Chapter 3

Data Modeling Using the Entity-Relationship (ER) Model

Chapter Outline

- Example Database Application (COMPANY)
- 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
- ER Diagram for COMPANY Schema
- Alternative Notations – UML class diagrams, others

Outline of Database Design

- **Requirements Collection and Analysis**: purpose is to produce a description of the users' requirements.
- **Conceptual Design**: purpose is to produce a *conceptual schema* for the database, including detailed descriptions of *entity types*, *relationship types*, and *constraints*.
- **Implementation**: purpose is to transform the conceptual schema into a *representational/implementational* model supported by whatever DBMS is to be used.
- **Physical Design**: purpose is to decide upon the internal storage structures, access paths (indexes), etc., that will be used in realizing the representational model produced in previous phase.
- In addition with these, application programs are designed and implemented as database transaction corresponding to the high level transaction specification

ER Model Concepts

- Entities and Attributes
  - The basic object the ER model represents is an ENTITY
  - 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, …

Cont..

- Types of Attributes:
  - Simple (Atomic) Vs. Composite
  - Single Valued Vs. Multi Valued
  - Stored Vs. Derived
  - Null Values

Cont..

- Simple Vs. Composite Attribute:
  - Simple

    Attribute that are not divisible are called simple or atomic attribute.

    For example, Age, 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 and some components are themselves composite.

single Valued Vs. Multi Valued

 Single Valued

An attribute having only one value.

For example, Age, Date of birth, Sex, SSN

 Multi-valued

An entity may have multiple values for that attribute.

For example, Color of a CAR or Previous Degrees of a STUDENT.

Denoted as {Color} or {PreviousDegrees}.

Phone number of an employee.

Stored Vs. Derived

- In some cases two are more attributes values are related
- For example the age and date of birth of a person.
- For a particular person entity, the value of age can be determined from the current (todays) date.

The Age attribute is hence called a **derived attribute** and is said to be derivable from the Birthdate attribute , which is called stored attribute.

Cont..

- Null Values

In some cases a particular entity may not have an applicable value for an attribute. For example, a college degree attribute applies only to persons with college degrees. For such situation, a special value called **null** is created.

Cont..

- Entity types : A entity type defines a set of entities that have the same attribute
- Entity sets : A collection of all entities of a particular entity type in the database at any point in time is called an entity set.

ENTITY SET corresponding to the ENTITY TYPE CAR

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

car_1
((ABC 123, TEXAS), TK629, Ford Mustang, convertible, 1999, (red, black))
car_2
((ABC 123, NEW YORK), WP9872, Nissan 300ZX, 2-door, 2002, (blue))
car_3
((VSY 720, TEXAS), TD729, Buick LeSabre, 4-door, 2003, (white, blue))

.

.

.

Entity Types and Key Attributes

- 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:

  Employee is identified with Employee Id and Adhar No

Relationships and Relationship Types (1)

- An attribute of one entity type refers to another entity type, some relationship exists
- 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.

Relationships of Higher Degree

- The degree of a relationship type is the number of participating entity types.
- 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

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

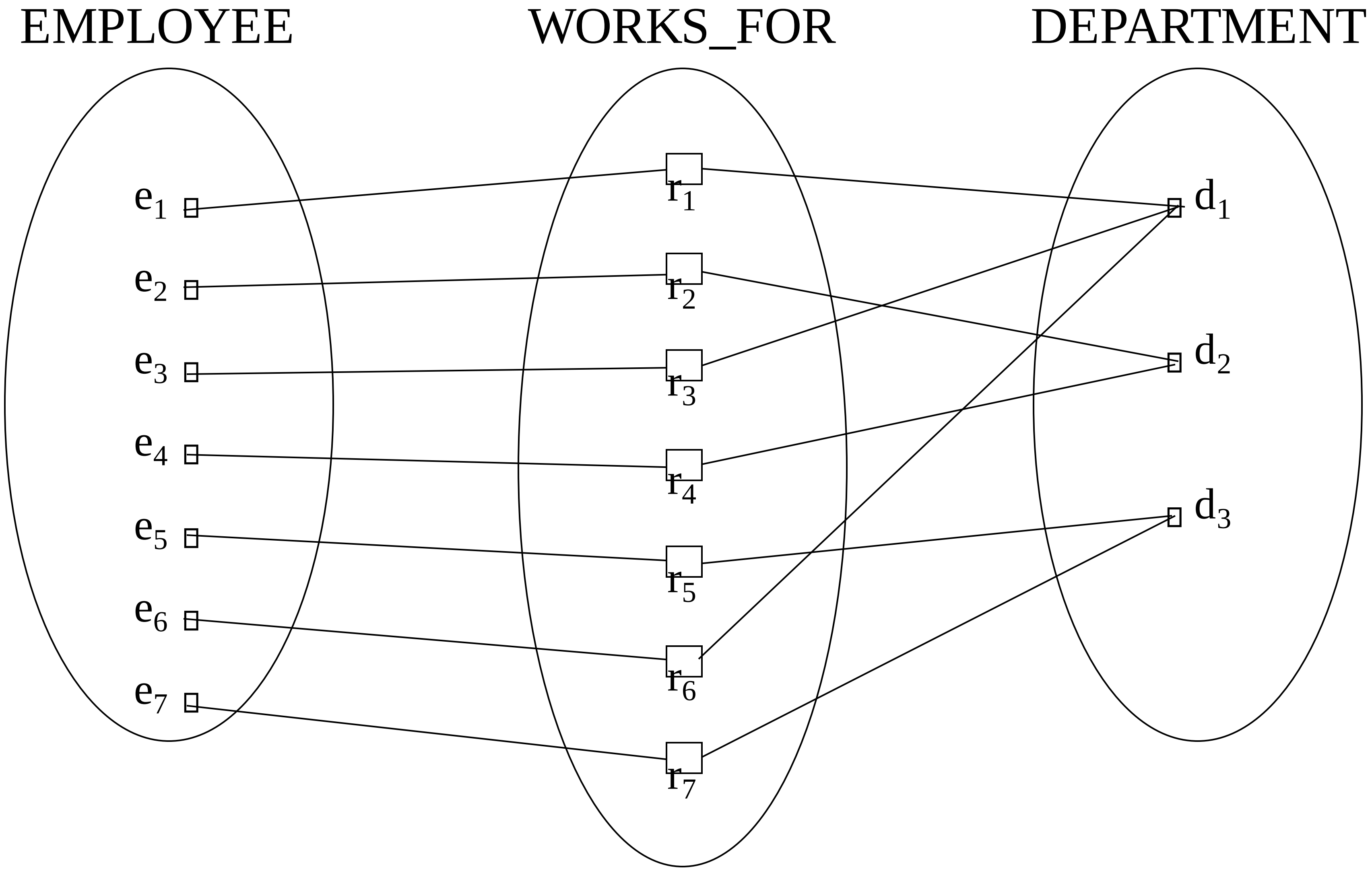
Example:

Suppose that a DEPENDENT entity is identified by the dependent's first name and birhtdate, *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

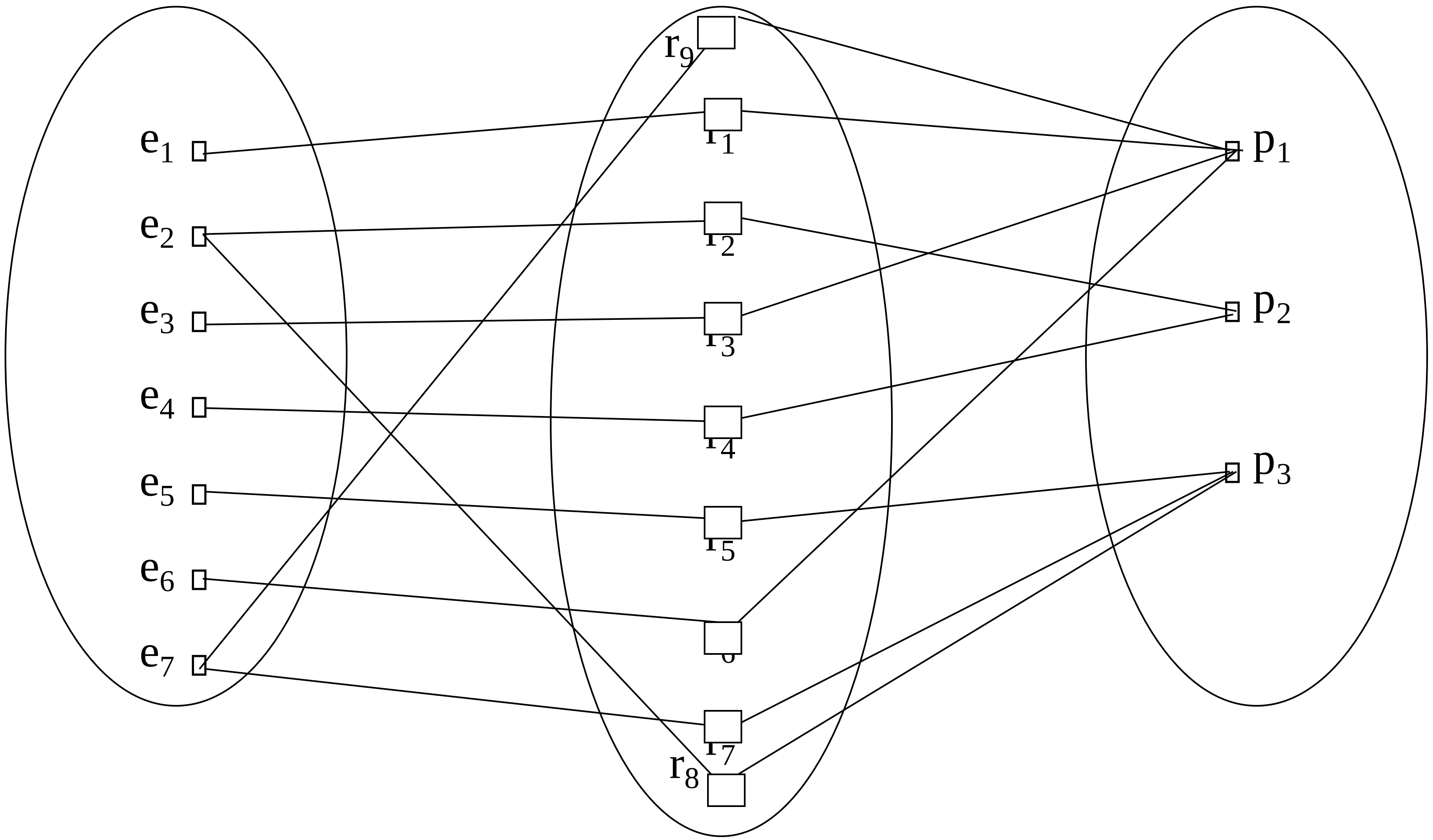
Constraints on Relationships

- Constraints on Relationship Types
  - (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)
    - zero (optional participation, not existence-dependent)
    - one or more (mandatory, existence-dependent)

Many-to-one (N:1) RELATIONSHIP



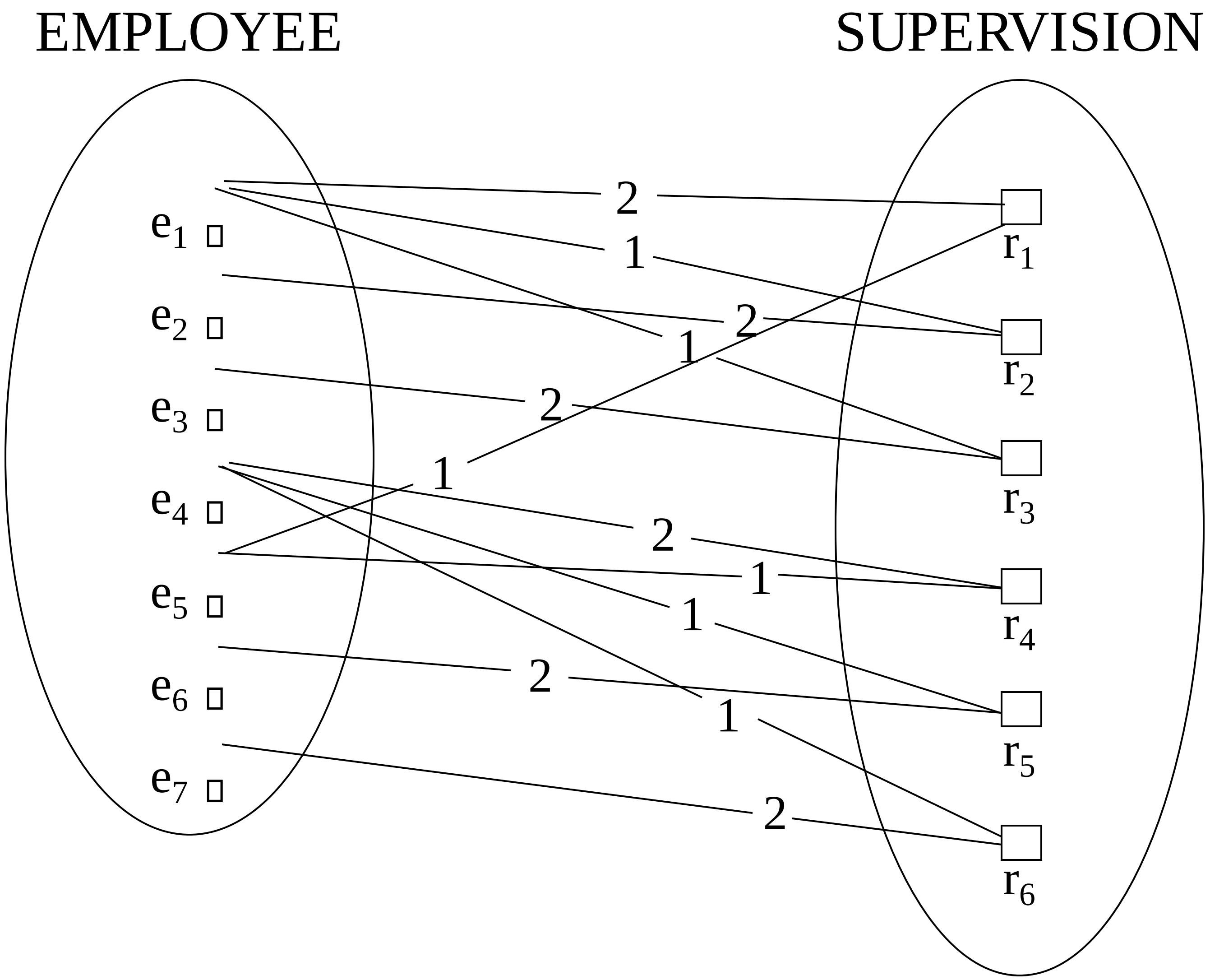
Many-to-many (M:N) RELATIONSHIP



Relationships and Relationship Types (3)

- We can also have a **recursive** relationship type.
- Both participations are same entity 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.

A RECURSIVE RELATIONSHIP SUPERVISION



© The Benjamin/Cummings Publishing Company, Inc. 1994, Elmasri/Navathe, Fundamentals of Database Systems, Second Edition

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.

Structural Constraints – one way to express semantics of relationships

<u>Structural constraints on relationships:</u>

- **Cardinality ratio** (of a binary relationship):

Specify the number of relationship instances that an entity can participate in.

cordinality ratio 1:1, 1:N, N:1, or M:N

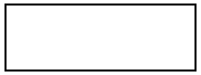
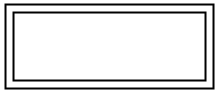
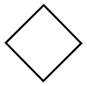
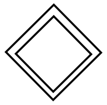
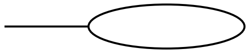
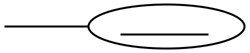
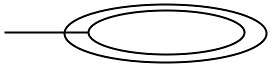
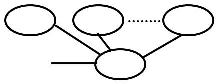
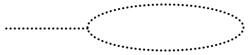
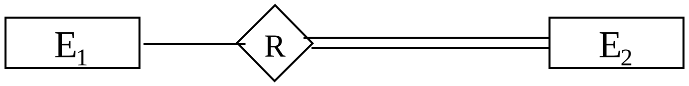
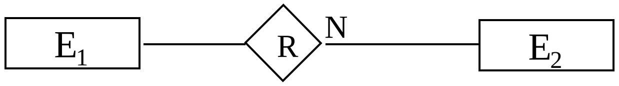
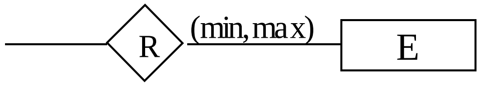
Cont..

- **Participation constraint :**.Specify whether the existence of an entity depends on its being related to another entity via the relationship**.**

Total participation : It specifies that each entity present in the entity set must mandatorily **participate** in at least one relationship instance of that relationship set, for this reason, it is also called as mandatory **participation EX : Every employee must work for Department**

Partial Participation : specifies that each entity in the entity set may or may not **participate** in the relationship instance in that relationship set. That is why, it is also called as optional **participation**. **Partial participation** is represented using a single line between the entity set and relationship set. Ex : WE do not expect every employee to manage department.

SUMMARY OF ER-DIAGRAM NOTATION FOR ER SCHEMAS

Symbol	Meaning
	ENTITY TYPE
	WEAK ENTITY TYPE
	RELATIONSHIP TYPE
	IDENTIFYING RELATIONSHIP TYPE
	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

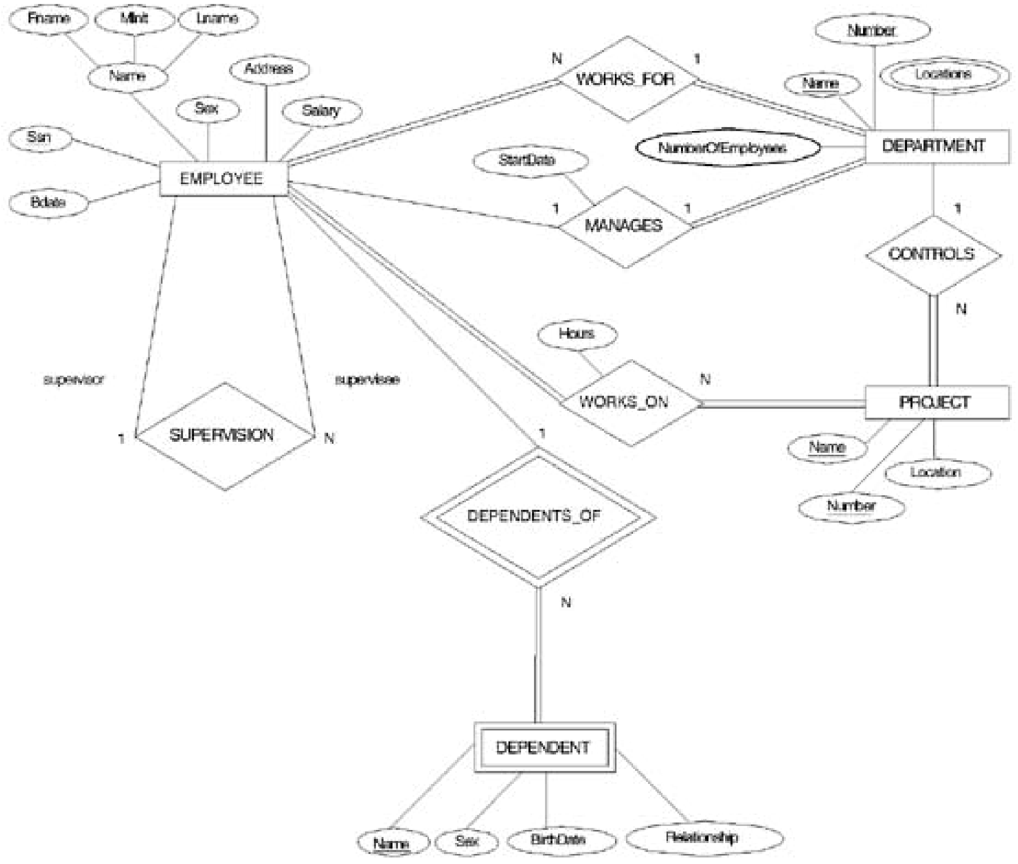
- Requirements of the Company (oversimplified for illustrative purposes)
  - The company is organized into DEPARTMENTs. Each department has a name, number and an employee who *manages* the department. We keep track of the start date of the department manager.

  Department is located at multiple location.
  - Each department *controls* a number of PROJECTs. Each project has a name, number and is located at a single location.

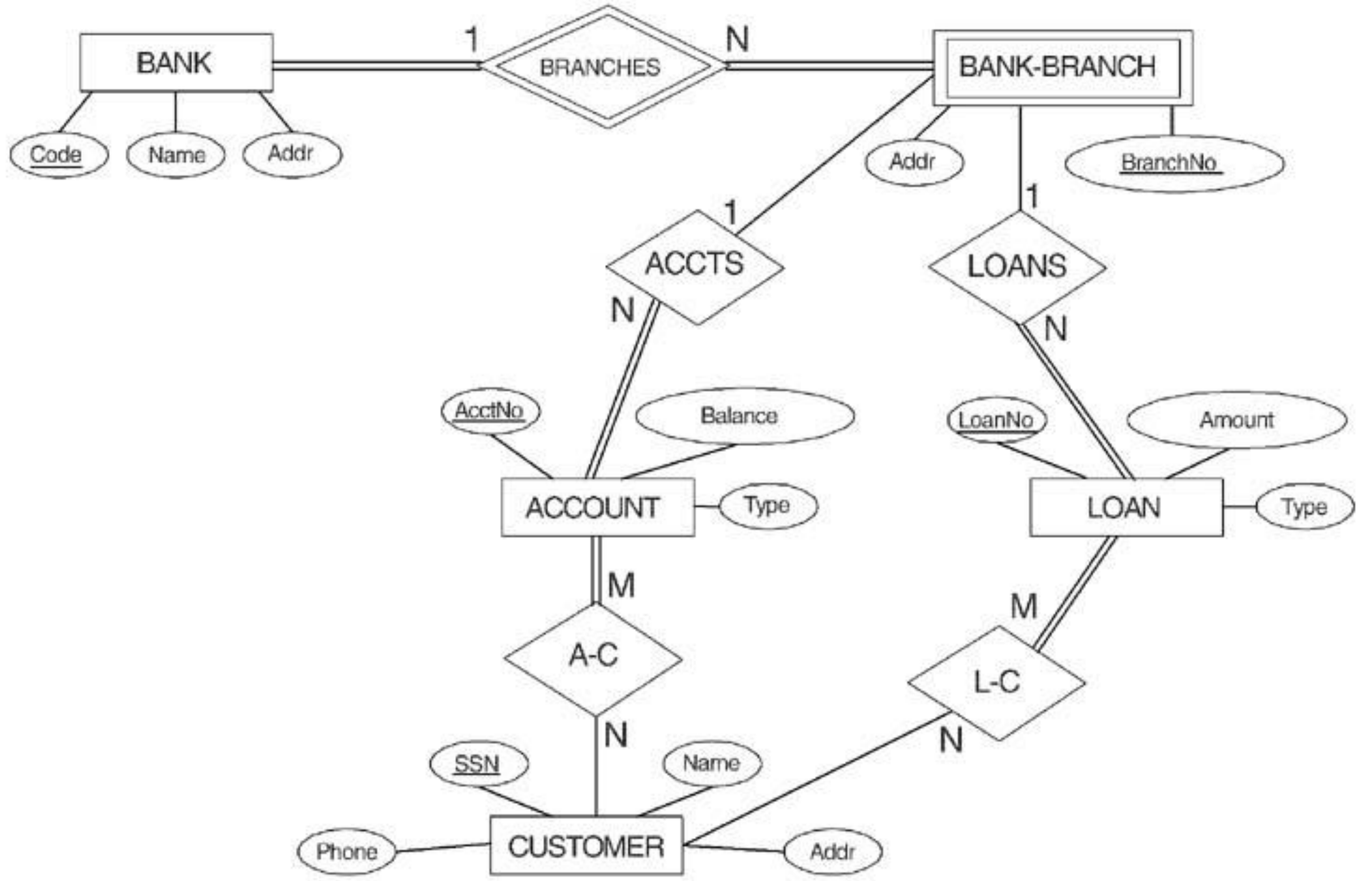
Example COMPANY Database (Cont.)

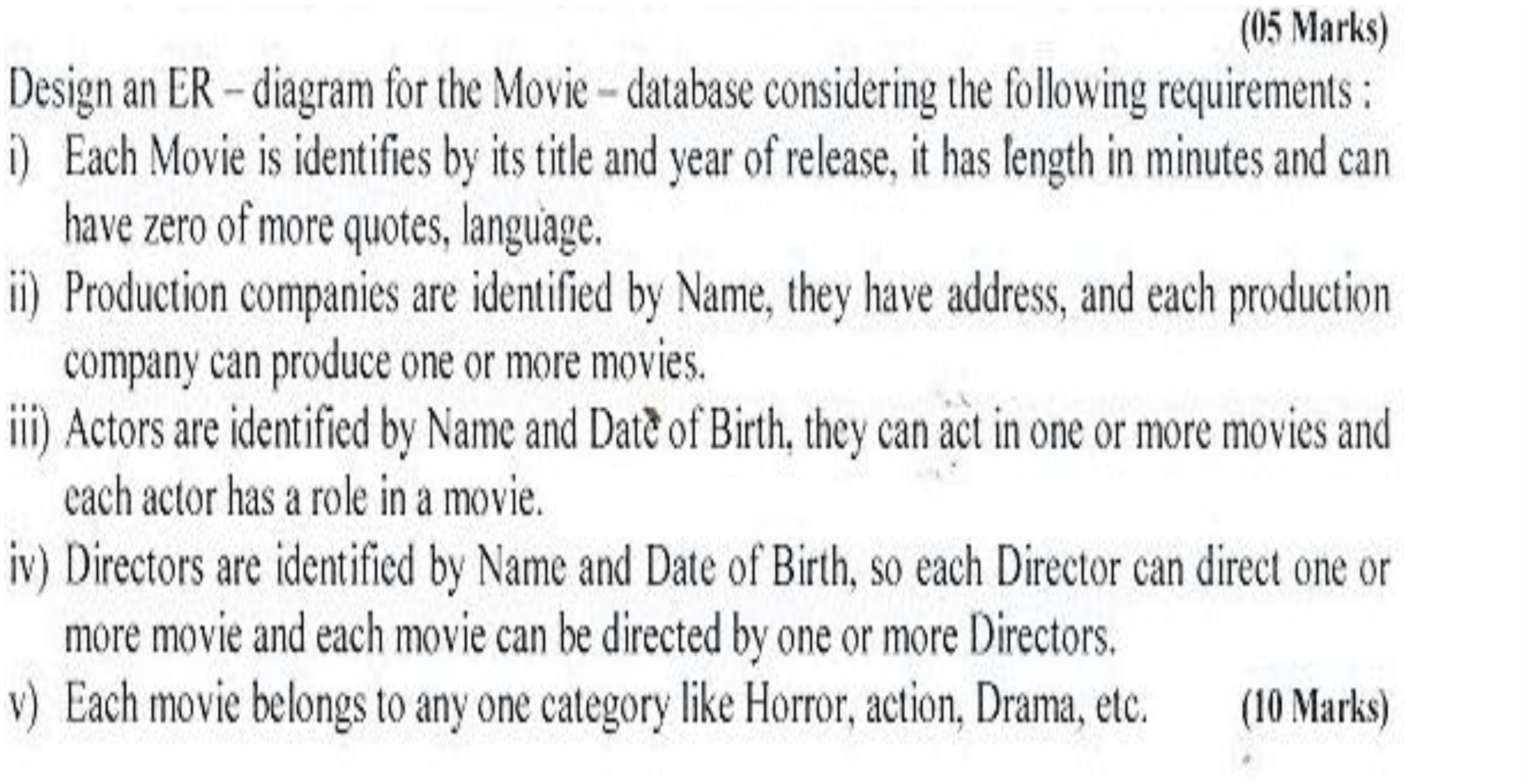
- We store each EMPLOYEE’s social security number, address, salary, sex, and birthdate. Each employee *works for* one department but may *work on* several projects. We keep track of the number of hours per week that an employee currently works on each project. We also keep track of the *direct supervisor* 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.

Cont..


Fname
Minit
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
1
MANAGES
1
1
CONTROLS
N
Hours
supervisor
supervisee
WORKS_ON
N
PROJECT
1
SUPERVISION
N
1
Name
Location
Number
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship

ER DIAGRAM FOR A BANK DATABASE





(05 Marks)

Design an ER – diagram for the Movie – database considering the following requirements :

i) Each Movie is identifies by its title and year of release, it has length in minutes and can have zero of more quotes, language.

ii) Production companies are identified by Name, they have address, and each production company can produce one or more movies.

iii) Actors are identified by Name and Date of Birth, they can act in one or more movies and each actor has a role in a movie.

iv) Directors are identified by Name and Date of Birth, so each Director can direct one or more movie and each movie can be directed by one or more Directors.

v) Each movie belongs to any one category like Horror, action, Drama, etc. **(10 Marks)**

PROBLEM with ER notation

THE ENTITY RELATIONSHIP MODEL IN ITS ORIGINAL FORM DID NOT SUPPORT THE SPECIALIZATION/ GENERALIZATION ABSTRACTIONS