



Database Management Systems

ICT1212

Data Modeling Using Entity-Relationship Model (Conceptual Data Modeling)

Department of ICT
Faculty of Technology
University of Ruhuna

Lecture 4

What we discuss Today.....

- An Example Database Application
- Entity Types, Entity Sets, Attributes, and Keys
- Relationships, Relationship Types, Roles, and Structural Constraints
- Weak Entity Types
- Refining the ER Design for the COMPANY Database
- ER Diagrams, Naming Conventions, and Design Issues

Example COMPANY Database

Requirements of the Company (oversimplified for illustrative purposes)

- The company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when that employee began managing the department. A department may have several locations.
- A department controls a number of projects, each of which has a unique name, a unique number, and a single location.
- We store each employee's name, identity card number, address, salary, sex(gender) and birth date. An employee is assigned to one department but may work on several projects, which are not necessarily controlled by the same department. We keep track of the number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee.
- We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent's first name, sex(gender), birth date, and relationship to the employee.

E-R Model Concepts

- Entities

Entities are specific objects or things in the mini-world with an independent existence that are represented in the database

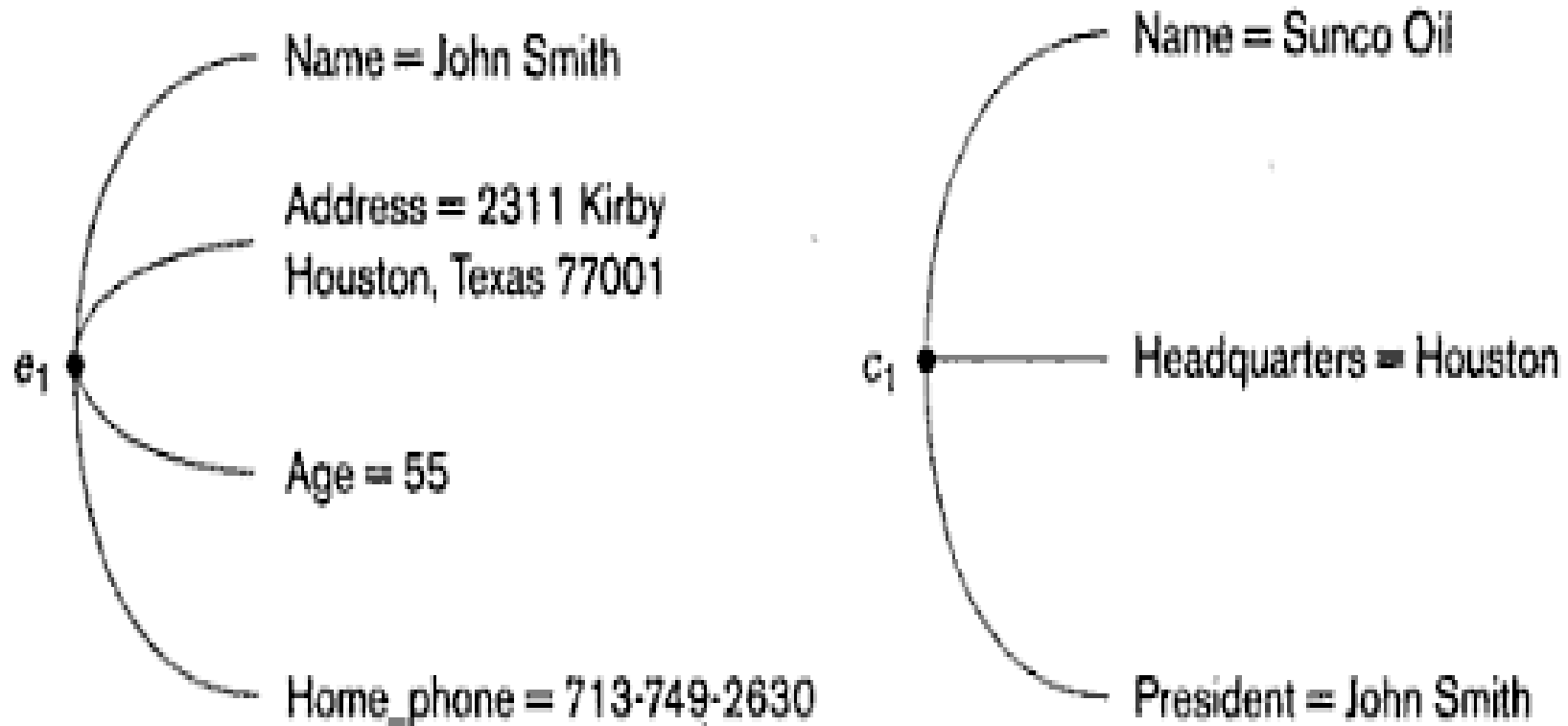
Ex :- student, car, a company, a job, or a university course

- Attributes

Each entity has **attributes** - the particular properties that describe it

Ex :- an employee entity may be described by the employee's name, age, address, salary, and job etc

Entity & attributes



Types of Attributes

- Simple(Atomic)

- Attributes that are not divisible are called simple or atomic attributes

Ex :- Sex(Gender)

- Composite

- The attribute may be composed of several components

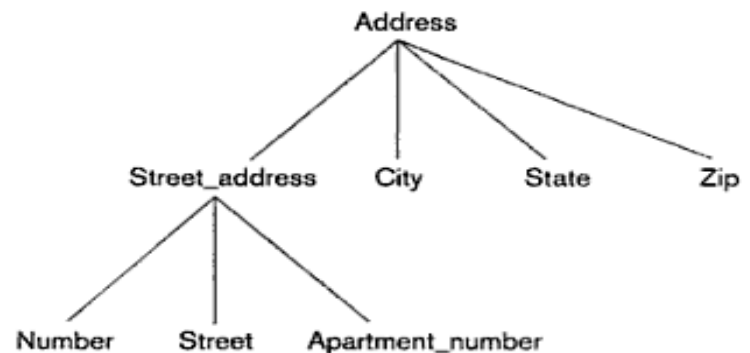
Ex :- Address

(Apartment no, House no, Street, City, State, Zip Code, Country)

Name

(First Name, Middle Name, Last Name)

- Composition may form a hierarchy where some components are themselves composite.



Types of Attributes

- Single Valued

- Have a single value for a particular entity

Ex :- National Identity Card No

- Multi Valued

- An attribute can have a set of values for the same entity

Ex :- Color of a CAR

Degrees of a STUDENT.

Types of Attributes

- **Stored Attributes**

- An attribute's value cannot be determined from the values of other attributes
- Attribute's value for an entity must be kept as part of the entity

Ex :- Date of Birth

- **Derived Attributes**

- An attribute's value can be determined from the values of other attribute/s
- Attribute is derivable

Ex :- Age

Null Values

- In some cases, a particular entity may not have an applicable value for an attribute

Ex :- Apartment Number attribute

- applies only to addresses that are in apartment buildings and not to other types of residences(single-family homes)

College Degree

- Only applies to people who are having college degrees

Height of a person

- When we do not know the value(missing)

Home Phone/ Land Phone

- When we do not know whether the value exists or not

Null

1. Not Applicable

2. Unknown

- It is known that the attribute value exists but is missing
- It is not known whether the attribute value exists

Complex Attributes

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

Ex :-

```
contact_details(  
    { phone_no },  
    { address  
        (  
            street_address(no, street, apartment_no),  
            city, state, zip  
        )  
    }  
)
```

Entity Types and Entity Sets

- Entity Type (Schema or Intension)
 - An entity type defines a *collection* (or set) of entities that have the same attributes
 - Each entity type in the database is described by its name and attributes
- Entity Set (Extension)
 - The collection of all entities of a particular entity type in the database at any point in time
 - The entity set is usually referred to using the same name as the entity type.

Entity Types and Entity Sets

Entity Type Name:

EMPLOYEE

COMPANY

Name, Age, Salary

Name, Headquarters, President

Entity Set:
(Extension)

e_1 •

(John Smith, 55, 80k)

e_2 •

(Fred Brown, 40, 30K)

e_3 •

(Judy Clark, 25, 20K)

•
•
•

c_1 •

(Sunco Oil, Houston, John Smith)

c_2 •

(Fast Computer, Dallas, Bob King)

•
•
•

Key Attributes

- An entity type usually has one or more attributes whose values are distinct for each individual entity in the entity set
- Its values can be used to identify each entity uniquely

Example ???

- Sometimes several attributes together form a key, meaning that the *combination* of the attribute values must be distinct for each entity

Example ???


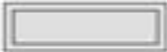










Value Sets of Attributes

- Also called as Domain of Values
- Specifies the set of values that may be assigned to that attribute for each individual entity

Ex :- Age (16-70 and only Integers)

- The value set provides all possible values
- Represent the data from the current state of the miniworld

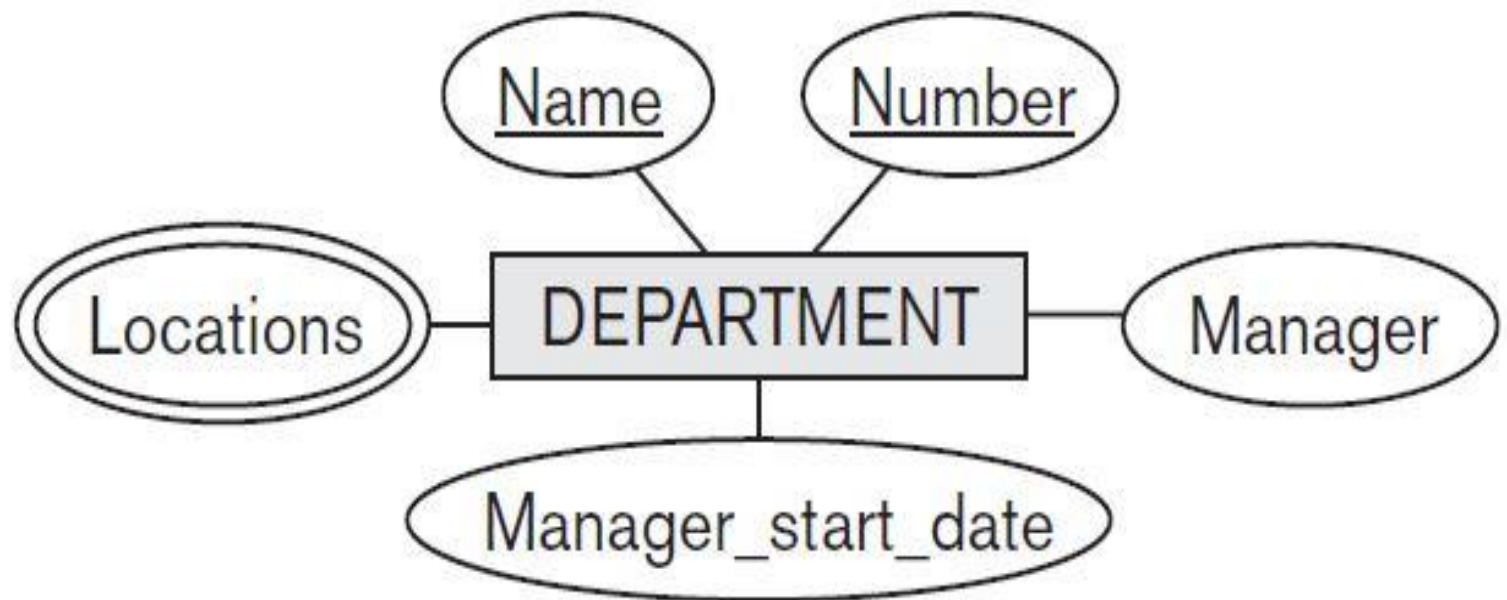
The Conventions for ER Diagrams

Symbol	Meaning
	Entity
	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

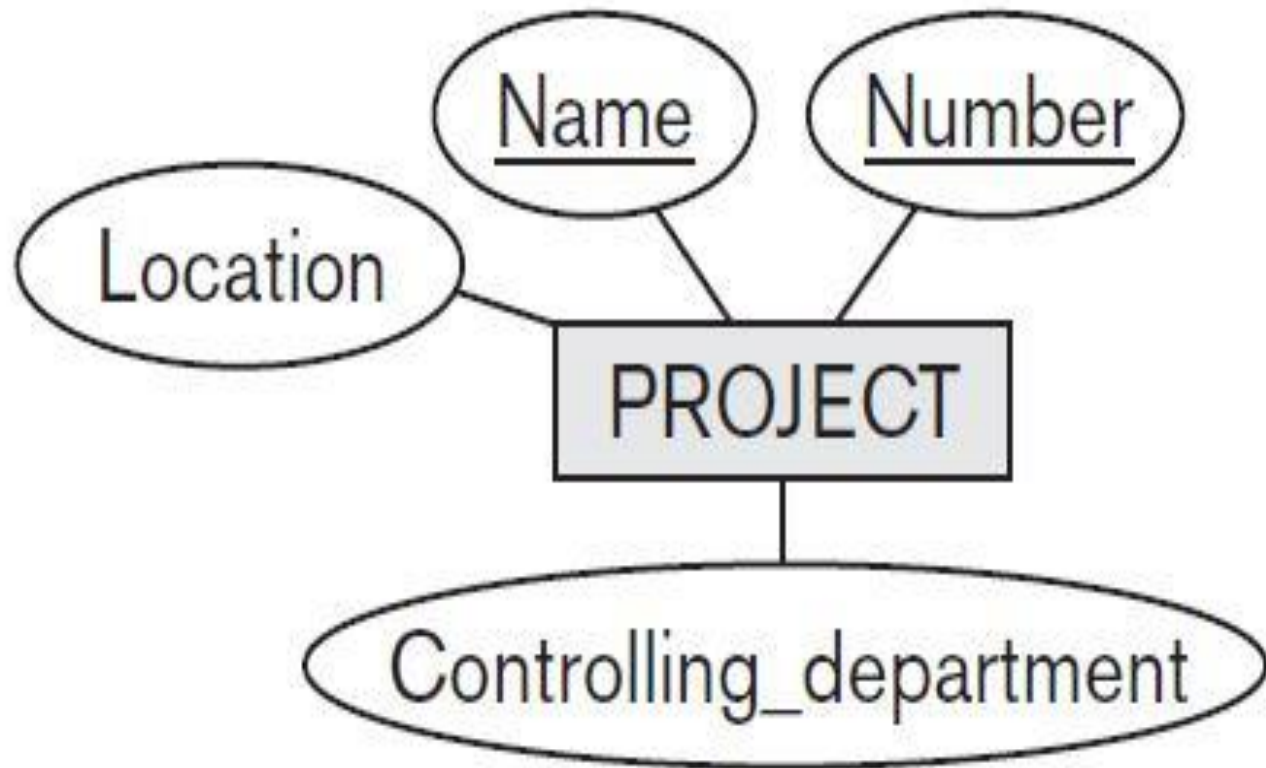
Let's Try IT

- Identify the Entities
- Identify the Attributes of Entities
- Use Proper notation and draw a **Preliminary Design** of Entity types for the COMPANY database

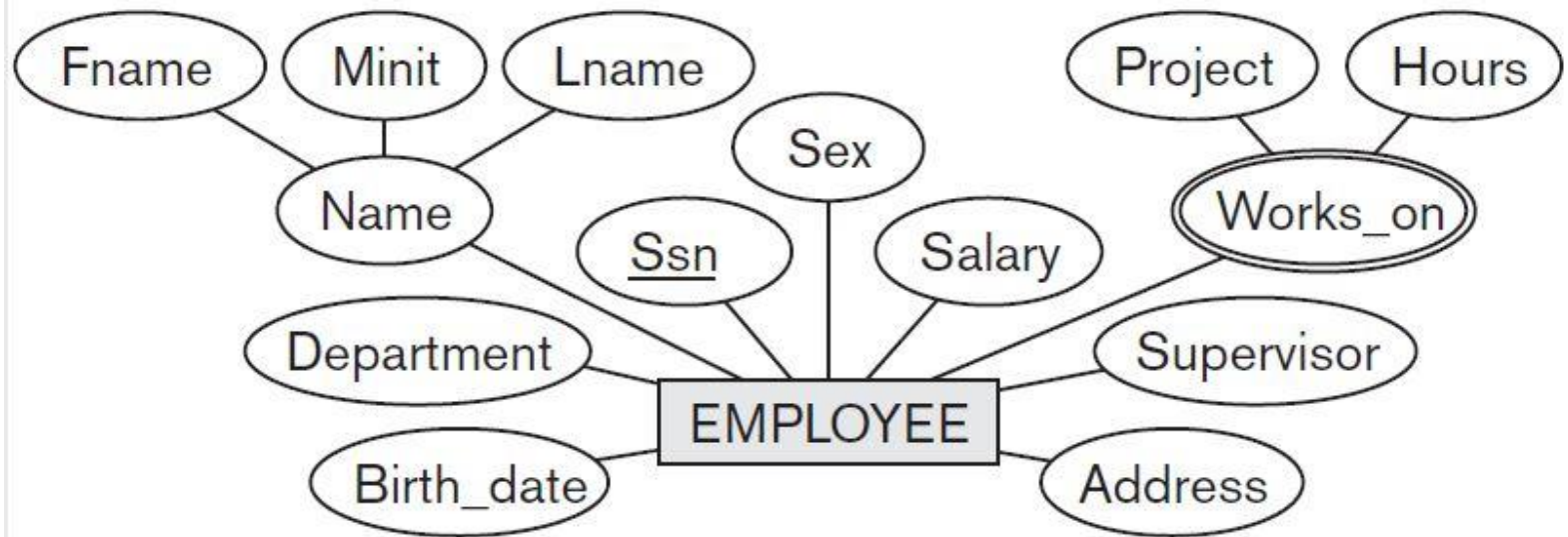
Preliminary Design - Department



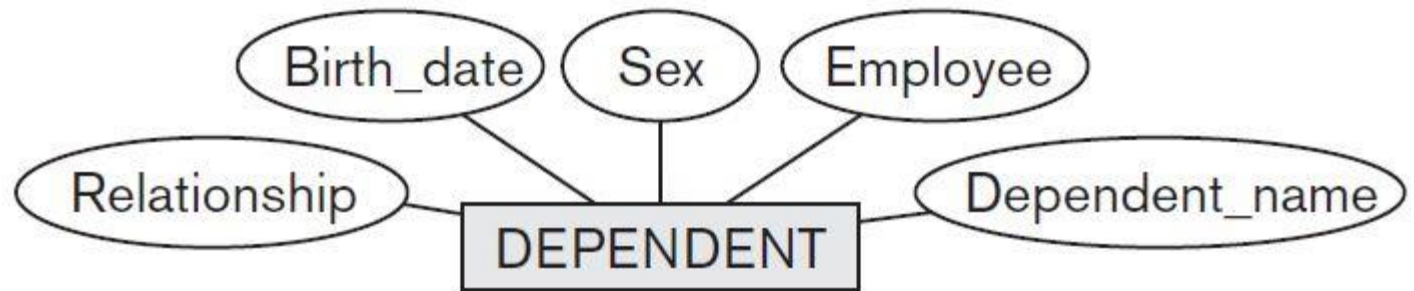
Preliminary Design - Project



Preliminary Design - Employee



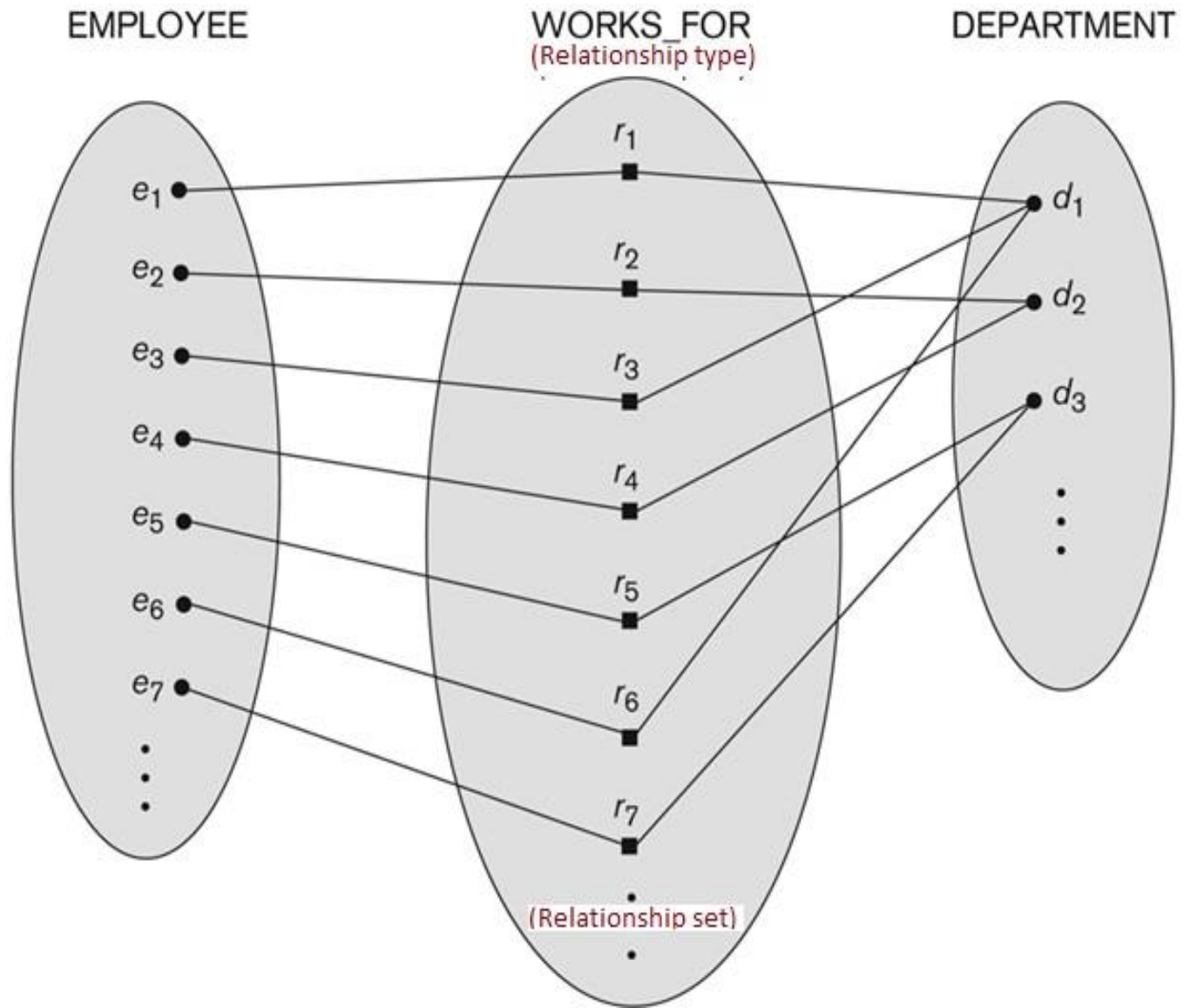
Preliminary Design - Dependent



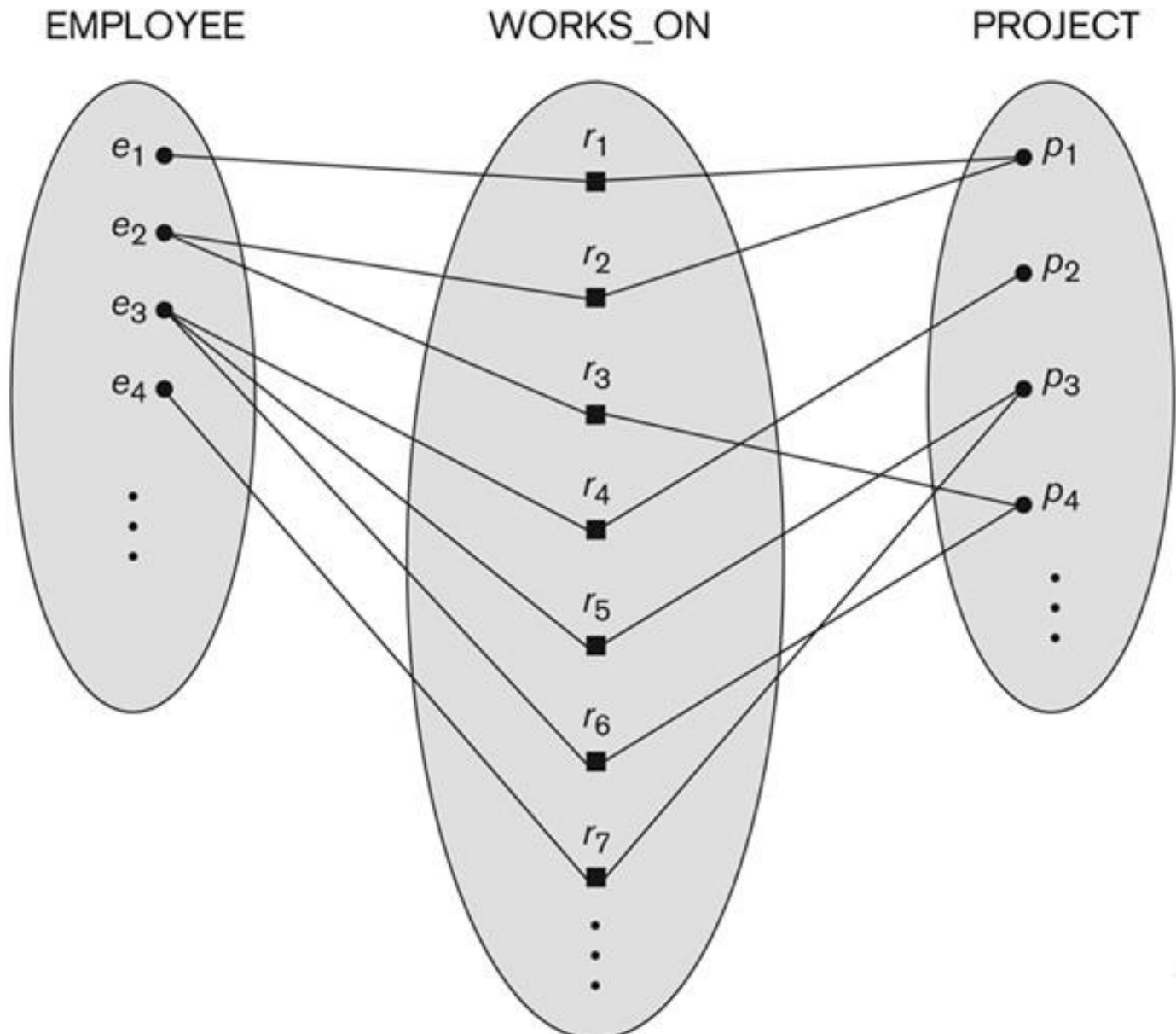
Relationships and Relationship Types

- 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 MANAGES and WORKS_ON are binary relationships.

Relationship - WORKS_FOR

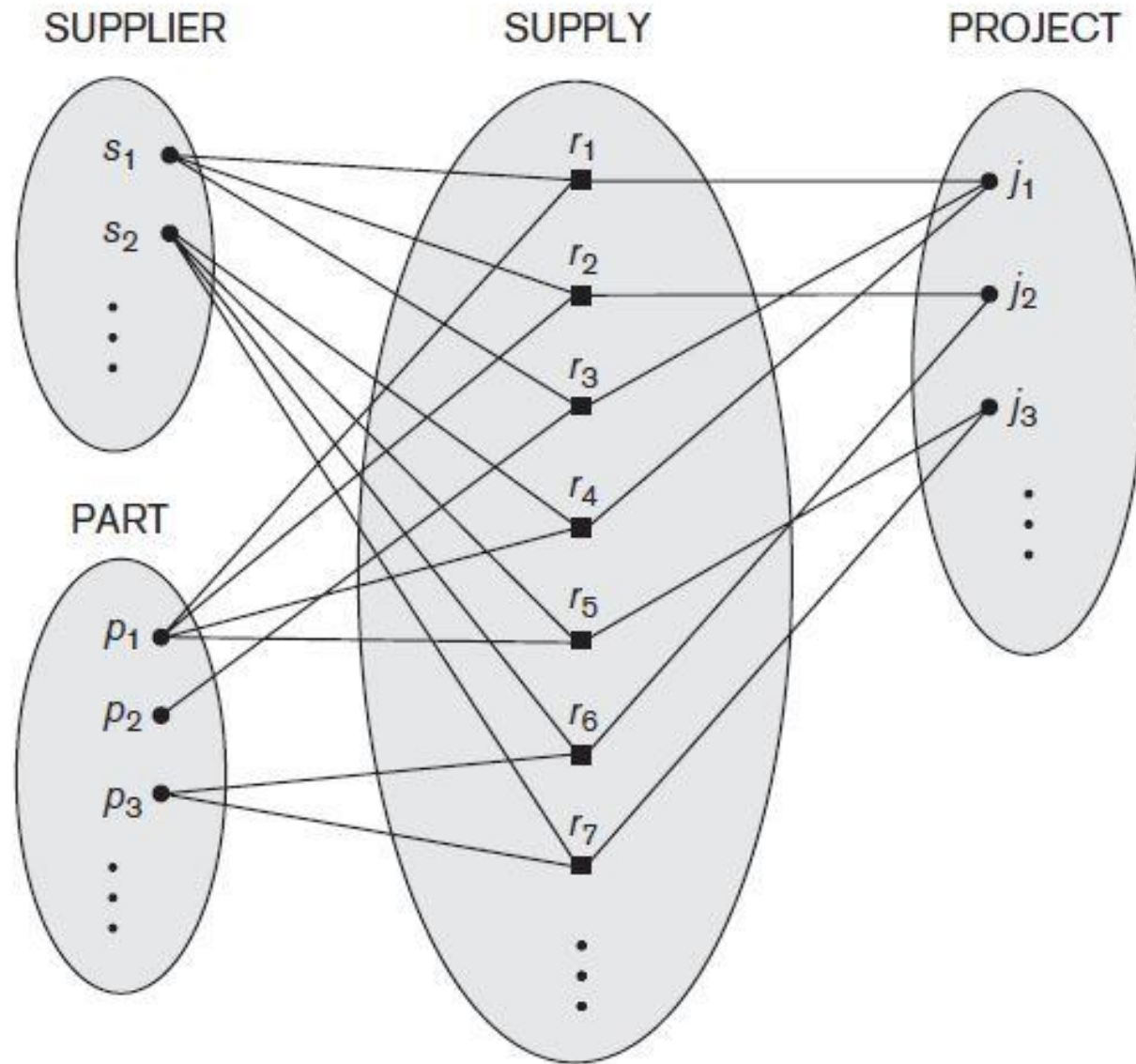


Relationship - WORKS_ON



An M:N relationship WORKS_ON

Relationship - SUPPLY



Let's Try

- By examining the requirements identify how many relationships exists in the given example
- List them down with their participating entity types
- Identify the degree of above listed relationships

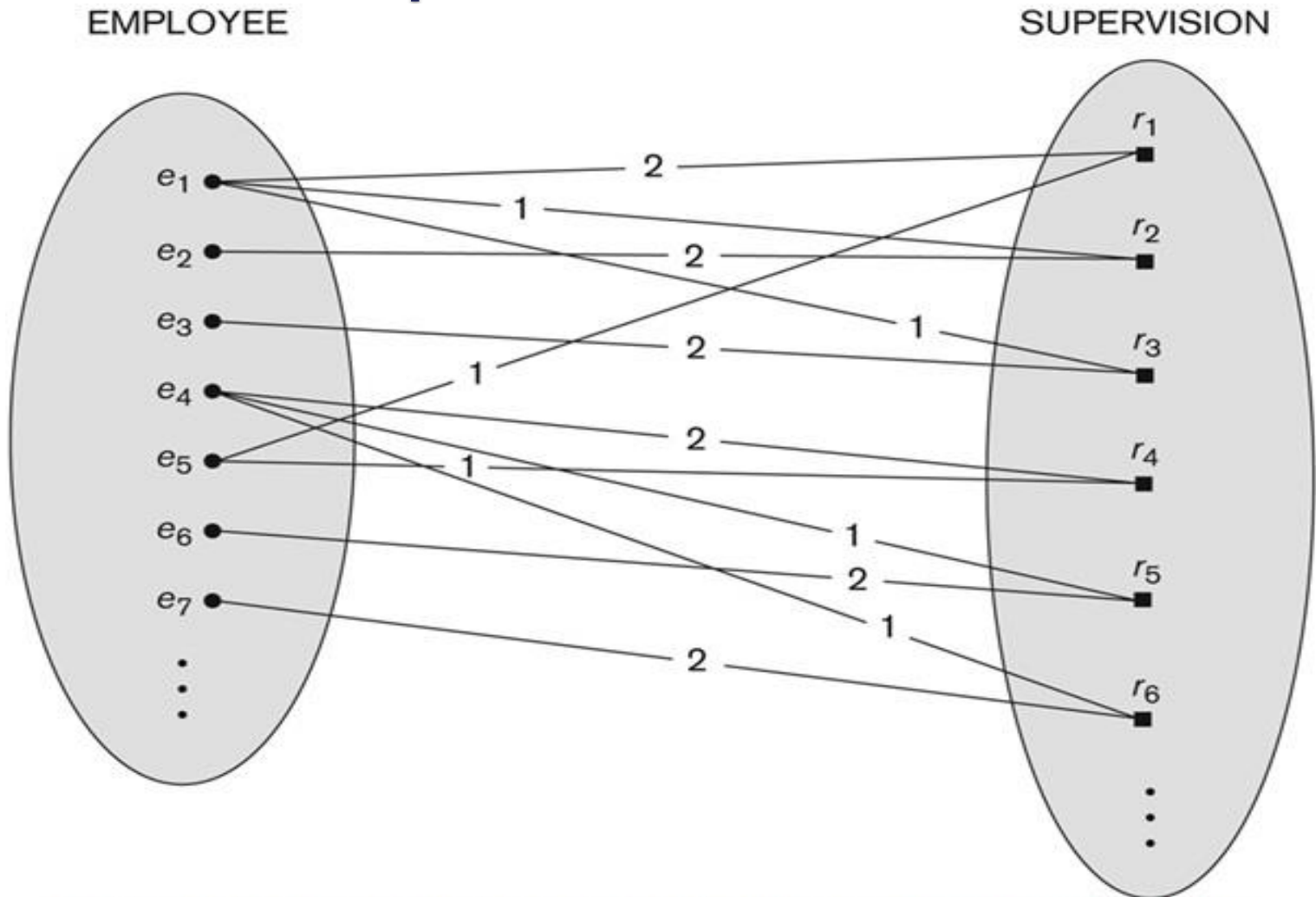
Let's Try

- By examining the requirements, six relationships are there
- 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)
- All are binary relationships(degree 2)

Relationships and Relationship Types

- It is sometimes convenient to think of a binary relationship type in terms of attributes
 - For example WORKS_FOR and Department attribute of EMPLOYEE entity
- 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

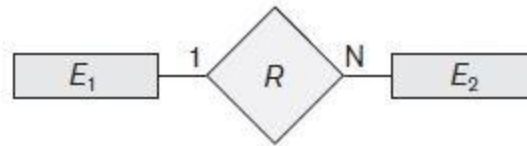
Role Names and Recursive Relationships



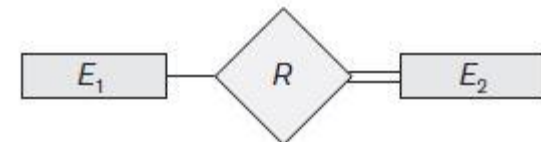
A recursive relation-ship SUPERVISION between EMPLOYEE in the **supervisor role(1)** and EMPLOYEE in the **subordinate role(2)**

Constraints of Binary Relationship Types

- Cardinality Ratios for Binary Relationships (Maximum Cardinality)
 - The possible cardinality ratios for binary relationship types are 1:1, 1:N, N:1, and M:N



- Participation Constraints and Existence Dependencies (Minimum Cardinality)
 - Two types of participation constraints
 - Total
 - Partial
 - Total participation is also called **existence dependency**



Let's Try

- By examining the requirements identify the cardinality ratios and participation constrains

Let's Try

- **MANAGES**
 - a 1:1 relationship type between EMPLOYEE and DEPARTMENT.
 - EMPLOYEE participation is partial. DEPARTMENT participation is not clear from the requirements. We question the users, who say that a department must have a manager at all times, which implies total participation. The attribute Start_date is assigned to this relationship type.
- **WORKS_FOR**
 - a 1:N relationship type between DEPARTMENT and EMPLOYEE. Both participations are total.
- **CONTROLS**
 - a 1:N relationship type between DEPARTMENT and PROJECT.
 - The participation of PROJECT is total, whereas that of DEPARTMENT is determined to be partial, after consultation with the users indicates that some departments may control no projects.

Let's Try

- **SUPERVISION**

- a 1:N relationship type between EMPLOYEE (in the supervisor role) and EMPLOYEE (in the supervisee role).
- Both participations are determined to be partial, after the users indicate that not every employee is a supervisor and not every employee has a supervisor.

- **WORKS_ON**

- determined to be an M:N relationship type with attribute Hours, after the users indicate that a project can have several employees working on it. Both participations are determined to be total.

- **DEPENDENTS_OF**

- a 1:N relationship type between EMPLOYEE and DEPENDENT
- which is also the identifying relationship for the weak entity type DEPENDENT. The participation of EMPLOYEE is partial, whereas that of DEPENDENT is total.

Attributes of Relationship Types

- Relationship types can also have attributes, similar to those of entity types
 - Attributes of 1:1 or 1:N relationship types can be migrated to one of the participating entity types
 - For M:N some attributes
 - may be determined by the *combination of participating entities* in a relationship instance
 - not by any single entity
 - Such attributes *must be specified as relationship attributes*
 - Examples
 - 1 : 1
 - 1 : N
 - M : N

Weak Entity Types

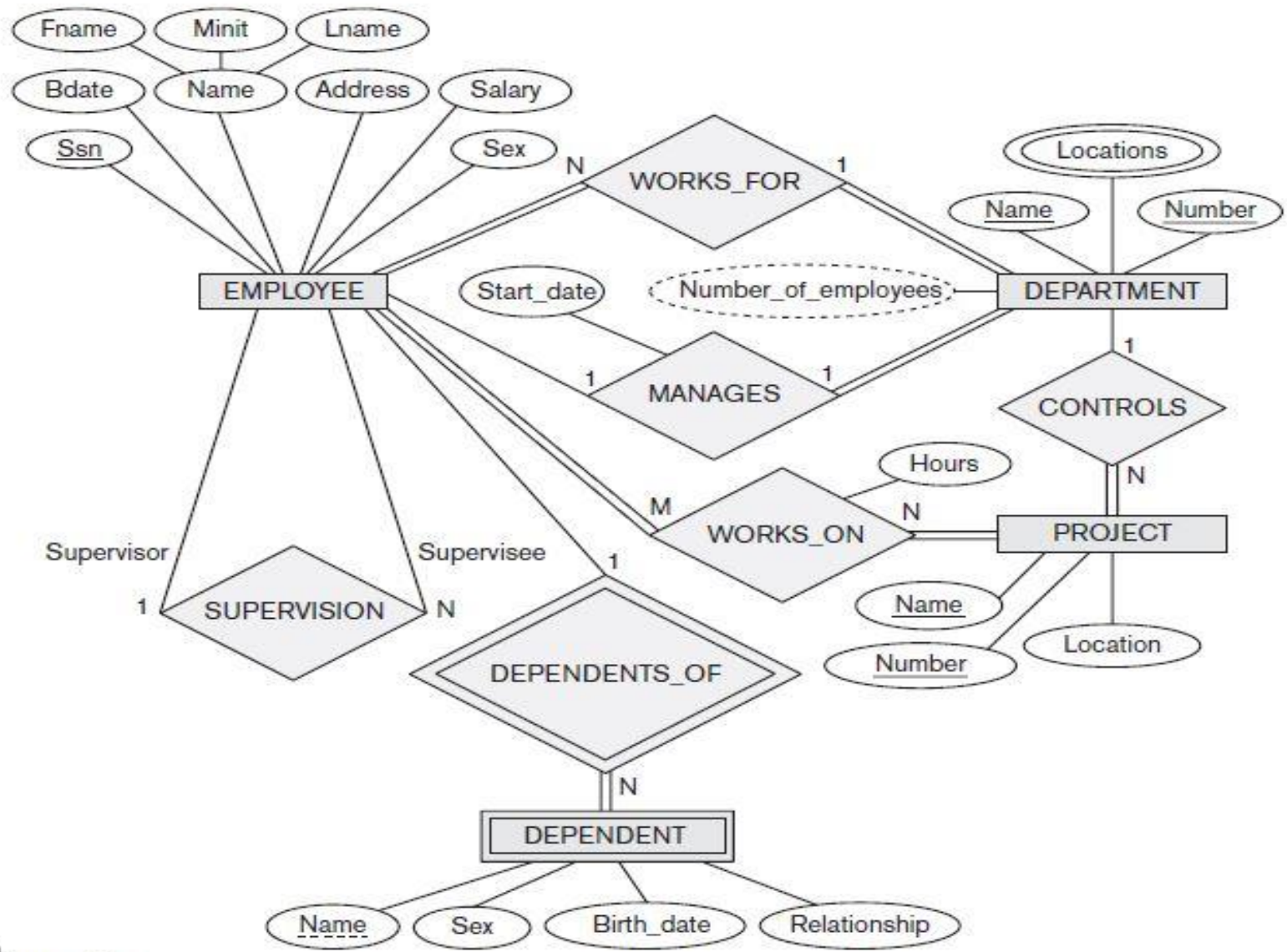
- An entity that does not have a key attribute
- Entities belonging to a weak entity type are identified by being related to specific entities from another entity type in combination with some of their attribute values.
- Entities are identified by the combination of
 - A partial key of the weak entity type
 - The particular entity they are related to in the identifying entity type

Ex:- 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**

Let's Try

- Identify the Weak entity type and the related relationship for that Entity
- Now Refine the ER Design for the COMPANY Database

Refined ER Design for the COMPANY Database



Reference

- *Chapter 7- Fundamentals of Database Systems*
(6th Edition) By Ramez Elmasri & Shamkant B. Navathe

Questions ???





Thank You