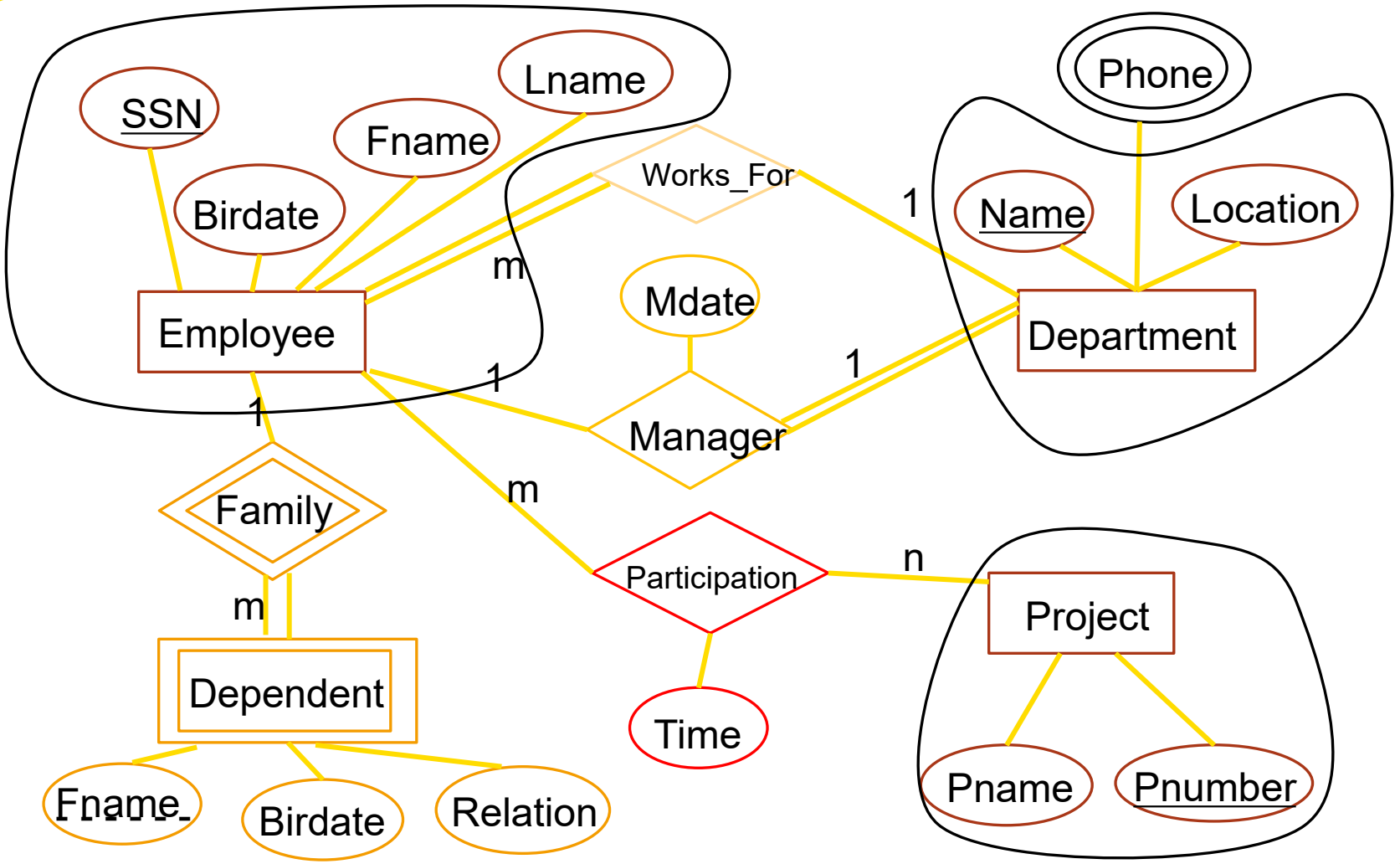
# Mapping Strong Entity Types

Step 1: For each ***strong entity*** (not weak entity) type E, create a new relation R with

- Attributes: all *simple attributes* (and simple components of composite attributes) of E.
- Key: key of E as the *primary key* for the relation.

# Mapping Strong Entity Types



# Mapping Strong Entity Types

Employee

| SSN | Fname | Lname | Birdate |
|---|---|---|---|

Department

| Name | Location |
|---|---|

Project

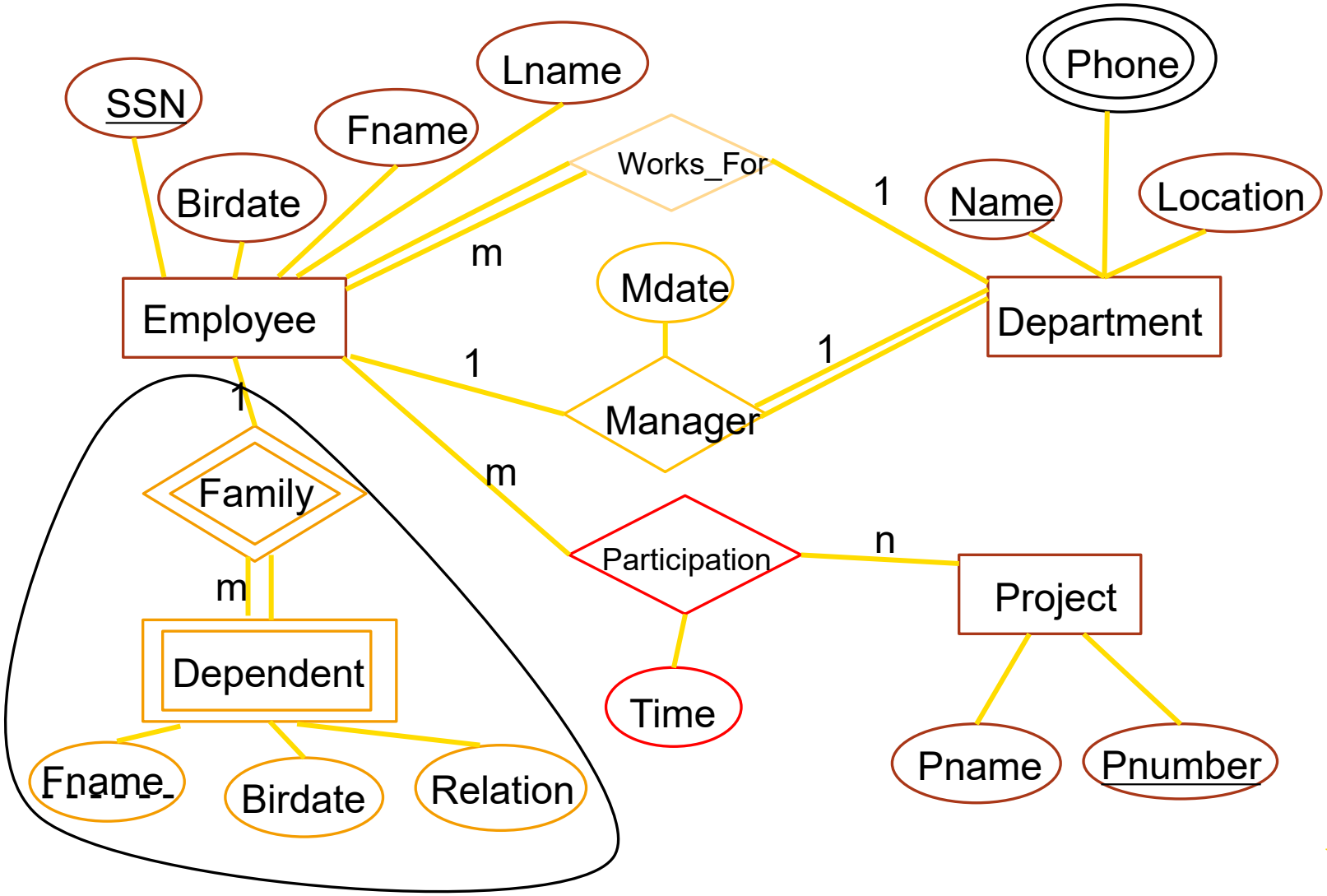
| Pnumber | Pname |
|---|---|

# Mapping Weak Entity Types

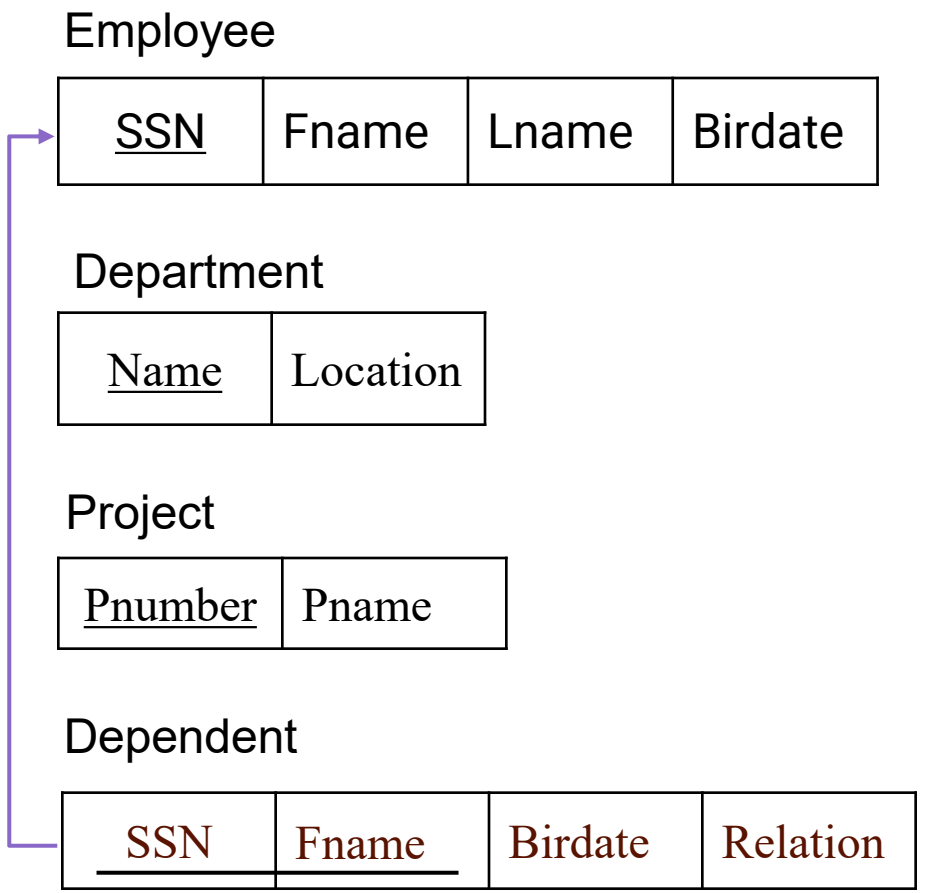
Step 2 : For each ***weak entity type*** W with the owner entity type E, create a new relation R with

- Attributes :
  - all simple attributes (and simple components of composite attributes) of W,
  - and include the primary key attributes of the relation derived from E as the foreign key.
- Key of R: foreign key to E and partial key of W.

# Mapping Weak Entity Types



# Mapping Weak Entity Types



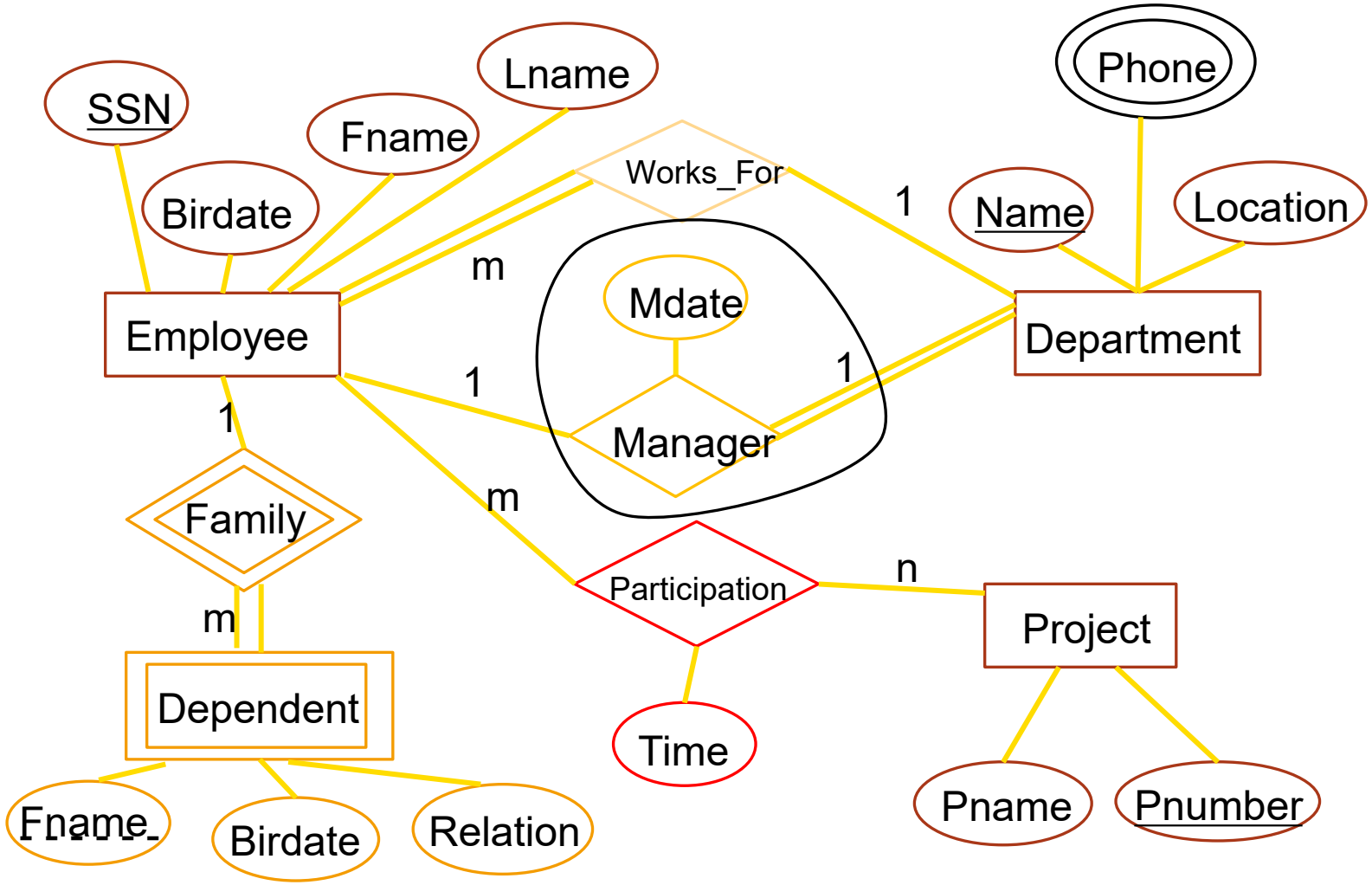
# Mapping 1:1 Relationship Types

Step 3 : For each ***1:1 relationship type*** B. Let E and F be the participating entity types. Let S and T be the corresponding relations.

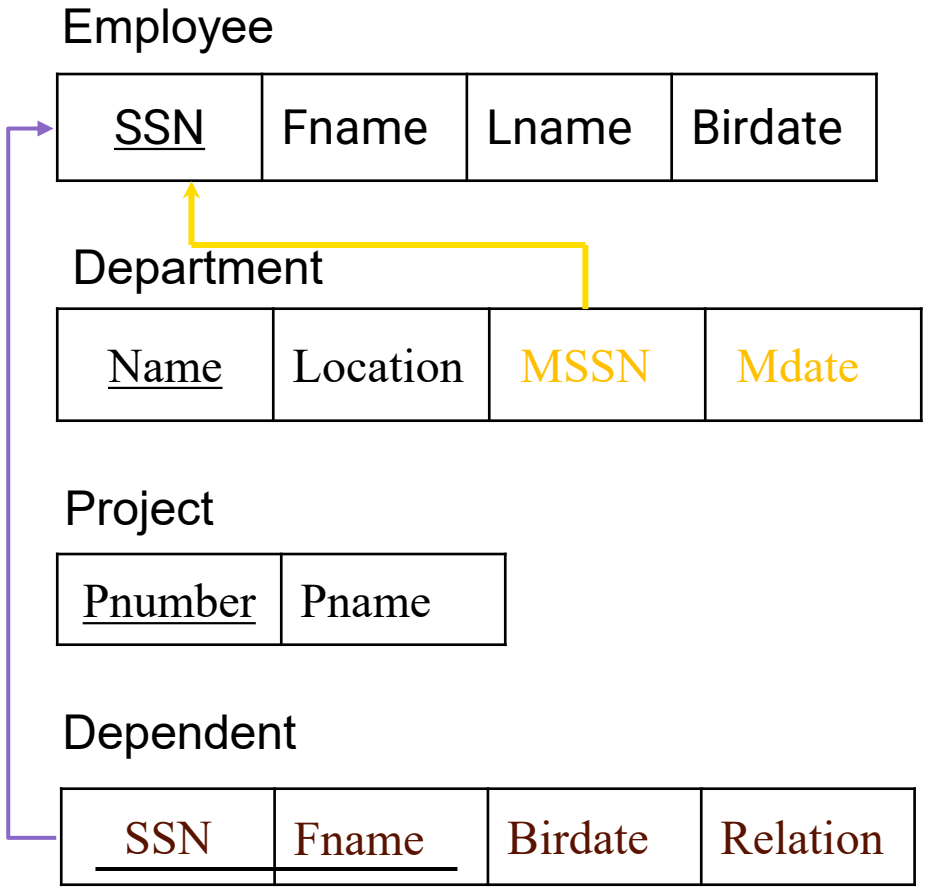
- Choose one of S and T (let S be the one that participates totally if there is one).
- Add attributes from the primary key of T to S as a foreign key.
- Add all simple attributes (and simple components of composite attributes) of B as attributes of S.

*(Alternatively, merge the two entity types and the relationship into a single relation, especially if* ***both participate totally and do not participate in other relationships****).*

# Relationship Types



# Mapping 1:1 Relationship Types



# Mapping 1:N Relationship Types

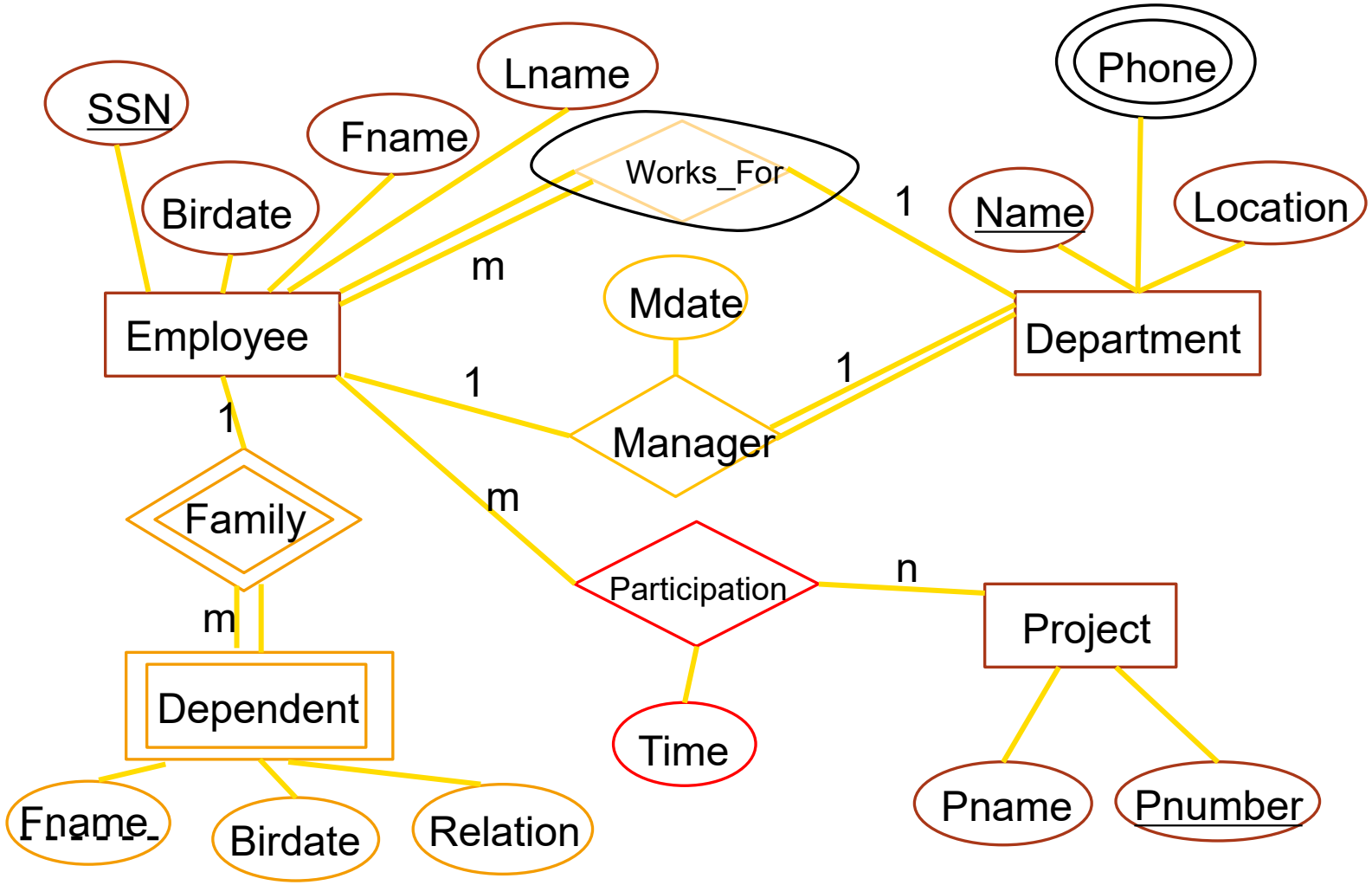
Step 4 : For each ***1:N relationship type*** B. Let E and F be the participating entity types. Let S and T be the corresponding relations. Let E be the entity on the 1 side and F on the N side.

Add to the relation belonging to entity T,

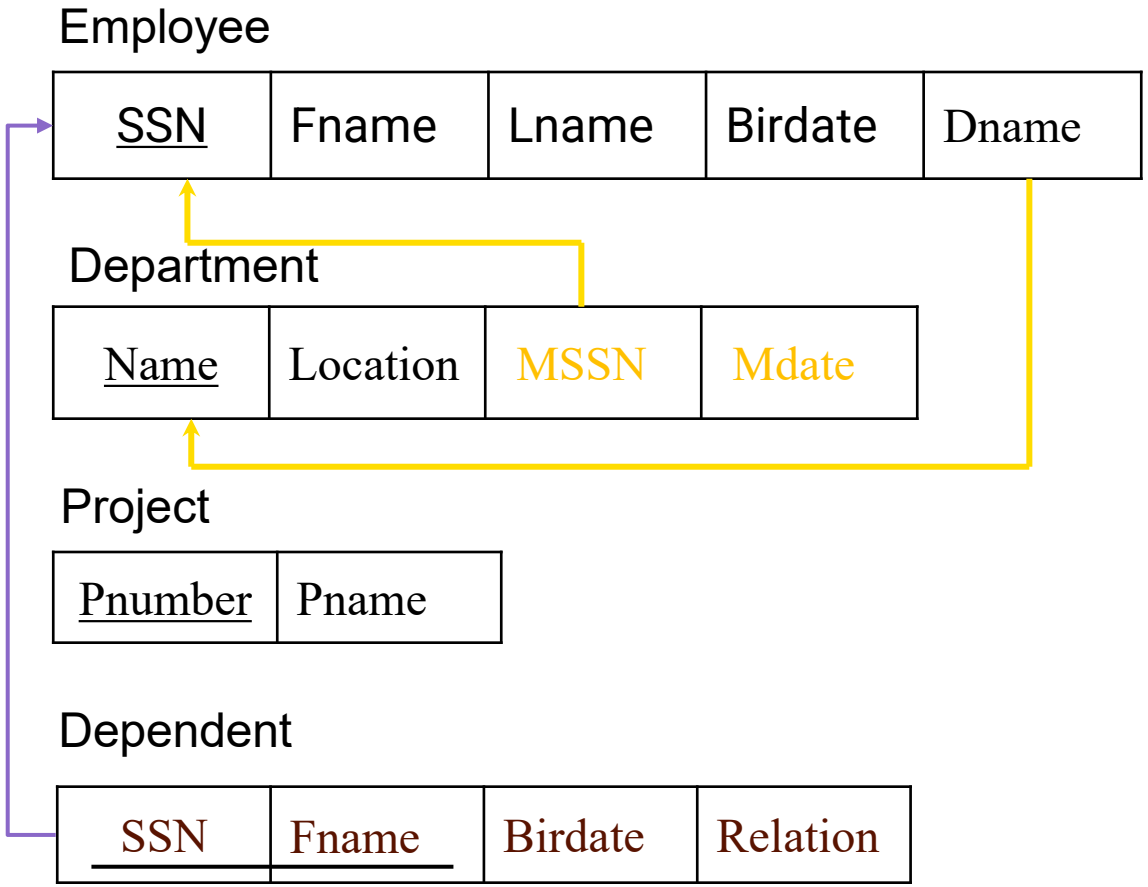
- the attributes from the primary key of S as a foreign key.
- any simple attributes (or simple components of composite attributes) from relationship B.

(Notice that this doesn't add any new tuples, just attributes.)

# Mapping 1:N Relationship Types



# Mapping 1:N Relationship Types



# Mapping M:N Relationship Types

Step 5: For each ***N:M relationship type*** B. Let E and F be the participating entity types. Let S and T be the corresponding relations

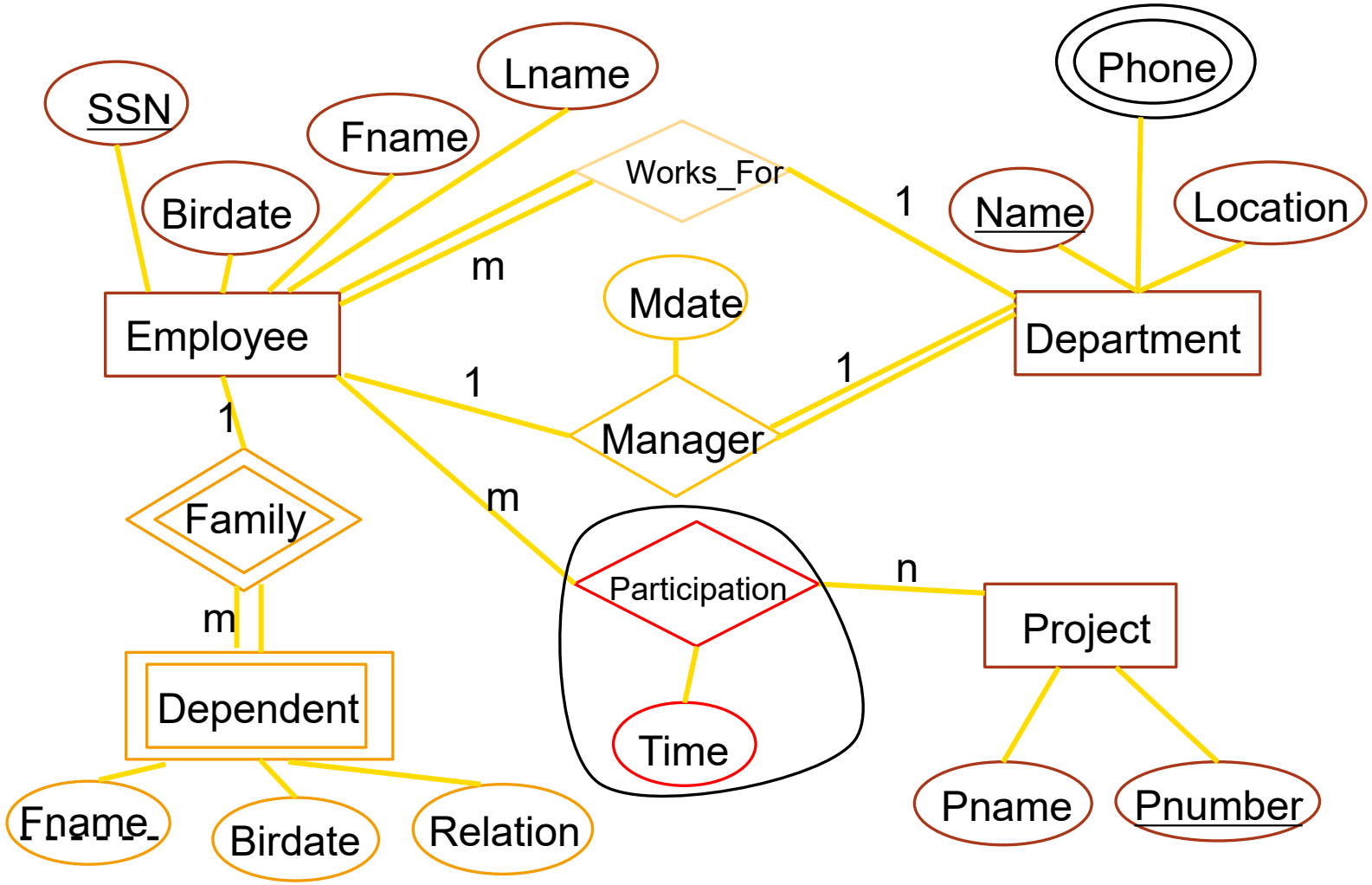
Create a new relation R (*cross-reference*) with

Attributes :

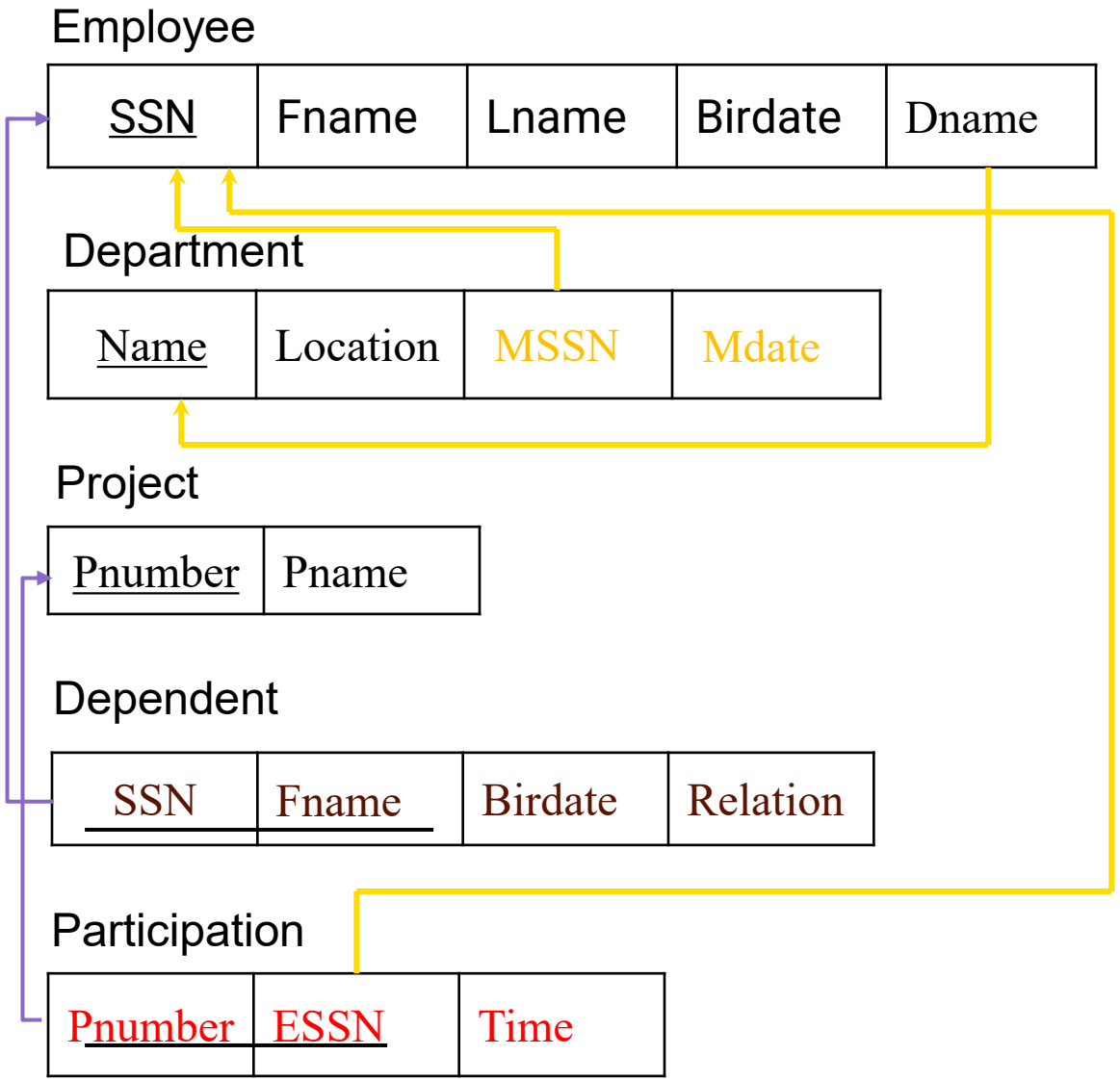
- Attributes from the key of S as a foreign key,
- Attributes from the key of T as a foreign key,
- Simple attributes and simple components of composite attributes of relation B.

Key: All attributes from the key of S and T.

# Mapping M:N Relationship Types



# Mapping M:N Relationship Types



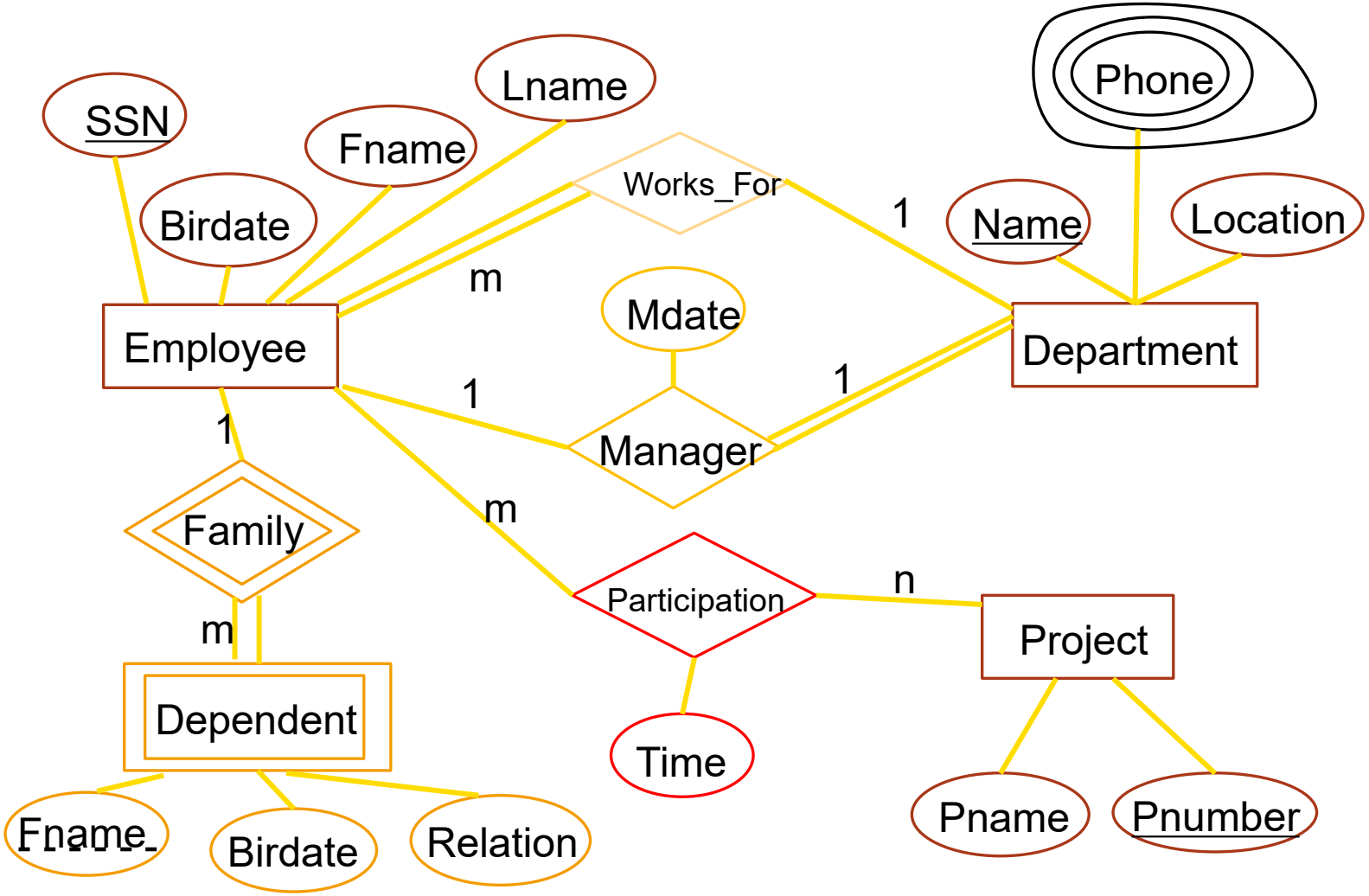
# Mapping Multivalued Attributes

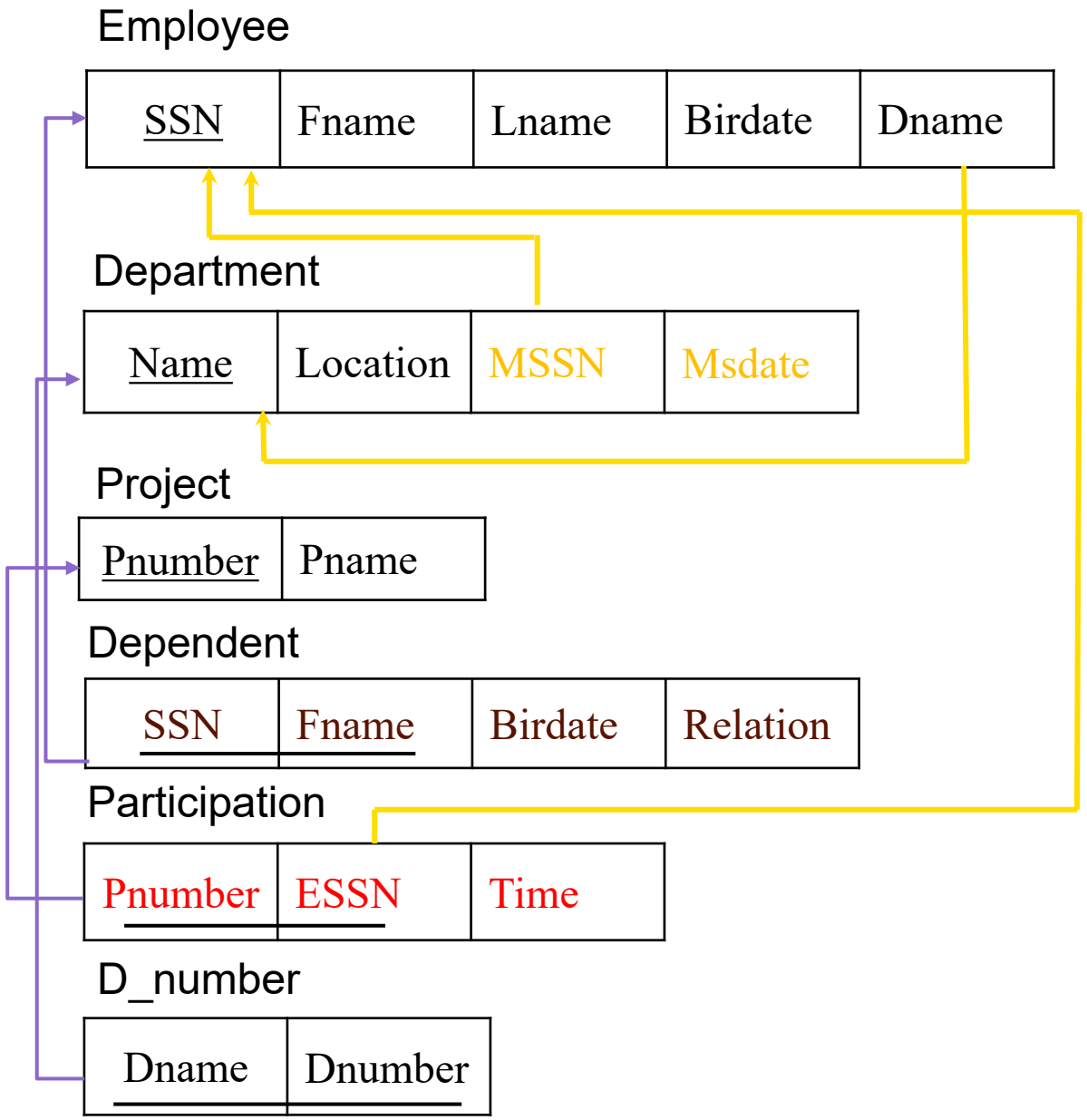
Step 6: For each ***multivalued attribute*** A, where A is an attribute of E, create a new relation R.

- *If A is a multivalued simple attribute,*
  - Attributes of R = Simple attribute A, and key of E as a foreign key.
- *If A is a multivalued composite attribute,*
  - Attributes of R = All simple components of A, and key of E as a foreign key.

In both cases, the primary key of R is the set of all attributes in R.

# Mapping Multivalued Attributes




Employee
SSN
Fname
Lname
Birdate
Dname
Department
Name
Location
MSSN
Msdate
Project
Pnumber
Pname
Dependent
SSN
Fname
Birdate
Relation
Participation
Pnumber
ESSN
Time
D_number
Dname
Dnumber

# Mapping N-ary Relationship Types

Step 7: For each ***N-ary relationship type*** (n > 2), create a new relation with

- Attributes: same as Step 5.
- Key: same as Step 5

*(Advice: binary relationships are simpler to model.)*