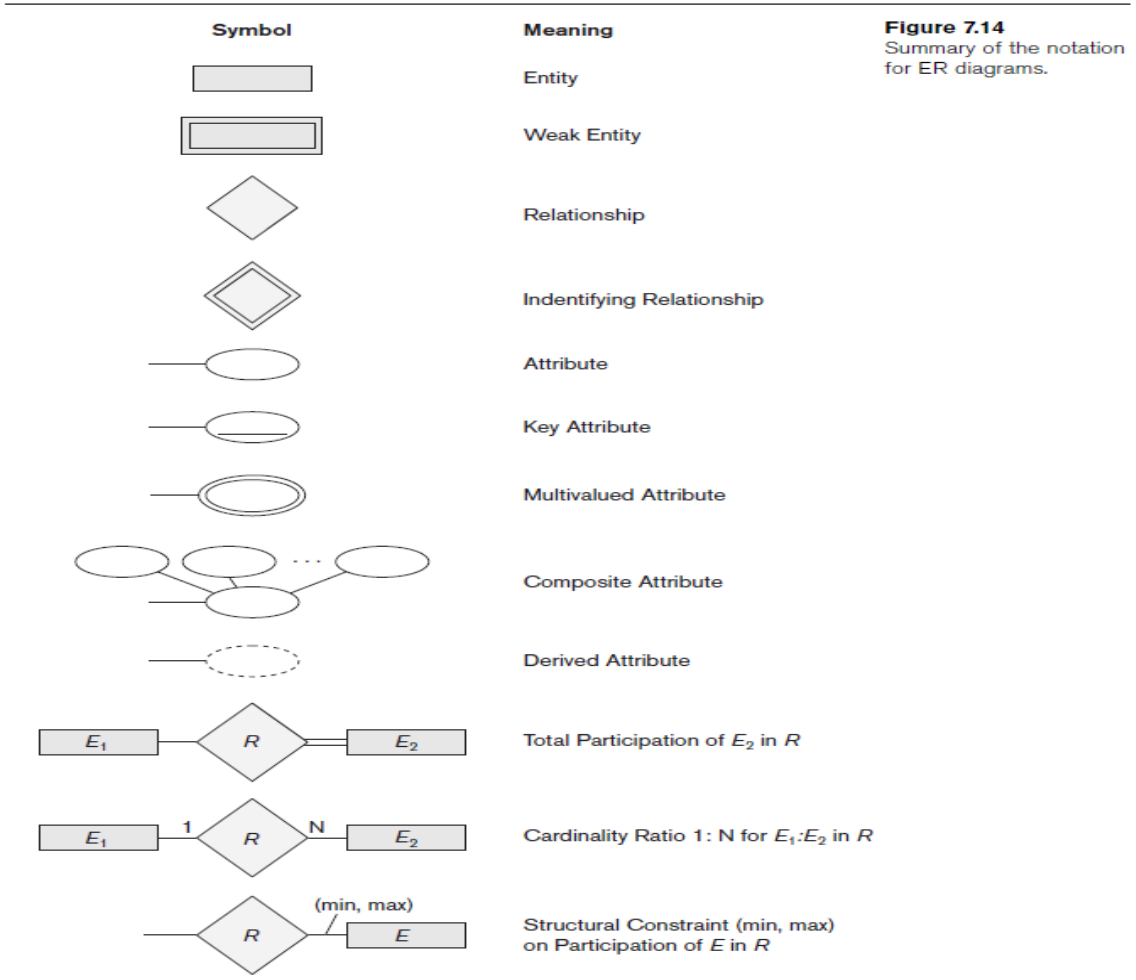


Figure 7.14
Summary of the notation for ER diagrams.

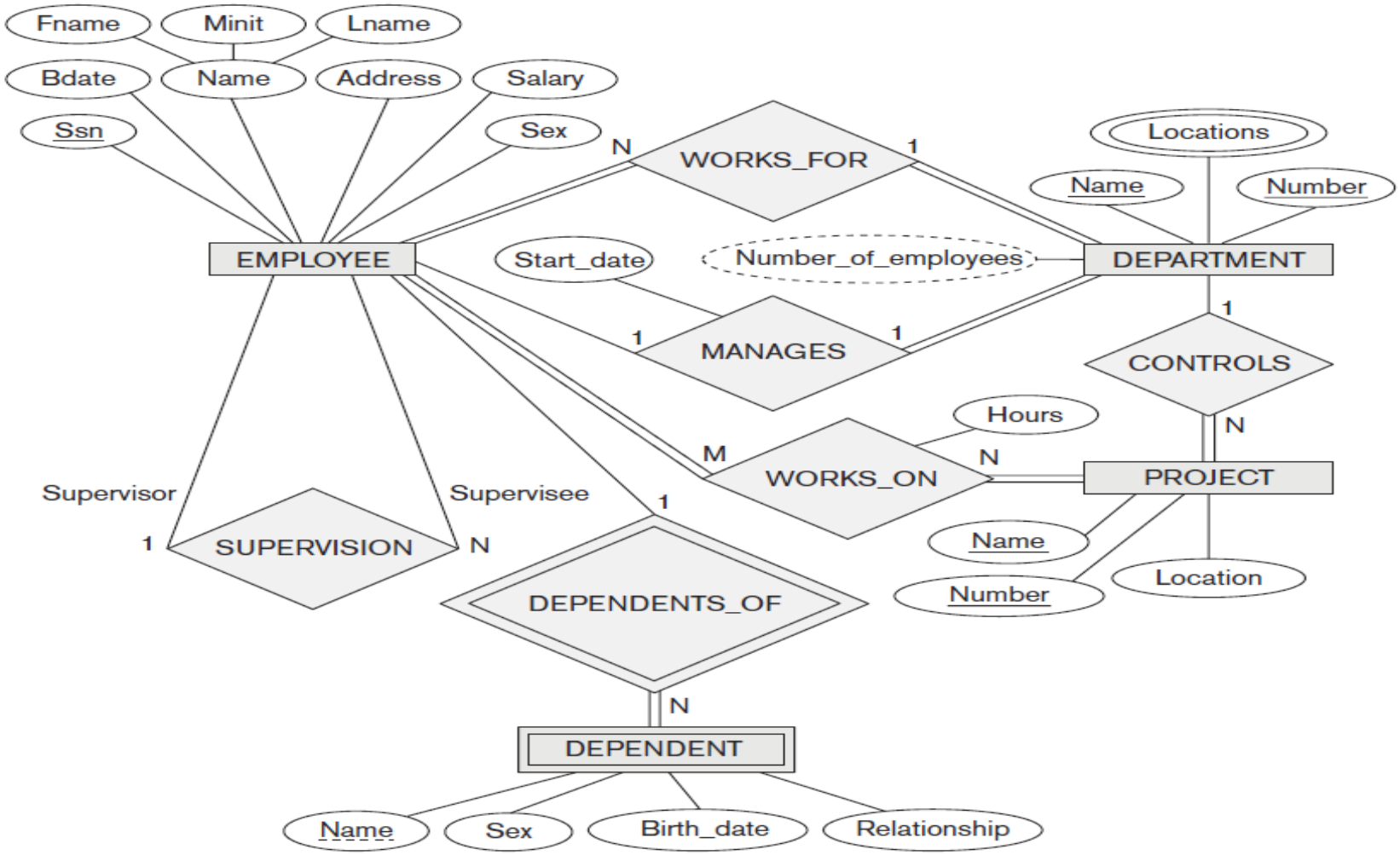
Proper Naming of Schema Constructs

- 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

Alternative Notations for ER Diagrams

- **Specify structural constraints on relationships**
  - **Replaces cardinality ratio (1:1, 1:N, M:N) and single/double line notation for participation constraints**
  - **Associate a pair of integer numbers (min, max) with each participation of an entity type *E* in a relationship type *R*, where 0 ≤ min ≤ max and max ≥ 1**
  - **It means for each entity e in E, e must participate in at least min and at most max relationship instances in R, at any point of time.**
  - **Min = 0, partial participation.**
  - **Min>0, total participation**
  - **Use either cardinality ratio/single/double line or (min,max)**

ER Diagram


Fname
Minit
Lname
Name
Bdate
Address
Salary
Ssn
Sex
N
WORKS_FOR
1
Locations
Name
Number
EMPLOYEE
Start_date
Number_of_employees
DEPARTMENT
1
MANAGES
1
1
CONTROLS
Hours
N
M
WORKS_ON
N
PROJECT
Supervisor
Supervisee
1
1
SUPERVISION
N
Name
Location
Number
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
Birth_date
Relationship

Alternative Notations for ER Diagrams

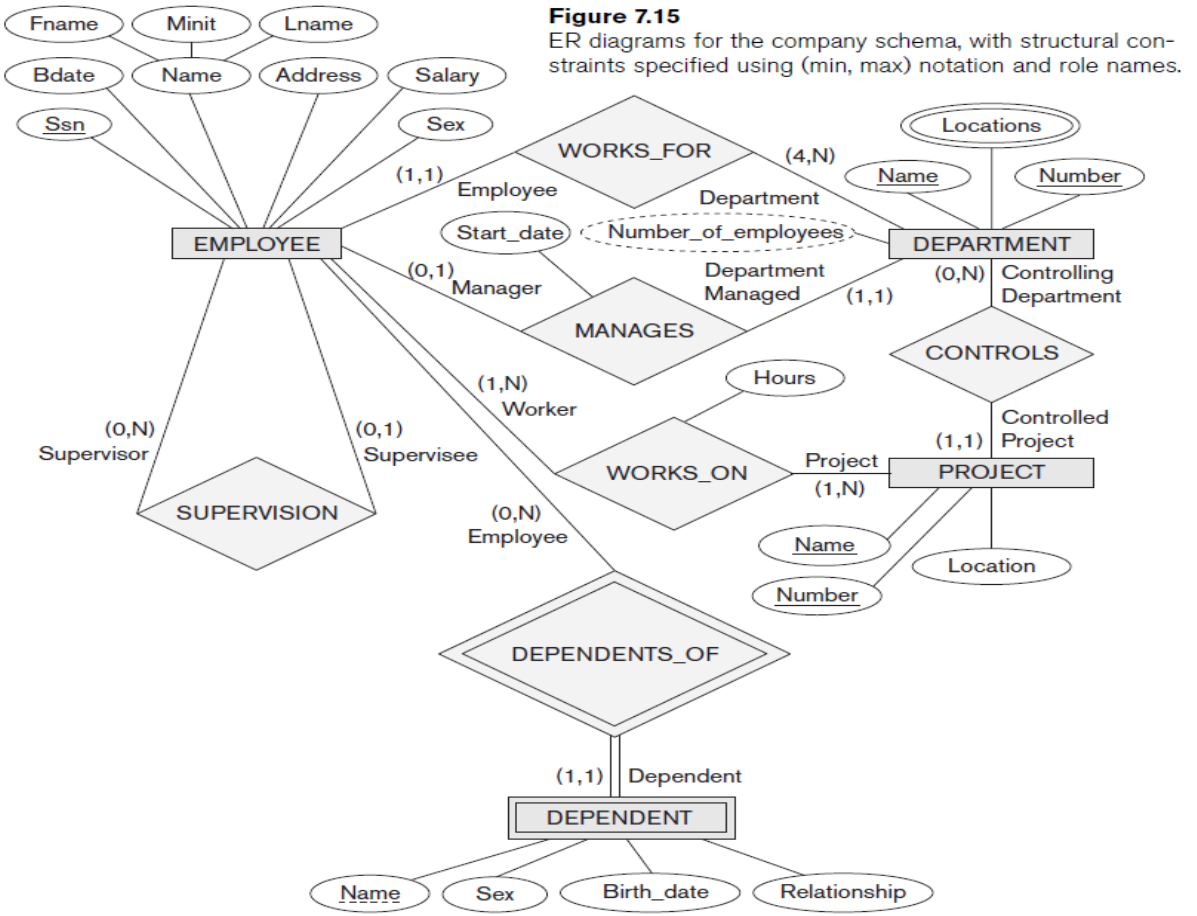


Figure 7.15
ER diagrams for the company schema, with structural constraints specified using (min, max) notation and role names.

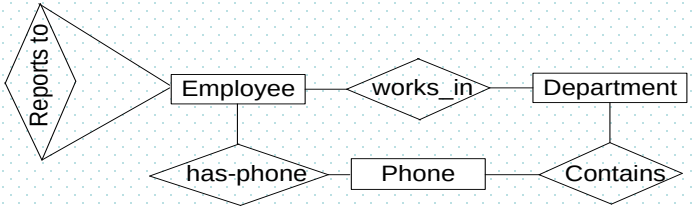
References

http://www.ischool.drexel.edu/faculty/song/courses/info605/appendix/AppendixA.PDF

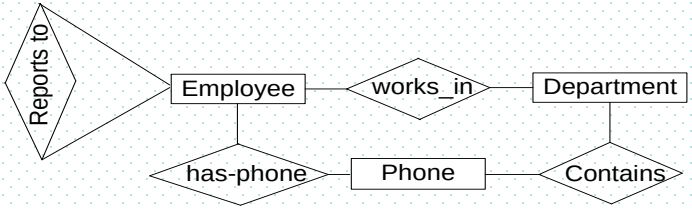
http://www.csc.lsu.edu/~chen/pdf/english.pdf

http://www.csc.lsu.edu/~chen/pdf/framework.pdf

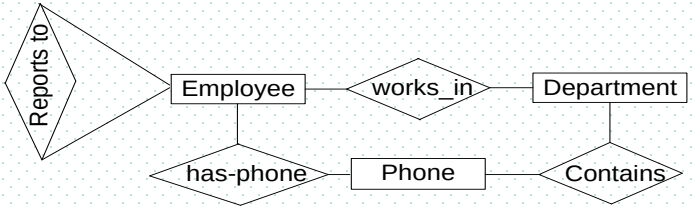
a. *All employees have a minimum of 1 phone and a maximum of 5 phones. Also only some departments have (contains) phones.*



b. *Only a few employees report to a supervisor and some employees are not assigned to any department.*



c. *A maximum of 5 departments and a minimum of 0 (Zero) departments contains phone and no employee has a phone.*



d. Modify the schema (provide all missing information) if: *All employees work in all departments, all departments contain all the phones, and no employee report to a supervisor.*