

Entity Relationship Model

Dr. Hari Seetha

Database Design Process

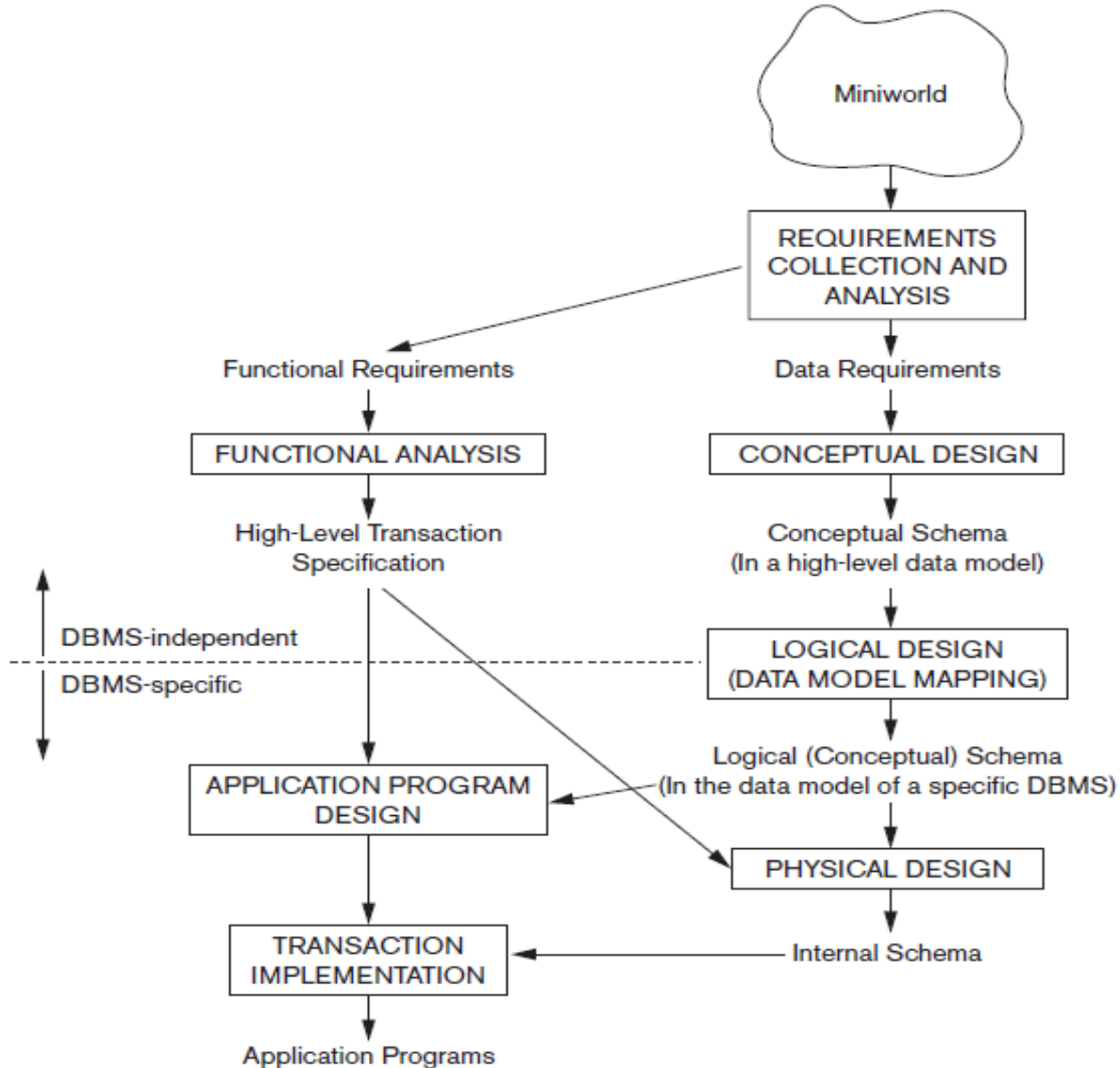


Figure 3.1

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

ER Model Concepts

- Entities and Attributes

- Entities are specific objects or things 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 a Name, SSN, Address, Sex, BirthDate

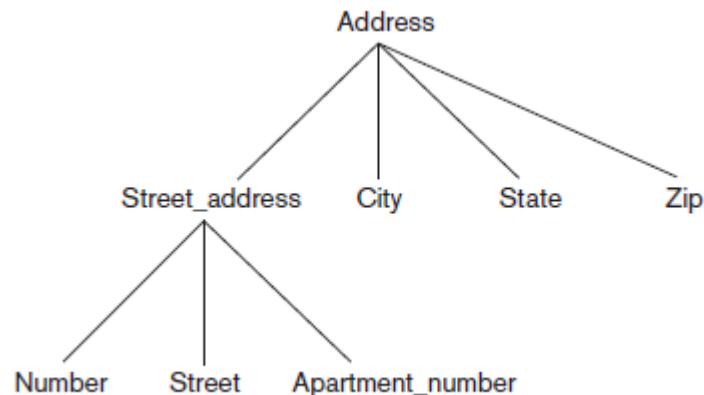
- A specific entity will have a value for each of its attributes.

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, subrange, enumerated type, ...

Types of Attributes (1)

- **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.



Types of Attributes (2)

- **Stored vs Derived Attribute:** In some cases, two (or more) attribute values are related—Age and Birthdate. Birthdate is stored and age is derived
- **NULL Value:** In some cases, a particular entity may not have an applicable value for an attribute.
- A NULL value is assigned in case of missing information, attribute NOT APPLICABLE, attribute value is unknown

Types of Attributes (3)

- **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}.

- **Complex**

- In general, composite and multi-valued attributes may be nested arbitrarily to any number of levels although this is rare.

For example,

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

Entity Types and Key Attributes

- Collection of Entities with the same basic attributes are grouped or typed into an **entity type**. For example, the EMPLOYEE entity type or the PROJECT entity type.
- 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. For example, VehicleTagNumber is a key of the CAR entity type with components (Number, State).
- An entity type may have more than one key. For example, the CAR entity type may have two keys:
 - VehicleIdentificationNumber (popularly called VIN) and
 - VehicleTagNumber (Number, State), also known as license_plate number.

ENTITY SET corresponding to the ENTITY TYPE CAR

CAR

Registration(RegistrationNumber, State), VehicleID, Make, Model, Year, (Color)

car₁

((ABC 123, TEXAS), TK629, Ford Mustang, convertible, 1999, (red, black))

car₂

((ABC 123, NEW YORK), WP9872, Nissan 300ZX, 2-door, 2002, (blue))

car₃

((VSY 720, TEXAS), TD729, Buick LeSabre, 4-door, 2003, (white, blue))

◦

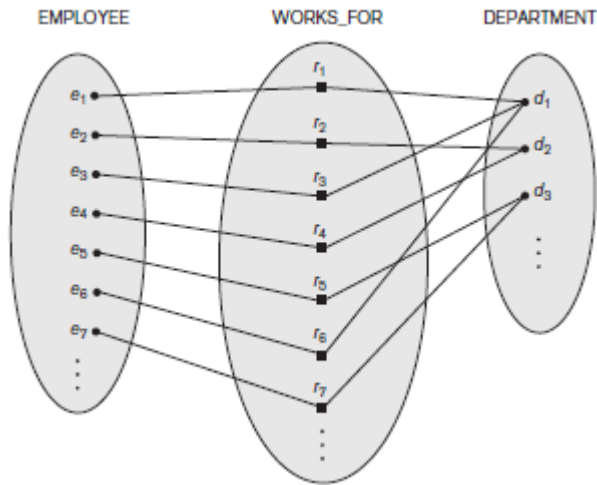
◦

◦

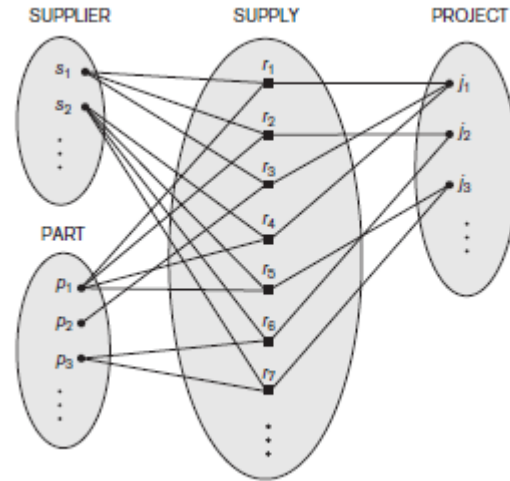
Relationships and Relationship Types (1)

- 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 MANAGES and WORKS_ON are binary relationships.

Degree of a Relationship

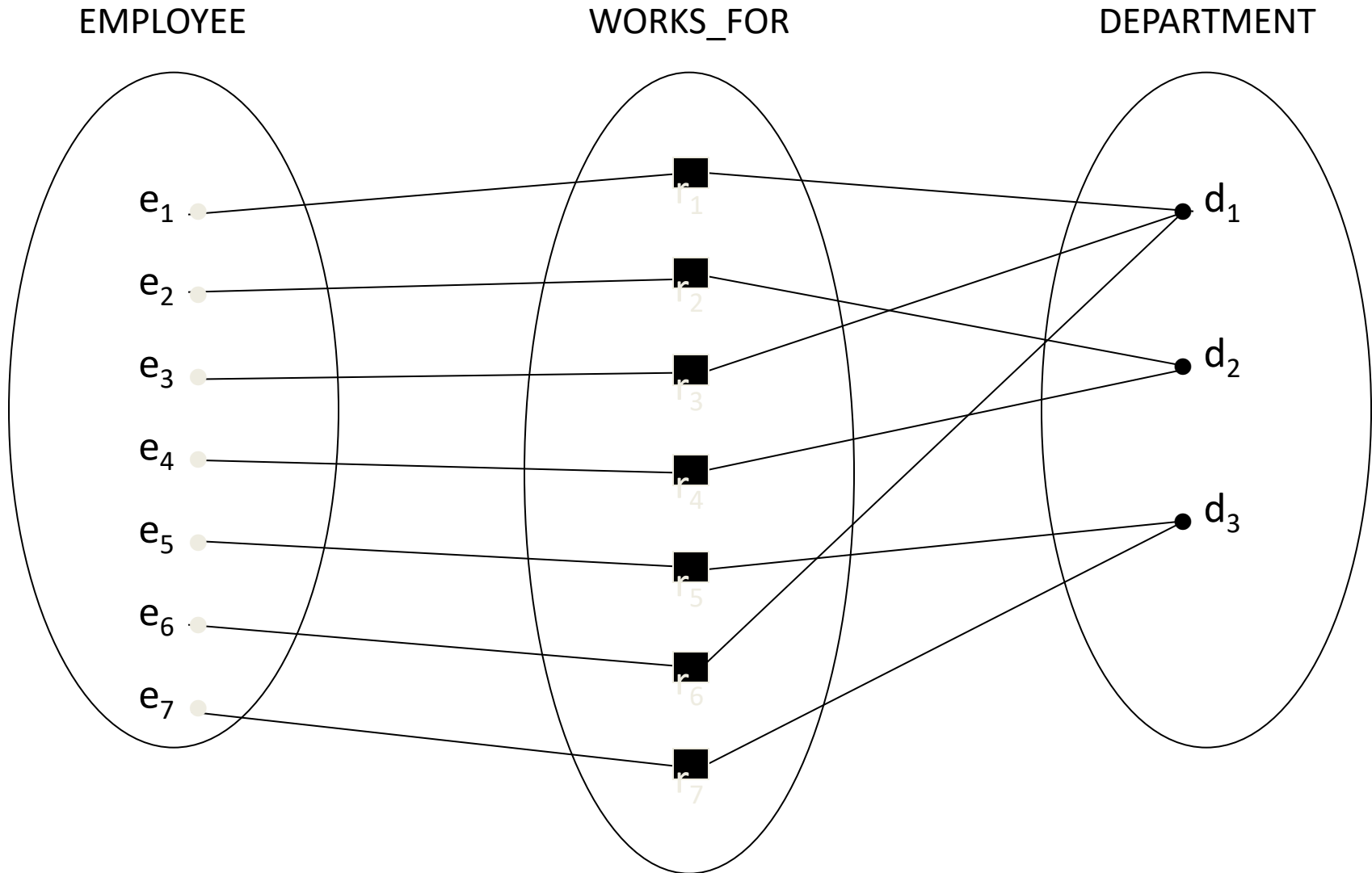


Binary

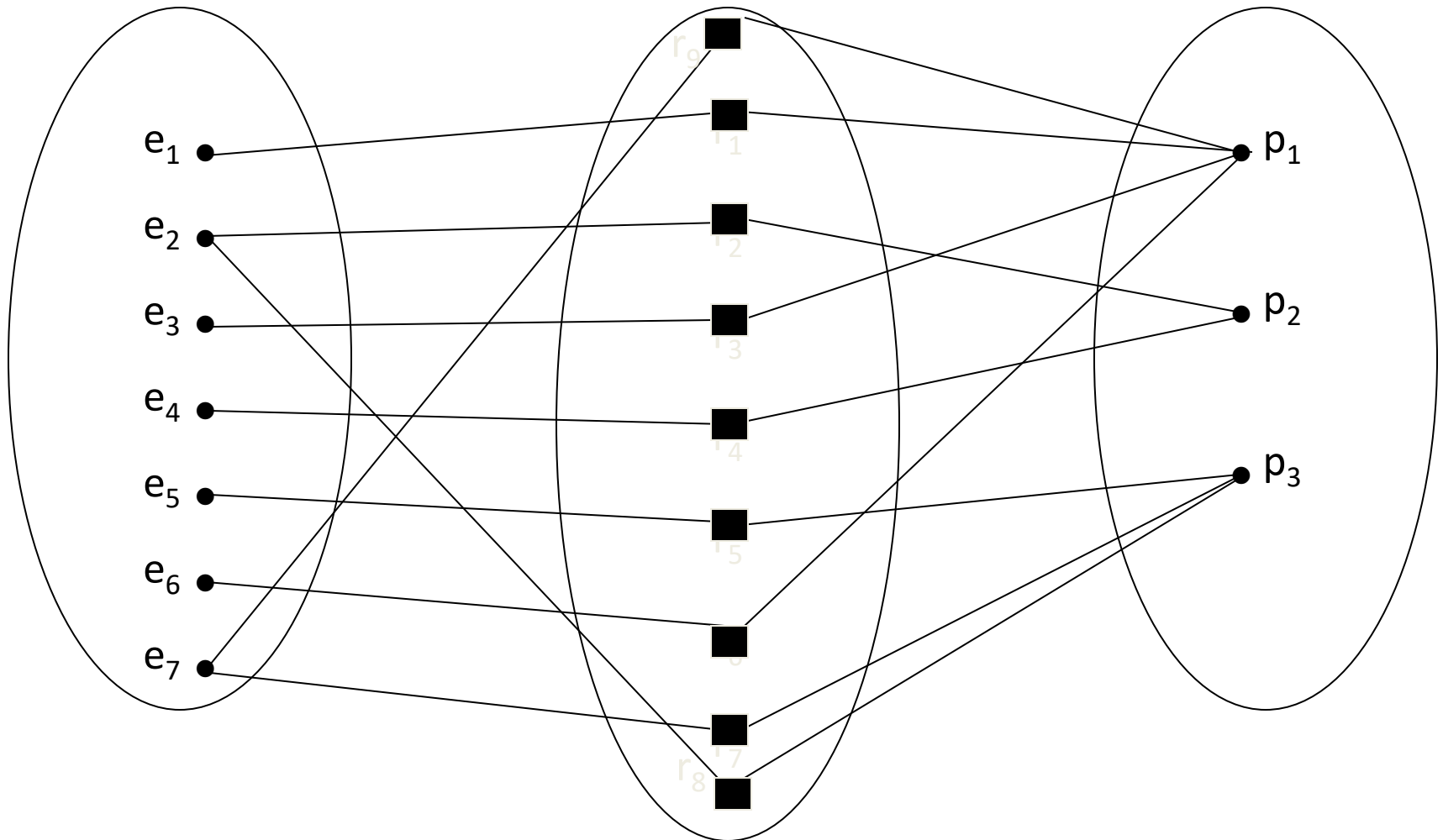


Ternary

Example relationship instances of the WORKS_FOR relationship between EMPLOYEE and DEPARTMENT



Example relationship instances of the WORKS_ON relationship between EMPLOYEE and PROJECT



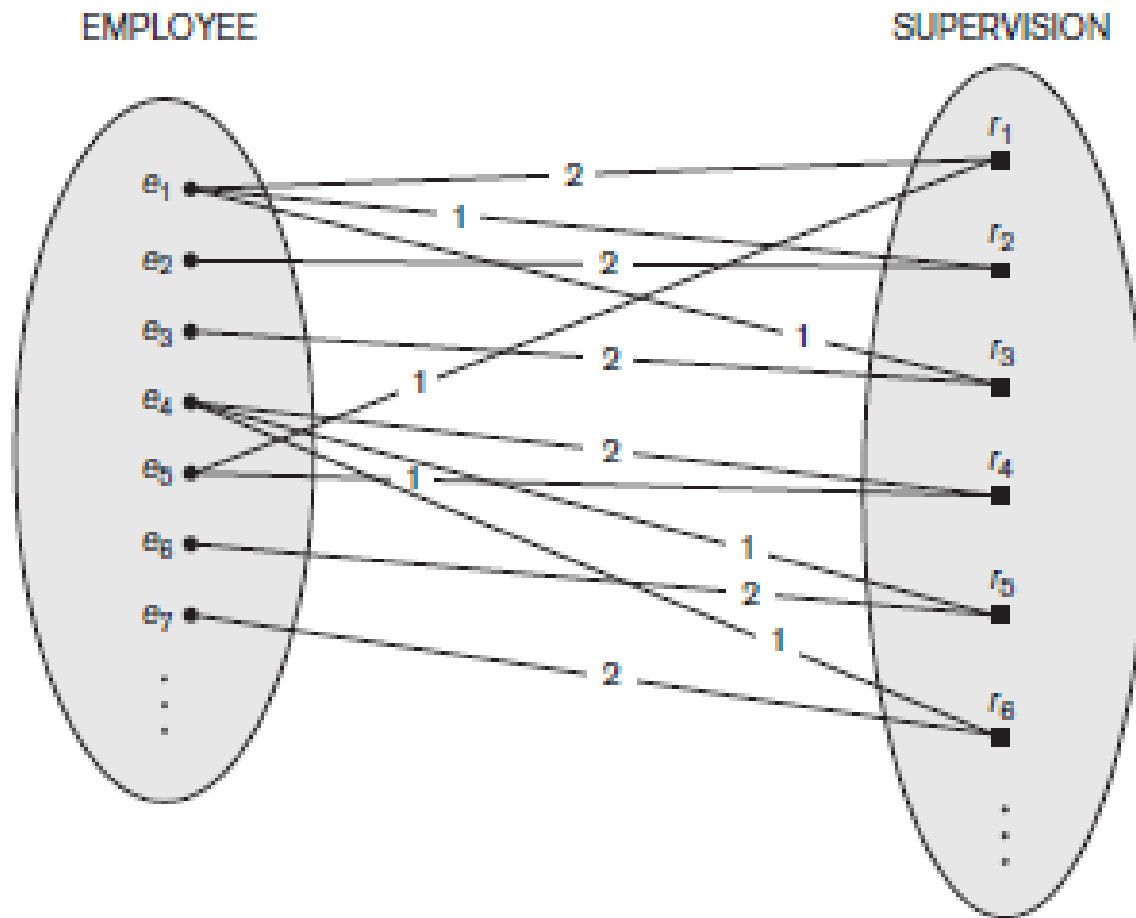
Relationships and Relationship Types (2)

- More than one relationship type can exist with the same participating entity types. For example, MANAGES and WORKS_FOR are distinct relationships between EMPLOYEE and DEPARTMENT, but with different meanings and different relationship instances.

Relationships and Relationship Types (3)

- We can also have a **recursive** relationship type.
- The *same* entity type participates more than once in a relationship type in *different roles*.
- For example, SUPERVISION relationships between EMPLOYEE (in role of supervisor or boss) and (another) EMPLOYEE (in role of subordinate or worker).
- In following figure, first role participation labeled with 1 and second role participation labeled with 2.
- In ER diagram, need to display **role names** to distinguish participations.
- The **role name** signifies the role that a participating entity from the entity type plays in each relationship instance, and it helps to explain what the relationship means

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



Constraints on Relationships

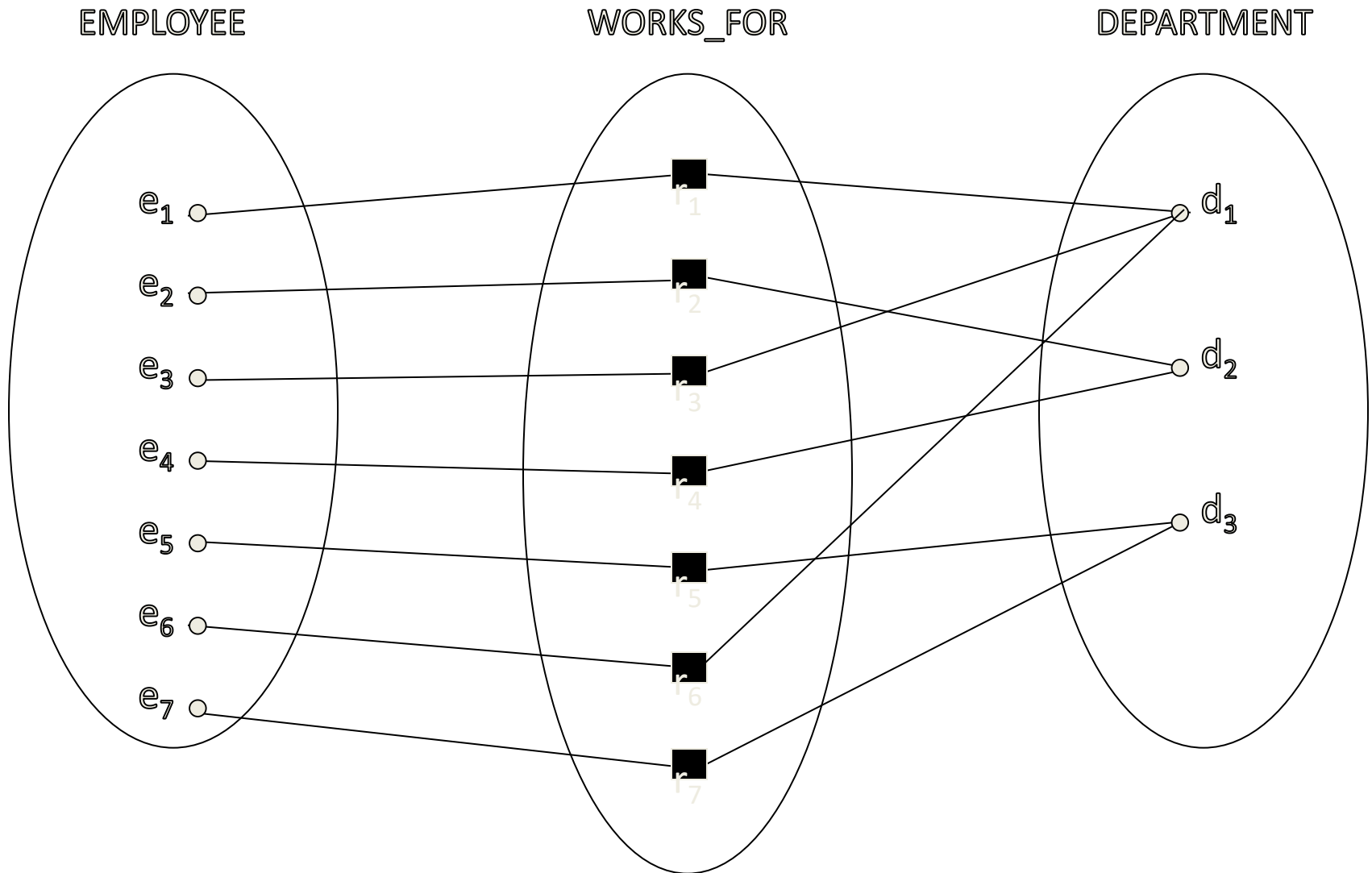
Constraints on Relationship Types

- The **cardinality ratio** for a binary relationship specifies the *maximum* number of relationship instances that an entity can participate in
 - (Also known as ratio constraints)
 - Maximum Cardinality
 - One-to-one (1:1)
 - One-to-many (1:N) or Many-to-one (N:1)
 - Many-to-many

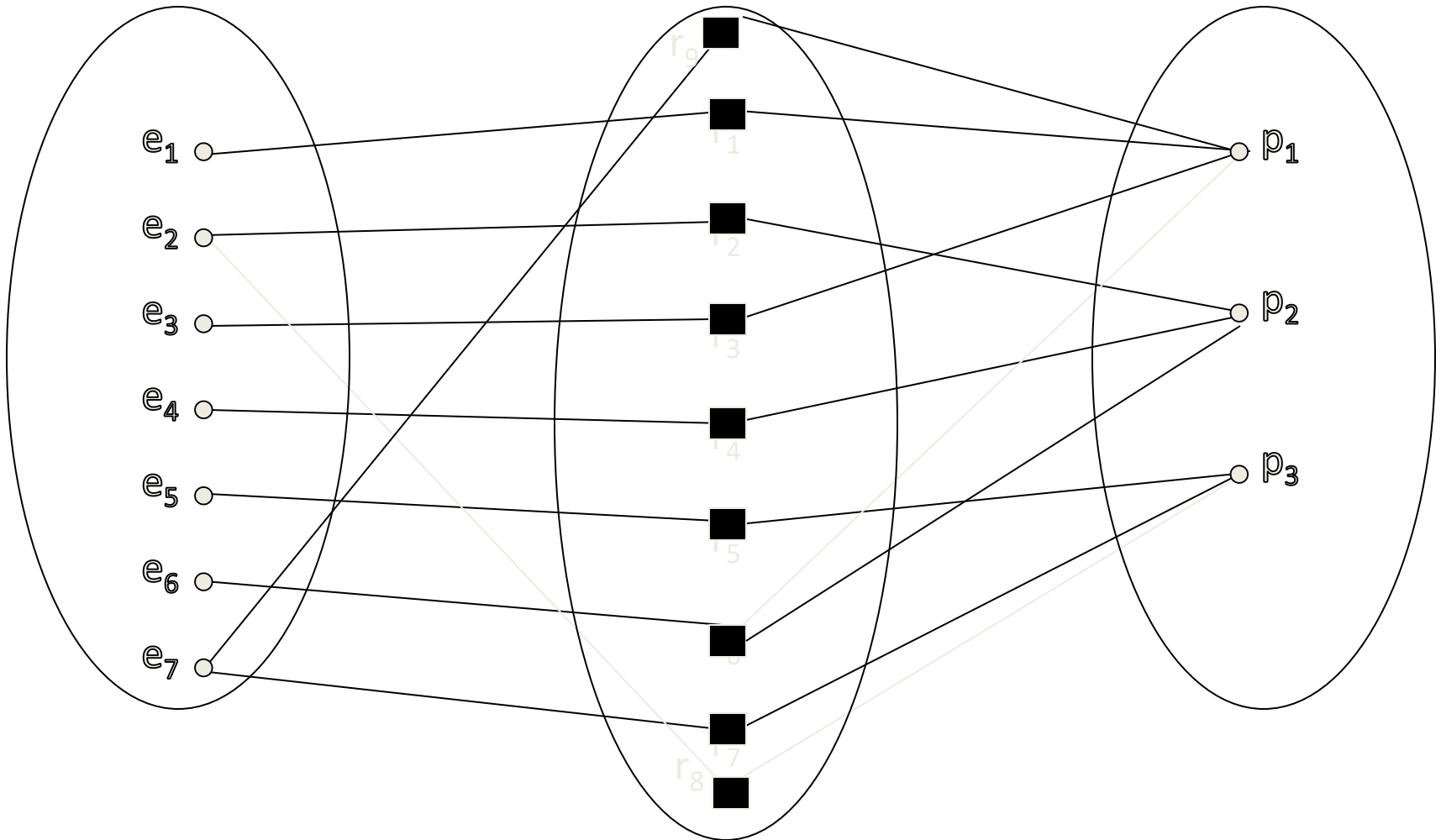
Minimum Cardinality (also called participation constraint or existence dependency constraints)

- The **participation constraint** 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 is sometimes called as the **minimum cardinality constraint**.
- There are two types of participation constraints—**total and partial**
 - zero (optional participation, not existence-dependent)
 - one or more (mandatory, existence-dependent)
- the cardinality ratio and participation constraints, taken together, as the **structural constraints** of a relationship type.

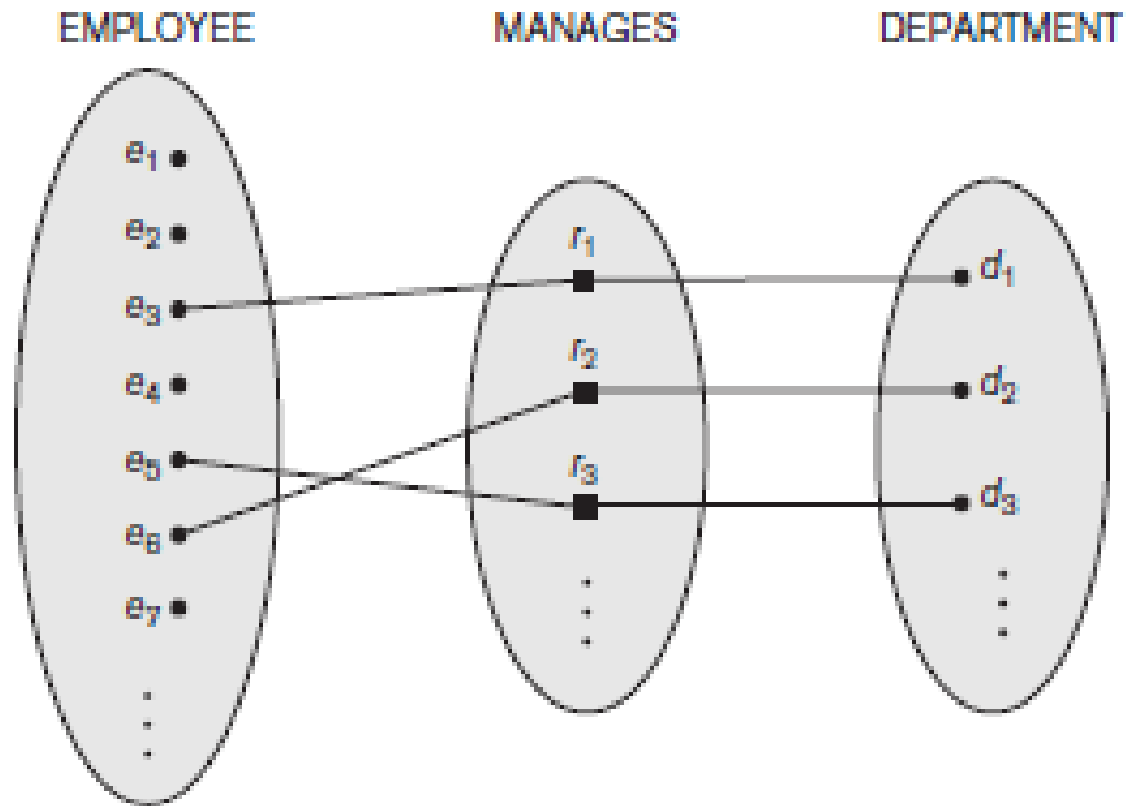
Many-to-one (N:1) RELATIONSHIP



Many-to-many (M:N) RELATIONSHIP



A 1:1 RELATIONSHIP




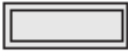






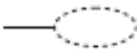



Weak Entity Types

- An entity that does not have a key attribute
- 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 entity type

Example:

Suppose that a DEPENDENT entity is identified by the dependent's first name and birthdate, *and* the specific EMPLOYEE that the dependent is related to. DEPENDENT is a weak entity type with EMPLOYEE as its identifying entity type via the identifying relationship type DEPENDENT_OF

SUMMARY OF ER-DIAGRAM NOTATION FOR ER SCHEMAS

Symbol	Meaning	F S n d
	Entity	F S n d
	Weak Entity	
	Relationship	
	Identifying 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	

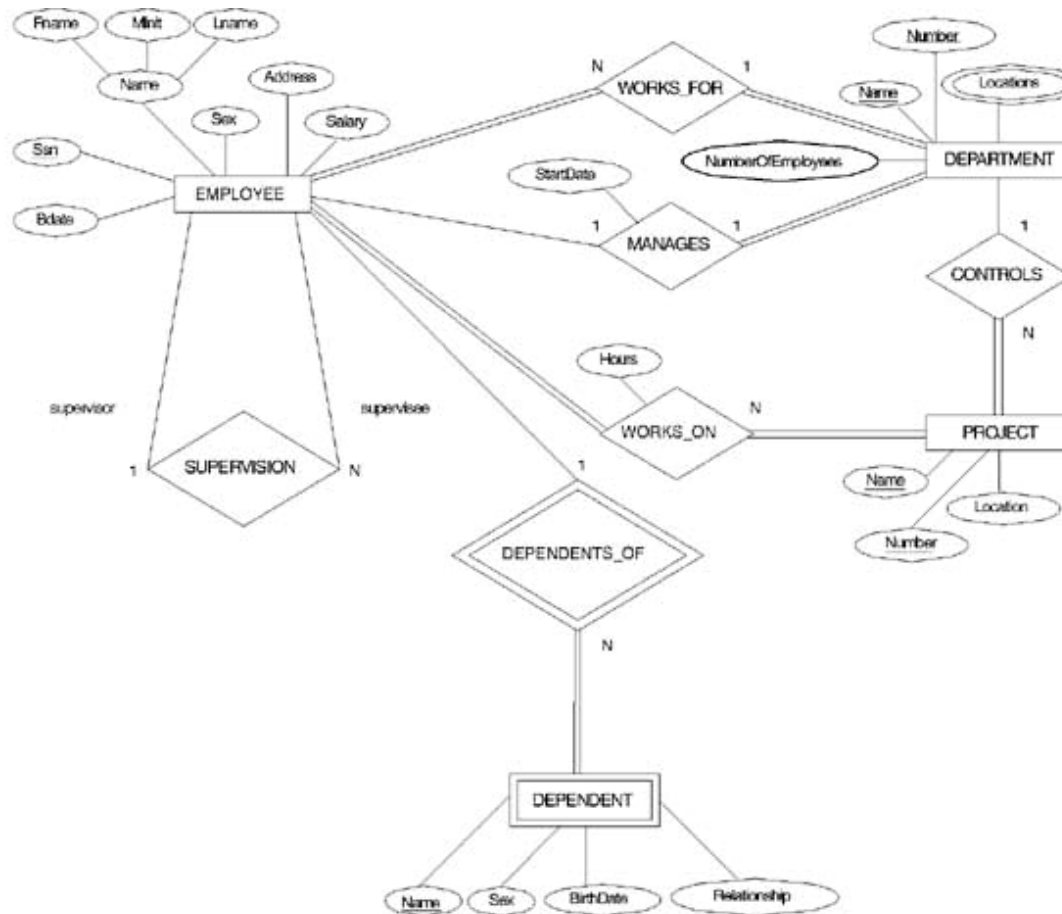
Example COMPANY Database

- Requirements of the Company (oversimplified for illustrative purposes)
 - 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.
 - Each department *controls* a number of PROJECTs. Each project has a name, number and is located at a single location.

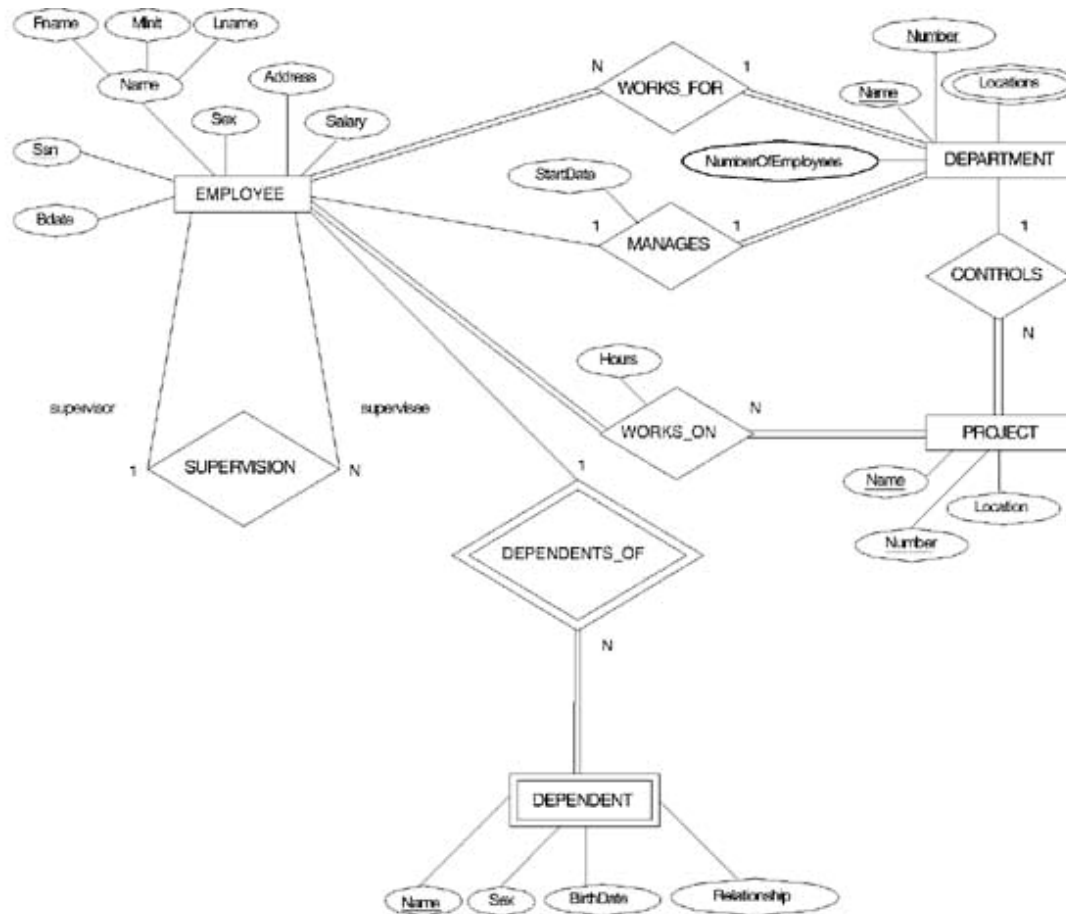
Example COMPANY Database (Cont.)

- We store each EMPLOYEE's social security number, address, salary, sex, and birthdate. Each employee *works for* one department but may *work on* several projects. We keep track of the number of hours per week that an employee currently works on each project. We also keep track of the *direct supervisor* of each employee.
- Each employee may *have* a number of DEPENDENTS. For each dependent, we keep track of their name, sex, birthdate, and relationship to employee.

ER DIAGRAM – Entity Types are: EMPLOYEE, DEPARTMENT, PROJECT, DEPENDENT

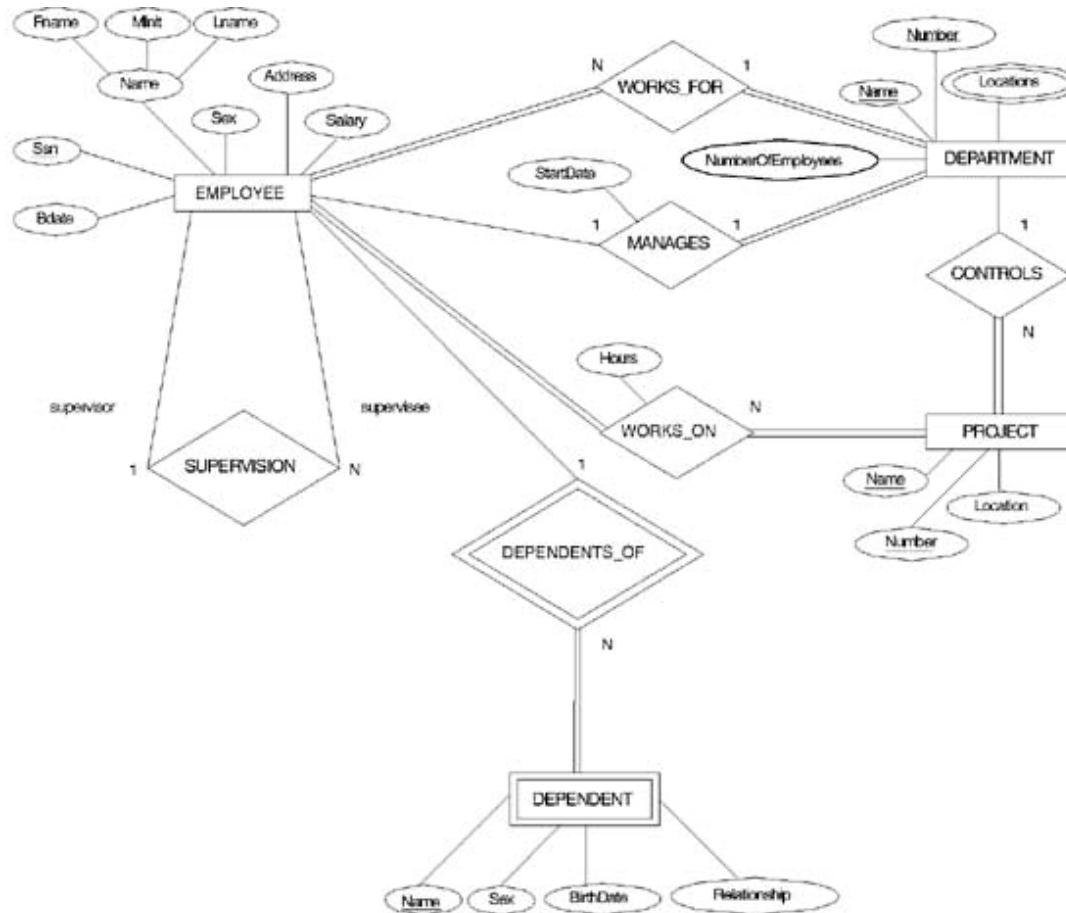


ER DIAGRAM – Relationship Types are: WORKS_FOR, MANAGES, WORKS_ON, CONTROLS, SUPERVISION, DEPENDENTS_OF



Weak Entity Type is: DEPENDENT

Identifying Relationship is: DEPENDENTS_OF



Recursive Relationship Type is: SUPERVISION (participation role names are shown)

