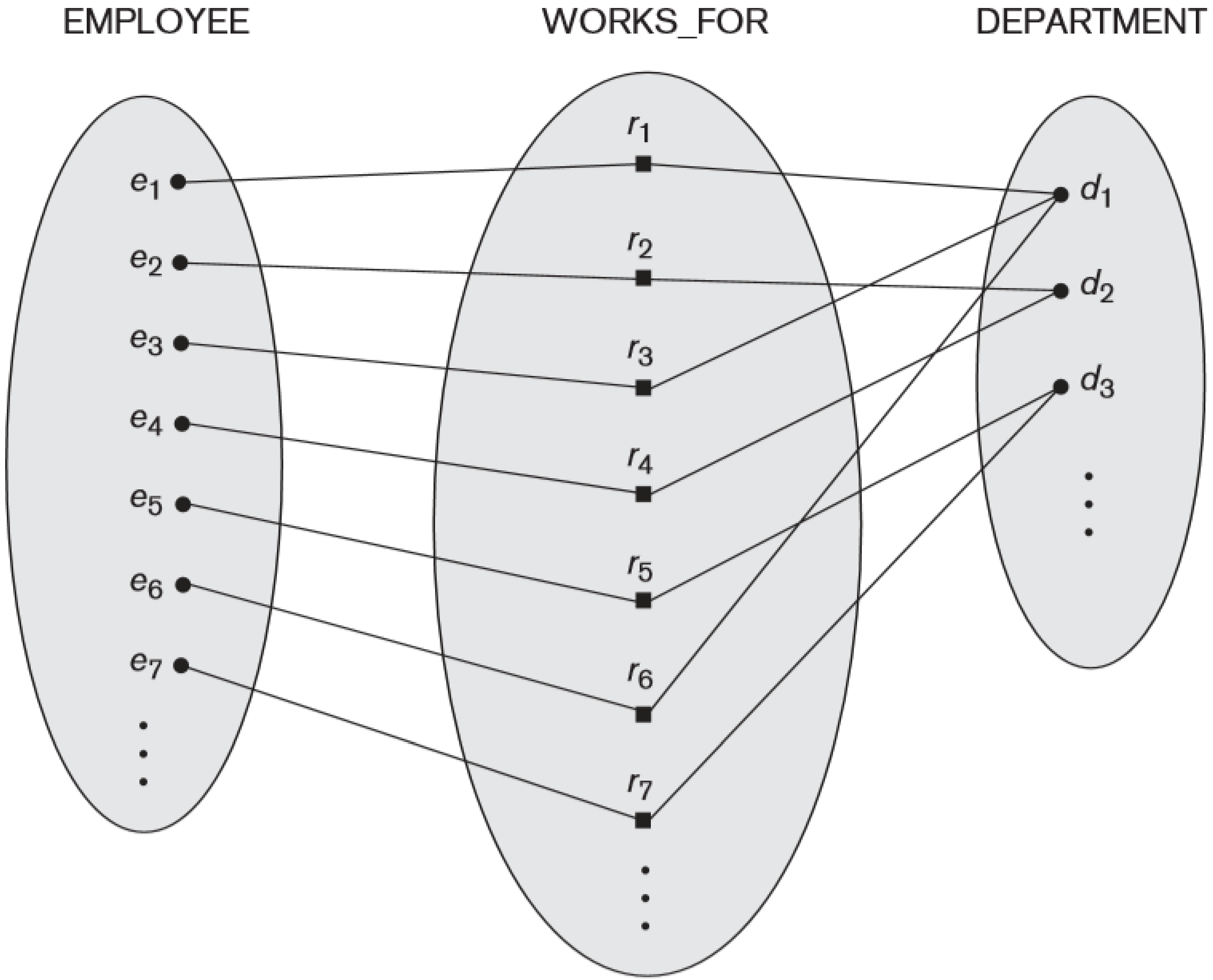
Data Modeling Using the Entity-Relationship (ER) Model

Part 2

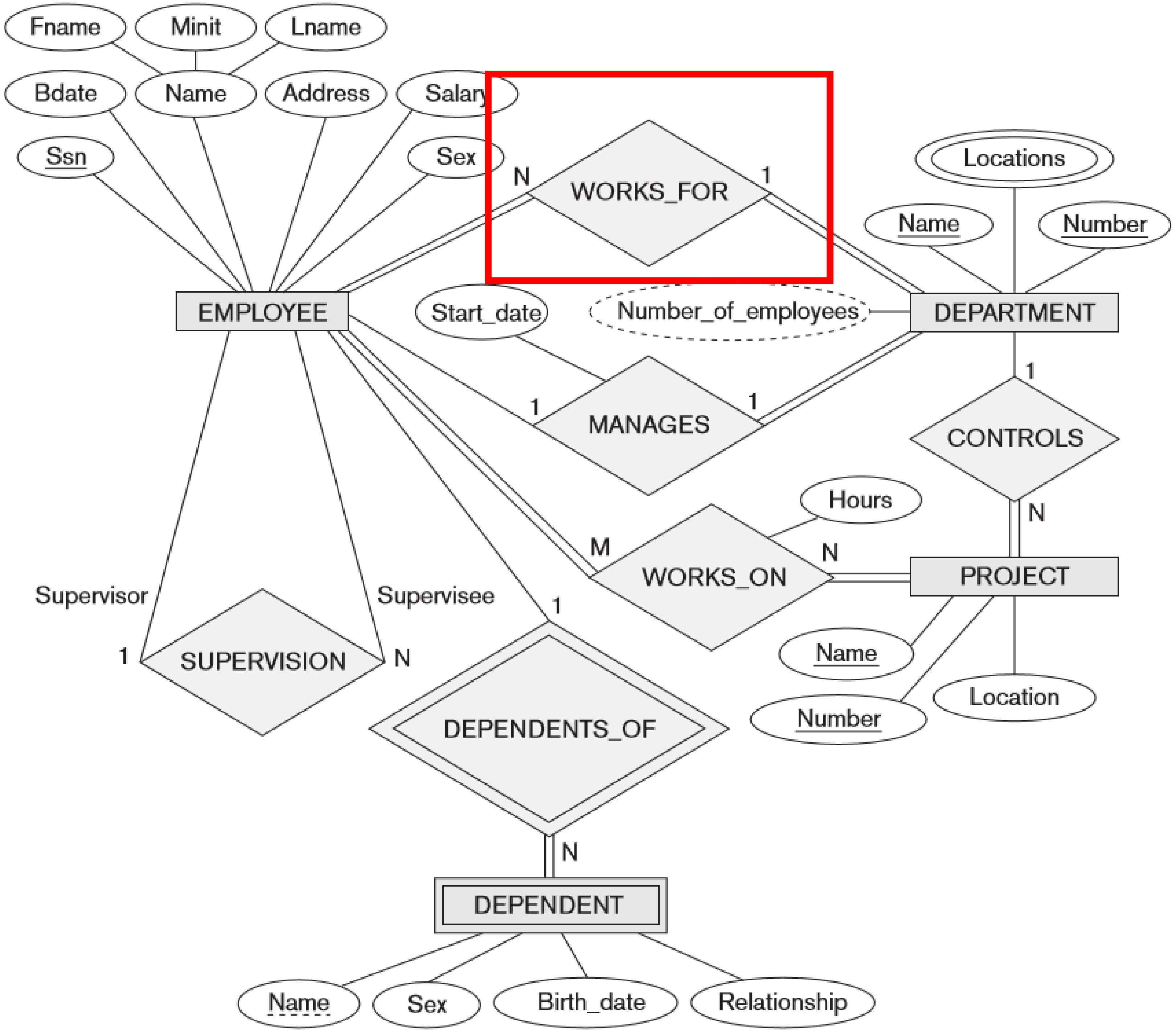
Relationship Types, Relationship Sets, Roles, and Structural Constraints

Relationship Types, Sets, and Instances

- A **relationship type** R among n entity types E_1, E_2, ..., E_n defines a set of associations—or a **relationship set**—among entities from these entity types.
- As for the case of entity types and entity sets, a relationship type and its corresponding relationship set are customarily referred to by the *same name*, R.
- Mathematically, the relationship set R is a set of relationship instances r_i, where each r_i associates n individual entities (e_1, e_2,..., e_n),and each entity e_j in r_i is a member of entity set E_j, $1 \leq j \leq n$.
- Hence, a relationship set is a subset of the Cartesian product of the entity sets $E_1 \times E_2 \times \ldots \times E_n$.
- Each of the entity types E_1, E_2, ..., E_n is said to participate in the relationship type R; similarly, each of the individual entities e_1, e_2, ..., e_n is said to participate in the relationship instance $r_i = (e_1, e_2, \ldots, e_n)$.

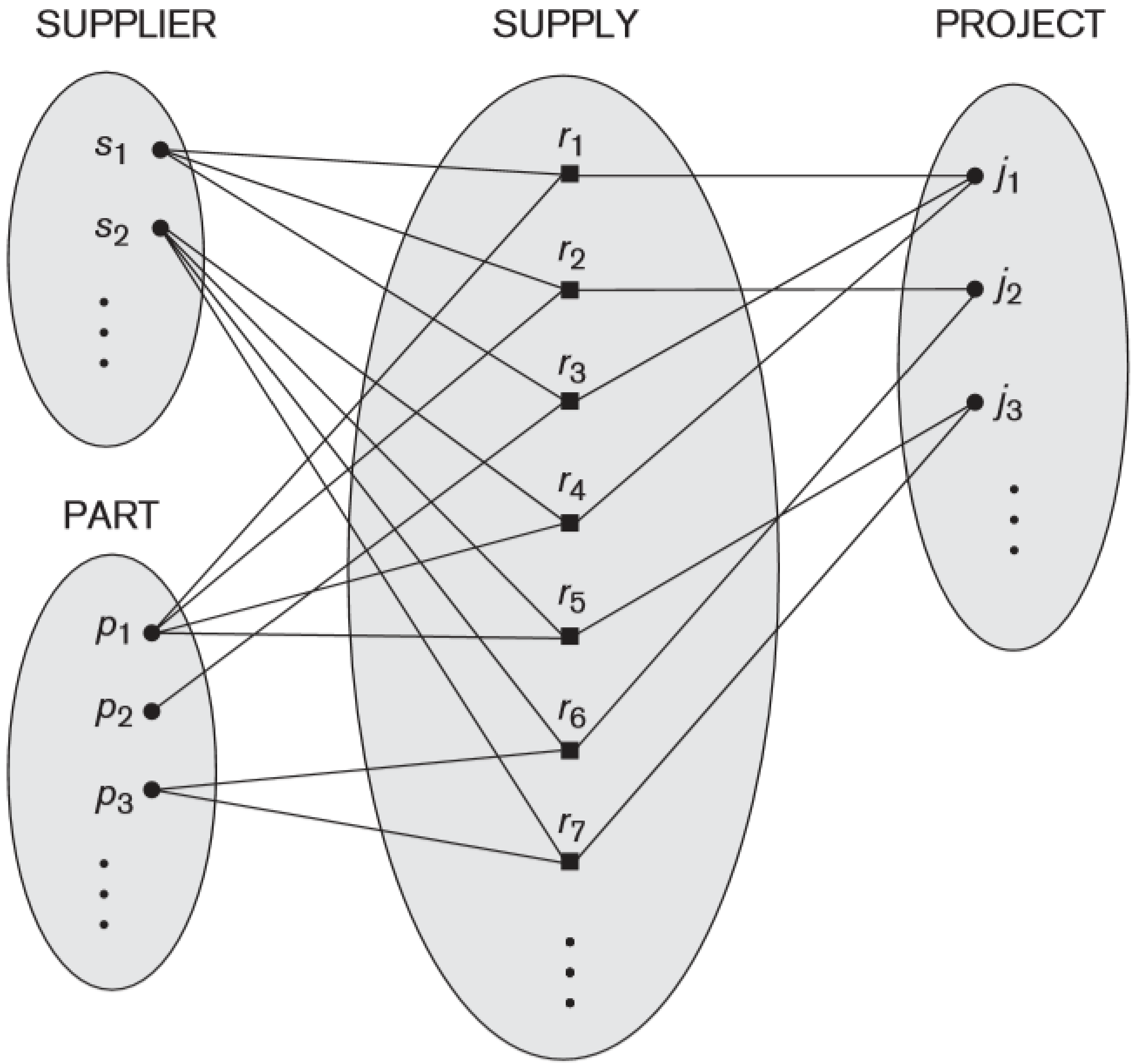

EMPLOYEE
WORKS_FOR
DEPARTMENT
e_1
e_2
e_3
e_4
e_5
e_6
e_7
r_1
r_2
r_3
r_4
r_5
r_6
r_7
d_1
d_2
d_3

- In ER diagrams, relationship types are displayed as diamond-shaped boxes, which are connected by straight lines to the rectangular boxes representing the participating entity types.
- The relationship name is displayed in the diamond-shaped box.


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

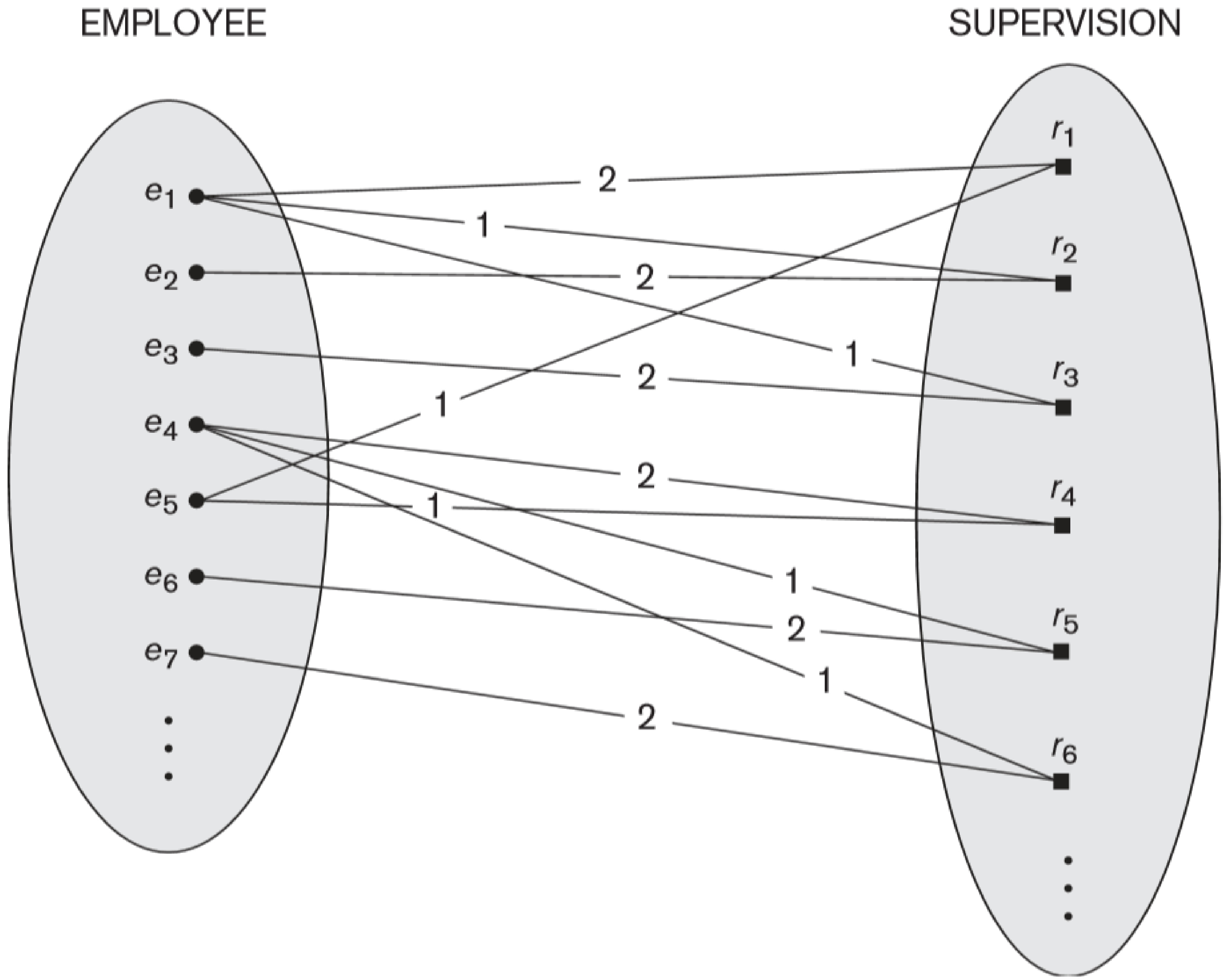
Relationship Degree, Role Names, and Recursive Relationships

- The **degree** of a relationship type is the number of participating entity types.
- Hence, the `WORKS_FOR` relationship is of degree two.
- A relationship type of degree two is called **binary**, and one of degree three is called **ternary**.


SUPPLIER
SUPPLY
PROJECT
PART
s_1
s_2
r_1
r_2
r_3
r_4
r_5
r_6
r_7
j_1
j_2
j_3
p_1
p_2
p_3

- Each entity type that participates in a relationship type plays a particular role in the relationship.
- The **role name** signifies the role that a participating entity from the entity type plays in each relationship instance, and helps to explain what the relationship means.

- Role names are not technically necessary in relationship types where all the participating entity types are distinct, since each participating entity type name can be used as the role name.
- However, in some cases the *same* entity type participates more than once in a relationship type in *different roles*.
- In such cases the role name becomes essential for distinguishing the meaning of the role that each participating entity plays.



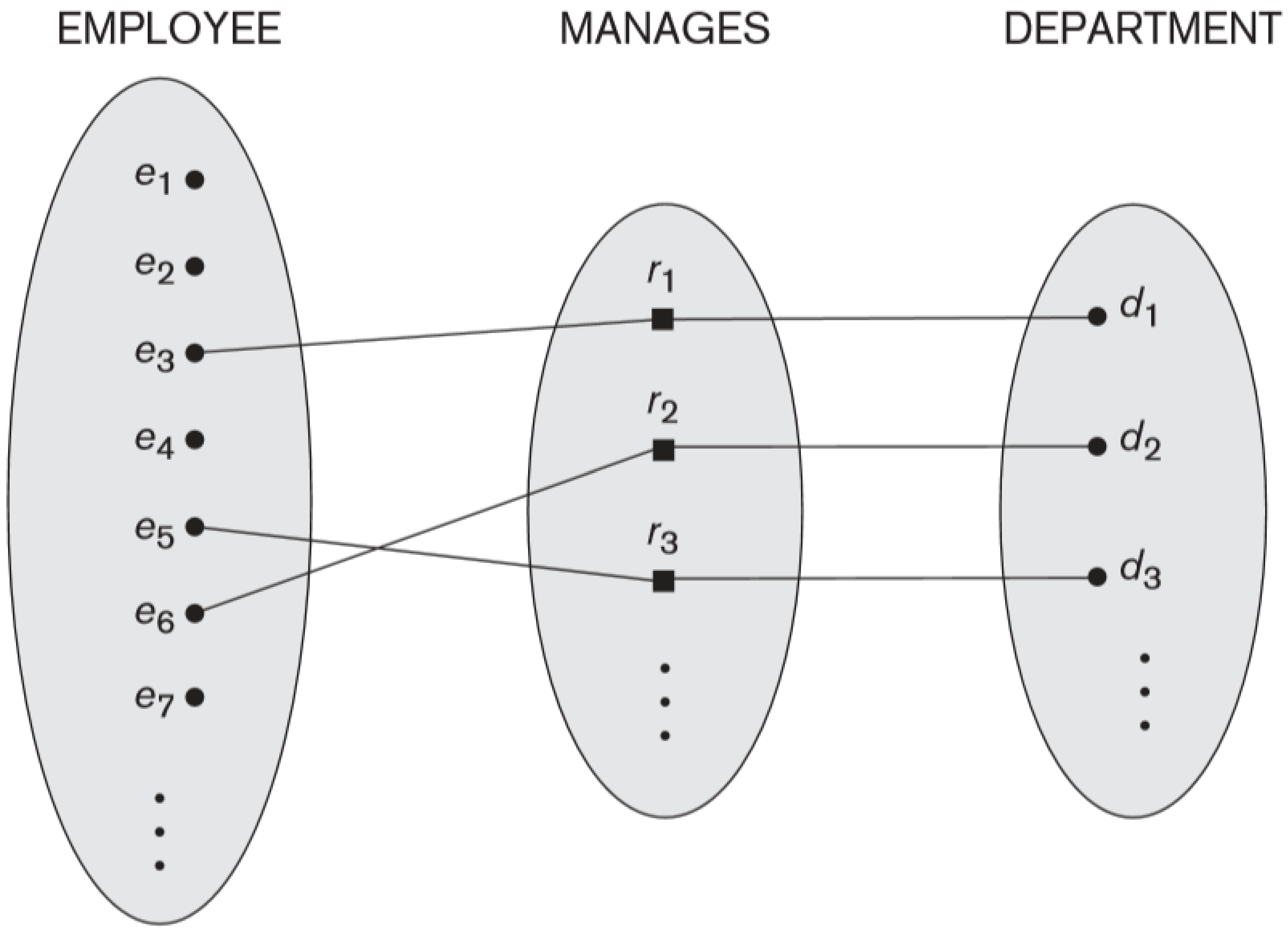
A recursive relationship SUPERVISION between EMPLOYEE in the *supervisor* role (1) and EMPLOYEE in the *subordinate* role (2).

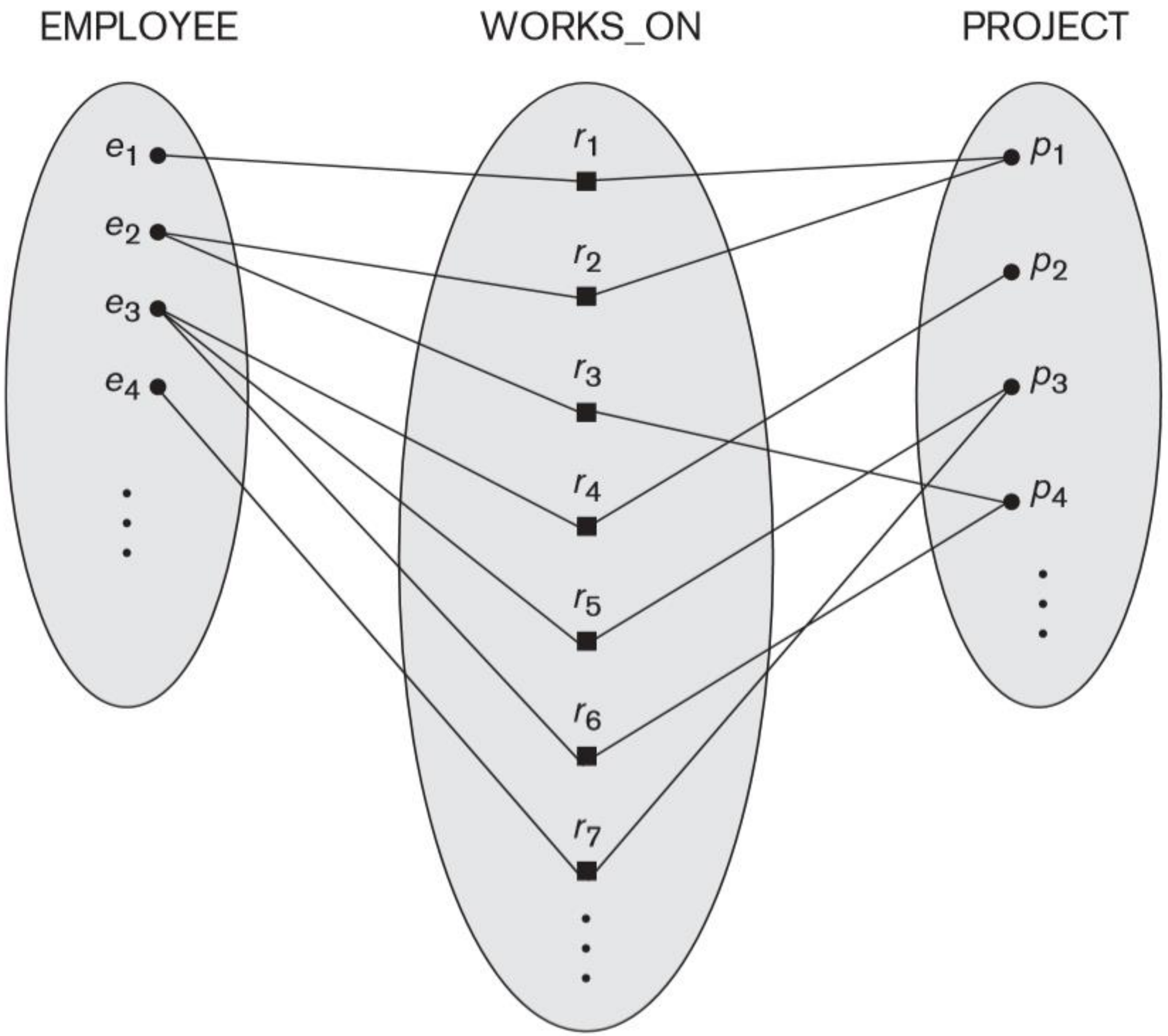
Constraints on Binary Relationship Types

- The **cardinality ratio** for a binary relationship specifies the *maximum* number of relationship instances that an entity can participate in.
- For example, in the `WORKS_FOR` binary relationship type, `DEPARTMENT`:`EMPLOYEE` is of cardinality ratio `1`:`N`, meaning that each department can be related to (that is, employs) any number of employees, but an employee can be related to (work for) only one department.

- The possible cardinality ratios for binary relationship types are `1:1`, `1:N`, `N:1`, and `M:N`.

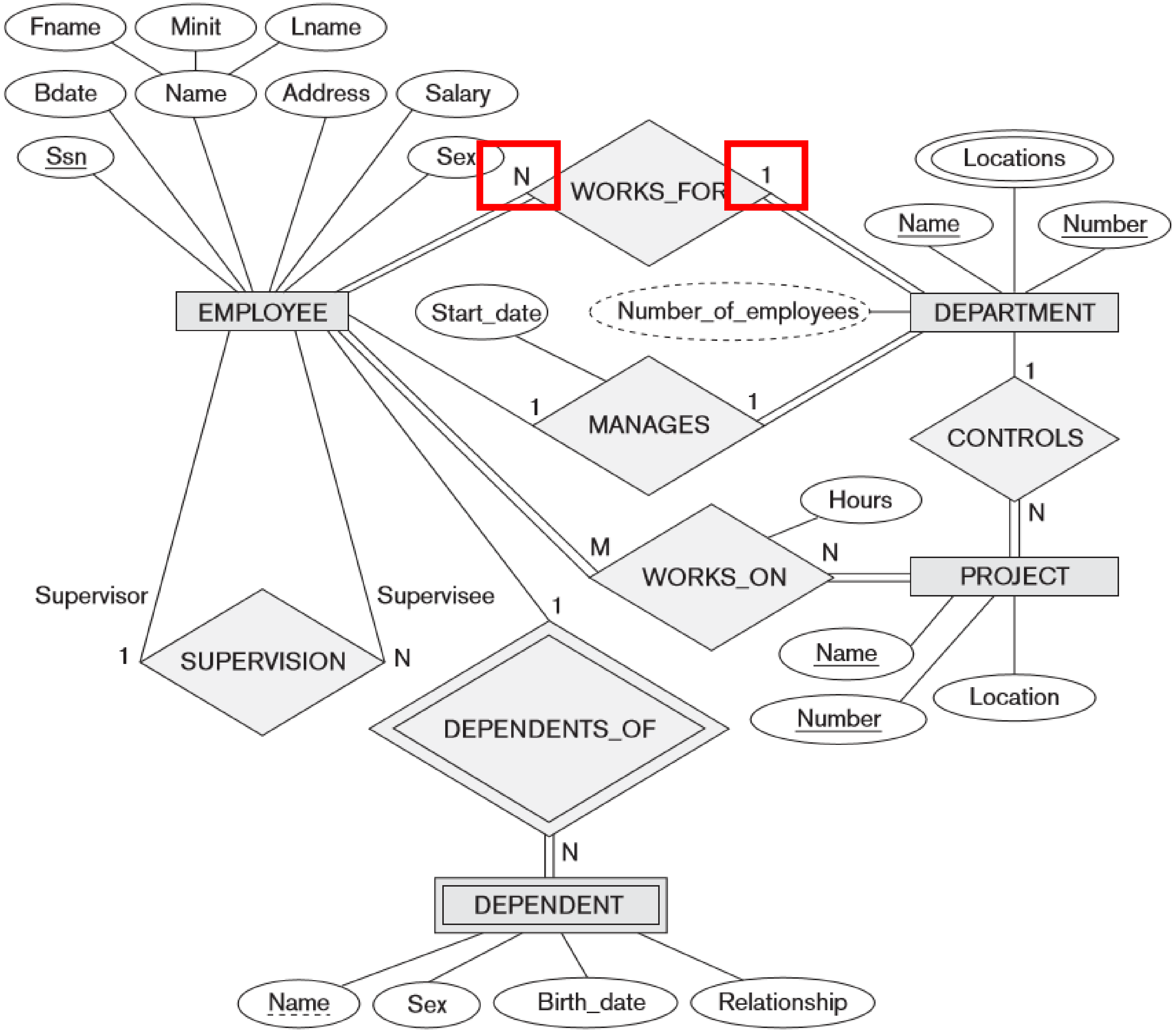
A 1:1 relationship, MANAGES.





An M:N relationship,
WORKS_ON.

- Cardinality ratios for binary relationships are represented on ER diagrams by displaying `1`, `M`, and `N` on the diamonds.


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

- 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 the **minimum cardinality constraint**.

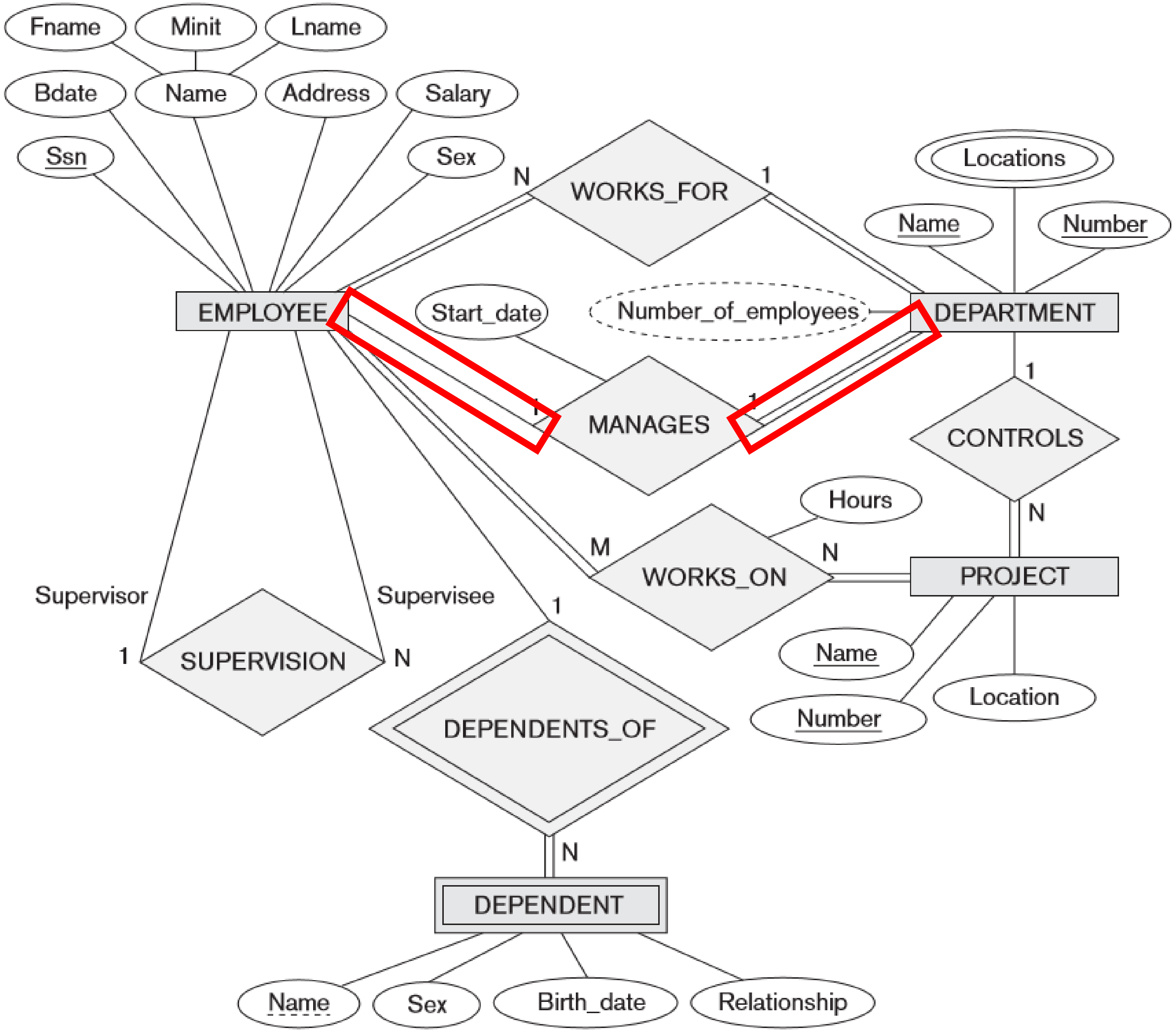
- There are two types of participation constraints—*total* and *partial*.

- If a company policy states that *every* employee must work for a department, then an employee entity can exist only if it participates in at least one `WORKS_FOR` relationship instance.
- Thus, the participation of `EMPLOYEE` in `WORKS_FOR` is called **total participation**, meaning that every entity in the *total set* of employee entities must be related to a department entity via `WORKS_FOR`.
- Total participation is also called **existence dependency**.

- We do not expect every employee to manage a department, so the participation of `EMPLOYEE` in the `MANAGES` relationship type is **partial**, meaning that *some* or *part of the set* of employee entities are related to some department entity via `MANAGES`, but not necessarily all.

- We will refer to the cardinality ratio and participation constraints, taken together, as the **structural constraints** of a relationship type.

- In ER diagrams, total participation (or existence dependency) is displayed as a *double line* connecting the participating entity type to the relationship, whereas partial participation is represented by a *single line*.


Fname
Minit
Lname
Bdate
Name
Address
Salary
Ssn
Sex
N
WORKS_FOR
1
Locations
Name
Number
EMPLOYEE
Start_date
Number_of_employees
DEPARTMENT
1
1
MANAGES
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

Attributes of Relationship Types

- Relationship types can also have attributes, similar to those of entity types.
- For example, to record the number of hours per week that an employee works on a particular project, we can include an attribute `Hours` for the `WORKS_ON` relationship type.
- Another example is to include the date on which a manager started managing a department via an attribute `Start_date` for the `MANAGES` relationship type.

- Notice that attributes of `1:1` or `1:N` relationship types can be migrated to one of the participating entity types.
- For example, the `Start_date` attribute for the `MANAGES` relationship can be an attribute of either `EMPLOYEE` or `DEPARTMENT`, although conceptually it belongs to `MANAGES`.
- This is because `MANAGES` is a `1:1` relationship, so every department or employee entity participates in at most one relationship instance.

- For a `1:N` relationship type, a relationship attribute can be migrated only to the entity type on the `N`-side of the relationship.
- For example, if the `WORKS_FOR` relationship also has an attribute `Start_date` that indicates when an employee started working for a department, this attribute can be included as an attribute of `EMPLOYEE`.
- This is because each employee works for only one department, and hence participates in at most one relationship instance in `WORKS_FOR`.

- In both `1:1` and `1:N` relationship types, the decision where to place a relationship attribute—as a relationship type attribute or as an attribute of a participating entity type—is determined subjectively by the schema designer.

- For `M:N` relationship types, 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*.
- An example is the `Hours` attribute of the `M:N` relationship `WORKS_ON`; the number of hours per week an employee currently works on a project is determined by an employee project combination and not separately by either entity.