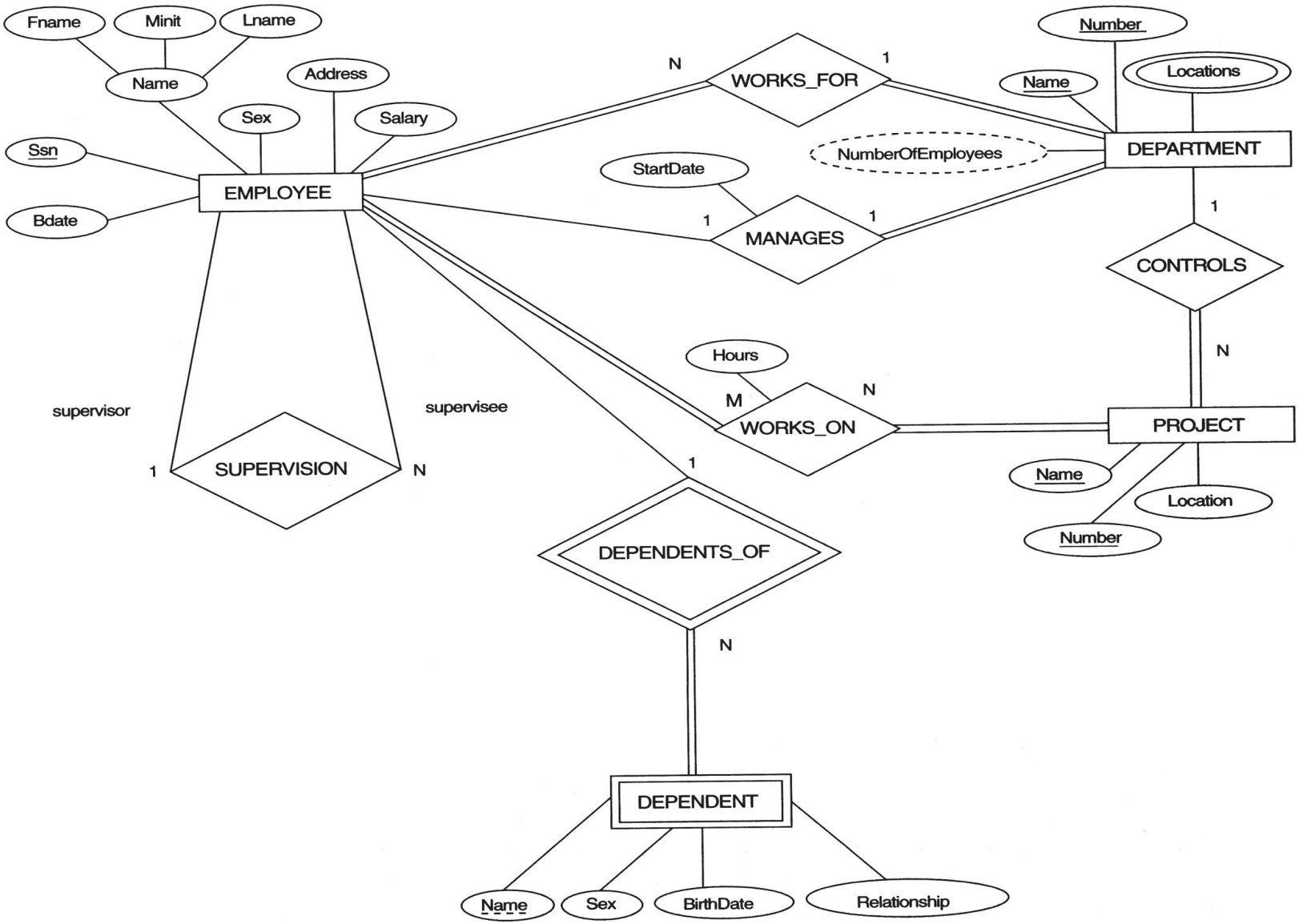
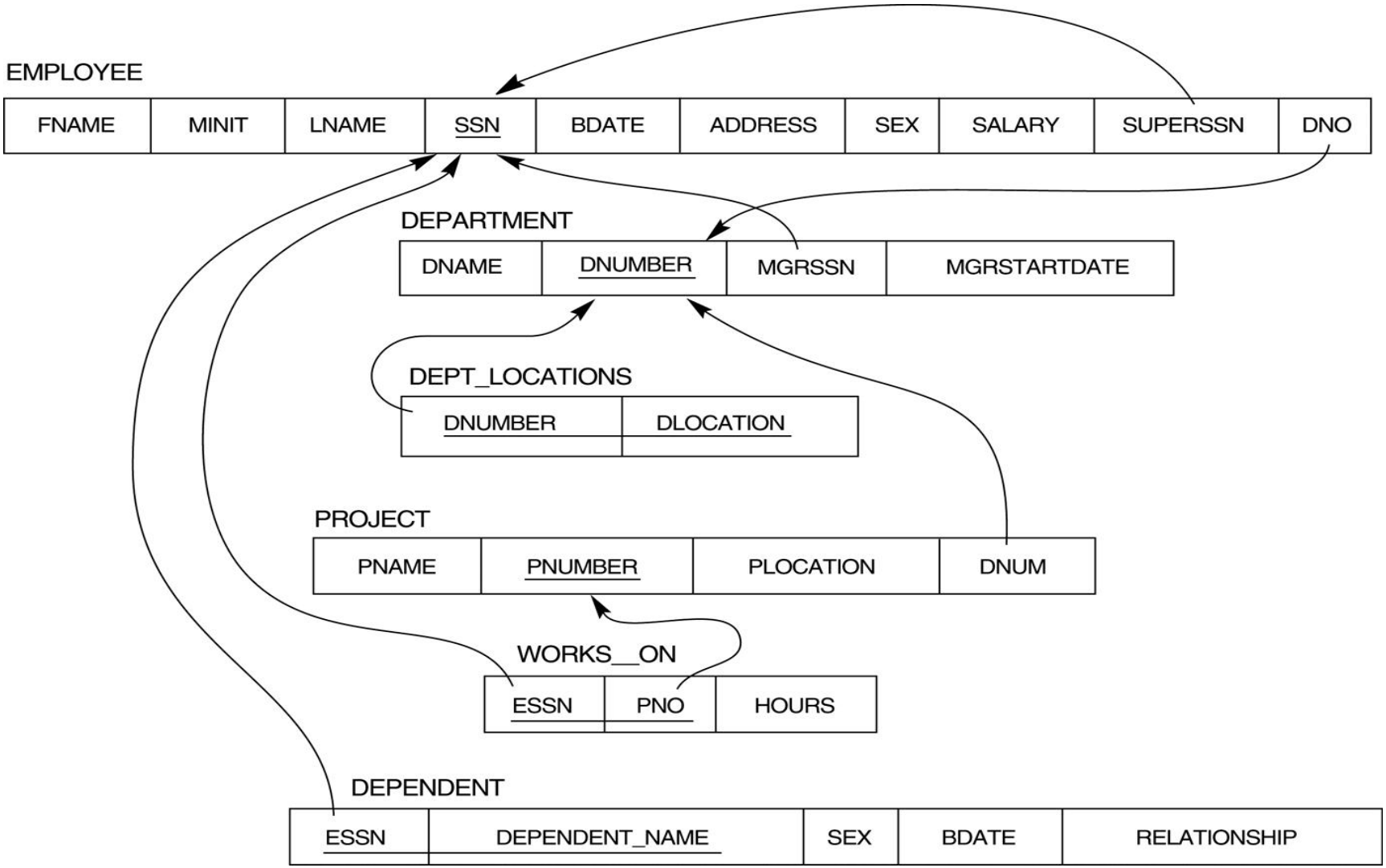
6. Relational Schema Design

ER to Relational Mapping

ER schema


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
M
N
WORKS_ON
PROJECT
Name
Location
Number
supervisor
supervisee
1
SUPERVISION
N
1
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship

Mapping Result : Relational schema



What to Map?

- (Regular) Entity Types
- Weak Entity Types
- Binary 1 : M Relationships
- Binary M : N Relationships
- Binary 1 : 1 Relationships
- Recursive Relationships
- Multi-Valued Attributes
- Ternary Relationships
- Superclass/Subclass : IS-A Relationship

Mapping Guidelines

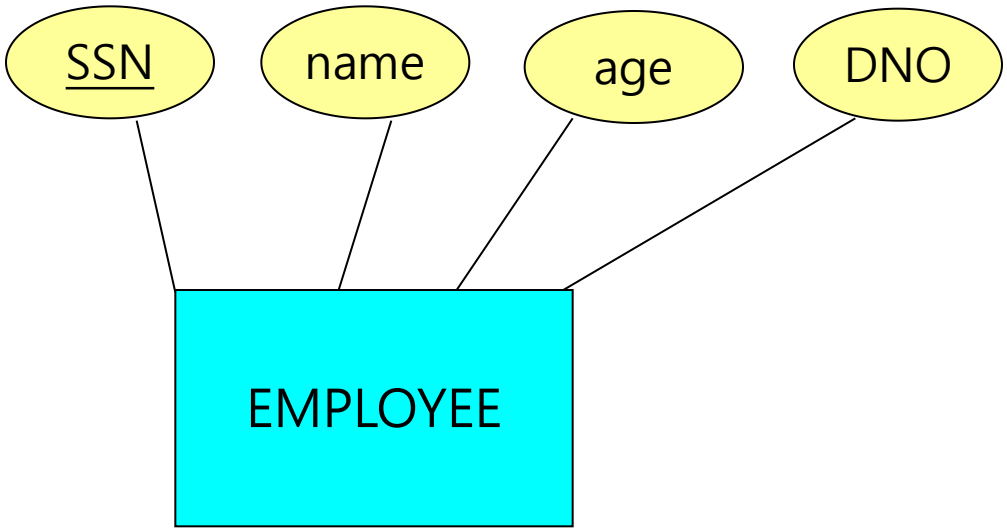
- Need to satisfy the following constraints;
  - 1 : M, M : N, 1 : 1 relationship
  - Total / Partial participation
  - Key Integrity
  - Referential Integrity
- Avoid many null values.
- Consider performance (= retrieval time).
- Avoid redundancy.

Entity Types

- For regular entity type **E**, create a relation **R** that includes all the simple attributes of **E**.
- Choose one of the keys of **E** as primary key (**PK**) for **R**.
- If the chosen key of **E** is composite, the set of simple attributes that form it will together form the **PK** of **R**.
- Each entity in **E** corresponds a row (tuple) in **R**
- Each attribute in **E** corresponds a column (attribute) in **R**

Entity Types

ER :

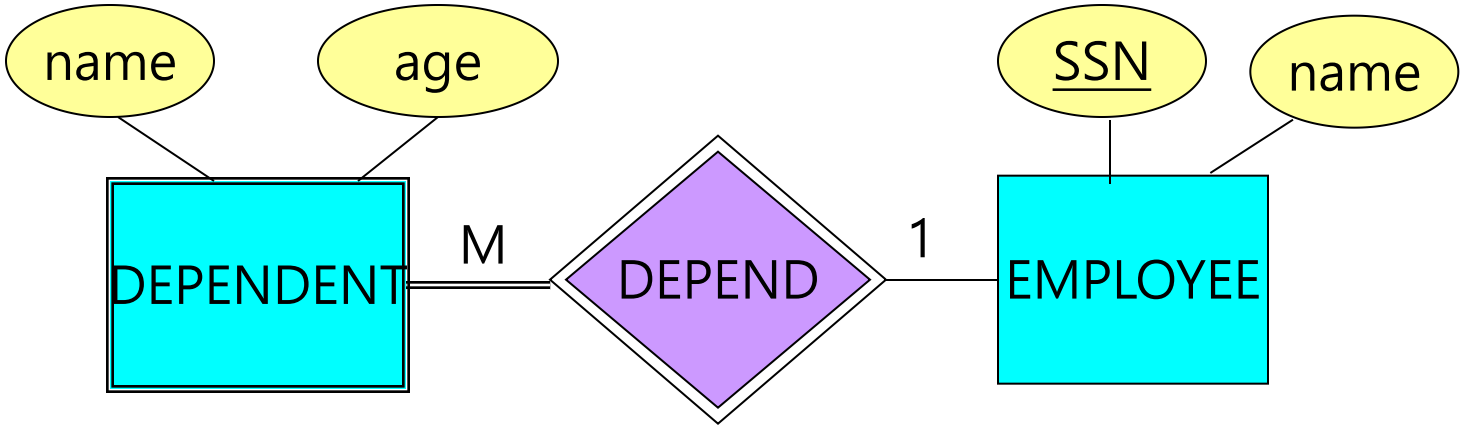


Relation : **EMPLOYEE (SSN, name, age, DNO)**

Weak Entity Types

- For weak entity type **W**, create a relation **R** and include all the attributes of **W**.
- Find **W**'s owner entity type **E**;
- Include as foreign key (**FK**) of **R** the **PK** of the owner **E**.
- **PK** of **R** is {**PK** of owner **E**, partial key of **R**}

Weak Entity Types: 예



- DEPENDENT is a weak type; EMPLOYEE is a owner type.

DEPENDENT (name, SSN, age)

EMPLOYEE (SSN, name)

1 : M Relationship : Total/Total

Case 1 **:** Both sides are **'Total':**

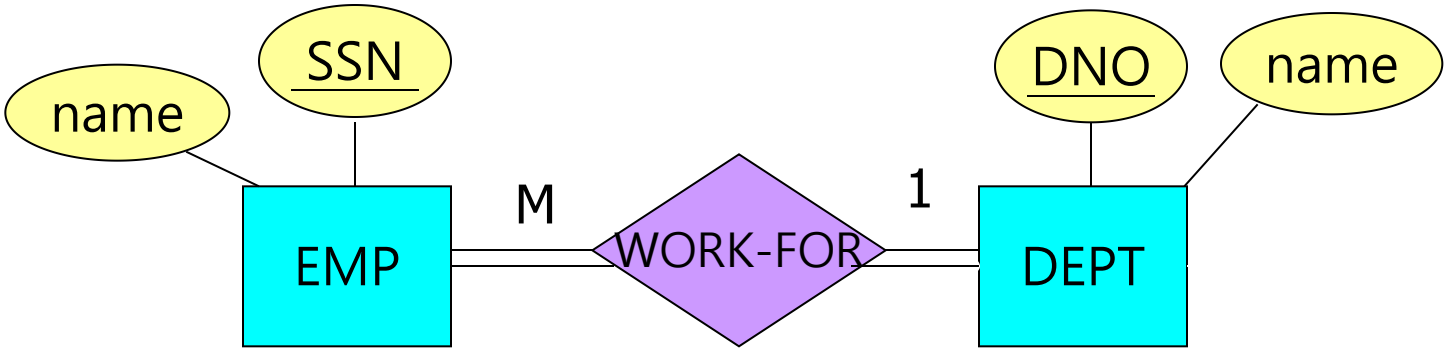
- For 1 : M relationship **R**, create a relation **S** representing the entity type at the "M-side" of the relationship.

- Include as **FK** in **S** the **PK** of the relation **T** that represents the entity type at the "1 – side".

  : Why? This is because we must satisfy key integrity;

- Include any simple attributes of the 1 : M relationship as attributes of **S**.

1 : M Relationship : Total/Total



EMP (SSN, name, DNO)

DEPT (DNO, name)

- Include the **PK** 'DNO' of DEPT relation (at the 1-side) as **FK** in EMP relation (at the M-side)
- WORK-FOR exists between DNO (in EMP) and DNO (in DEPT).

1 : M Relationship : Partial/Partial

Case 2 : Both sides are **'Partial'**

- For 1 : M relationship **R**, create a **new** relation **S** to represent **R**.

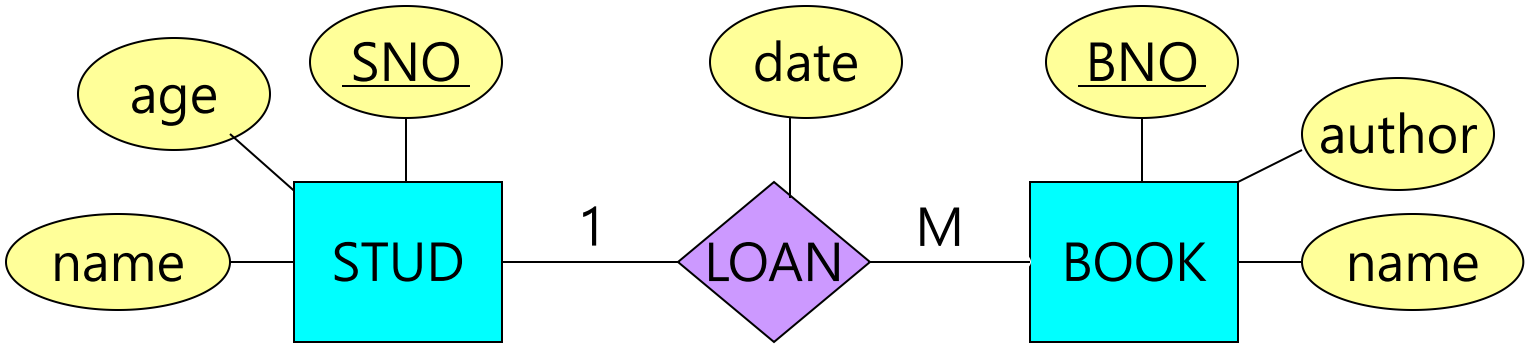
- Include as **FK** in **S** the **PK**s of each relations that represent the participating entity types;

  : Why? This is because we need to avoid many null values.)

- Also, include any simple attributes of the relationship type **R** as attributes of **S**.

1 : M Relationship : Partial/Partial

- Method A :



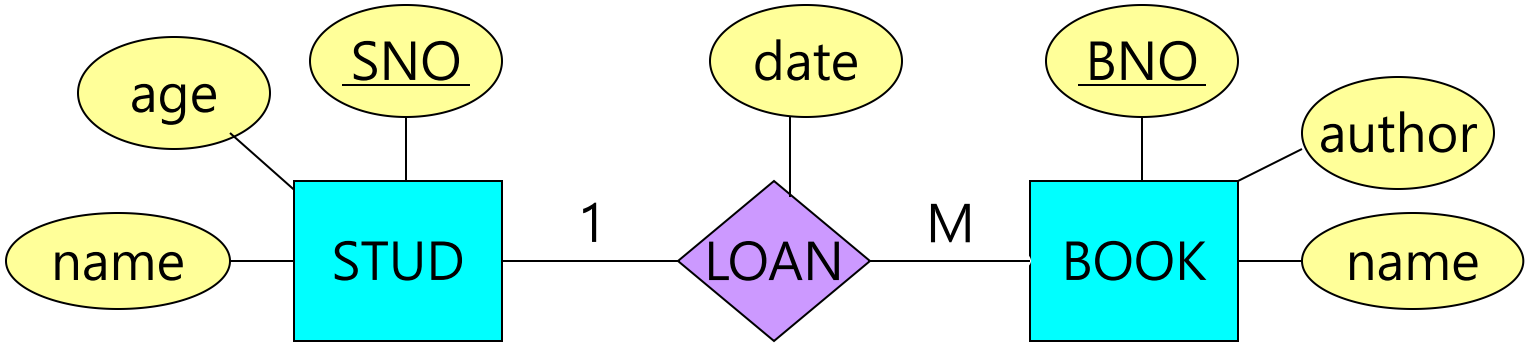
STUD (SNO, name, age)

BOOK (BNO, name, author, date, SNO)

- Problem : There may exist many null values in a BOOK relation; Why?

1 : M Relationship : Partial/Partial

- Method B :



STUD (SNO, name, age)

BOOK (BNO, name, author)

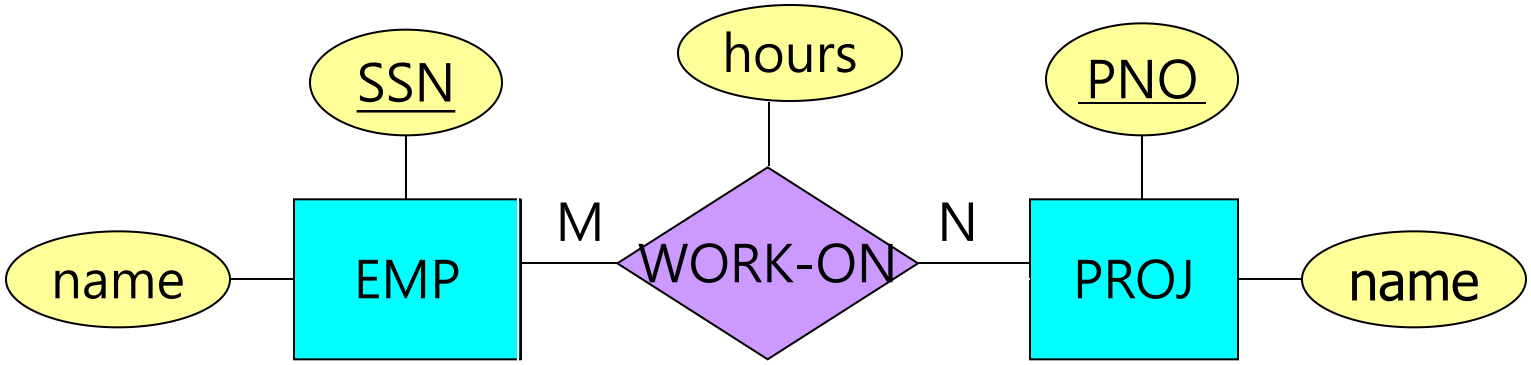
LOAN (BNO, SNO, date)

- Create a new relation LOAN, where each one of {SNO, BNO} is **FK**; BNO is **PK**; Why?
- No more null values! But, we need two join operations.

M : N Relationship

- For M : N relationship **R**, create a **new** relation **S** to represent **R**.
- Include as **FK** in **S** the **PK**s of each relations that represent the participating entity types;
- The combination of each **FK**s will form the **PK** of **S**.
- Also, include any simple attributes of the M : N relationship type as attributes of **S**.

M : N Relationship



EMP (SSN, name)

PROJ (PNO, name)

WORK-ON (SSN, PNO, hours)

- Create a new relation WORK_ON. Each **PK** of PROJ and EMP are included as **FK**s in WORK_ON.
- {SSN, PNO} is **PK** of WORK-ON relation.

1 : 1 Relationship

- For 1 : 1 relationship **R**, create the relations **S** and **T** that correspond to the entity types participating in **R**.

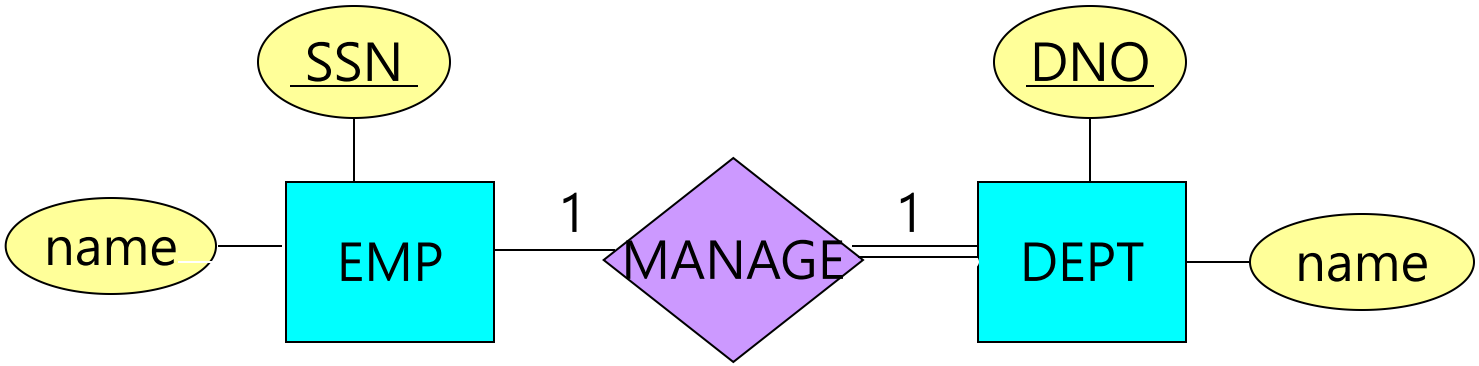
Case 1. **Foreign Key** : Two Relations
- The one side (say, **S**) is total and the other side (say, **T**) is partial.
- Include as **FK** in relation **S** the **PK** of relation **T**.

Case 2. **Merge relations** : Single Relation
- Both sides (say, **S** and **T**) are total.
- Merge two relations **S** and **T** and their relationship into a single relation.

Case 3. **Create a new relation** : Three Relations
- Both sides (say, **S** and **T**) are partial.
- Create a new relation **R** by including the **PK**s of the relations **S** and **T**.

1 : 1 Relationship

Case 1: The only one side (i.e., DEPT) is total:

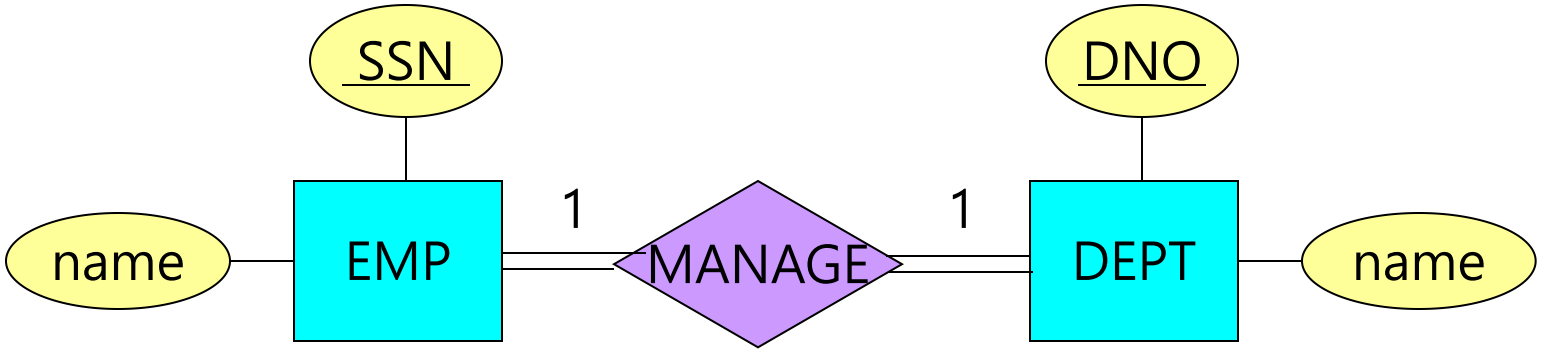


EMP (SSN, name)

DEPT (DNO, name, SSN)

1 : 1 Relationship

Case 2: The both sides are total:

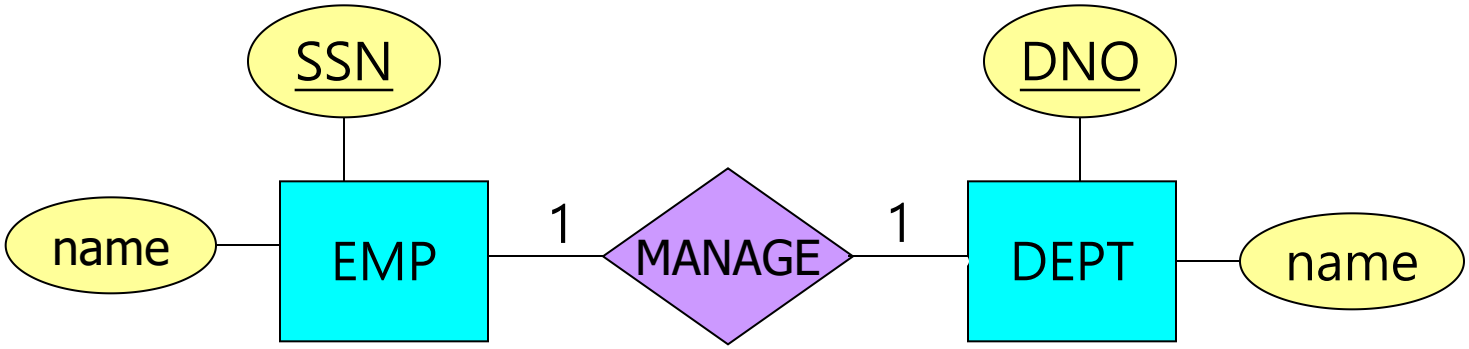


MANAGE (SSN, ename, DNO, dname)

- We merge into a single relation MANAGE.
- Either SSN or DNO (in MANAGE) is **PK**; No foreign key!
- In this case, we must rename 'name' as 'ename' and 'dname'.

1 : 1 Relationship

Case 3: The both sides are partial:



EMP (SSN, name)

DEPT (DNO, name)

MANAGE (SSN, DNO)

- Either SSN or DNO (in MANAGE) is **PK**.