COMP9311:
DATABASE SYSTEMS

Term 1 2024

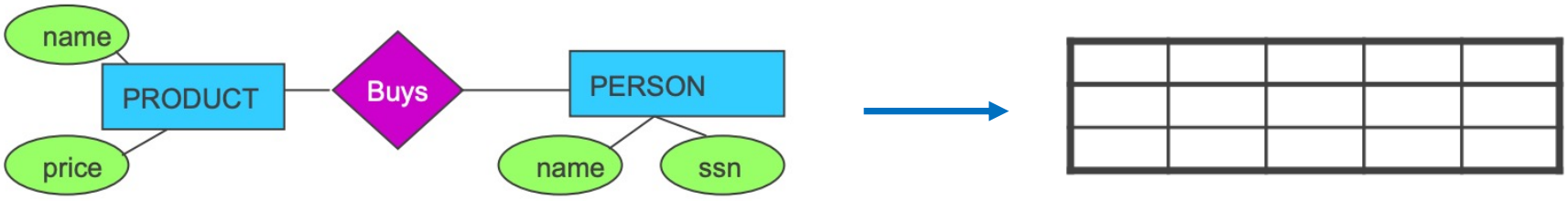
Week 2 – ER to Relational Data Model

By Xiaoyang Wang, CSE UNSW

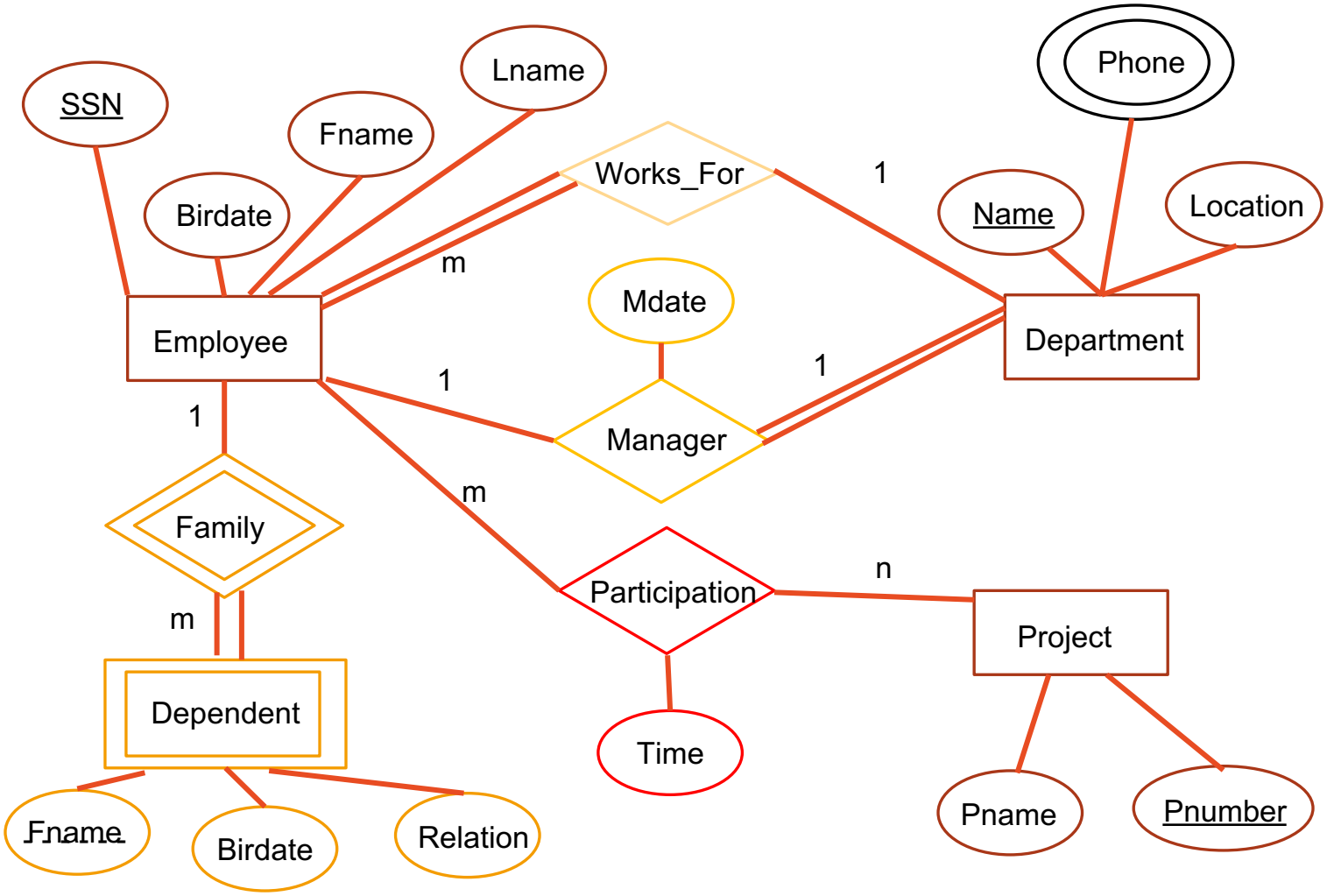
Disclaimer: the course materials are sourced from previous offerings of COMP9311 and COMP3311

ER to Relational Data Model Mapping

- One technique for database design is to first design a conceptual schema using a high-level data model,
- and then map it to a conceptual schema in the DBMS data model for the chosen DBMS.
- Here we look at a way to do this mapping from the ER to the relational data model.
- It involves the following **7 steps**.



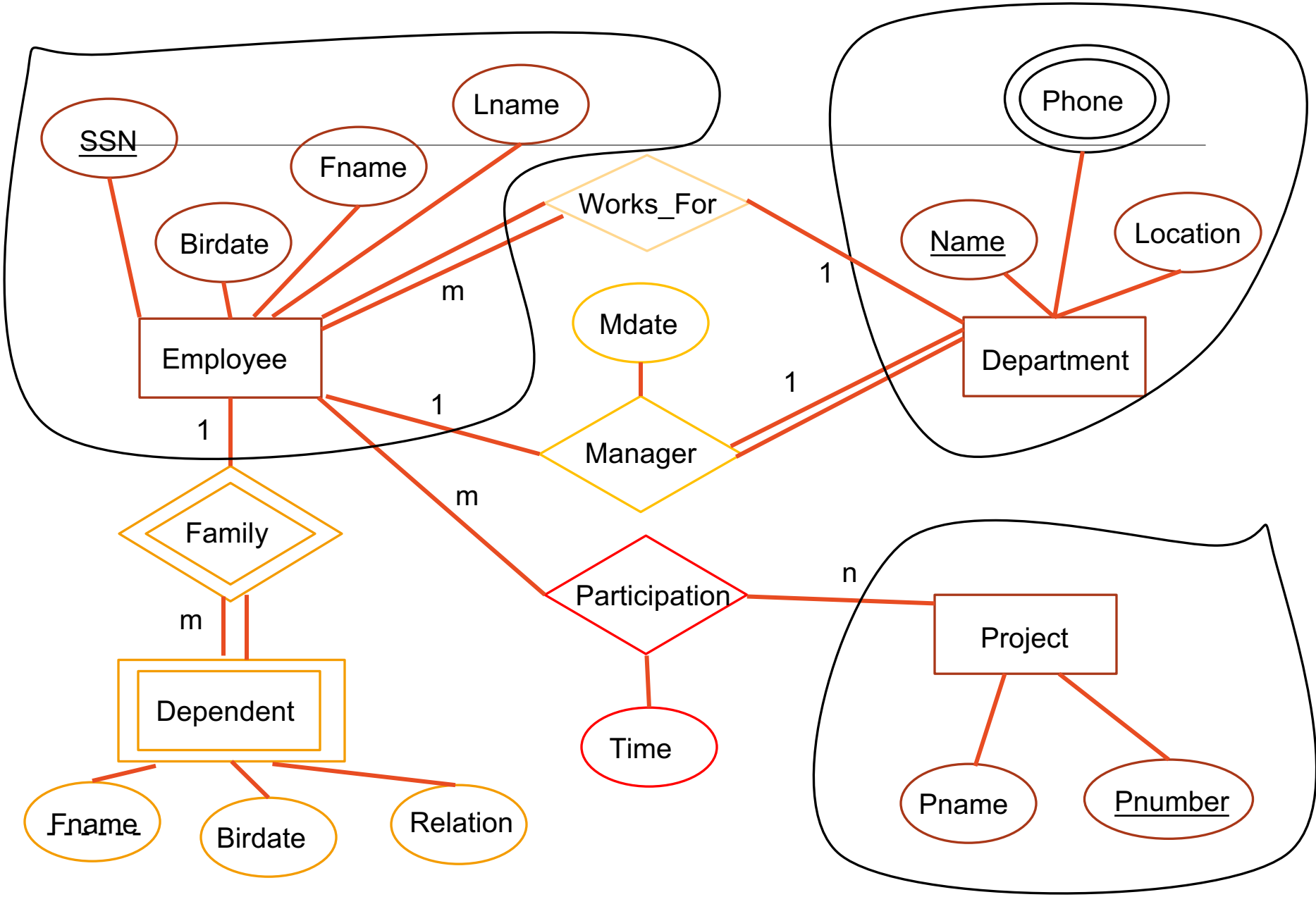
Guiding Example



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.


SSN
Lname
Fname
Birdate
Works_For
Phone
Name
Location
Employee
Mdate
Department
Manager
Family
Participation
Project
Dependent
Time
Fname
Birdate
Relation
Pname
Pnumber
m
1
1
1
m
m
n

Mapping Strong Entity Types

Employee

SSN	Fname	Lname	Birdate

Department

Name	Location

Project

Pnumber	Pname

Mapping Weak Entity Types

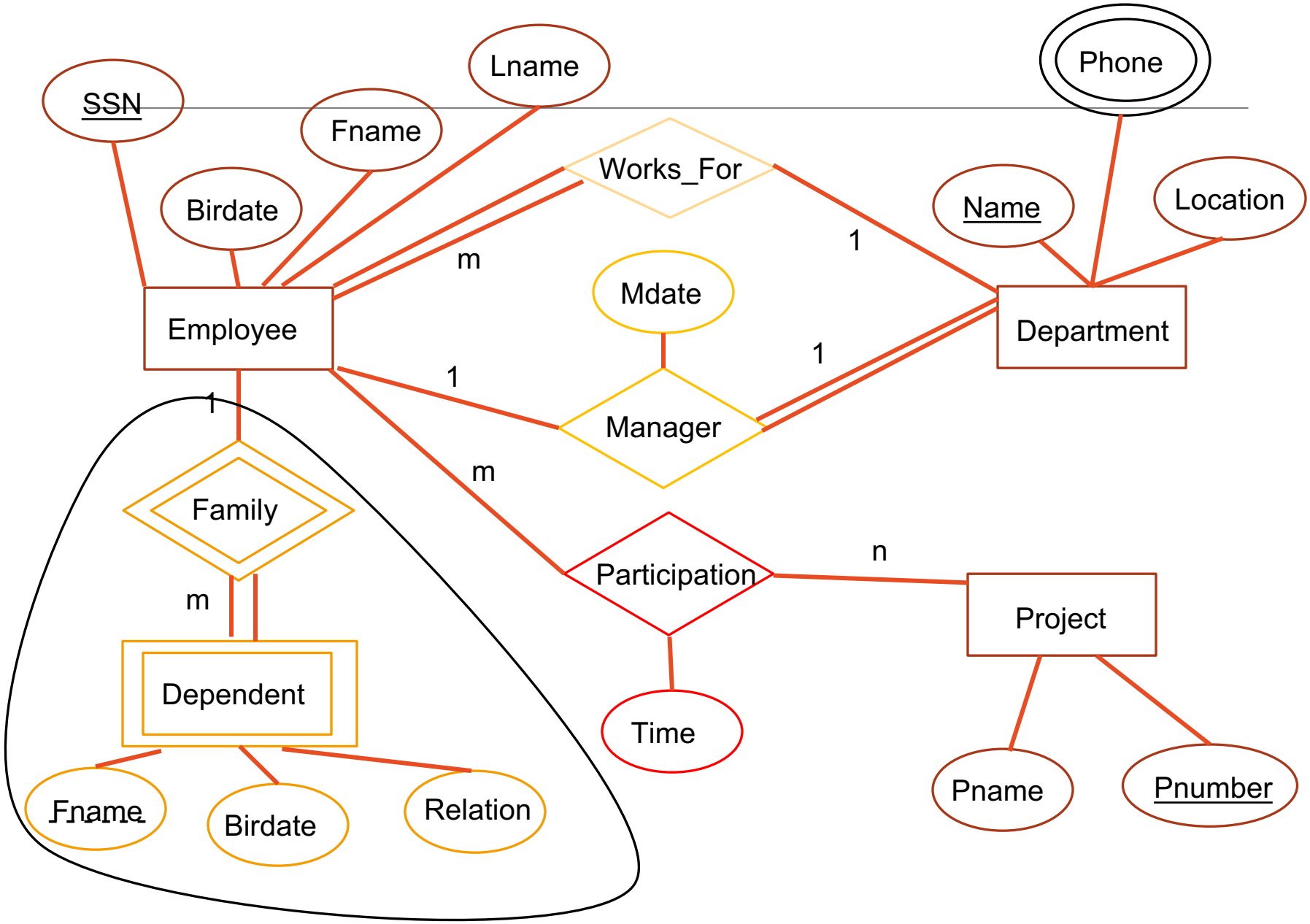
Step 2: For each *weak entity type* W with the owner entity type E, create a new relation R with

Attributes :

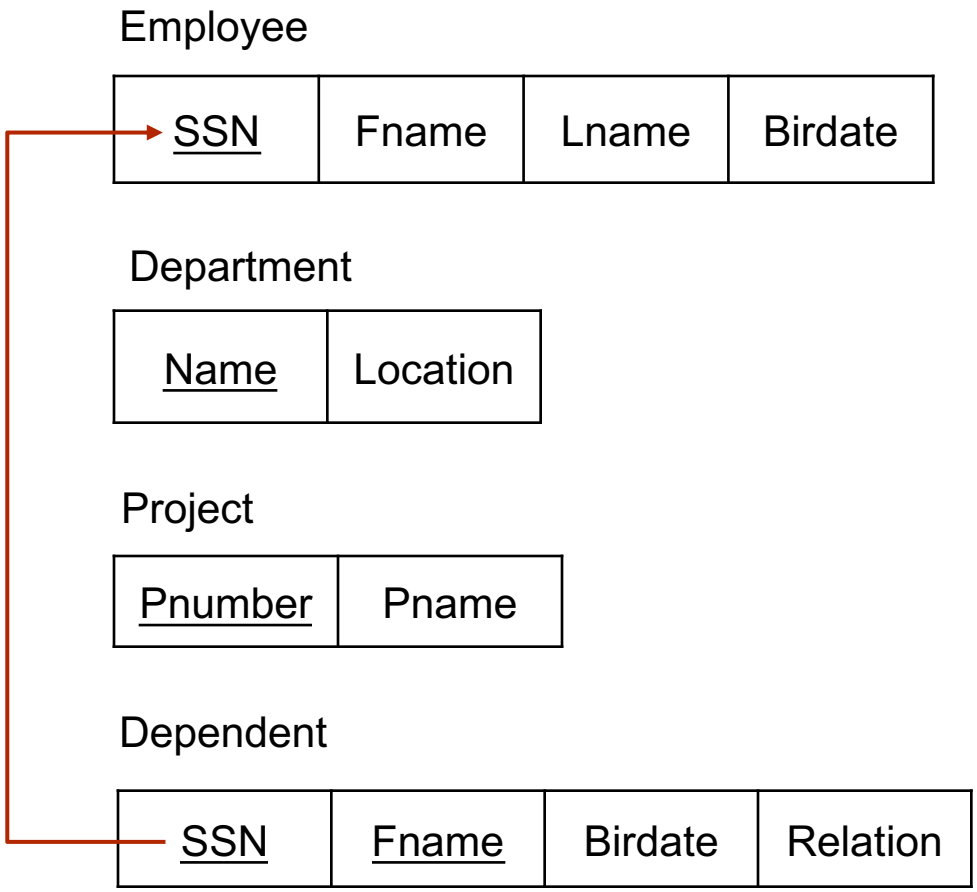
- all simple attributes (and simple components of composite attributes) of W,
- and include the primary key attributes of the relation derived from E as the foreign key.

Key of R:

- foreign key to E and partial key of W.


SSN
Lname
Fname
Phone
Works_For
Birdate
Name
Location
m
1
Mdate
Employee
Department
1
1
1
Manager
m
Family
Participation
n
m
Project
Dependent
Time
Fname
Birdate
Relation
Pname
Pnumber

Mapping Weak Entity Types

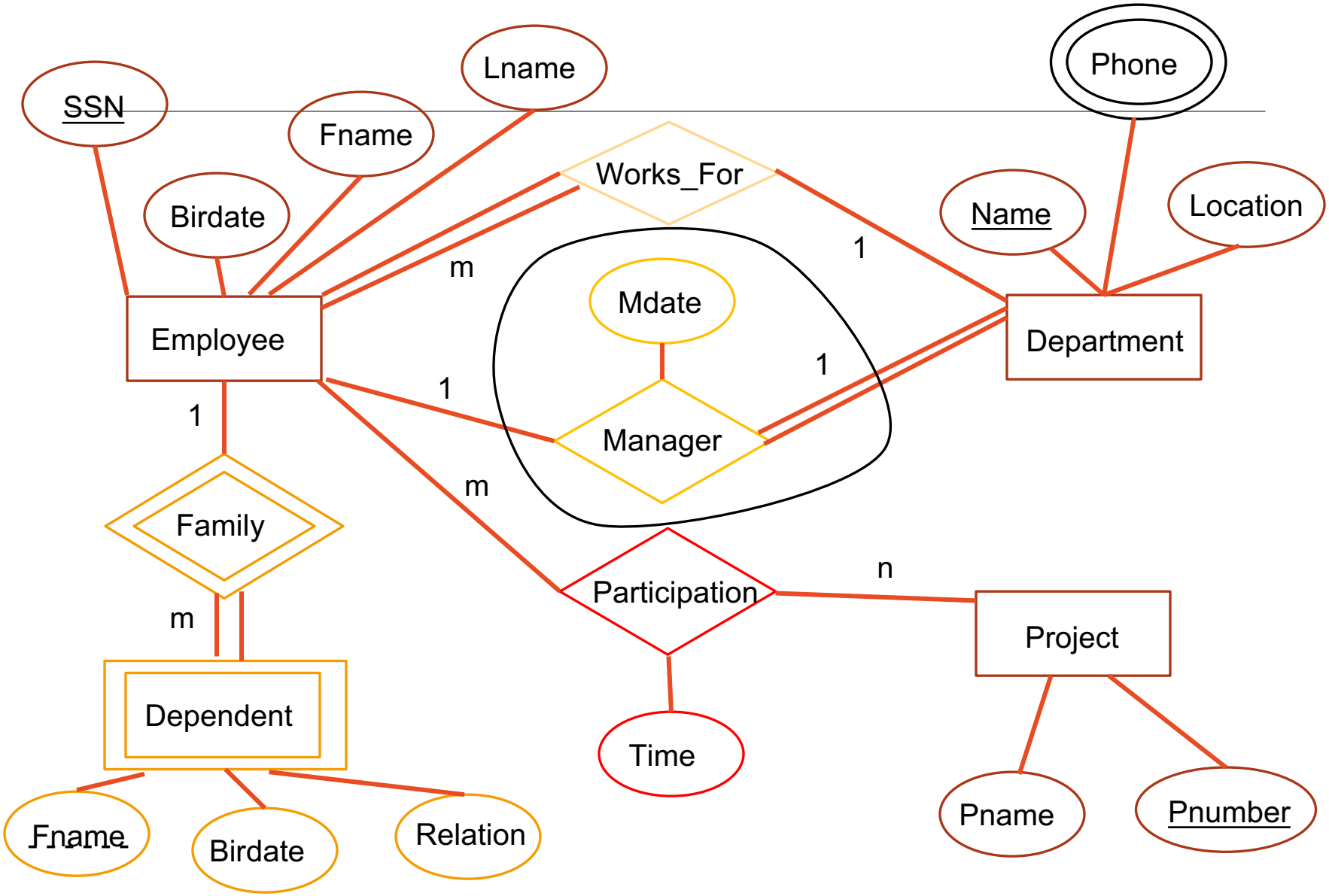


Mapping 1:1 Relationship Types

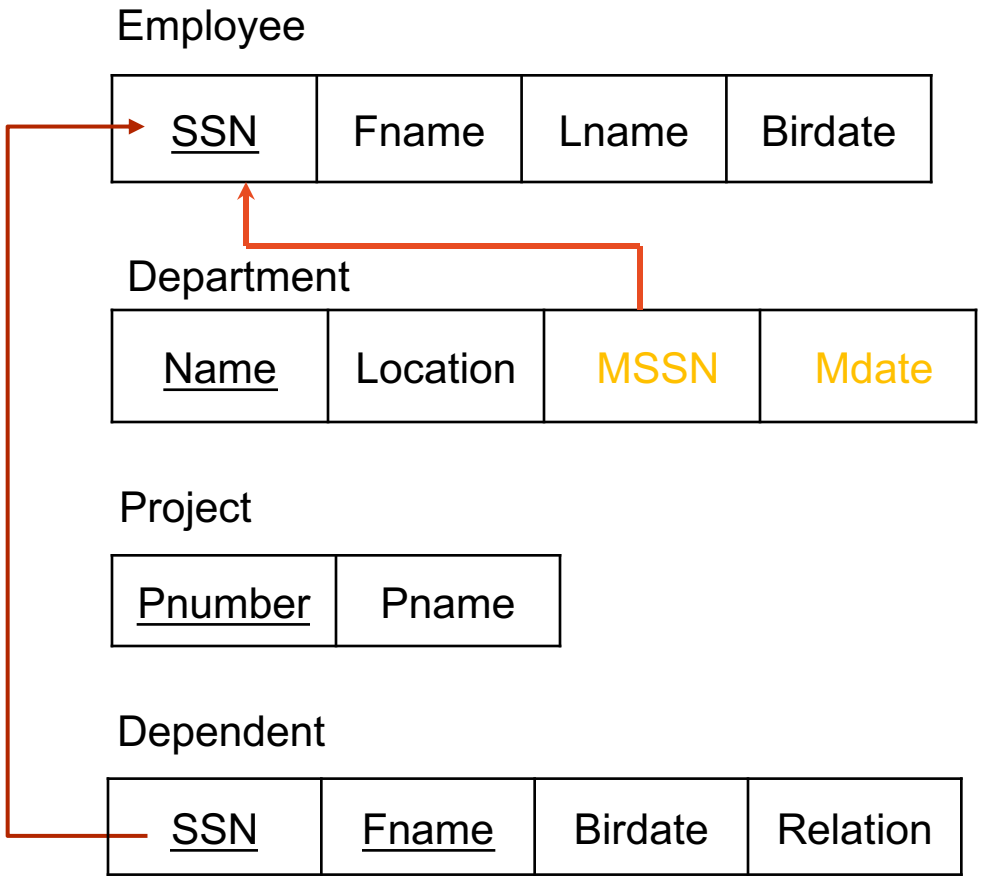
Step 3: For each *1:1 relationship type* B, let E and F be the participating entity types. Let S and T be the corresponding relations.

- 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****).*


SSN
Lname
Fname
Phone
Works_For
Birdate
Name
Location
m
1
Mdate
Employee
Department
1
1
1
Manager
m
Family
Participation
n
m
Project
Dependent
Time
Fname
Birdate
Relation
Pname
Pnumber

Mapping 1:1 Relationship Types



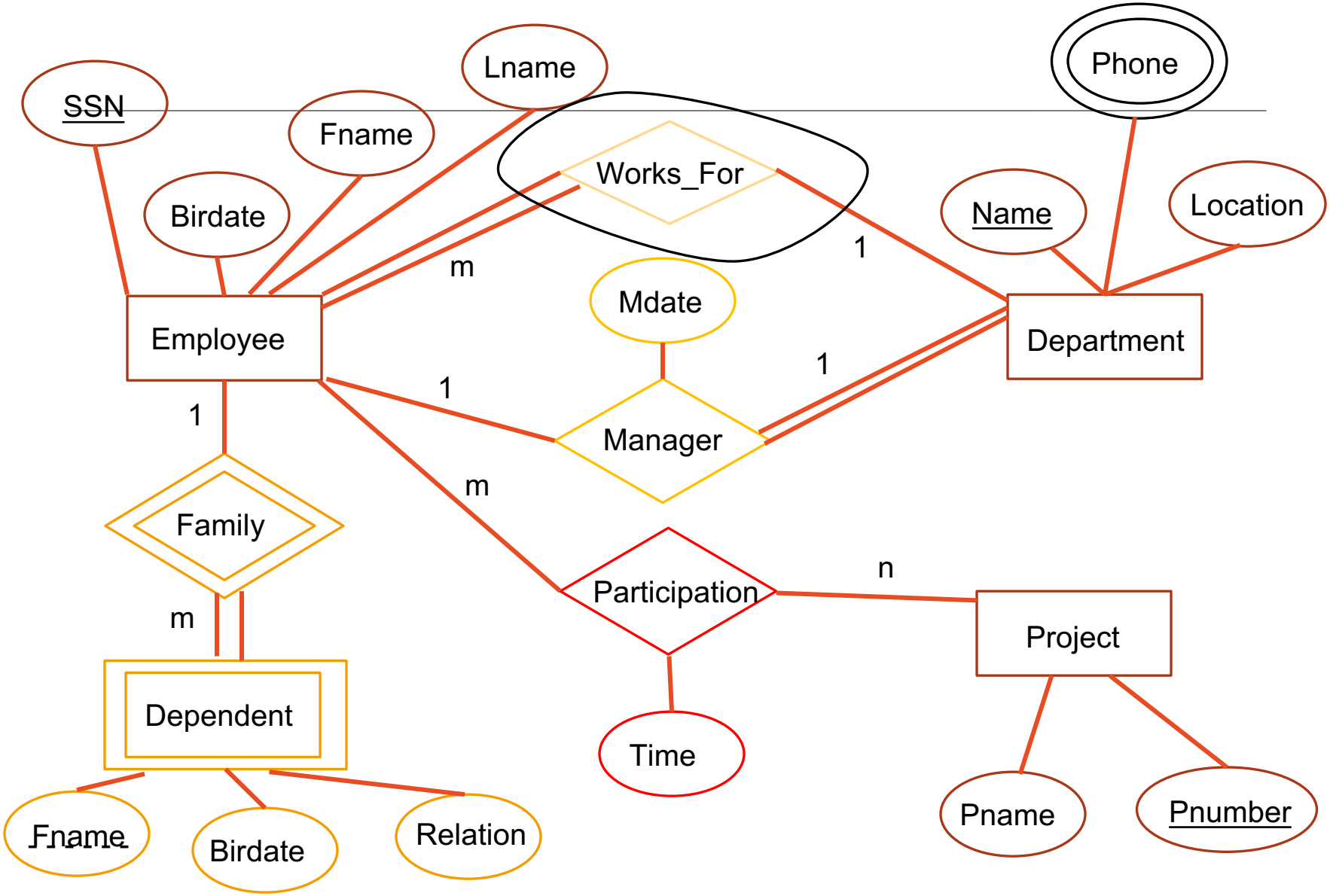
Mapping 1:N Relationship Types

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. E is the entity on the 1 side and F on the N side.

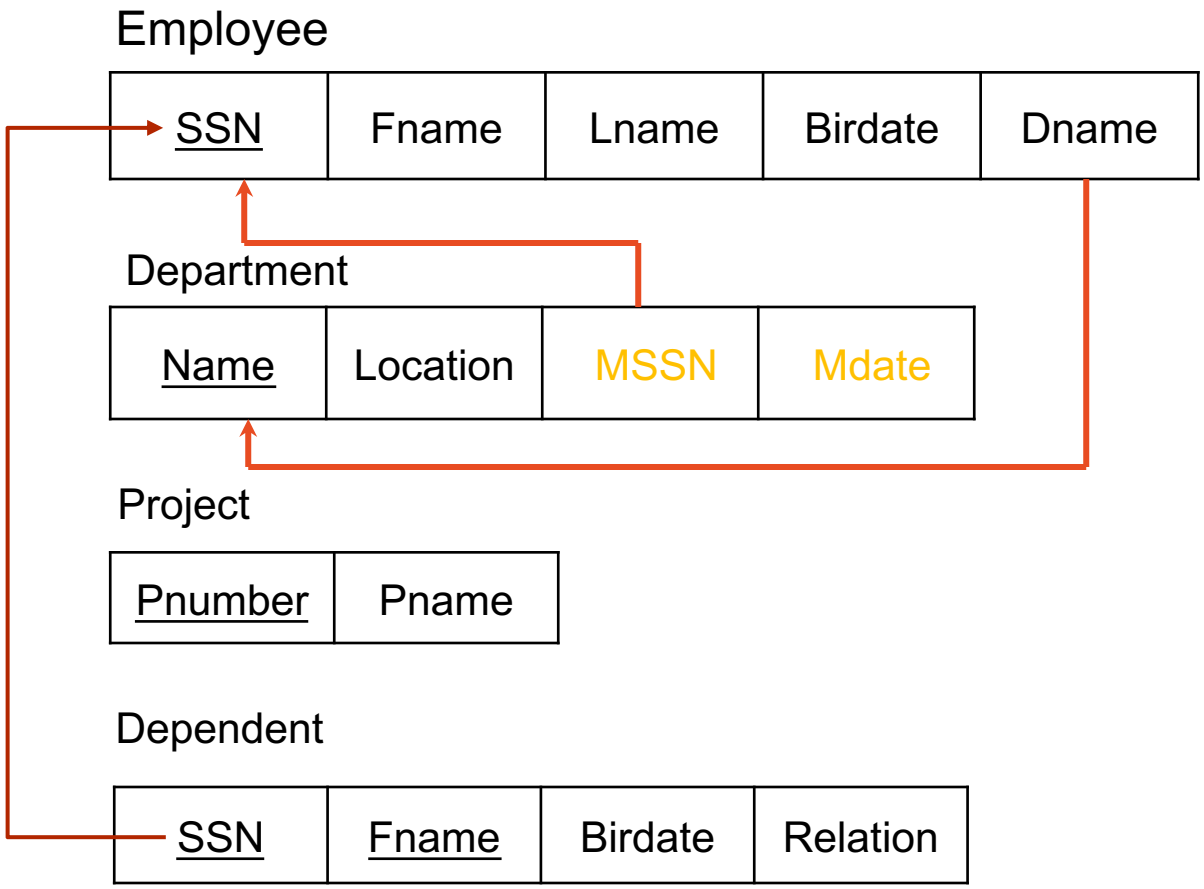
Add to the relation belonging to entity T,

- 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.)


SSN
Lname
Fname
Phone
Works_For
Birdate
Name
Location
m
1
Mdate
Employee
Department
1
1
1
Manager
m
Family
Participation
n
m
Project
Dependent
Time
Fname
Birdate
Relation
Pname
Pnumber

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

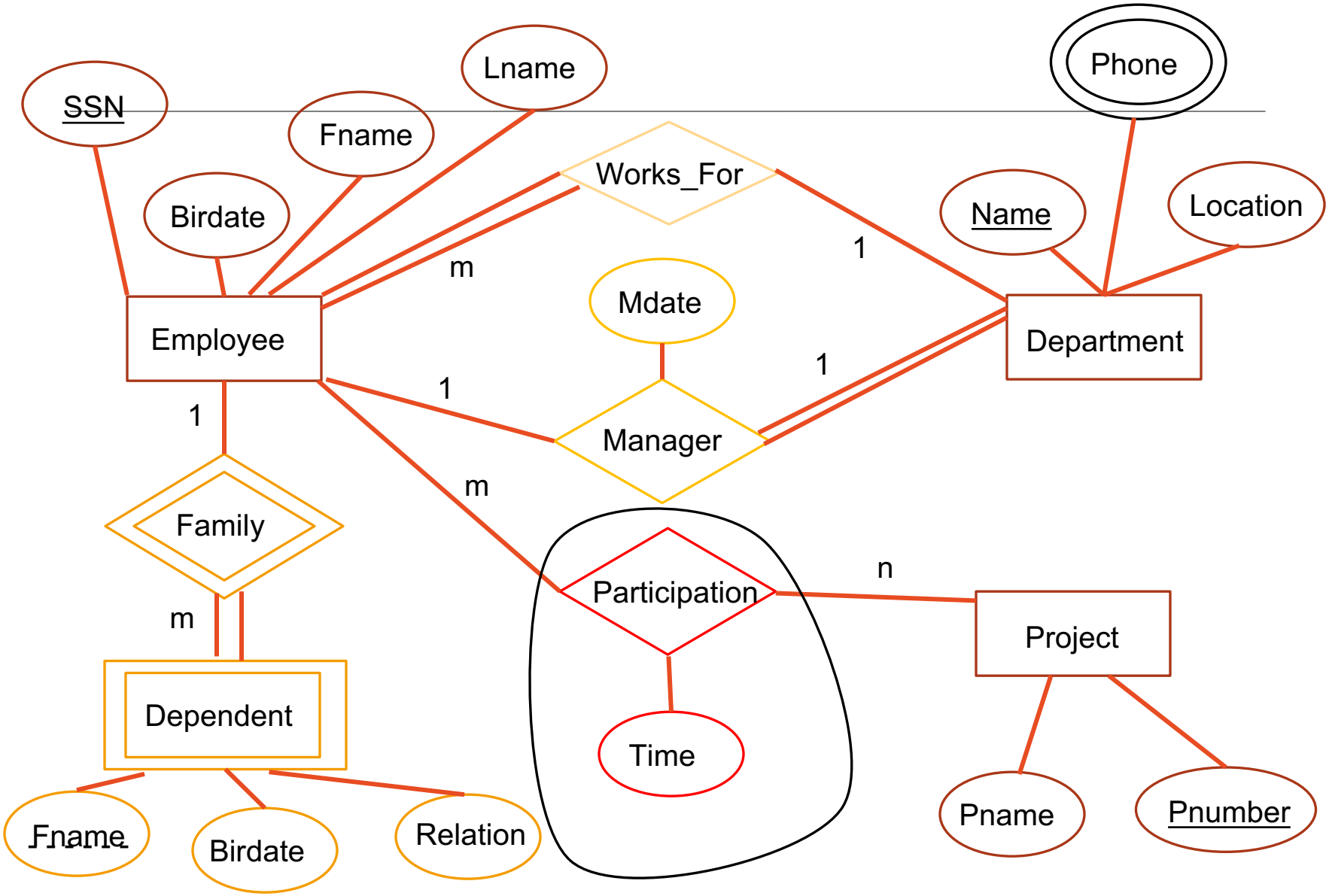
Create a new relation R (*cross-reference*) with

Attributes:

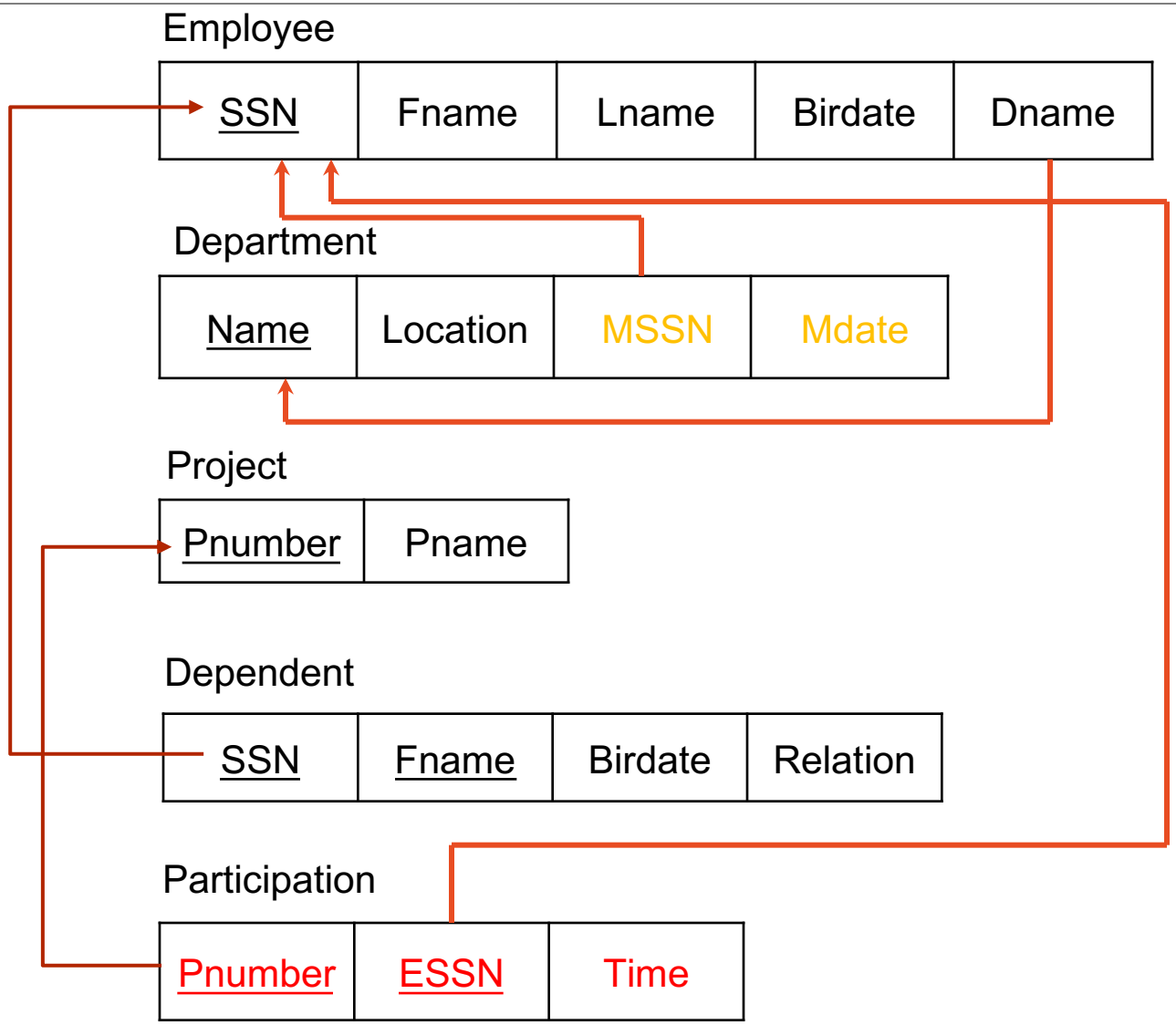
- Attributes from the key of S as foreign key,
- And attributes from the key of T as foreign key,
- And simple attributes, and simple components of composite attributes of relation B.

Key:

- All attributes from the key of S and the key of T.


SSN
Lname
Fname
Phone
Works_For
Birdate
Name
Location
m
1
Mdate
Employee
Department
1
1
1
Manager
m
Family
Participation
n
m
Project
Dependent
Time
Fname
Birdate
Relation
Pname
Pnumber

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.

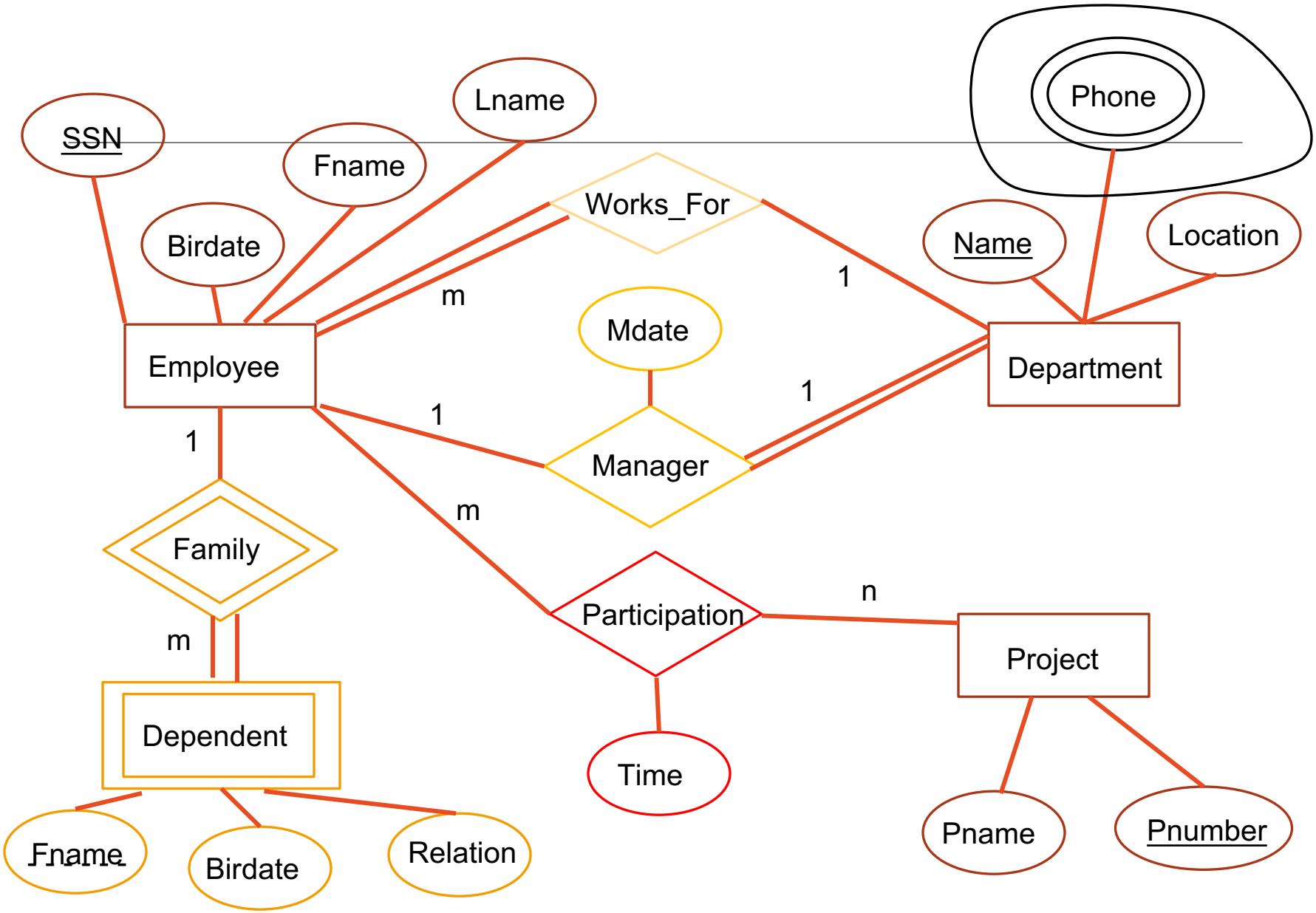
If A is a multivalued simple attribute,

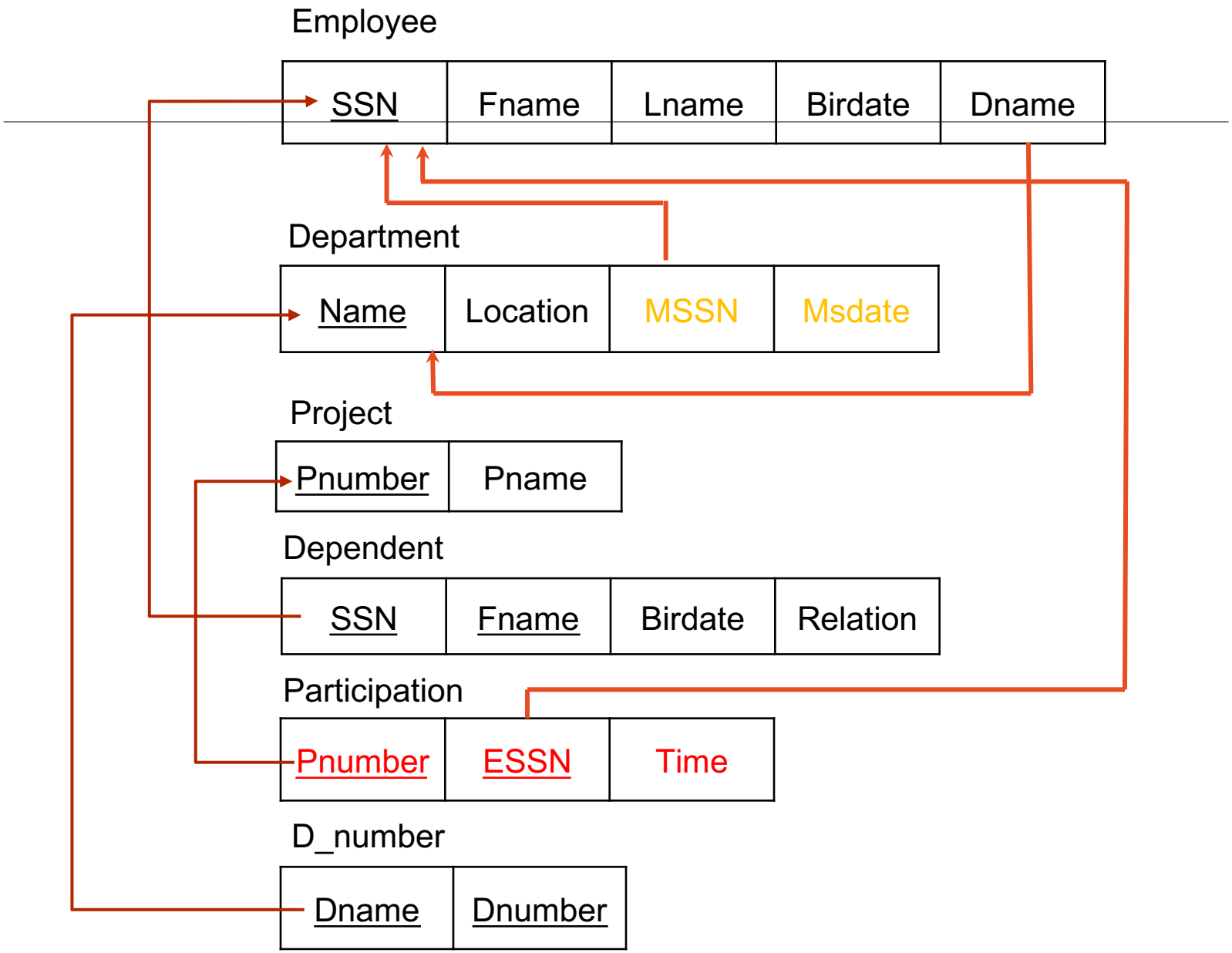
- Attributes of R = Simple attribute A, and key of E as a foreign key.

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.


SSN
Lname
Fname
Phone
Works_For
Birdate
Name
Location
m
1
Mdate
Employee
Department
1
1
Manager
1
m
Family
Participation
n
m
Project
Dependent
Time
Fname
Birdate
Relation
Pname
Pnumber


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 S with

Attributes:

- Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types.
- Also include any simple attributes of the n-ary relationship type (or simple components of composite attributes) as attributes of S.

Key:

- All attributes from the primary keys of the participating entity types

(Advice: binary relationships are simpler to model)

Summary of Mapping

Map Entities first

- Strong Entity Types (Step 1)
- Weak Entity Types (Step 2)

Map Relationship

- 1:1 Relationship Types (Step 3)
- 1:N Relationship Types (Step 4)
- M:N Relationship Types (Step 5)
- N-ary Relationship Types (Step 7)

Mapping

- Multivalued Attributes (Step 6)

ER vs Relational Model

ER MODEL	RELATIONAL MODEL
Entity Type	Entity relation
1:1 or 1:N relationship type	Foreign key (or relationship relation)
M:N relationship type	Relationship relation and two foreign key
n-ary relationship type	Relationship relation and n foreign key
Simple Attribute	Attribute
Composite Attribute	Set of simple component attributes
Multivalued Attribute	Relation and foreign key