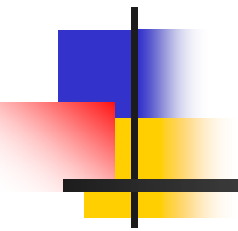
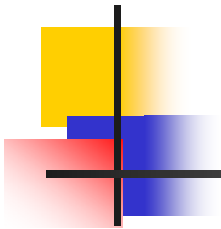
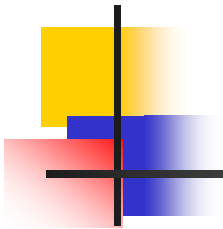


ER to Relational Mapping



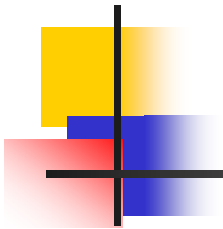
Objectives

- Introduction +
- Mapping Entity Types +
- Mapping Relationship Types +
- Mapping Multi-valued Attributes+



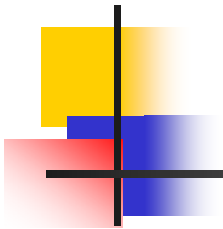
- Introduction

- In the previous lectures we looked at conceptual database design using the ER diagram.
- Here we are going to discuss in detail the process of mapping, or converting and ER design into a relational model. This is the logical design, or data model mapping, step as shown in the database design procedure.
- We will start by describing how we map an ER diagram into a relational schema.



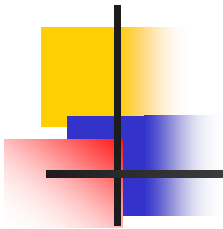
- Mapping Entity Types

- Mapping of Regular Entity Types +
- Mapping of Weak Entity Types +



-- Mapping of Regular Entity Types ...

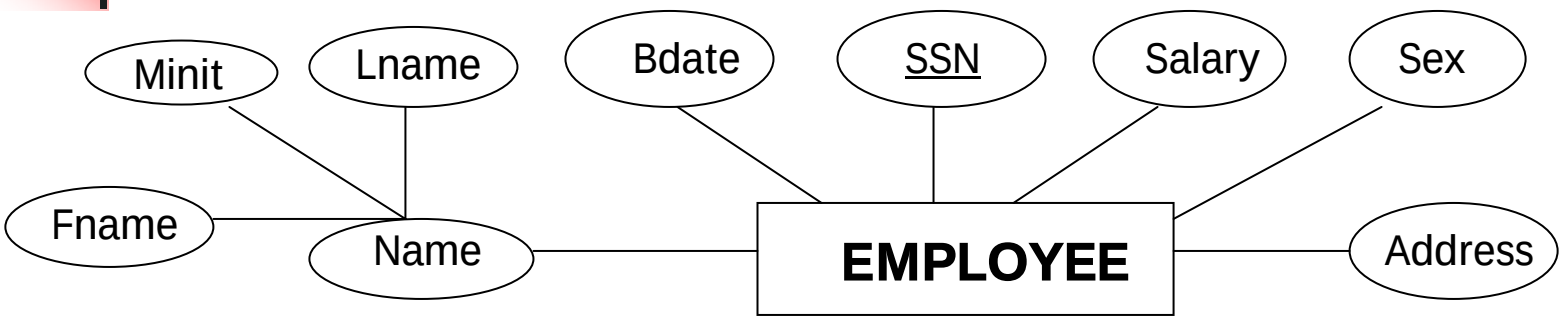
- For each regular (strong) entity type E in the ER schema:
  - Create a relation R that includes all the simple attributes of E.
  - Include only the simple components of any composite attribute.
  - Choose one of the key attributes of E as primary key for R.
  - If the chosen key of E is composite, then the set of simple attributes that form it will together form the primary key of R.



... -- Mapping of Regular Entity Types

- In the company database example, we create the relations EMPLOYEE, DEPARTMENT, and PROJECT to correspond to the regular entity types EMPLOYEE, DEPARTMENT, and PROJECT .
- The foreign key and relationship attributes, if any, are not included yet, they will be added during the subsequent steps. This includes the attributes of SUPERSSN and DNO of EMPLOYEE; MGRSSN and MGRSTARTDATE of DEPARTMENT; and DNUM of PROJECT.
- We choose SSN, DNUMBER and PNUMBER, as primary key for relations EMPLOYEE, DEPARTMENT, and PROJECT respectively

--- Mapping Regular Entity EMPLOYEE



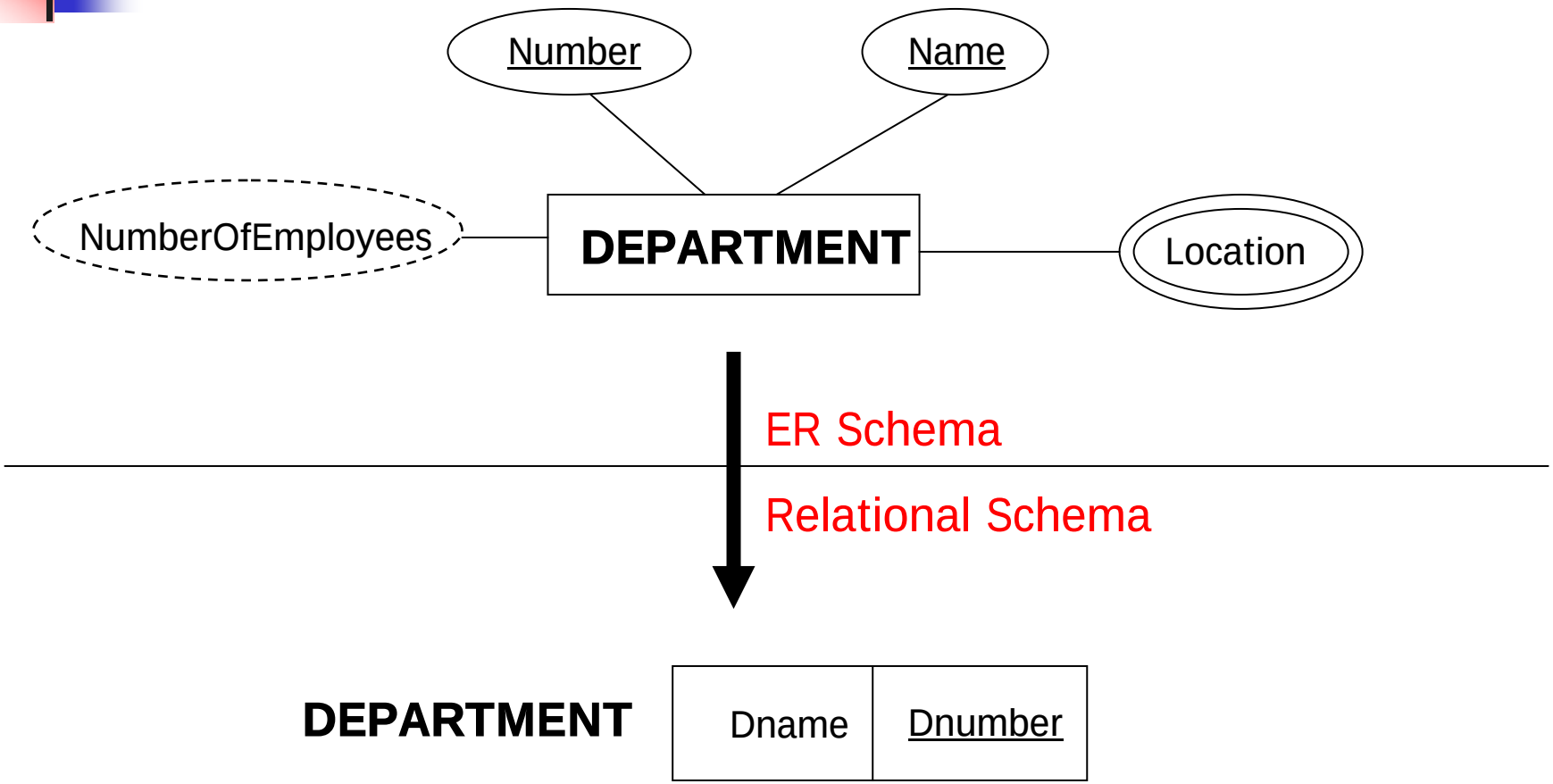
ER Schema

Relational Schema

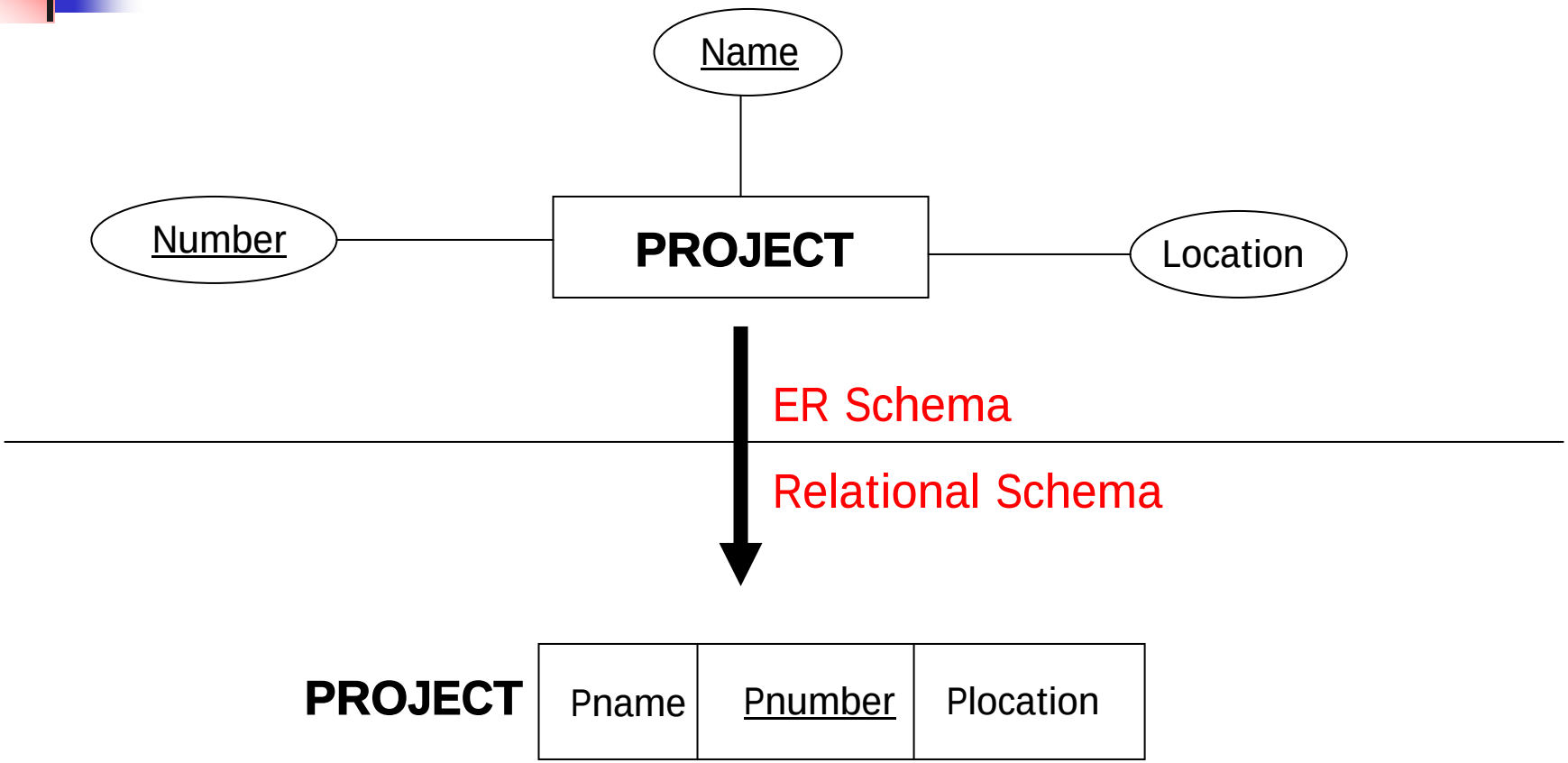
EMPLOYEE

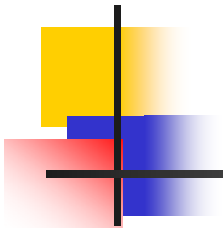
Fname	Minit	Lname	SSN	Bdate	Sex	Address	Salary

--- Mapping Regular Entity DEPARTMENT



--- Mapping Regular Entity PROJECT

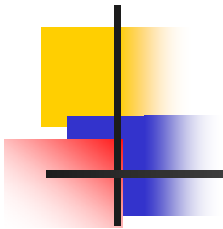




-- Mapping of Weak Entity Types ...

- For each weak entity type W in the ER schema with owner entity type E, create a relation R that includes all the simple attributes of W (or simple components of composite attributes) as attributes of R.

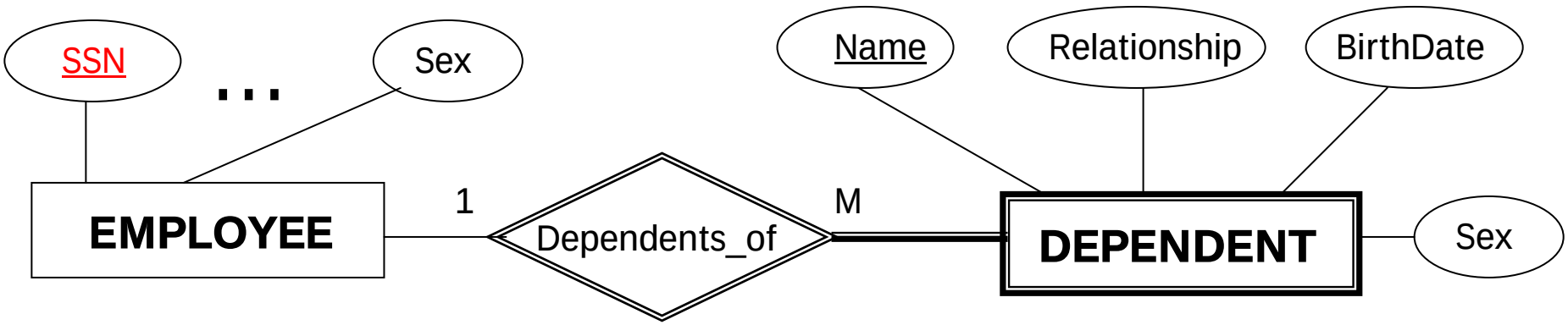
- In addition, include as foreign key attributes of R the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s); this takes care of the identifying relationship type of W. The primary key of R is the combination of the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any.



... -- Mapping of Weak Entity Types

- In the COMPANY database example, we created the relation DEPENDENT, in this step to correspond to the weak entity type DEPENDENT. We include the primary key SSN of the EMPLOYEE relation – which correspond to the owner entity type as a foreign key attribute of DEPENDENT; we renamed it ESSN; although this is not necessary. The primary key of dependent relation is the combination {ESSN, DEPENDENT_NAME} because DEPENDENT_NAME is the partial key of the weak entity DEPENDENT.

--- Mapping Weak Entity DEPENDENT

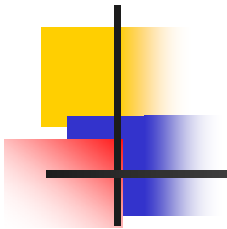


ER Schema

Relational Schema

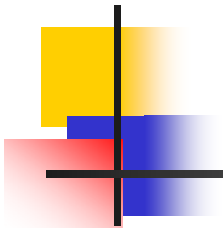
DEPENDENT

ESSN	DependentName	Sex	Bdate	Relationship



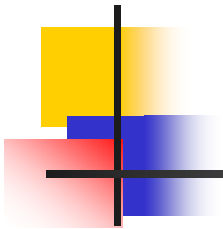
- Mapping Relationship Types

- Mapping of 1:1 Relationship Types +
- Mapping of 1:N Relationship Types +
- Mapping of M:N Relationship Types +
- Mapping of N-ary Relationship Types +



-- Mapping of 1:1 Relationship Types ...

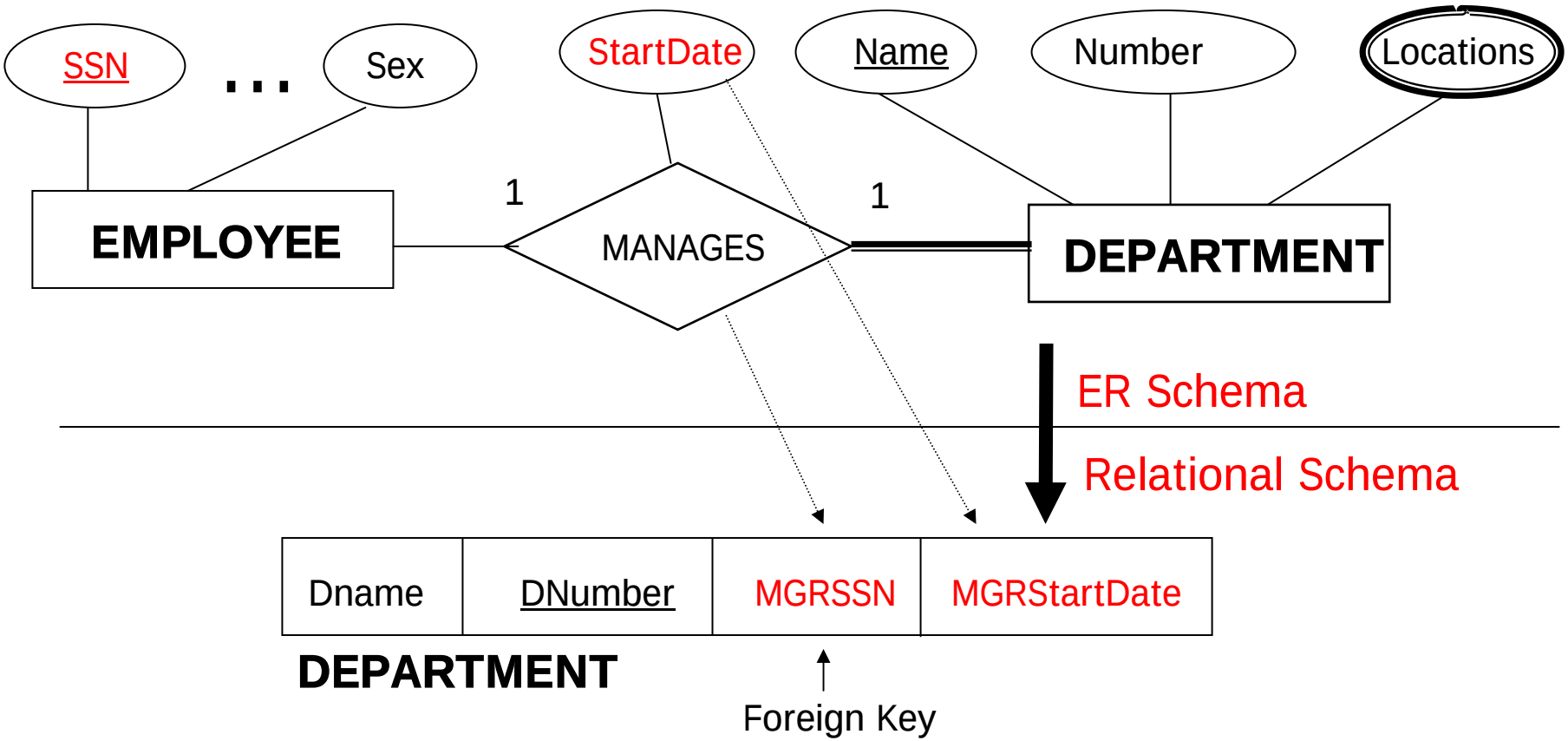
- For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R. Choose one of the relations, say S, and include as the foreign key in S the primary key of T. It is better to choose S as the entity type with total participation in R.

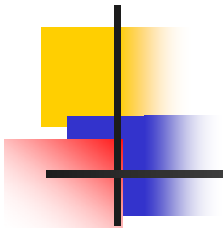


... -- Mapping of 1:1 Relationship Types ...

- In the COMPANY database example we map the 1:1 relationship type MANAGES, by choosing the participating entity type DEPARTMENT to serve in the role of S, because its participation in the MANAGES relationship type is total (Every department has a manager).
- We include the primary key of EMPLOYEE relation as a foreign key in the DEPARTMENT relation and we name it MRGSSN.
- We also include the simple attribute StartDate of the MANAGES relationship type in the DEPARTMENT relation and rename it as MGRSTARTDATE.

... -- Mapping of 1:1 Relationship Types





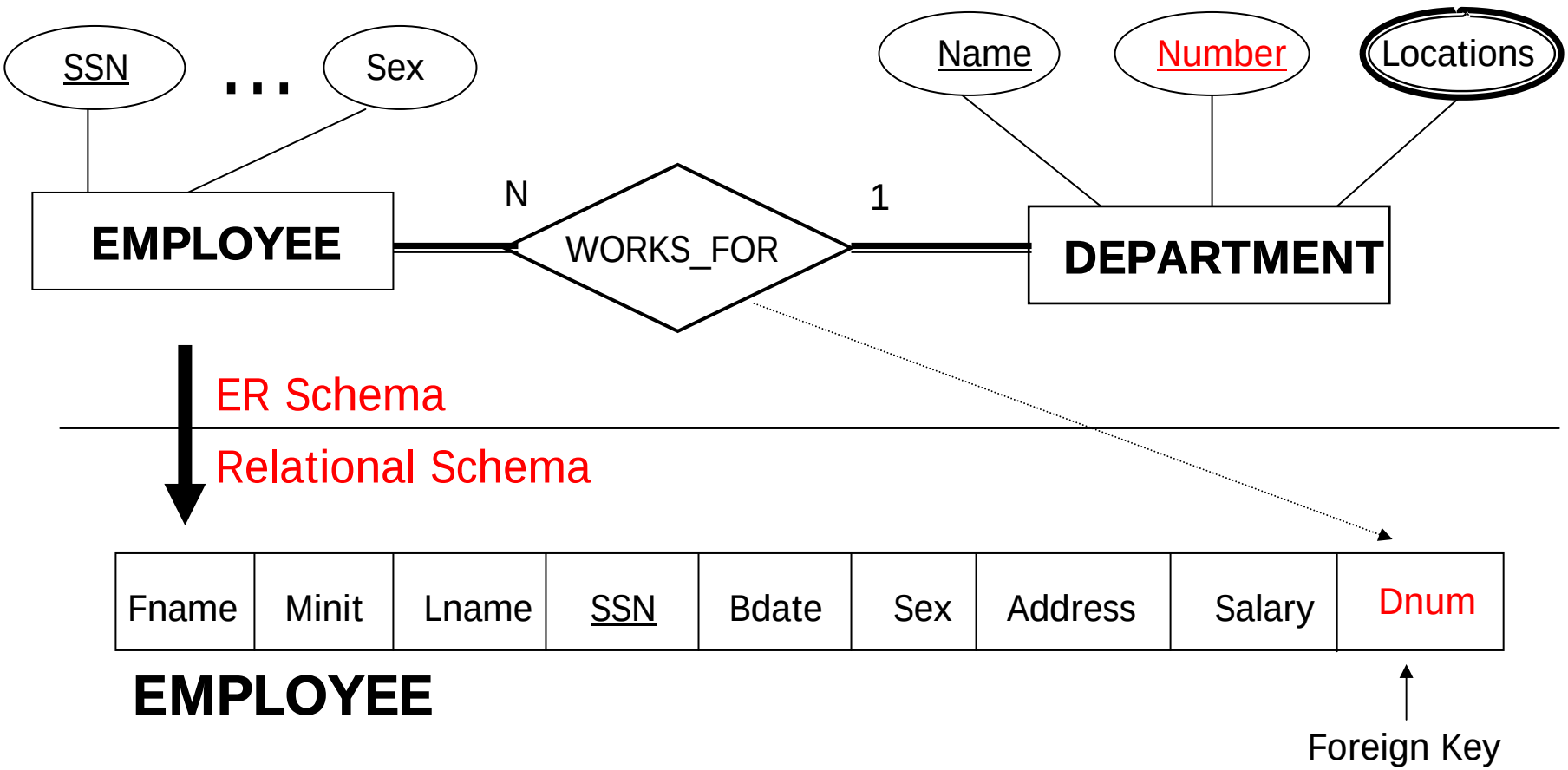
-- Mapping of 1:N Relationship Types ...

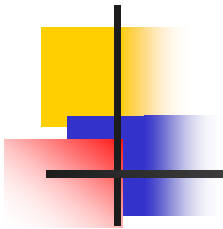
- For each regular binary 1:N relationship type R in the ER schema,
  - identify the relations S that represent the participating entity type at the N-side of R.
  - Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R; this is because each entity instance on the N-side is related to at most one entity instance one the 1-side of the relationship type.
  - Include any simple attributes (or simple components of composite attributes) of the 1:N relationship type as attributes of S.

... -- Mapping of 1:N Relationship Types ...

- In the COMPANY database example, we now map the 1:N relationship types WORKS_FOR, CONTROLS, and SUPERVISION.
  - For WORKS_FOR we include the primary key DNUMBER of the DEPARTMENT relation as a foreign key in the EMPLOYEE relation and call it DNO.
  - For CONTROLS we include the primary key DNUMBER of the DEPARTMENT relation as a foreign key in the project relation and call it DNUM.
  - For SUPERVISION we include the primary key SSN of the EMPLOYEE relation as a foreign key in the EMPLOYEE relation itself and call it SUPERSSN.

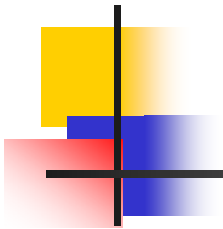
... -- Mapping of 1:N Relationship Types





-- Mapping of M:N Relationship Types ...

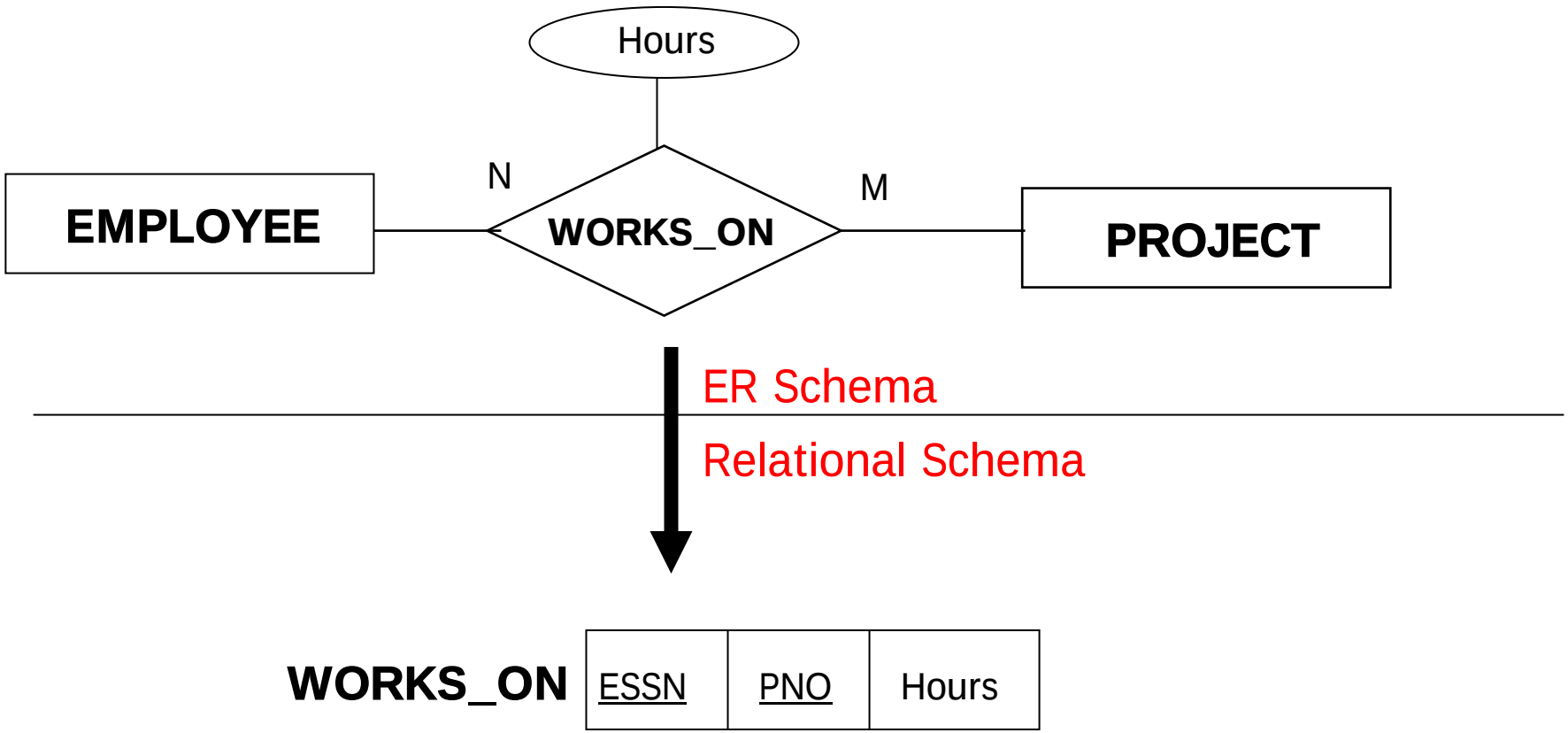
- For each binary M:N relationship type R in the ER schema
  - Create a new relation S to represent R.
  - Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types R.
  - The combination of these keys will represent the primary key for S.
  - Also include any simple attributes of R (or simple components of composite attributes) as attribute of S.

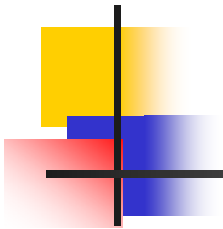


... -- Mapping of M:N Relationship Types

- In the COMPANY database example
  - we map M:N relationship type WORKS_ON by creating a new relation WORKS_ON.
  - We include the primary keys of the PROJECT and EMPLOYEE relations as foreign keys in WORKS_ON and rename them PNO and ESSN, respectively.
  - We also include the attribute HOURS in WORKS_ON to represent Hours attribute of the relationship type.
  - The primary key of the WORKS_ON relation is the combination {ESSN, DNO}

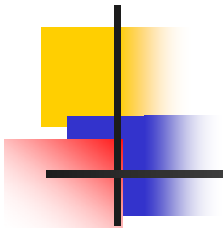
... -- Mapping of M:N Relationship Types





-- Mapping of N-ary Relationship Types ...

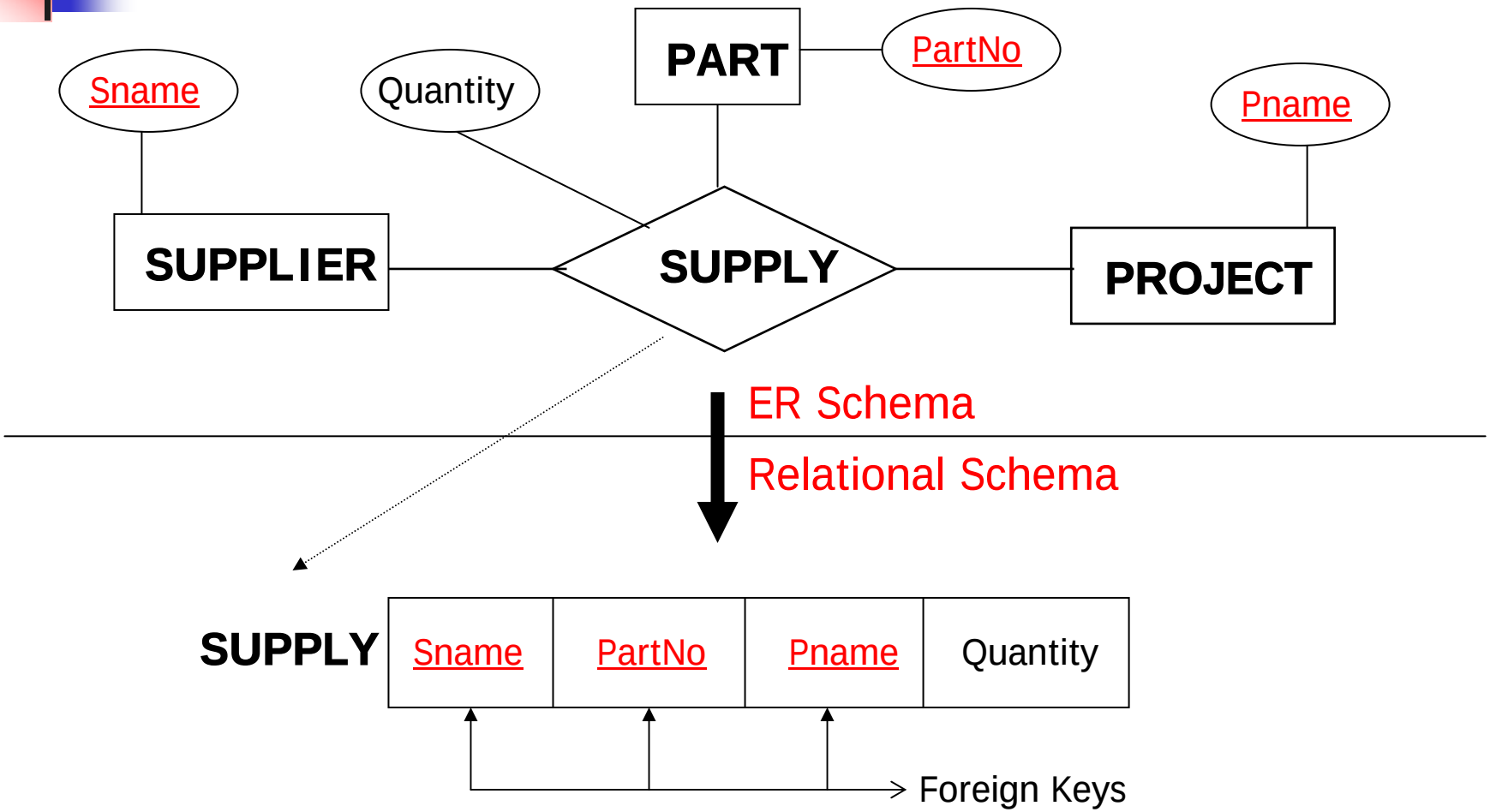
- For each n-ary relationship type R in the ER schema, where n > 2:
  - create a new relation S to represent R.
  - Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types in R.
  - The combination of these keys will represent the primary key of S.
  - Also, include any simple attributes of R (or simple components of composite attributes) as attributes of S.

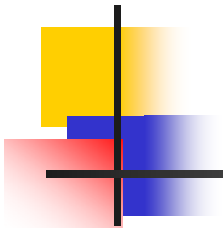


... -- Mapping of N-ary Relationship Types

- For example, the relationship type SUPPLY (shown in the following figure) can be mapped to the relation SUPPLY whose primary key is the combination of foreign keys {SNAME, PARTNO, PROJNAME}

... -- Mapping of N-ary Relationship Types





- Mapping Multi-valued Attributes ...

- For each multi-valued attribute Am create anew relation R.
- This relation R will include an attribute corresponding to A plus the primary key attribute K – as a foreign key in R – of the relation that represents the entity type or relationship type that has A as an attribute.
- The primary key of R is the combination of A and K.
- If the multi-valued attribute is composite, we include its simple components.

... - Mapping Multi-valued Attributes ...

- In the COMPANY database example:
  - We create a relation DEPT_LOCATIONS.
  - The attribute DLOCATION represents the multi-valued attribute Locations of DEPARTMENT.
  - DNUMBER – as foreign key – represents the primary key of the DEPARTMENT relation.
  - The primary key of DEPT_LOCATION is the combination {DNUMBER, DLOCATION}.
  - A separate tuple will exist in DEPT_LOCATIONS for each location that a department has.

... - Mapping Multi-valued Attributes