Chapter 7: Entity Relationship Model

Database Design Process

•Use a high-level conceptual data model (ER Model).

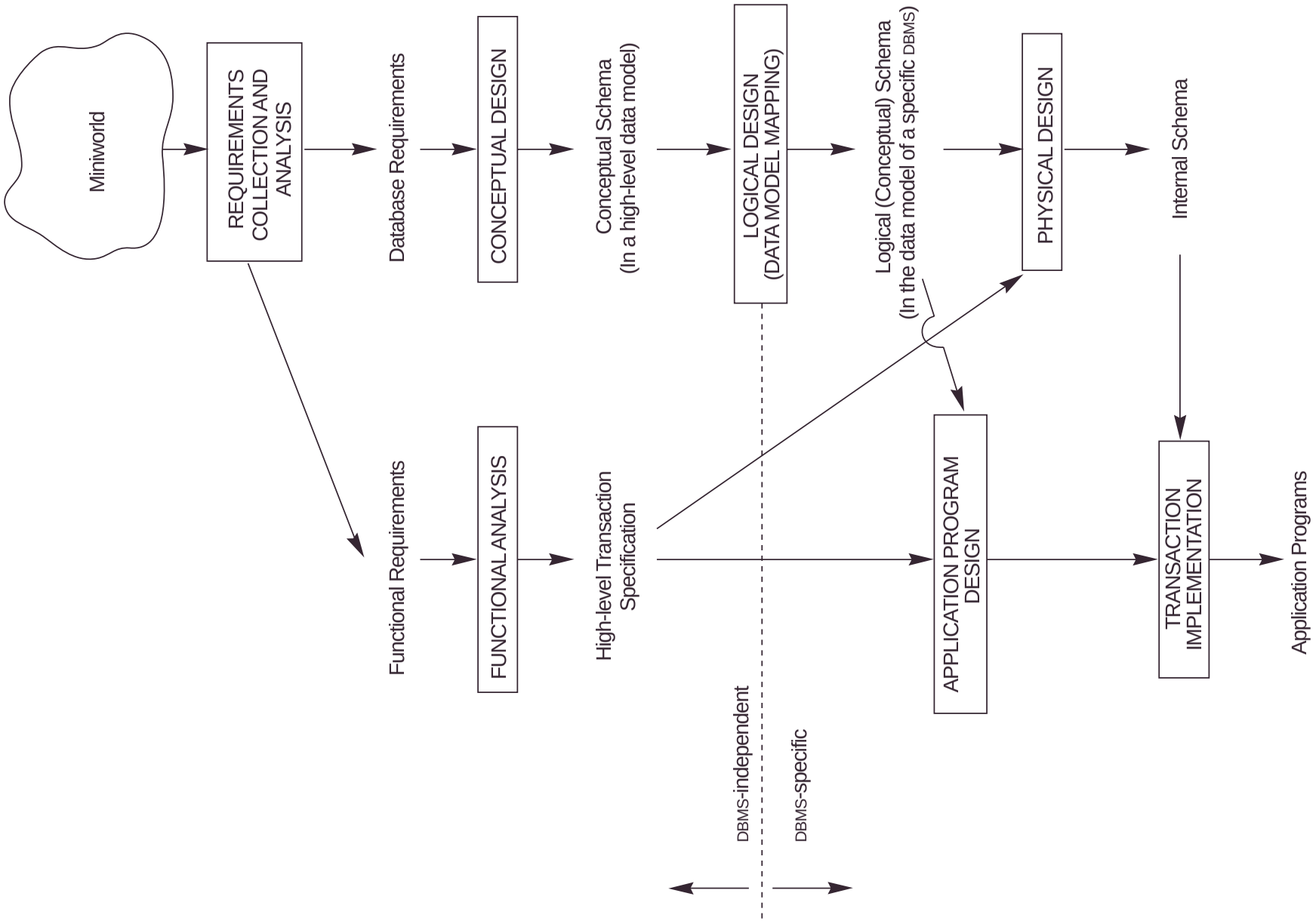
• Identify objects of interest (entities) and relationships between these objects

•Identify constraints (conditions)

•End result is an E-R Diagram that captures all entity, relationship types and constraints

Figure 3.1 Phases of Database Design

Figure 3.1 A simplified diagram to illustrate the main phases of database design.



Example Database Application (Company Database)

Company organized into **DEPARTMENT**s. Each department has unique name and a particular employee who manages the department. Start date for the manager is recorded. Department may have several locations.

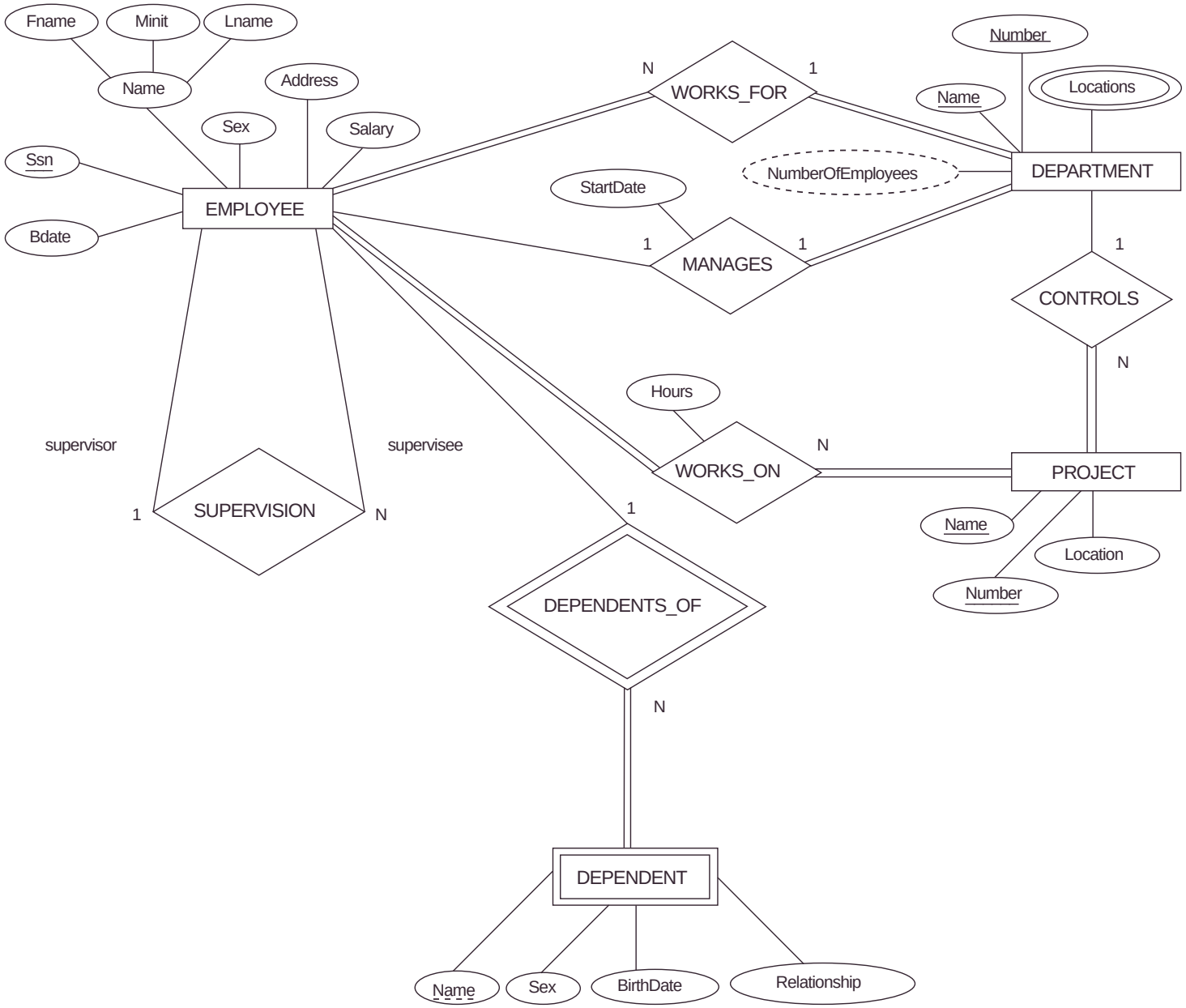
A department controls a number of **PROJECT**s. Projects have a unique name, number and a single location.

Company's **EMPLOYEE**s name, ssno, address, salary, sex and birth date are recorded. An employee is assigned to one department, but may work for several projects (not necessarily controlled by her dept). Number of hours/week an employee works on each project is recorded; The immediate supervisor for the employee.

Employee's **DEPENDENT**s are tracked for health insurance purposes (dependent name, birthdate, relationship to employee).

Figure 3.2: ER Diagram

Figure 3.2 ER schema diagram for the company database.



Entities and Attributes

Entity: an object in the real world with an independent existence.

Attribute: Property that describes an aspect of the entity.

Figure 3.3

Attribute types:

Simple vs Composite (Figure 3.4)
Single-valued vs Multi-valued (e.g. Locations for DEPARTMENT)
Stored vs Derived (e.g. NumberOfEmployees for DEPARTMENT)

Figure 3.5: example of a complex attribute with multi-valued and composite components

Null values for attributes: Not applicable, Unknown (Missing; not known if applicable)

Figure 3.3 Two entities, an employee e_1 and a company c_1, and their attribute values.

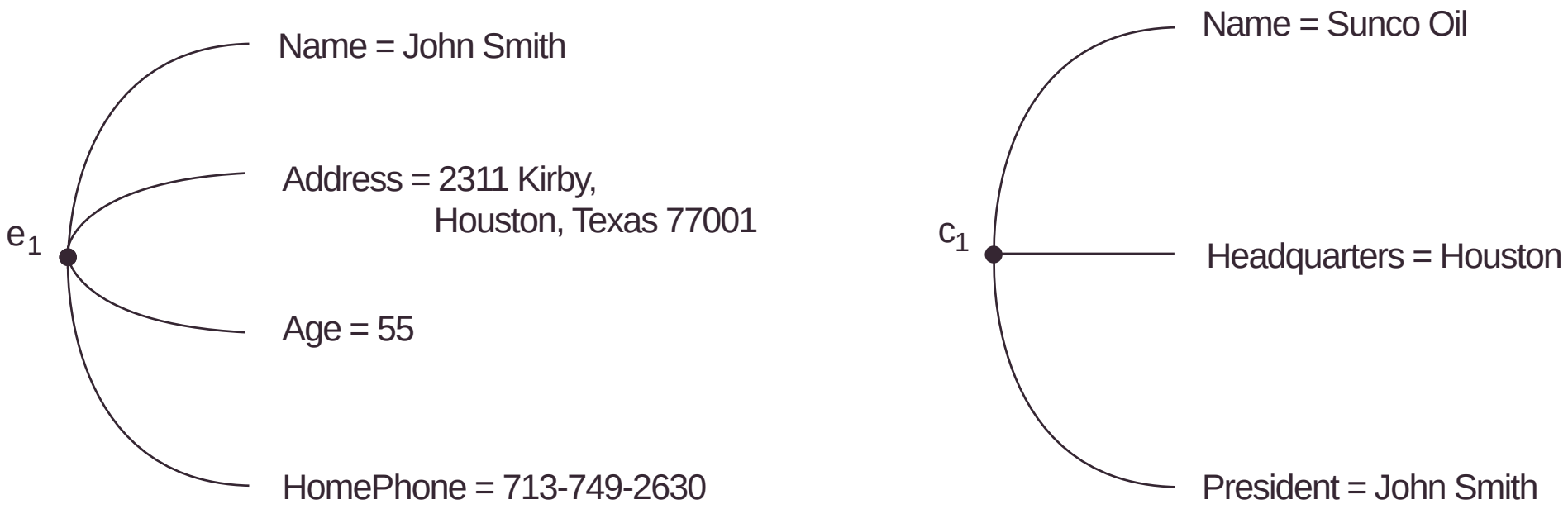


Figure 3.4 A hierarchy of composite attributes; the StreetAddress component of an Address is further composed of Number, Street, and ApartmentNumber.

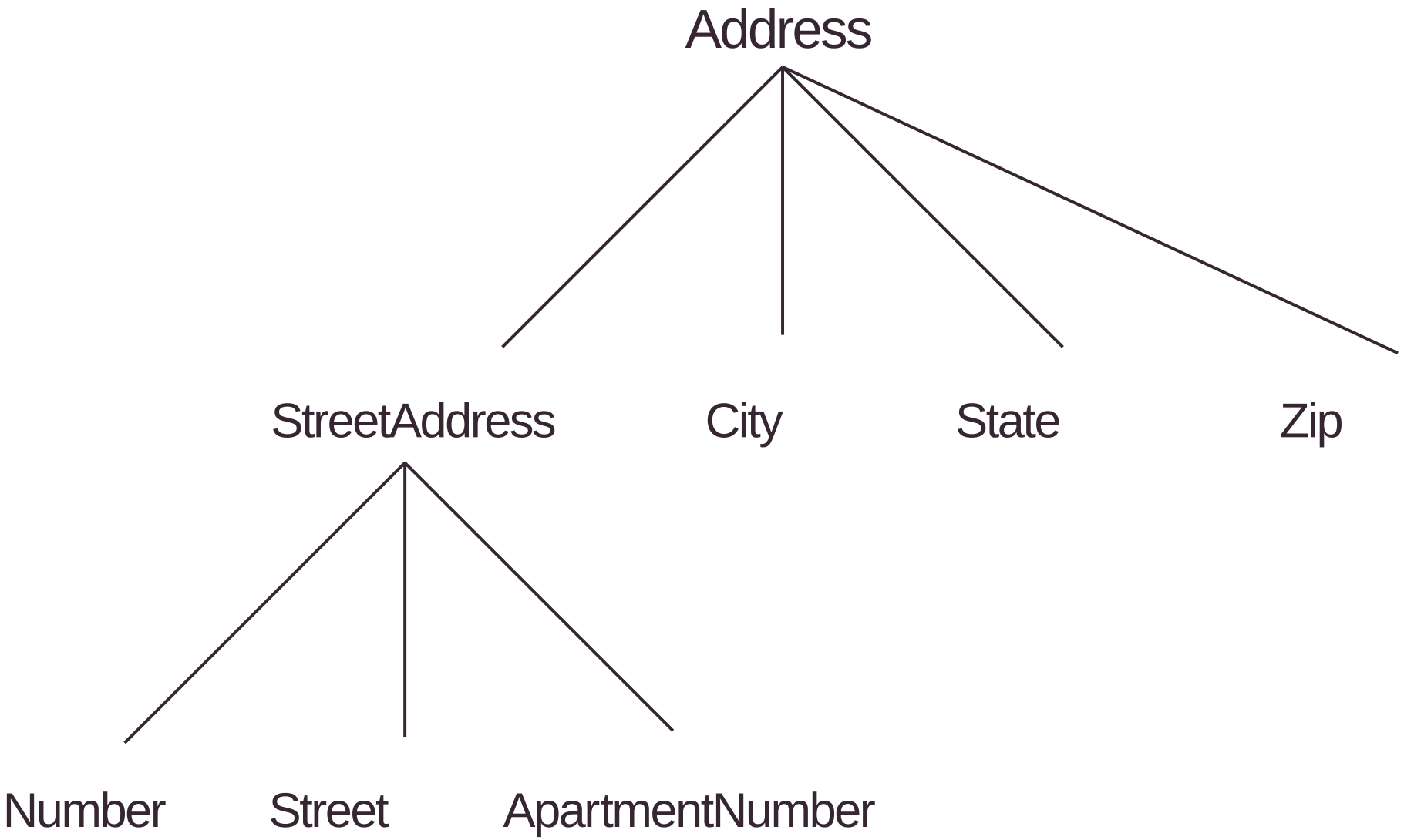


Figure 3.5 A complex attribute AddressPhone with multivalued and composite components.

{AddressPhone({Phone(AreaCode,PhoneNumber)},
Address(StreetAddress(Number,Street,ApartmentNumber),
City,State,Zip)) }

Entity Types, Value Sets, Key Attributes

An entity type defines a set of entities that have the same attributes.
See Figure 3.6

Rectangular box in ER Diagram denotes Entity Types
Ovals denote Attributes
    double-ovals: multi-valued attribute
    tree structured ovals: composite attribute

Entity Set = Set of all entities of the same type.

Key Attributes: uniquely identify each entity within an entity set
    (these are underlined in the ER Diagram)

Value Sets: or Domains for attributes.
    A: E -> P(V) A:attribute, E:Entity set, V: Value set
        V = P(V1) x ... x P(Vn) for composite attributes
    A(e) denotes the value of attribute A for entity e
Figure 3.7 Car Entity Type

Figure 3.6 Two entity types named EMPLOYEE and COMPANY, and some of the member entities in the collection of entities (or entity set) of each type.

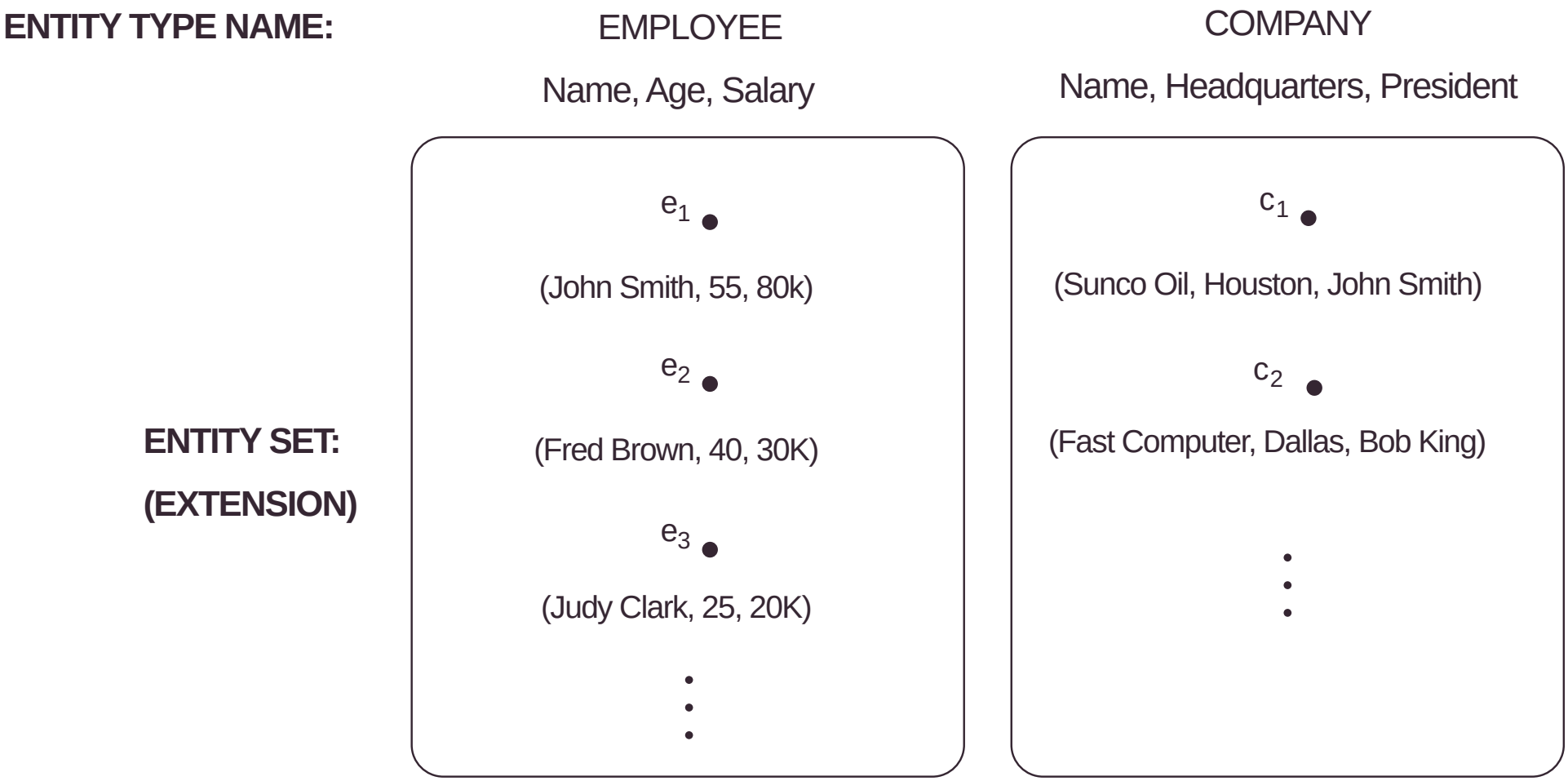
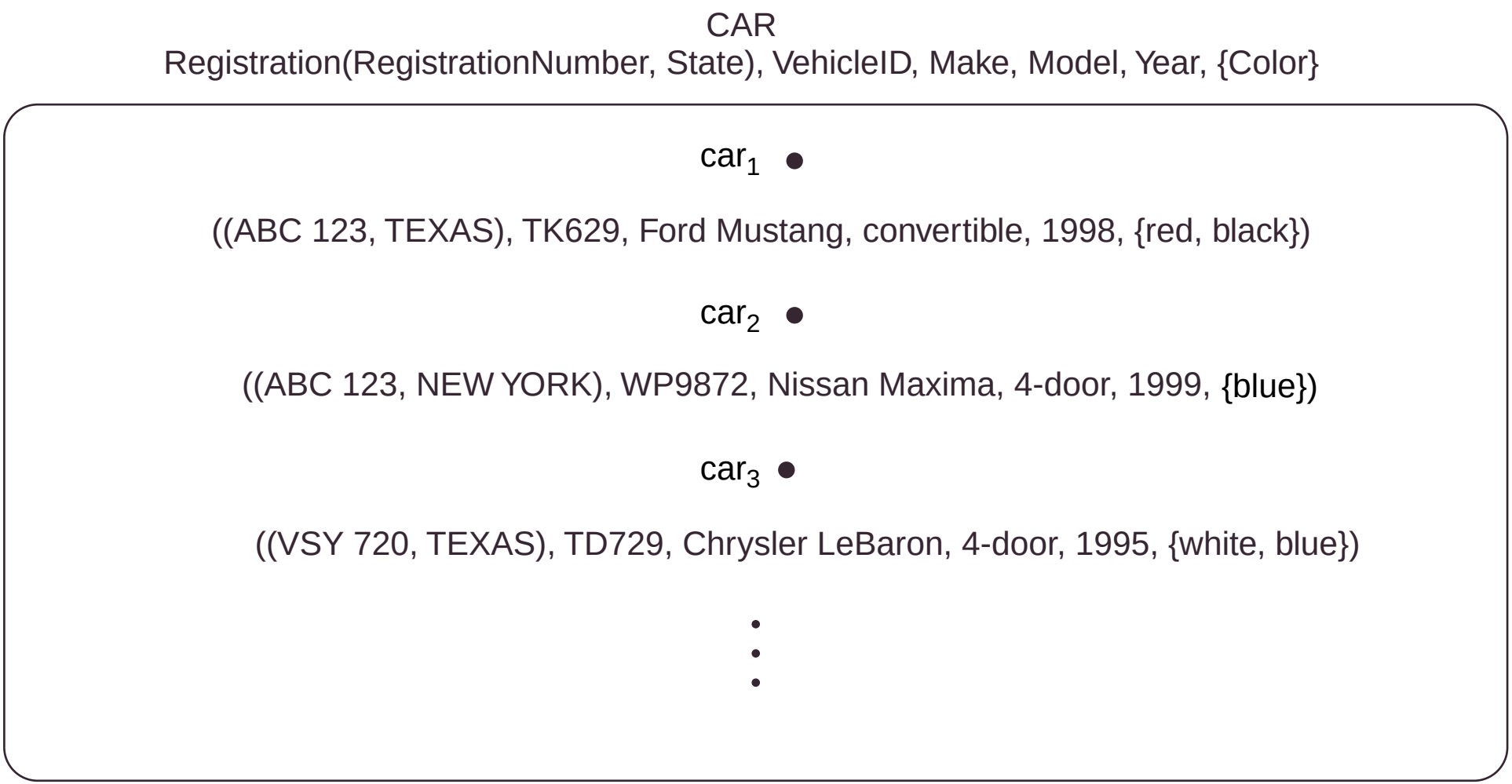


Figure 3.7 The CAR entity type, with two key attributes Registration and VehicleID. Multivalued attributes are shown between set braces {}. Components of a composite attribute are shown between parentheses ().



Initial design of the COMPANY DATABASE

Figure 3.8

4 entity types:

DEPARTMENT
EMPLOYEE
PROJECT
DEPENDENT

Not a perfect design! because it captures relationships between entities as attributes (in general not a good idea)

Figure 3.8 Preliminary design of entity types for the COMPANY database whose requirements are described in Section 3.2.

DEPARTMENT
Name, Number, {Locations}, Manager, ManagerStartDate

PROJECT
Name, Number, Location, ControllingDepartment

EMPLOYEE
Name (FName, MInit, LName), SSN, Sex, Address, Salary,
BirthDate, Department, Supervisor, {WorksOn (Project, Hours)}

DEPENDENT
Employee, DependentName, Sex, BirthDate, Relationship

Relationships, Roles, Structural Constraints

Relationship type: R among n Entity types E1, ..., En
defines a set of associations among entities from these types.
each association will be denoted as:
(e1, ..., en) where ei belongs to Ei, 1 <= i <= n.
ex. WORKS_FOR relationship in Figure 3.9

Degree of relationship = n (usually n = 2, binary relationship)

Ternary relationship: Figure 3.10

Relationships as attributes (e.g. Dept -- Empl relationship
can be viewed as two attributes one in Dept. and the other in Empl)

Figure 3.9 Some instances of the WORKS_FOR relationship between EMPLOYEE and DEPARTMENT.

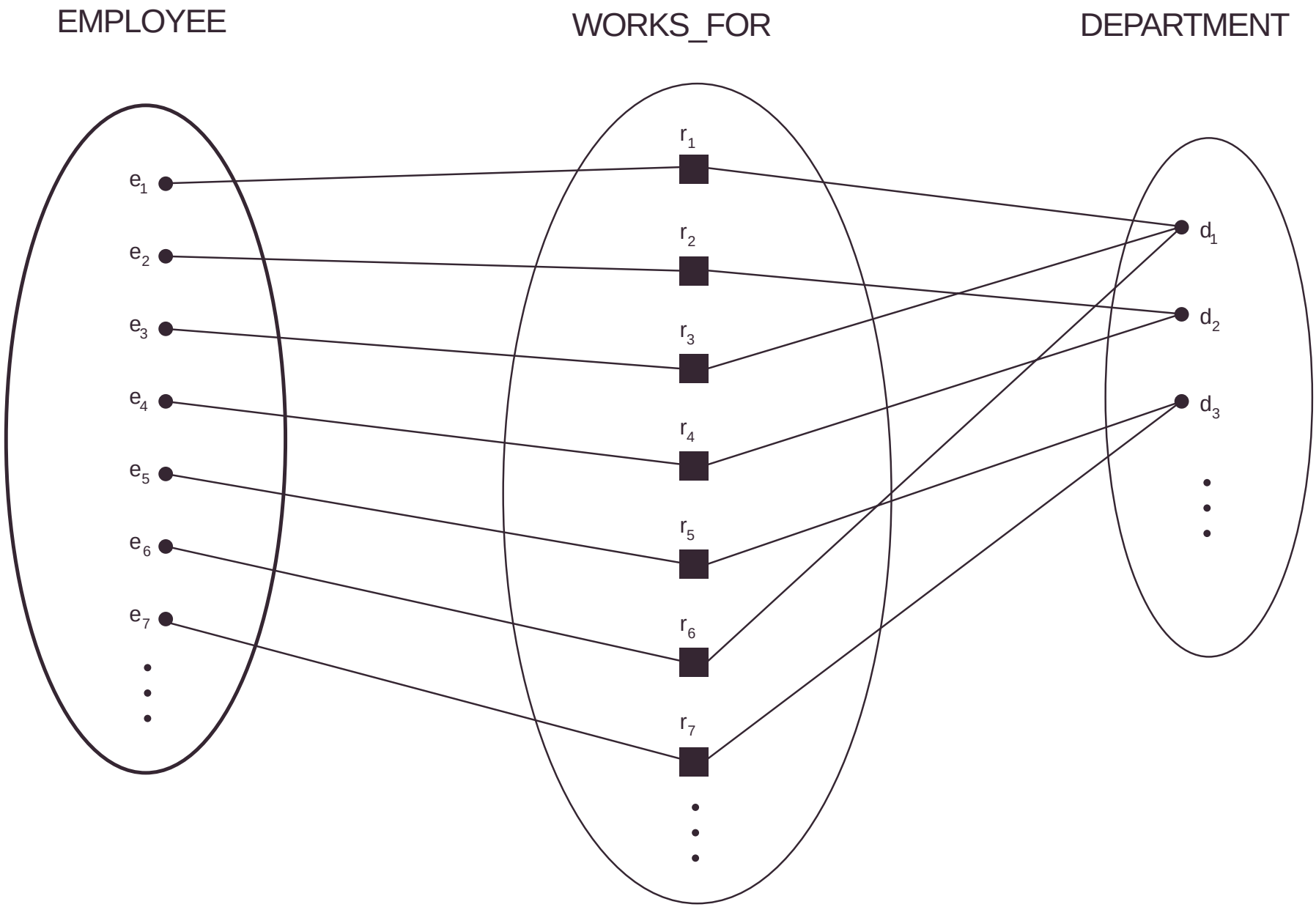
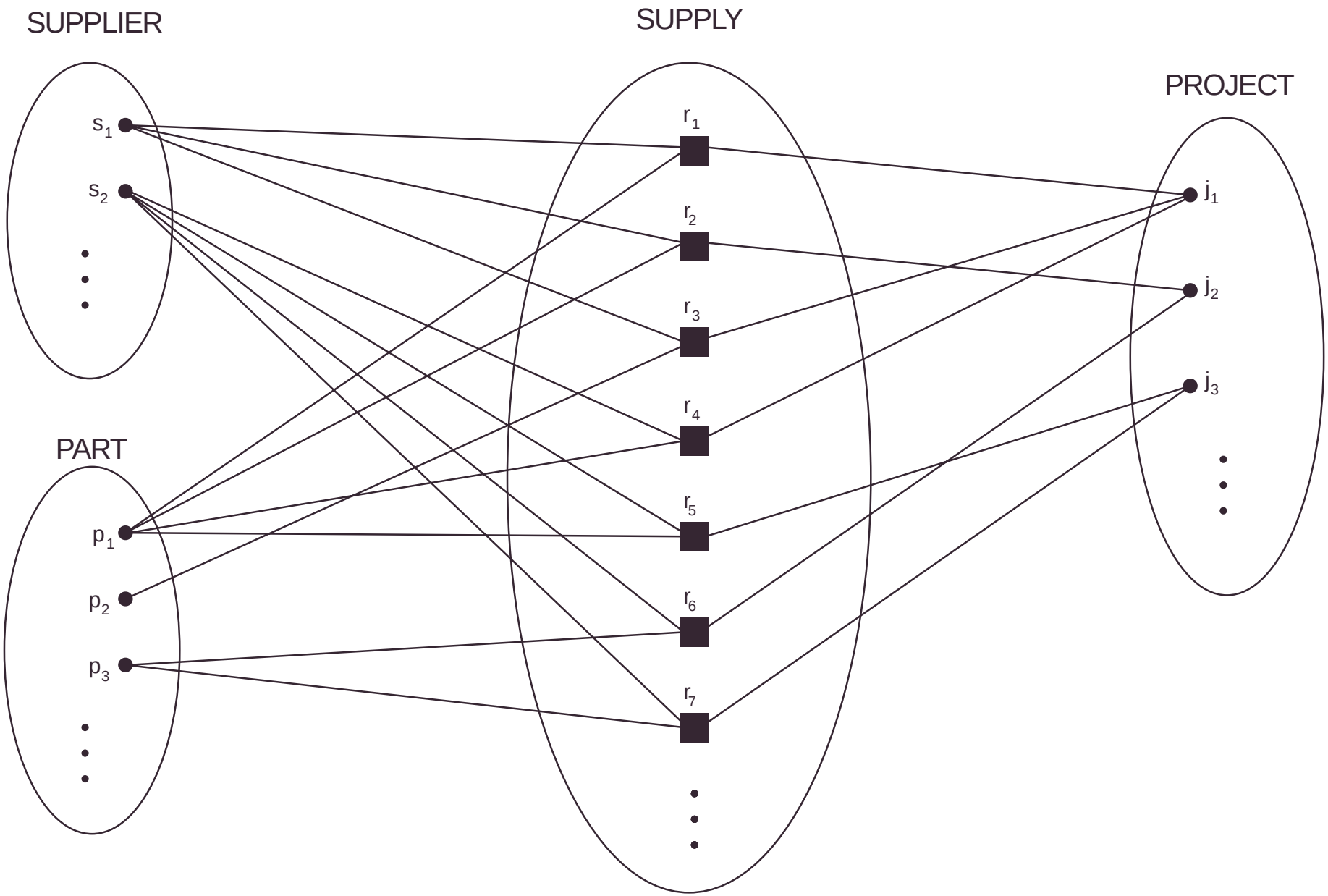


Figure 3.10 Some relationship instances of a ternary relationship SUPPLY.

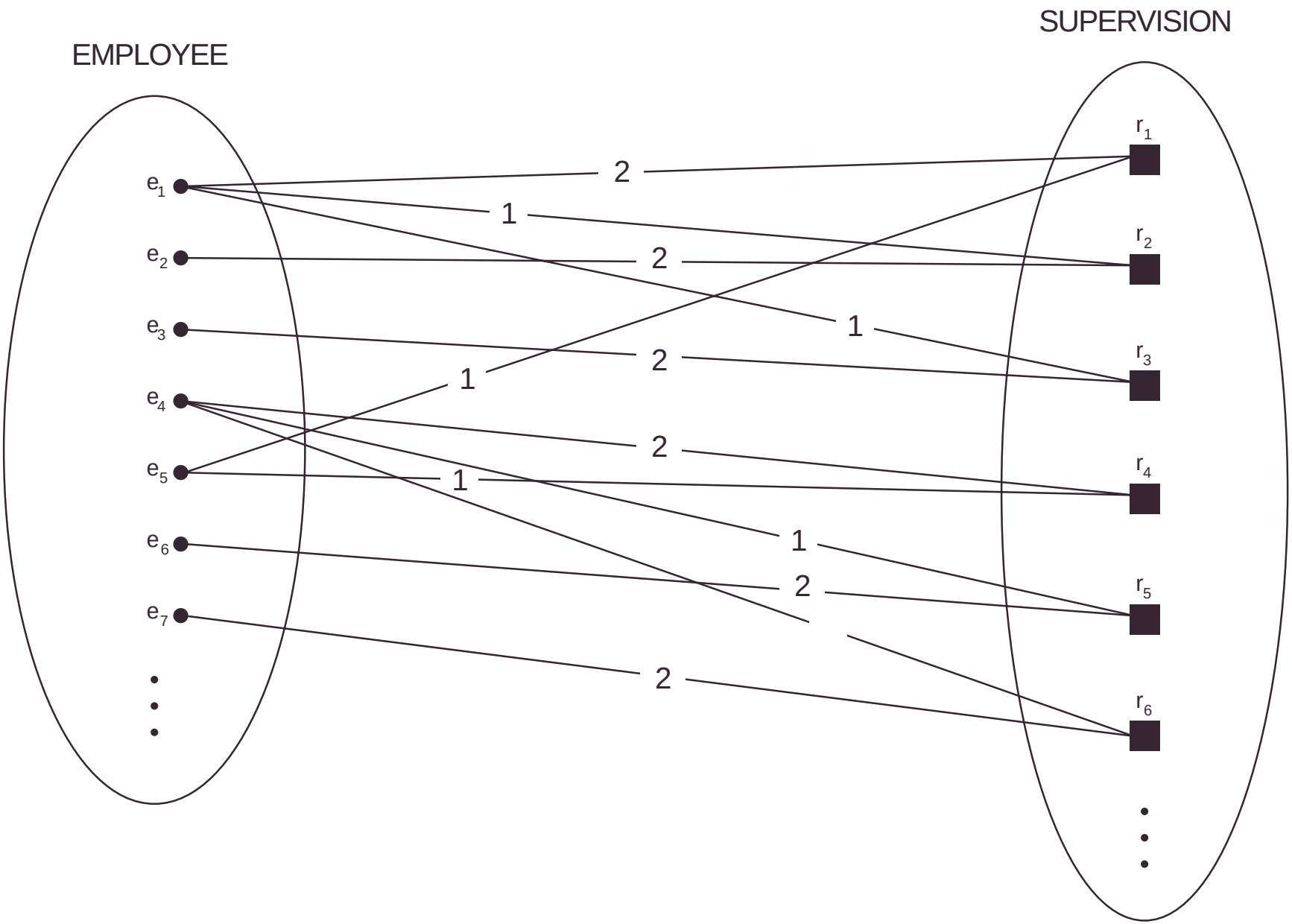


Role names

Each entity participating in a relationship has a ROLE.
e.g. Employee plays the role of worker and Department
plays the role of employer in the WORKS_FOR relationship type

Role names are more important in recursive relationships
ex. Figure 3.11

Figure 3.11 The recursive relationship SUPERVISION, where the EMPLOYEE entity type plays the two roles of supervisor (1) and supervisee (2).



Structural Constraints on Relationships

2 types:

Cardinality Ratio Constraint (1-1, 1-N, M-N)

Participation Constraint

* total participation (existence dependency)

* partial participation

Figures 3.12 and 3.13

In ER Diagrams:
total participation is denoted by double line and
partial participation by single line
cardinality ratios are mentioned as labels of edges

Figure 3.12 The 1:1 relationship MANAGES, with partial participation of employee and total participation of DEPARTMENT.

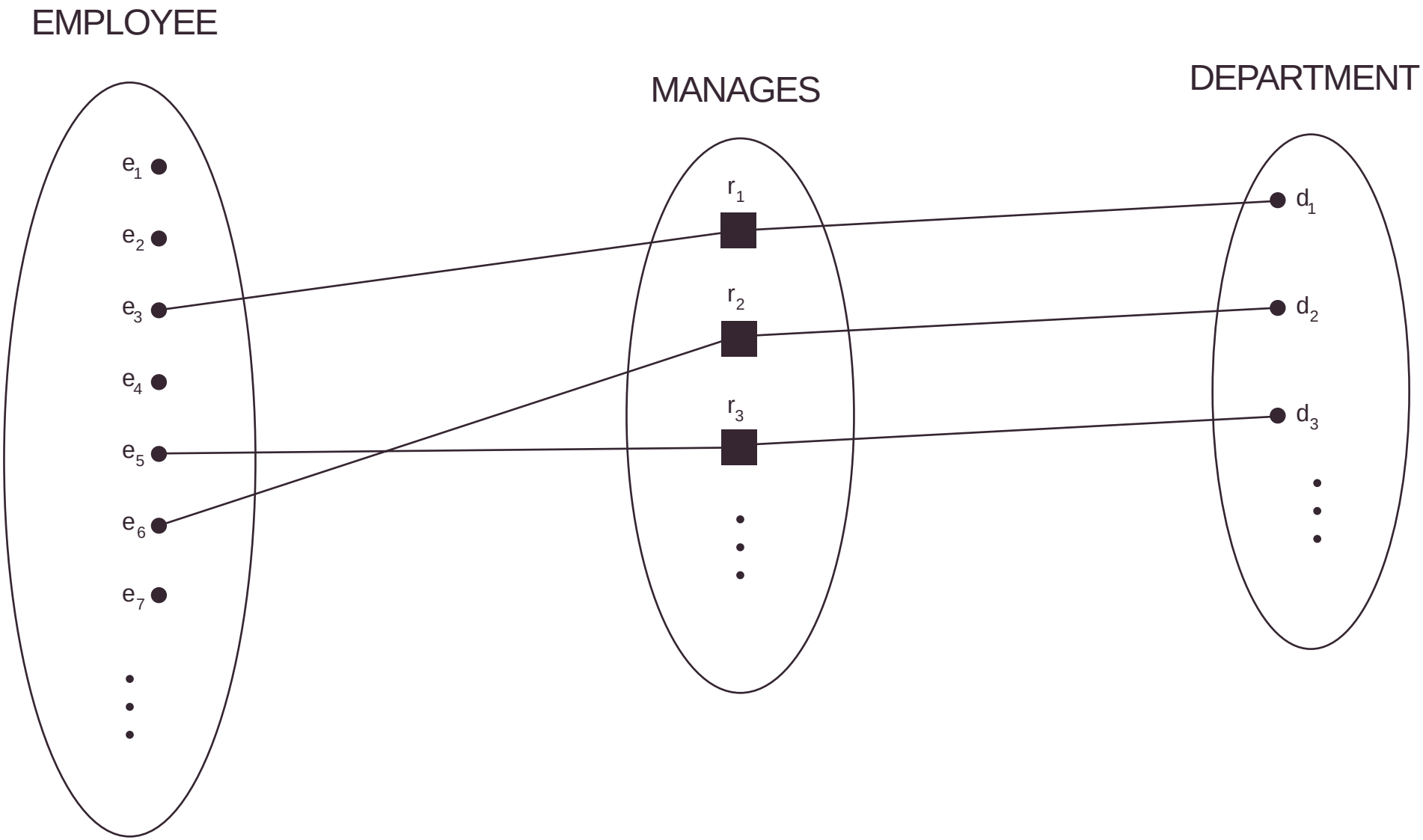
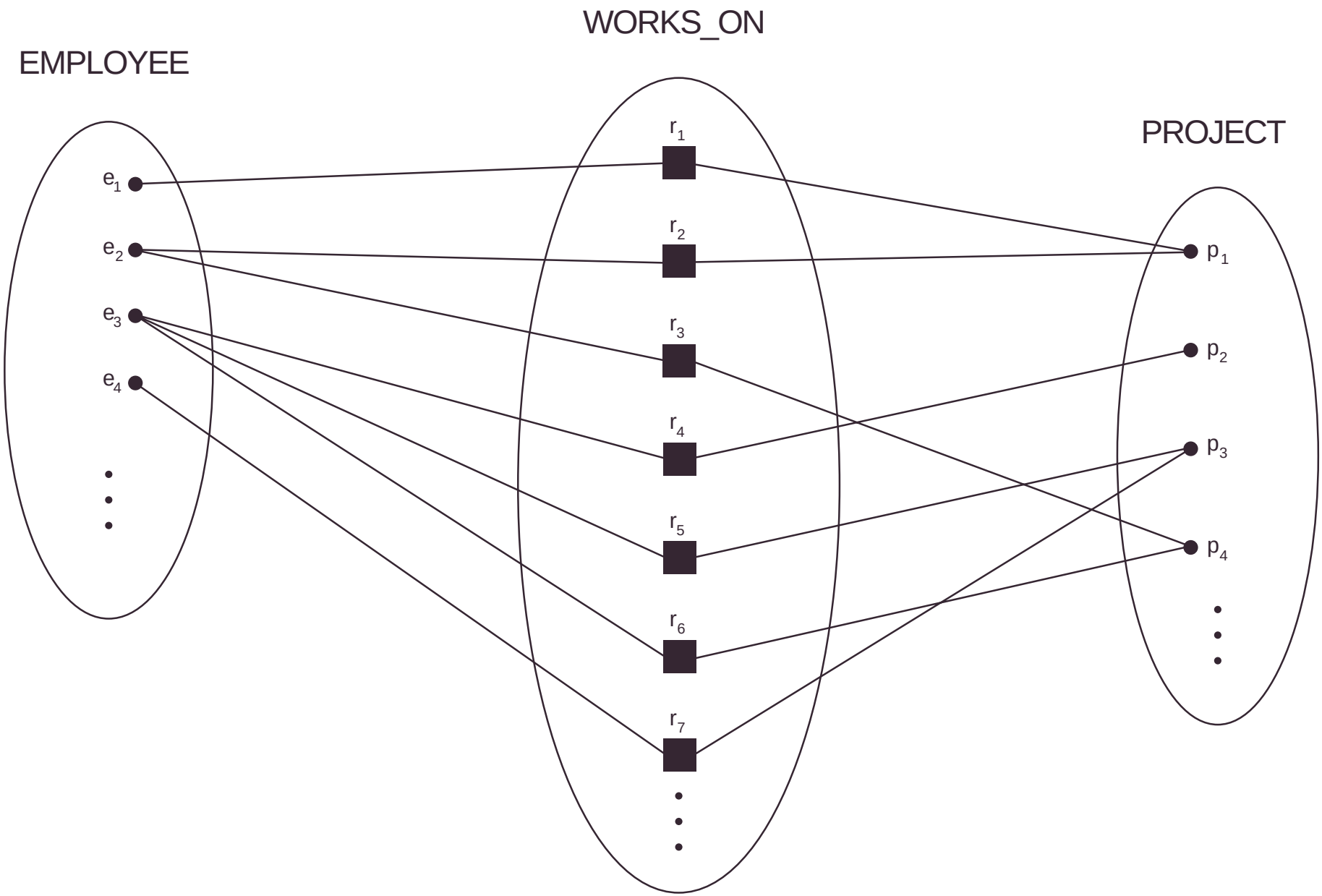


Figure 3.13 The M:N relationship WORKS_ON between EMPLOYEE and PROJECT.



Attributes of relationships:

e.x. Hours attribute for WORKS_ON relationship

If relationship is 1-N or 1-1, these attributes can be be migrated to the entity sets involved in the relationship.

1-N: migrate to N side

1-1: migrate to either side

Weak Entity Types:

Entity Types that do not have key attributes.

Such entities are identified by being related to other entity sets called **identifying owner**.
This relationship is called **identifying relationship**.

Partial key: attributes that uniquely identify entities within the identifying relationship.

A weak entity type always has TOTAL participation.

Ex. DEPENDENT is a weak entity type.

Notation for ER Diagrams:

Fig 3.14, 3.15

(Min,Max) notation encapsulates both types of structural constraints. This is more general than previous notation.

Min >=1 implies TOTAL participation

Min = 0 implies PARTIAL participation

Alternative Notations : UML based, ERD (Crow-Foot ERD model), others

Figure 3.14 Summary of ER diagram notation.

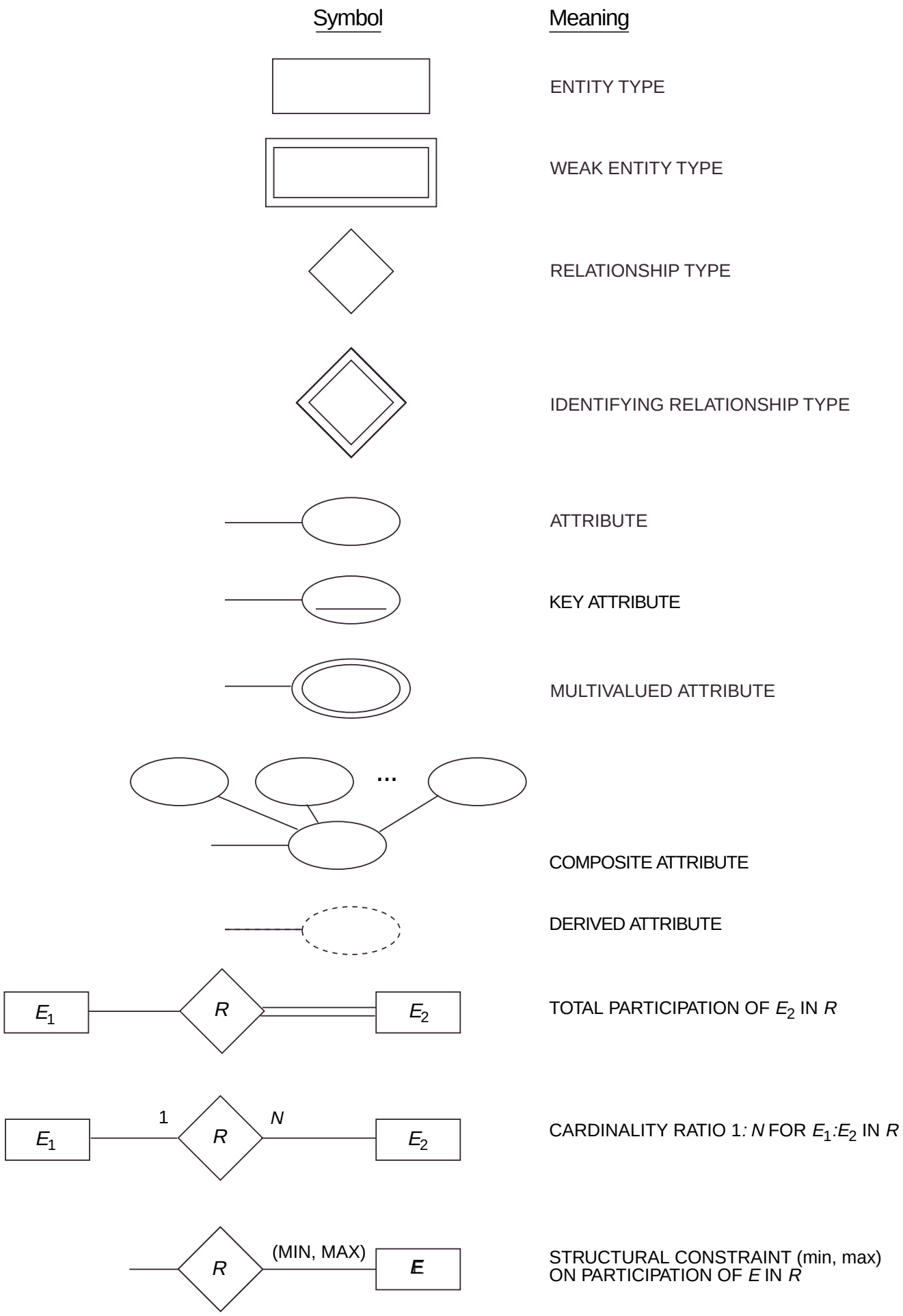
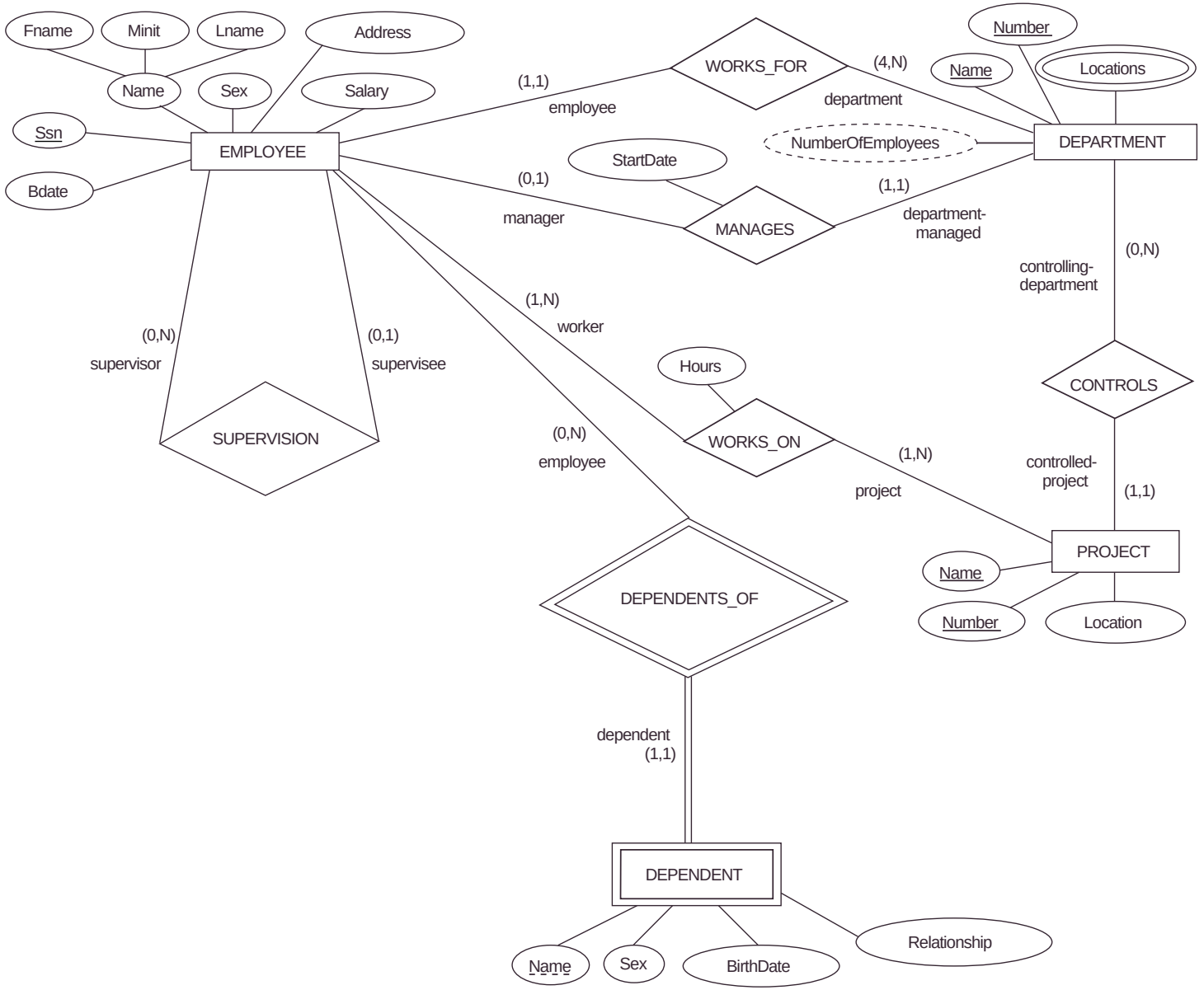


Figure 3.15 ER diagram for the COMPANY schema, with all role names included and with structural constraints on relationships specified using the alternate notation (min, max).

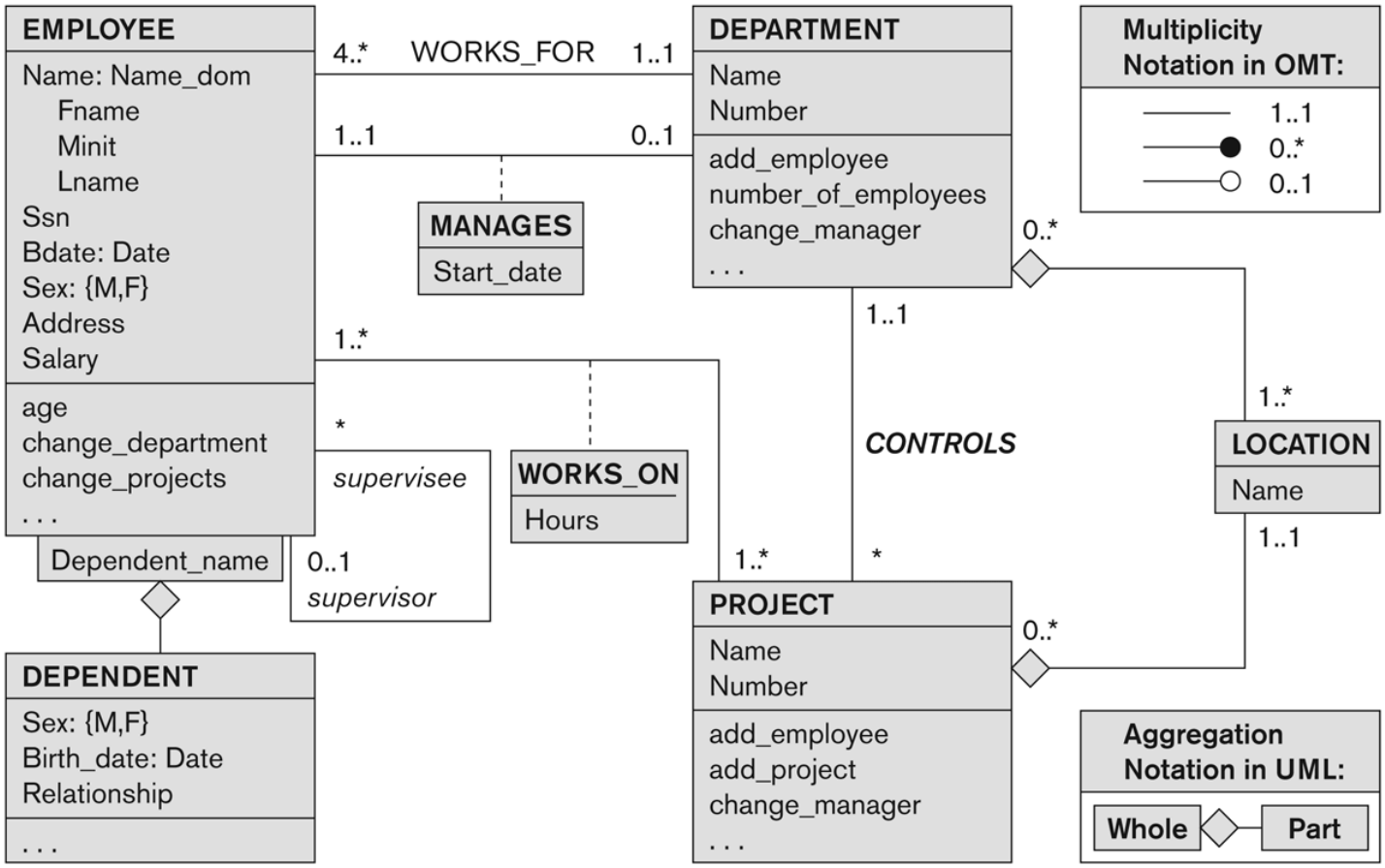


Some of the Currently Available Automated Database Design Tools

COMPANY	TOOL	FUNCTIONALITY
Embarcadero Technologies	ER Studio	Database Modeling in ER and IDEF1X
	DB Artisan	Database administration, space and security management
Oracle	Developer 2000/Designer 2000	Database modeling, application development
Popkin Software	System Architect 2001	Data modeling, object modeling, process modeling, structured analysis/design
Platinum (Computer Associates)	Enterprise Modeling Suite: Erwin, BPWin, Paradigm Plus	Data, process, and business component modeling
Persistence Inc.	Pwertier	Mapping from O-O to relational model
Rational (IBM)	Rational Rose	UML Modeling & application generation in C++/JAVA
Resolution Ltd.	Xcase	Conceptual modeling up to code maintenance
Sybase	Enterprise Application Suite	Data modeling, business logic modeling
Visio	Visio Enterprise	Data modeling, design/reengineering Visual Basic/C++

UML class diagram for COMPANY database schema

Figure 3.16
The COMPANY conceptual schema in UML class diagram notation.



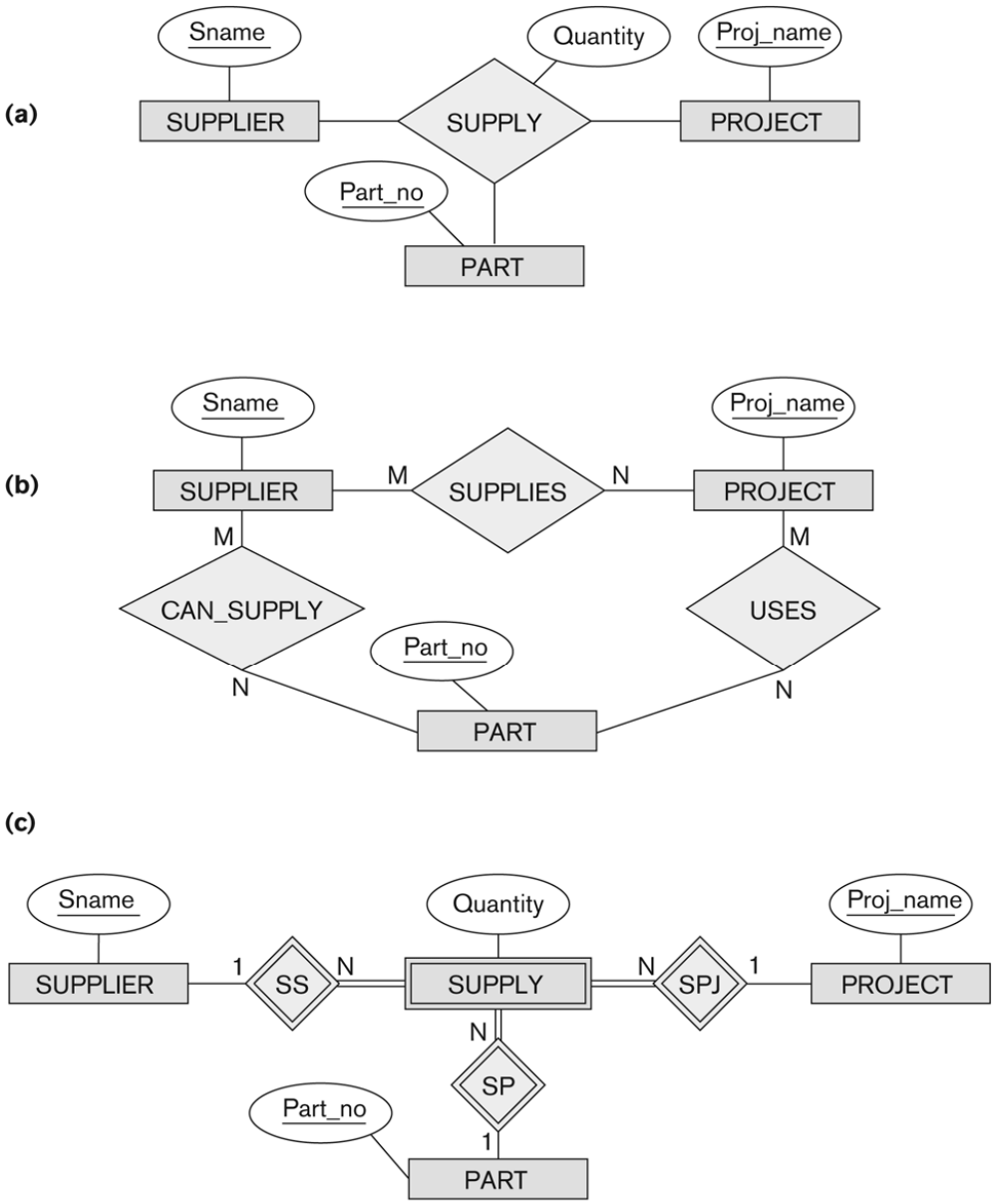
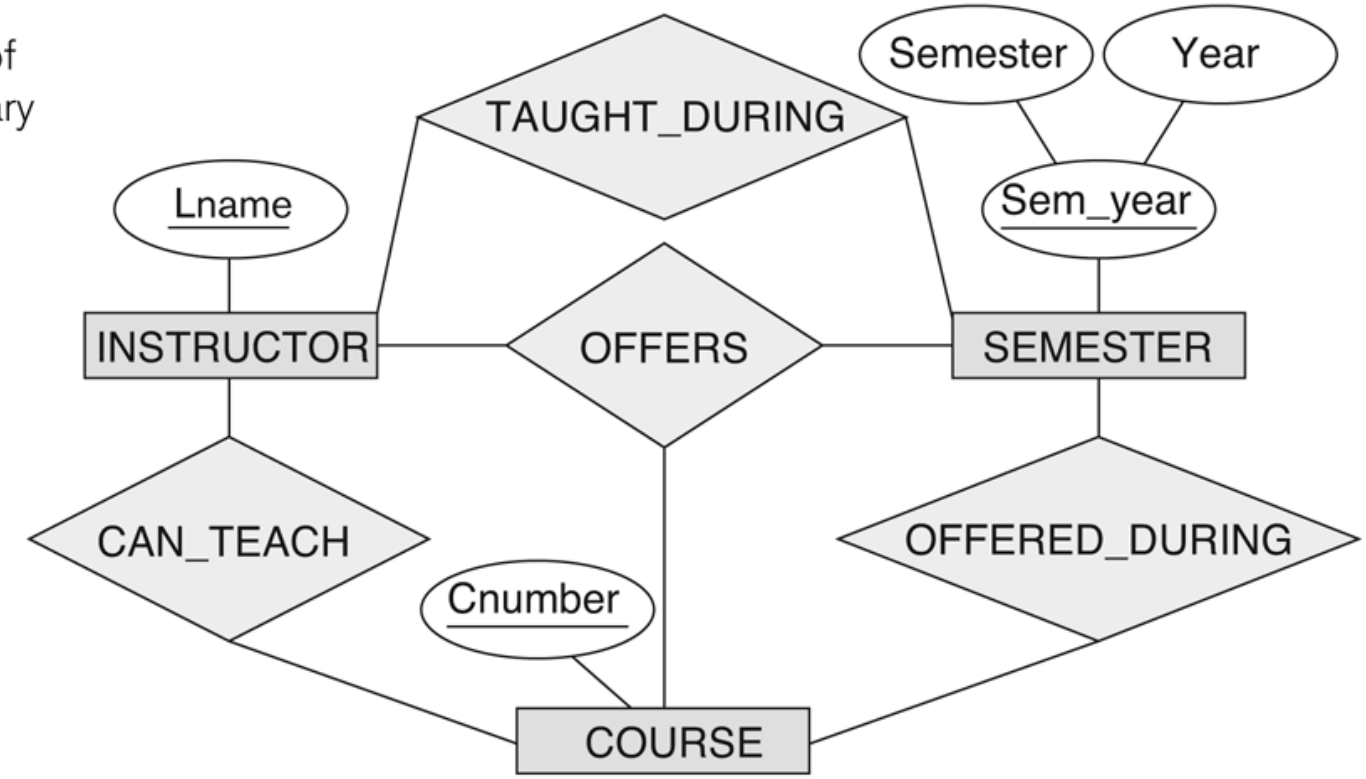


Figure 3.17
Ternary relationship types. (a) The SUPPLY relationship. (b) Three binary relationships not equivalent to SUPPLY. (c) SUPPLY represented as a weak entity type.

Figure 3.18
Another example of ternary versus binary relationship types.



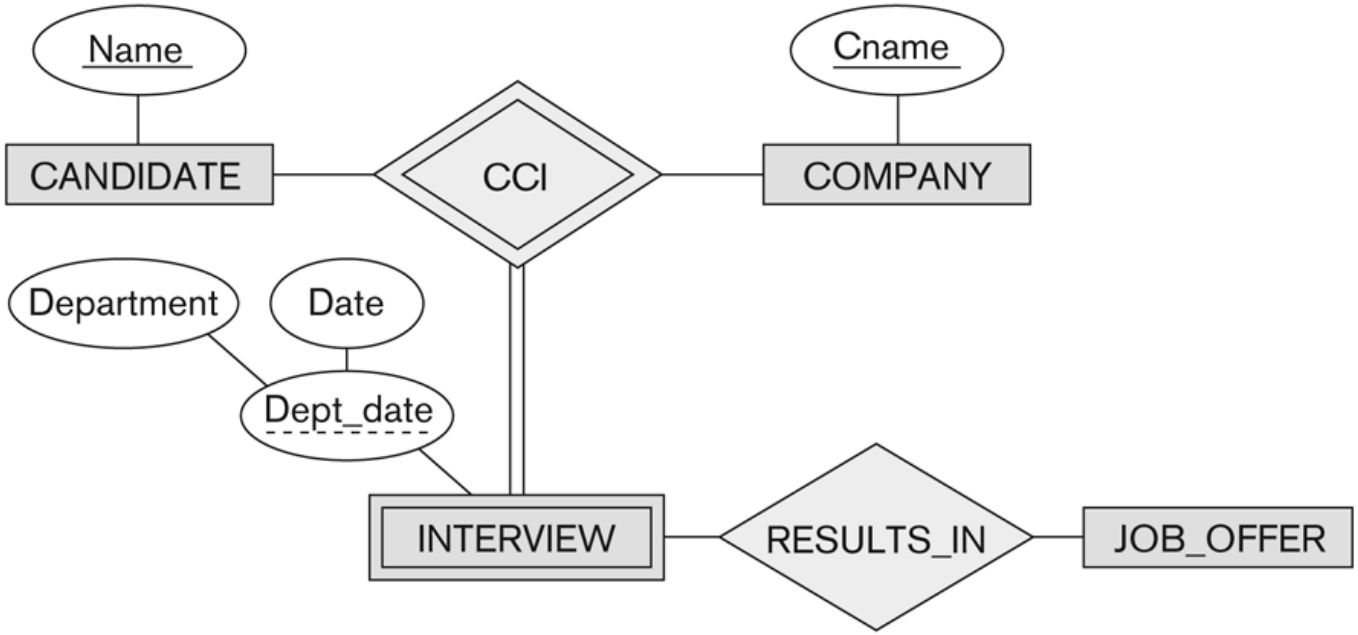


Figure 3.19
A weak entity type INTERVIEW with a ternary identifying relationship type.