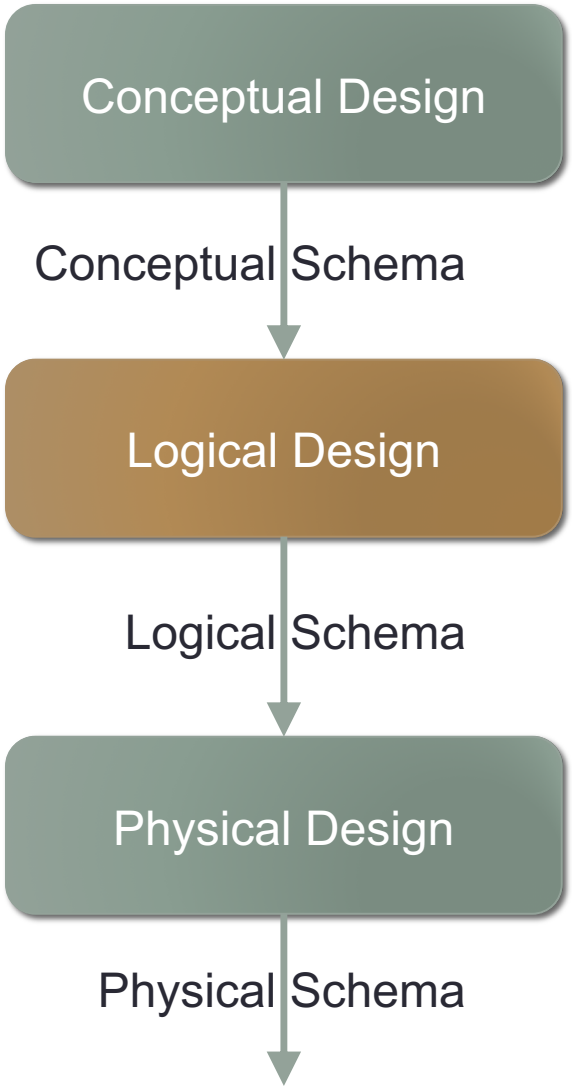
LECTURE 10:
LOGICAL DESIGN
MAPPING ER/EER DIAGRAMS TO RELATIONS

COMP2004J: Databases and Information Systems

Dr. Ruihai Dong (ruihai.dong@ucd.ie)
UCD School of Computer Science
Beijing-Dublin International College

The Phases of Database Design



An ER/EER diagram is a **conceptual schema** of the database.

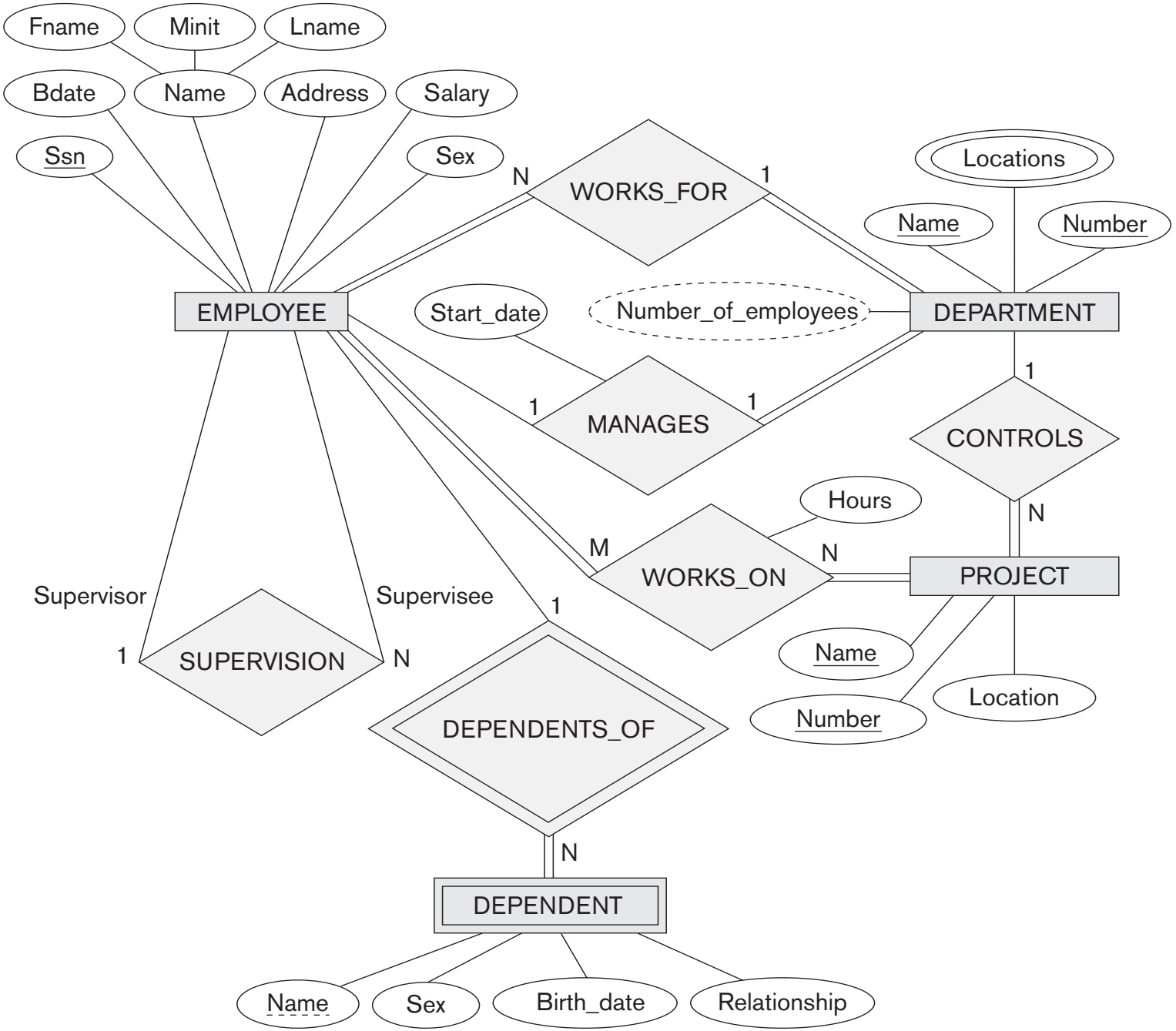
Next, we need to map this to a **logical schema**: i.e. the tables and attributes that will be created to store the data in the database.

Mapping to Relations

- In this lecture, we will look at a method to convert an ER/EER diagram into a logical model (i.e. the relations that will be used to create a database).

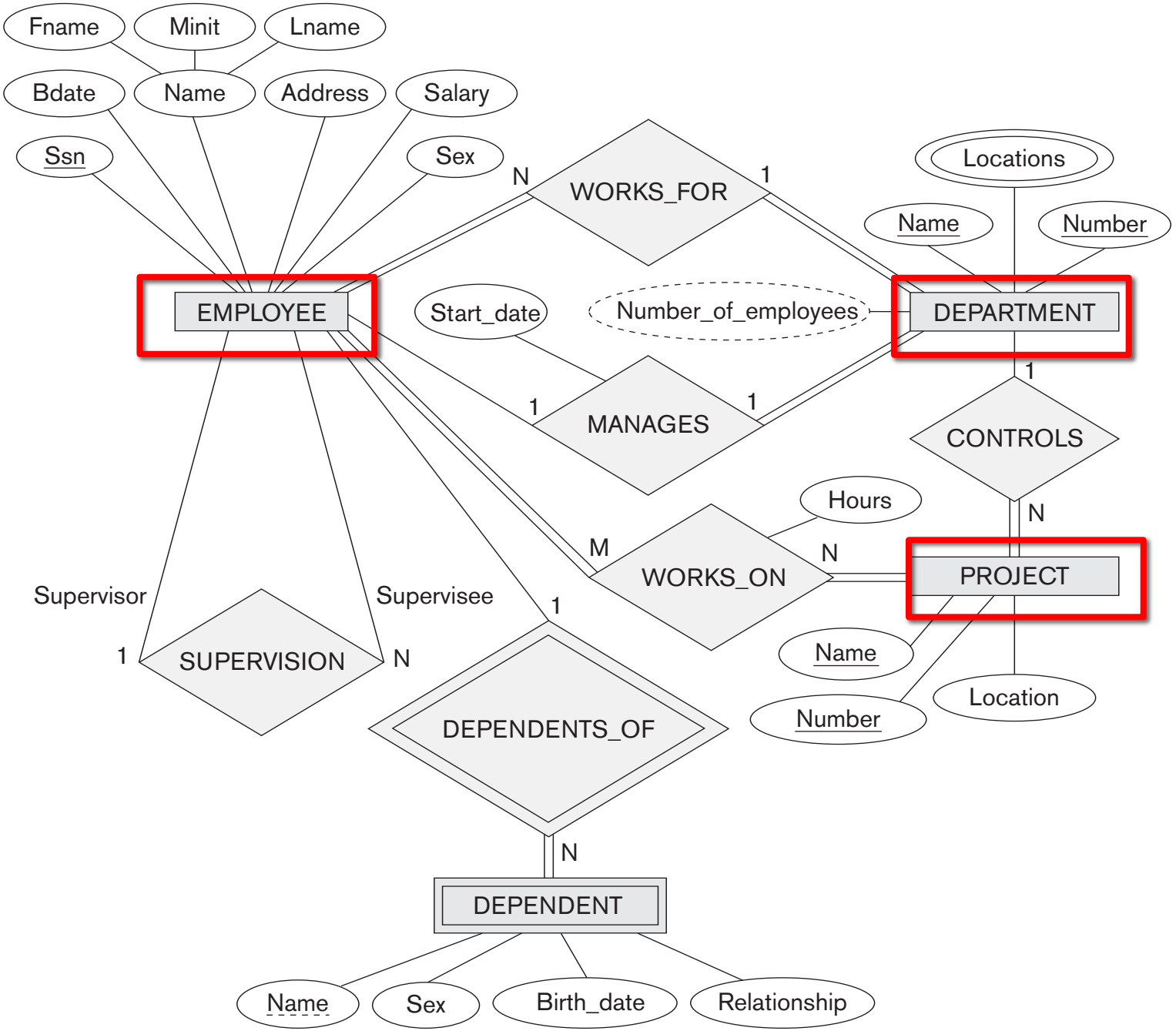
- This follows an 8-step algorithm.
  1. Mapping Regular Entity Types
  2. Mapping Weak Entity Types
  3. Mapping 1:1 Relationships
  4. Mapping 1:N Relationships
  5. Mapping M:N Relationships
  6. Mapping Multivalued Attributes
  7. Mapping *N*-ary Relationships
  8. Mapping supertypes/subtypes

- Chapter 9 of Elmasri & Navathe Book


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

1. Mapping Regular Entity Types

- We begin with regular (strong) entity types.
- Create a **relation** for each strong entity.
- Include all simple attributes:
  - For any composite attributes, only include its component attributes.
  - For multi-valued attributes, leave these out until later.
- Select a primary key (possibly a composite primary key consisting of multiple attributes).
- We do not consider foreign keys and relationships until later in the process.


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

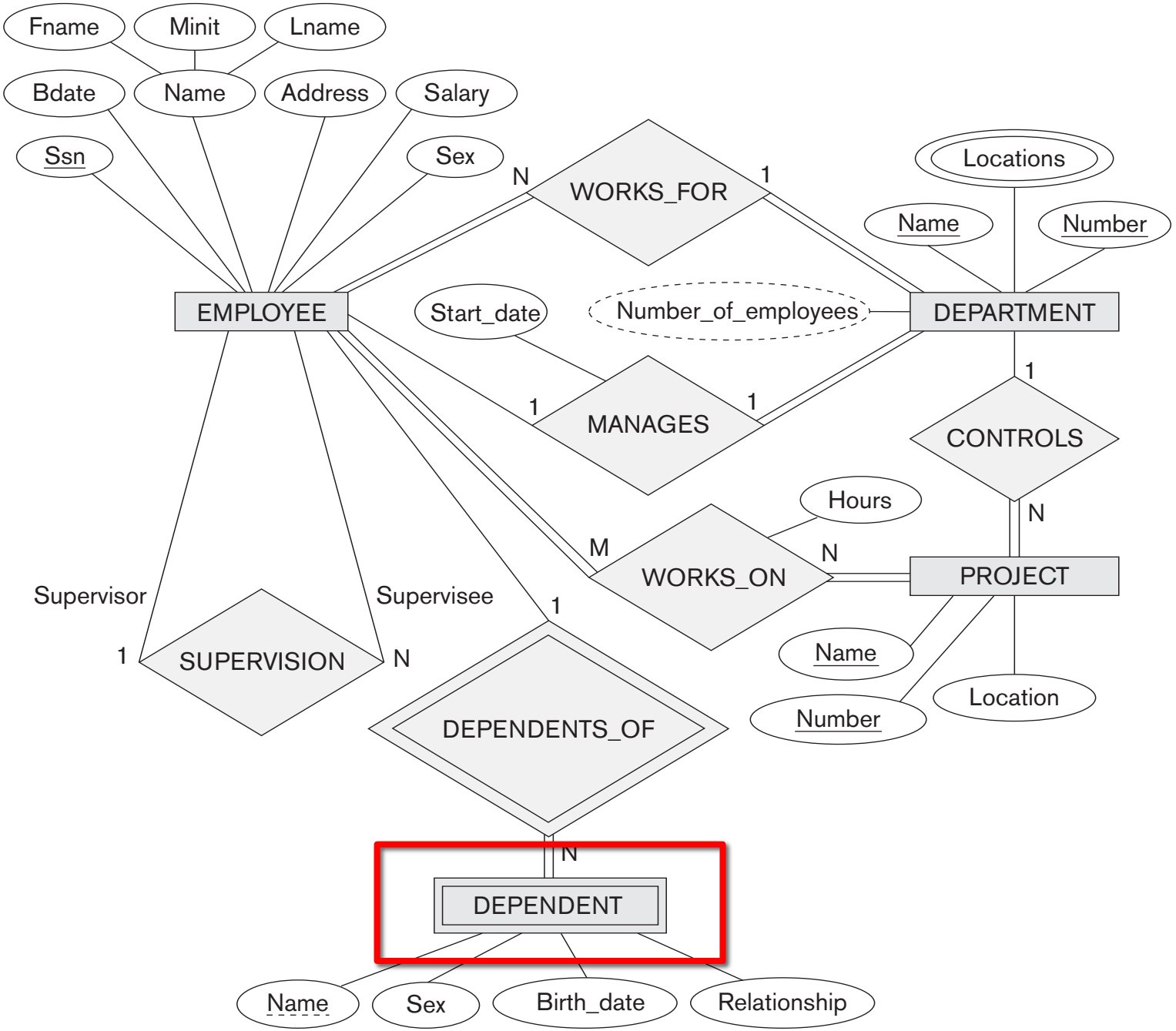
1. Mapping Regular Entity Types

Employee	Fname	Minit	Lname	Ssn	birth_date	address	sex	salary
Department	Dname	Dnumber						
Project	Pname	Pnumber	Plocation					

- **Notes:**
  - Composite attribute “Name” from EMPLOYEE is not included: only its component attributes: Fname (first name), Minit (middle initial) and Lname (last name).
  - For DEPARTMENT, either Dname or Dnumber would be OK to choose as a primary key. We choose only one (primary keys are underlined).
  - We do not yet include the multi-valued attribute “Locations” for DEPARTMENT (we will do that later).
  - No relationships/foreign keys are included yet.

2. Mapping Weak Entity Types

- For any weak entity types, create a relation.
- As with step 1, include any simple attributes it has (including component attributes of composite attributes).
- Include the primary key of the owner attribute as a foreign key.
  - The primary key of this relation is a combination of this key and the weak entity's partial key.
- If the owner is also a weak entity type, that should be mapped first (so that we know its primary key).
- Usually, we use the CASCADE option for referential integrity, because the weak entity cannot exist without its owner.


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
N
WORKS_ON
PROJECT
Supervisor
Supervisee
1
1
SUPERVISION
N
Name
Location
Number
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
Birth_date
Relationship

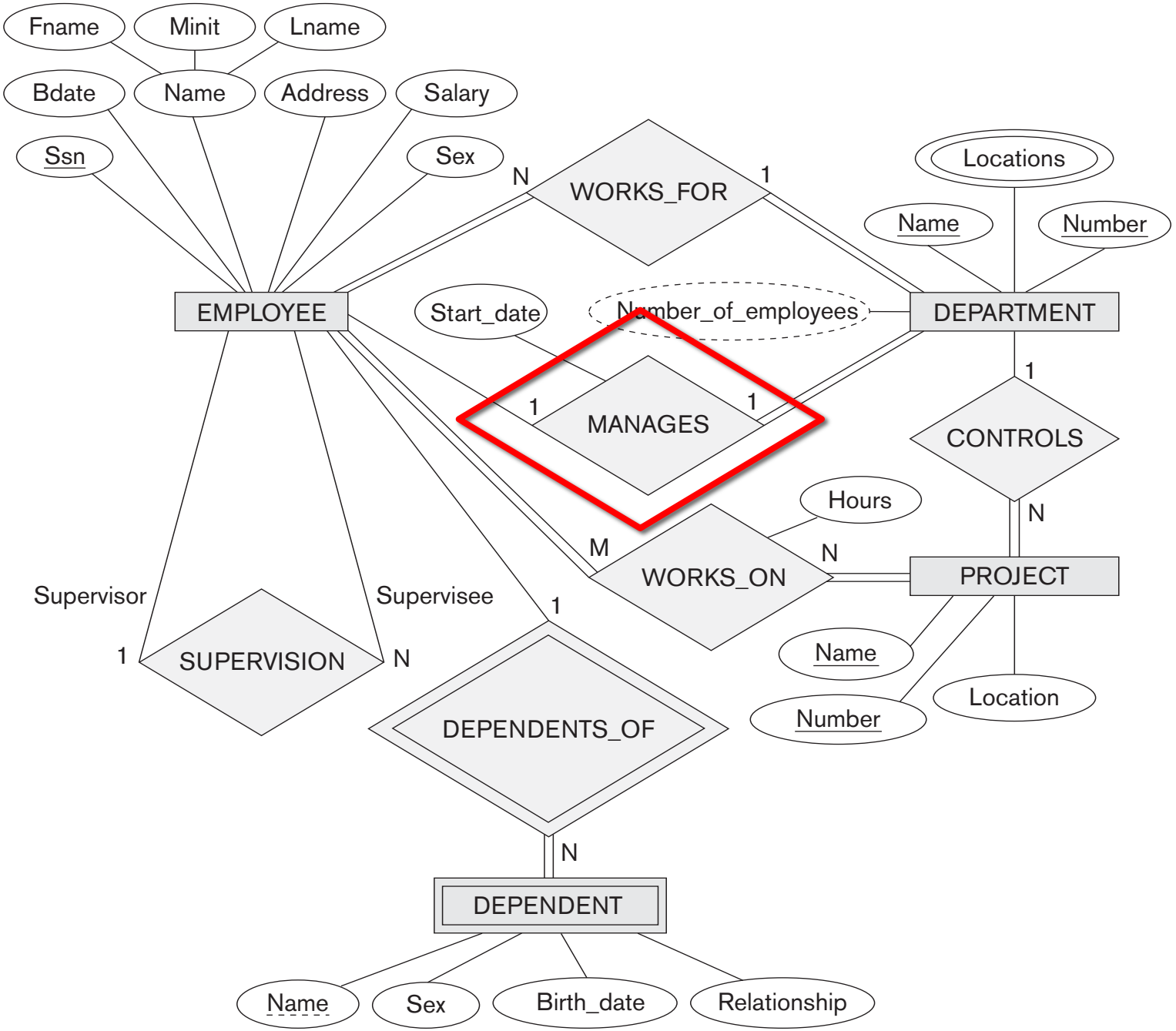
2. Mapping Weak Entity Types

Employee	Fname	Minit	Lname	Ssn	birth_date	address	sex	salary
Department	Dname	Dnumber						
Project	Pname	Pnumber	Plocation					
Dependent	*Essn*	Dependent_name	Sex	Bdate	Relationship			

- **Notes:**
  - Essn is a foreign key to the ”Ssn” attribute in EMPLOYEE (and so we show it in italics).
  - The primary key of DEPENDENT is the combination of its owner’s primary key (Essn) and its own partial key (Dependent_name).

3. Mapping 1:1 Relationships

- To map a 1:1 relationship, there are 3 approaches to choose from. The first is most common:
  1. **Foreign key approach:** Choose one of the entity types, and include the primary key of the other as a foreign key.
     - It is best to include the foreign key in an entity type that has mandatory participation in the relationship (avoids storing many NULL values).
     - If the relationship has any simple attributes, they can be included in the same relation.
  2. **Merge the relations:** if both entity types have mandatory participation in the relationship, they can be merged into one relation.
  3. **Relationship relation:** Create a separate table to represent the relationship. This is rare for 1:1 relationships but very common for M:N relationships, as we will see later.


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
N
WORKS_ON
PROJECT
Supervisor
Supervisee
1
1
SUPERVISION
N
Name
Location
Number
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
Birth_date
Relationship

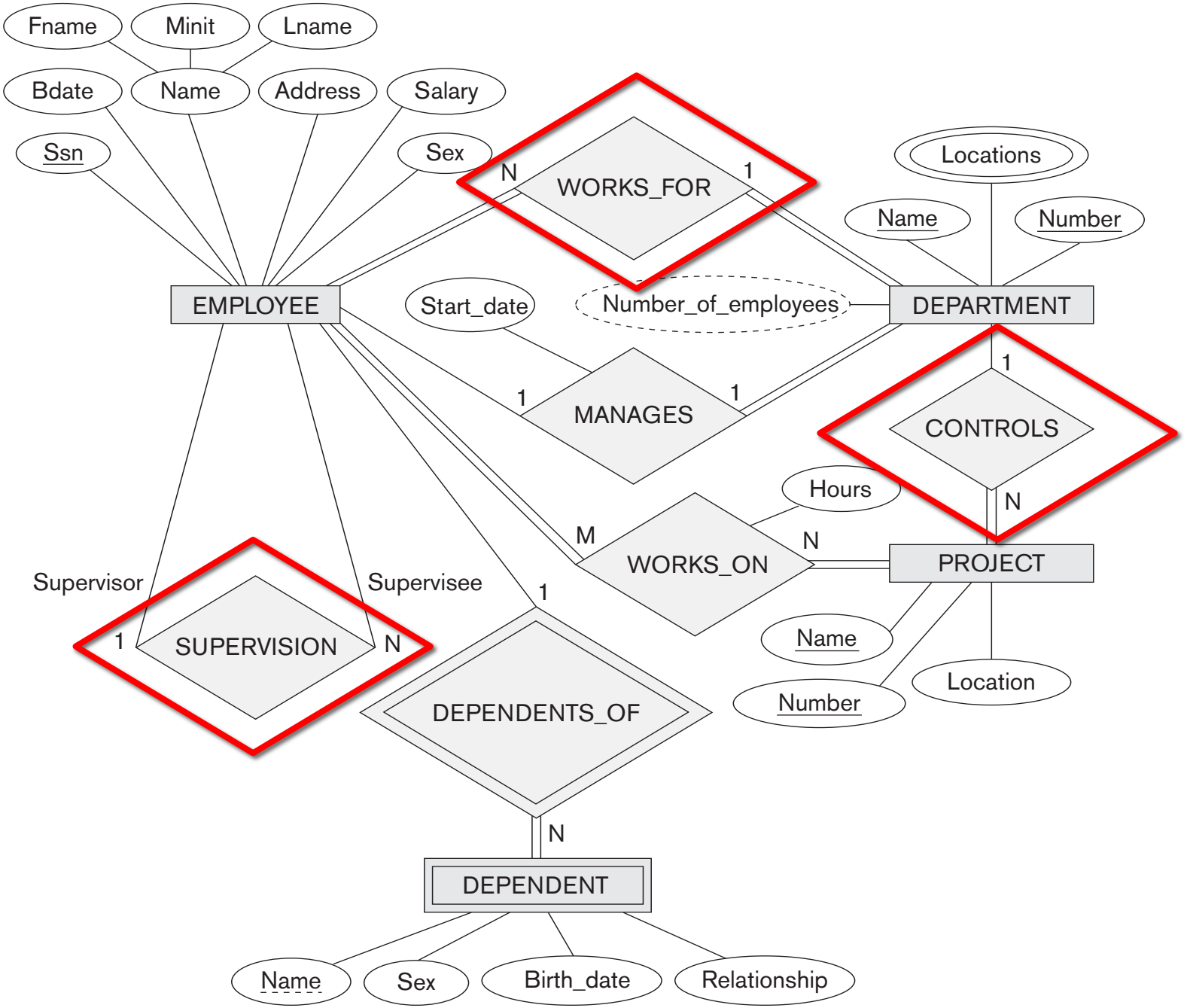
3. Mapping 1:1 Relationships

Employee	Fname	Minit	Lname	Ssn	birth_date	address	sex	salary
Department	Dname	Dnumber	*Mgr_ssn*	*Mgr_start_date*				
Project	Pname	Pnumber	Plocation					
Dependent	*Essn*	Dependent_name	Sex	Bdate	Relationship			

- **Notes:**
  - DEPARTMENT has mandatory participation in the MANAGES relationship (i.e. every department must have a manager).
  - Therefore it makes most sense to include a foreign key in DEPARTMENT that refers to the primary key of EMPLOYEE (i.e. ”Mgr_ssn” in DEPARTMENT refers to “Ssn” in EMPLOYEE).
  - If we instead included a foreign key in EMPLOYEE that referred to DEPARTMENT, then this would be NULL for all the employees that are not managers: because we wish to avoid too many NULL values, it’s better to include the foreign key in DEPARTMENT.
  - The “start_date” attribute of the MANAGES relationship can also be included as an attribute.
  - Because the participation is mandatory, we would use a NOT NULL constraint on the foreign key when we turn this into a database.

4. Mapping 1:N Relationships

- In the entity type on the *N* side, include a foreign key that refers to the primary key of the attribute on the *1* side.
- If the relationship has simple attributes, these can also be included in the same relation.
- Alternatively, we could use a separate table to store the relationship, but again this is rare for 1:N relationships.
  - It might be more useful if only a few tuples participate in the relationship, to avoid many NULL values.


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

4. Mapping 1:N Relationships

Employee	Fname	Minit	Lname	Ssn	birth_date	address	sex	salary	*Dno*	*Super_ssn*

Department	Dname	Dnumber	*Mgr_ssn*	*Mgr_start_date*

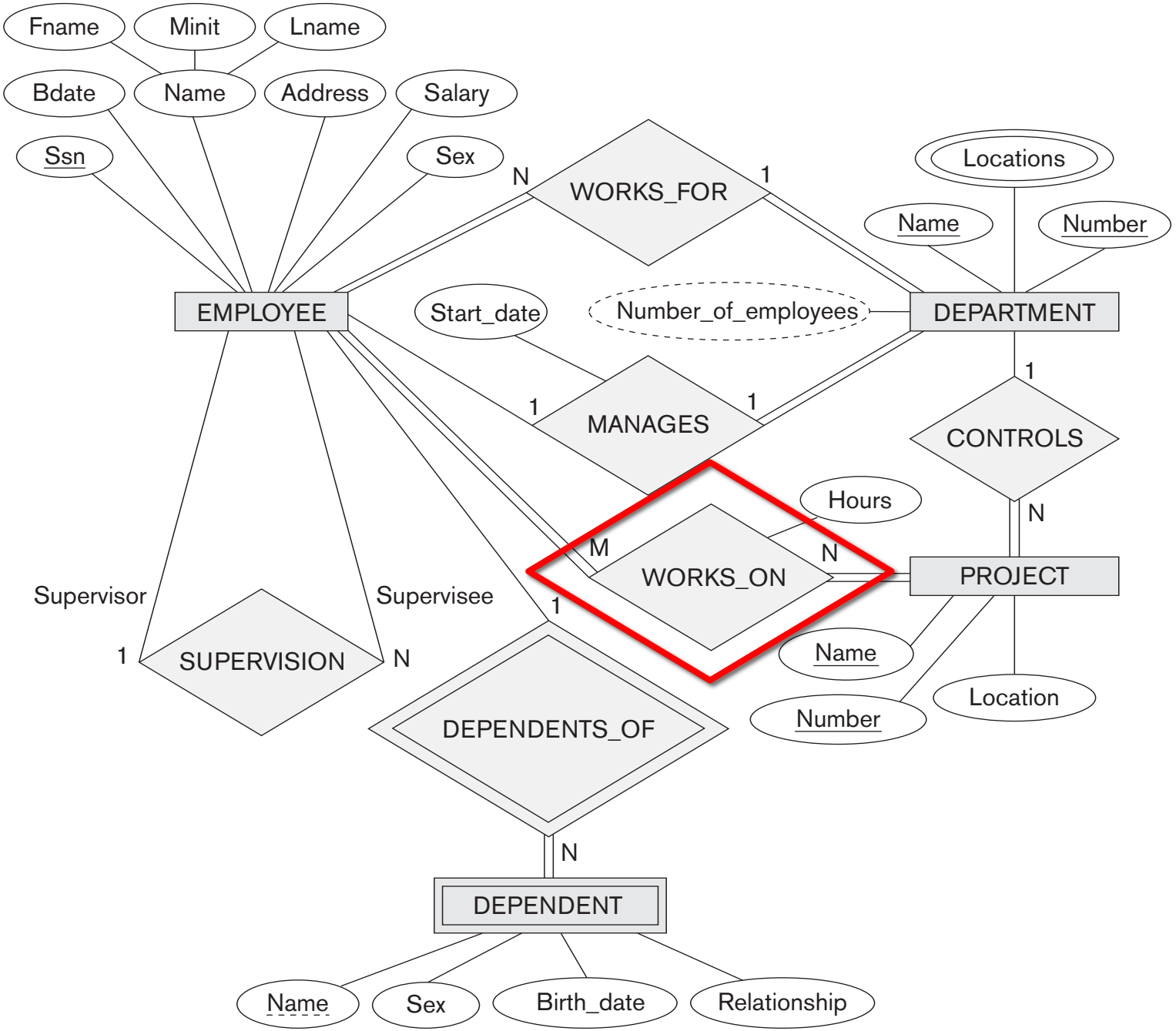
Project	Pname	Pnumber	Plocation	*Dnum*

Dependent	*Essn*	Dependent_name	Sex	Bdate	Relationship

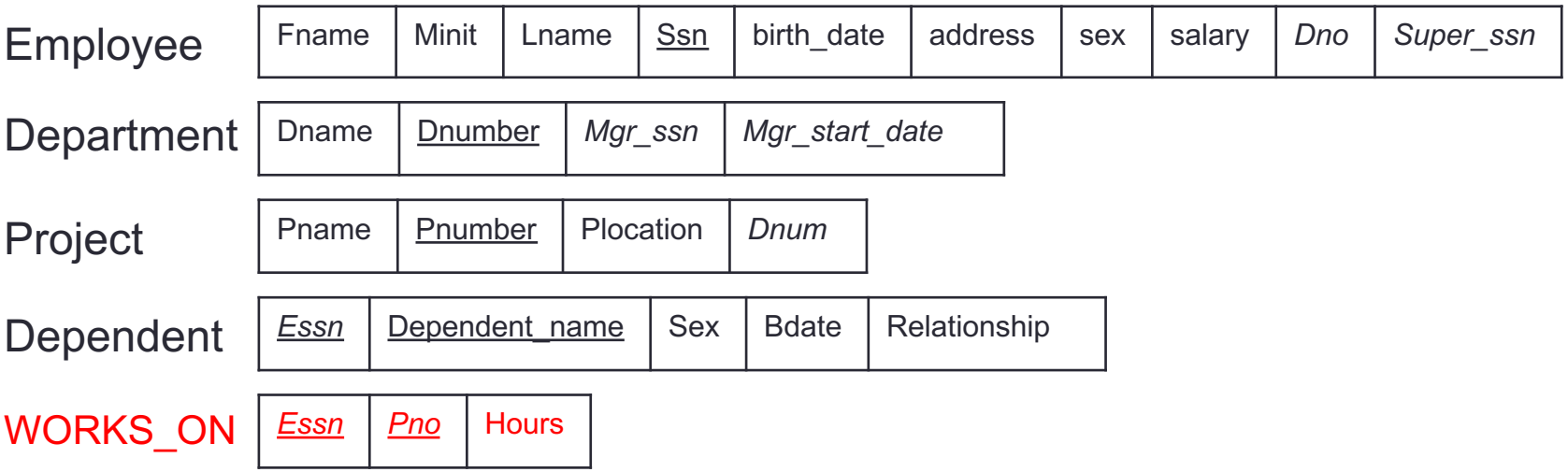
- **Notes:**
  - In the EMPLOYEE relation, we include the foreign key of DEPARTMENT to show the department they work for (*Dno* should be NOT NULL to show that each employee must work for a department).
  - In the EMPLOYEE relation, we include the “Super_ssn” attribute to be a foreign key that refers to the “Ssn” attribute of each employee’s supervisor (who is also an employee). As this is an optional relationship, it can be NULL for employees that don’t have a supervisor.
  - In the PROJECT relation, we include the foreign key of DEPARTMENT to show which department controls a project. Again, this will be NOT NULL because every project must be controlled by a department.

5. Mapping M:N Relationships

- For a M:N relationship, it is not possible to simply include a foreign key in one of the entity types involved.
  - A foreign key can only store one value, which would mean that each entity can only be related to one other entity. But here, all entities can be linked to many other entities.
- Instead, we must create a new relation that represents the relationship.
  - Include foreign keys to refer to **both** of the entity types involved.
  - The primary key of this table will be the combination of both of these foreign keys.
  - If the relationship has any simple attribute, include them in this new relation.
- We can also use this approach for 1:1 and 1:N relationships, but this is rarely done in practice.


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
N
WORKS_ON
PROJECT
Supervisor
Supervisee
1
1
SUPERVISION
N
Name
Location
Number
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
Birth_date
Relationship

5. Mapping M:N Relationships



- **Notes:**
  - The WORKS_ON relationship involves an EMPLOYEE and a PROJECT, so we include foreign keys that refer to **both** of these.
  - The primary key of WORKS_ON is a combined primary key that includes both of the foreign keys.
  - The attribute “Hours” is also added to this relation.
  - Because WORKS_ON depends on both, we would use a CASCADE option for referential integrity.

6. Mapping Multivalued Attributes

- For each multi-valued attribute, create a new relation to represent it.
  - If the multi-valued attribute is a composite attribute, we include only its simple components.
- This new relation contains:
  - An attribute to store the multi-valued attribute itself (or several attributes if it was a composite attribute).
  - A foreign key that refers to the entity type that the attribute belongs to.
- The primary key of this relation is the combination of the foreign key and the attribute itself.

6. Mapping Multivalued Attributes

Employee	Fname	Minit	Lname	Ssn	birth_date	address	sex	salary	*Dno*	*Super_ssn*

Department	Dname	Dnumber	*Mgr_ssn*	*Mgr_start_date*

Project	Pname	Pnumber	Plocation	*Dnum*

Dependent	*Essn*	Dependent_name	Sex	Bdate	Relationship

Works_on	*Essn*	*Pno*	Hours

Dept_Locations	*Dnumber*	Dlocation

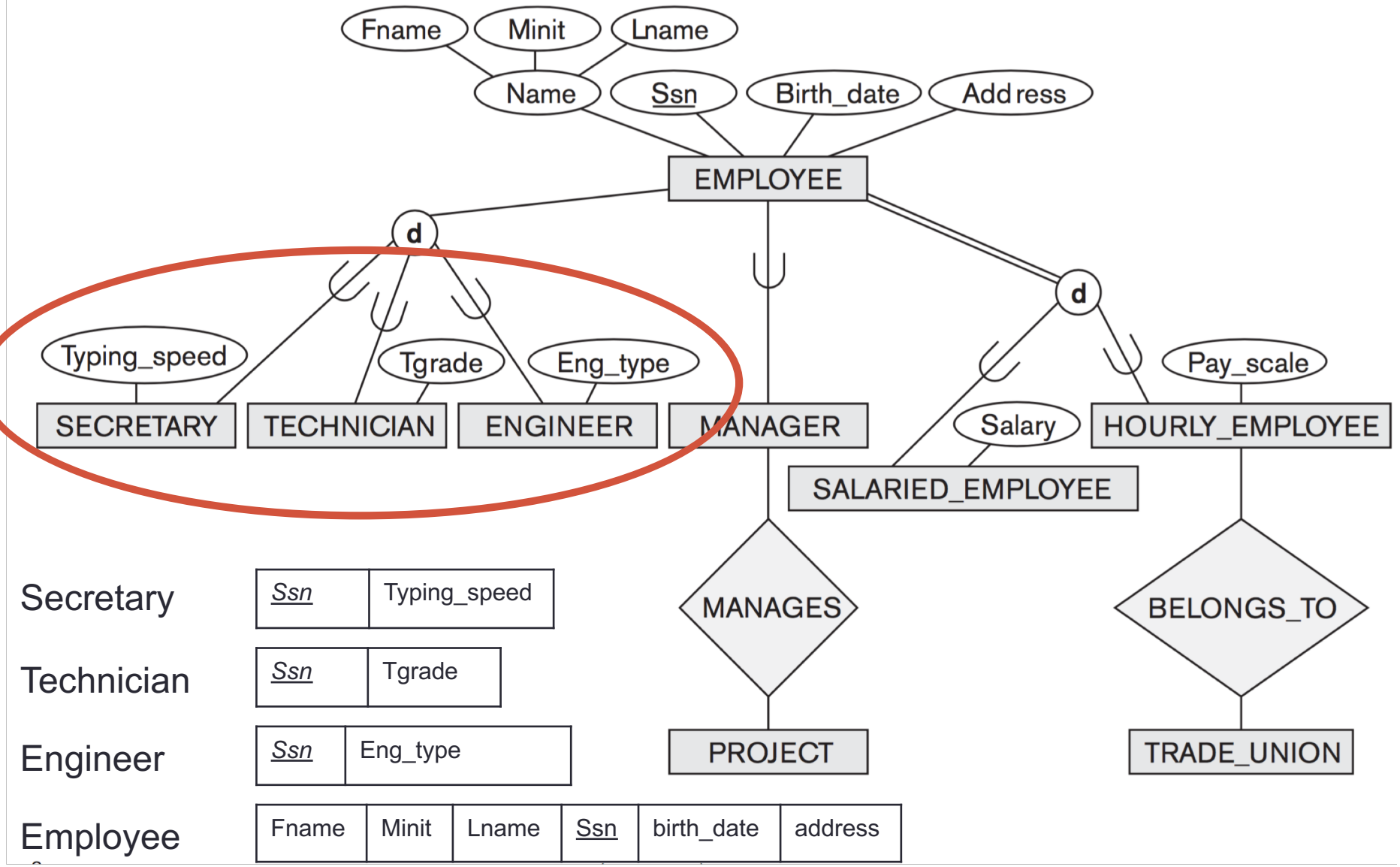
- **Notes:**
  - DEPARTMENT has a multivalued attribute named “locations”, which we represent as a new relation.
  - This includes an attribute “Dlocation” to store the location itself.
  - It also includes a foreign key “Dnumber” that refers to the department this location belongs to.
  - The primary key of DEPT_LOCATIONS is the combination of both of these.
  - Again we would use a CASCADE option because the department location depends on the existence of the department it belongs to.

7. Mapping *N*-ary Relationships

- If your model includes any relationships with more than 2 entity types, these must also be mapped to a new relation.
- Include foreign keys to refer to all the participating entity types.
- Include any attributes that the relationship has.
- The primary key of this relation is usually the combination of the foreign keys.

8. Mapping Supertypes/Subtypes

- There are a number of different options we can use for this mapping.
- The most general is:
  - Create a relation for the supertype.
  - For each subtype, create a relation and include a foreign key that refers to the supertype's primary key.
  - This foreign key attribute becomes the primary key of the subtype.



Secretary	*Ssn*	Typing_speed				
Technician	*Ssn*	Tgrade				
Engineer	*Ssn*	Eng_type				
Employee	Fname	Minit	Lname	*Ssn*	birth_date	address

Summary

- After creating an ER/EER diagram to represent a data model, the next step is to map this to a **relational schema** that can be stored in a database.
- By following the 8-step algorithm above, this can be achieved using a well-defined method.
- At this stage, we can also think about the constraints that we can use in the final database:
  - NOT NULL (when relationship participation is mandatory).
  - Options for ON DELETE and ON UPDATE to maintain referential integrity, depending on the nature of the relationships being modelled.