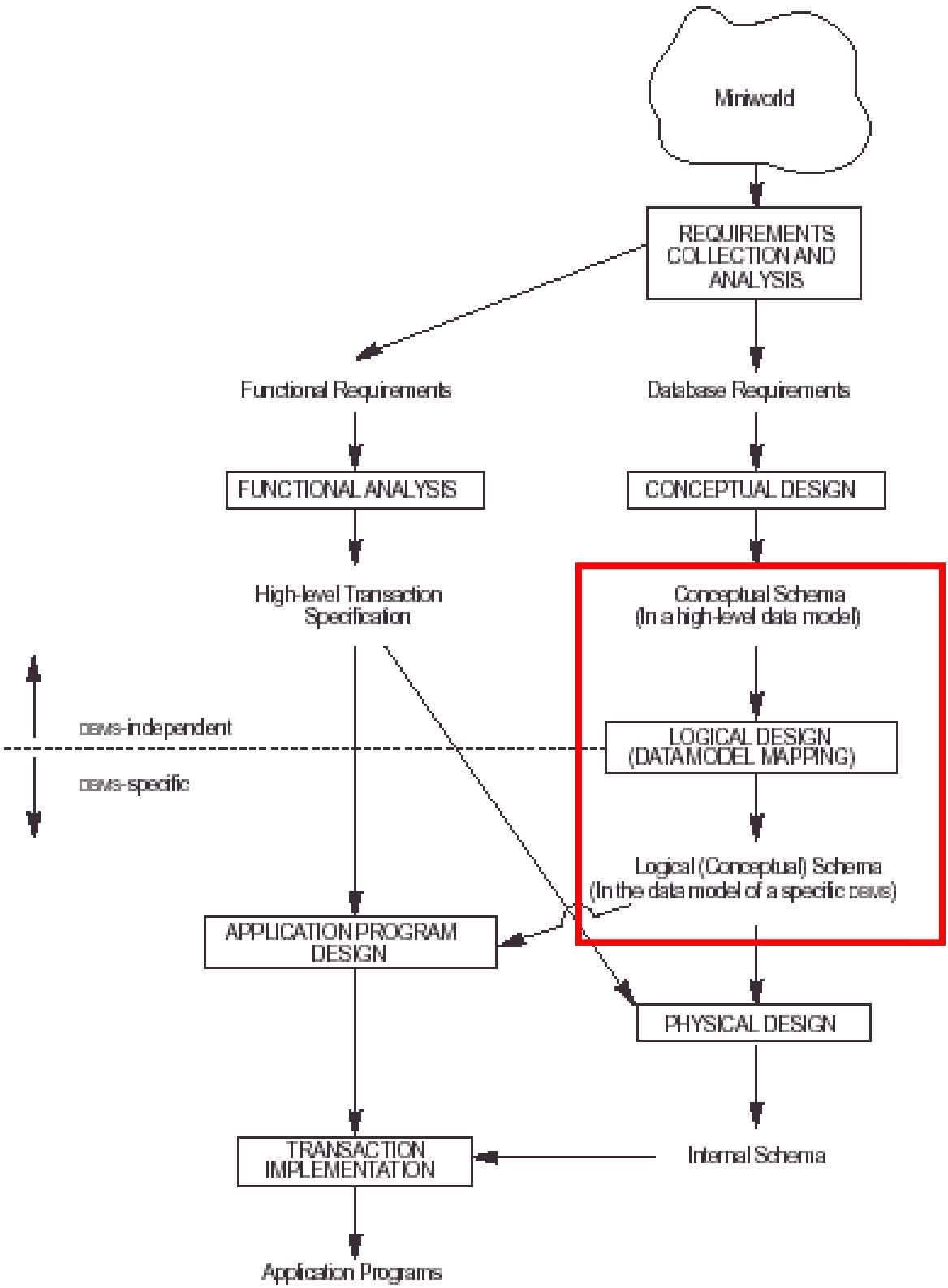
FROM CONCEPTUAL DATA MODELS TO LOGICAL DATA MODELS

MAPPING THE ER AND EER MODELS TO THE RELATIONAL MODEL

0. Database Design Process

The main phases of database design:



1. Relational Database Design Using EER to Relational Mapping

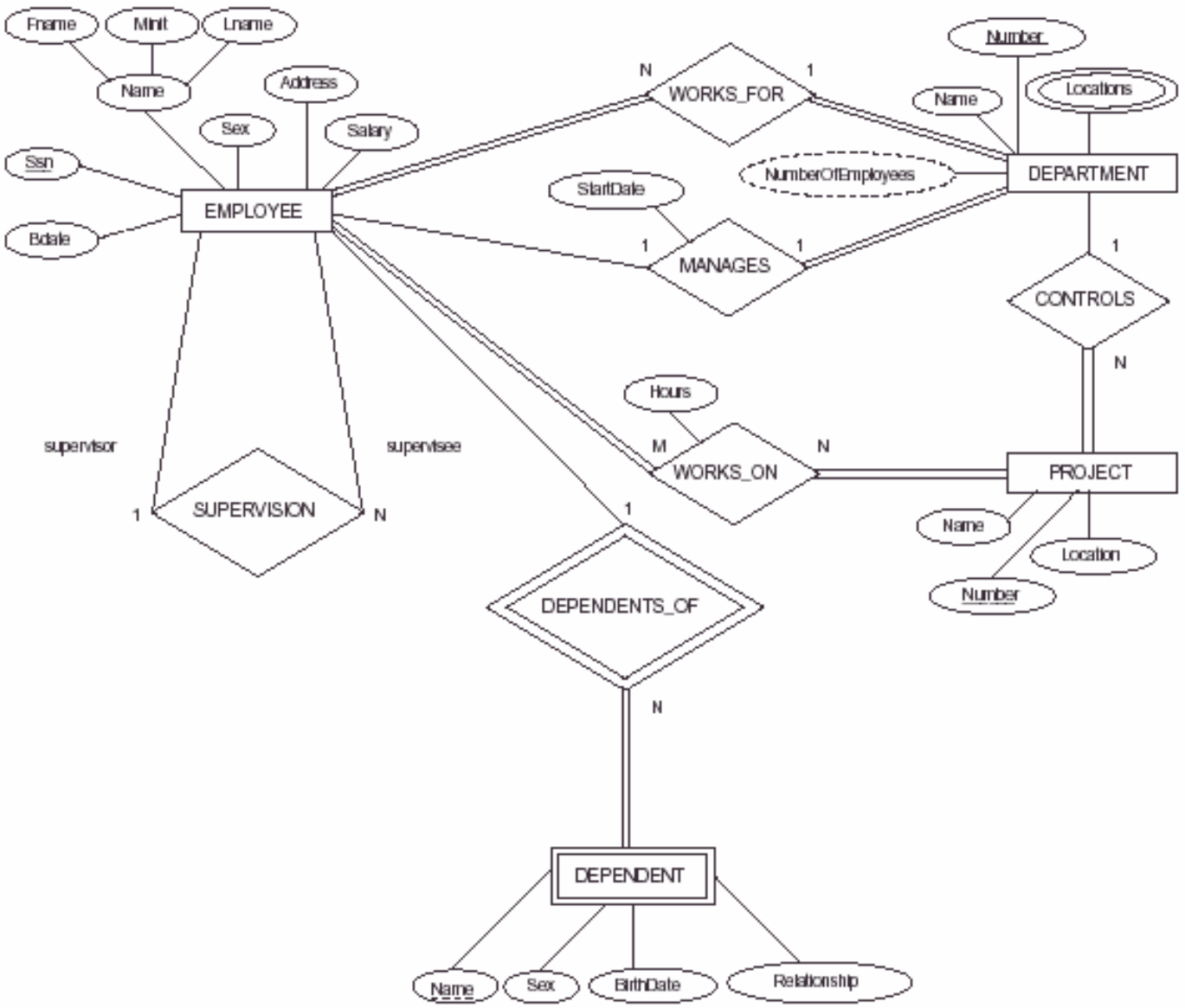
- The logical database design step in the design of a database maps EER concepts to Relational Model concepts.

- Many CASE tools automatically convert ER diagrams into Relational database schemas in the DDL of a specific relational DBMS.

- We outline an algorithm that maps an ER schema into the corresponding Relational Schema. The steps of the algorithm follow.

1. Relational Database Design Using EER to Relational Mapping

- We use the ER schema of the COMPANY database as a running example.



1.1 Mapping regular entity types (1)

- **Step 1:**

  For each *regular (non-weak) entity type* E in the ER schema, create a relation R that includes all the *simple (single-valued)* attributes of E. Include only the simple components of composite attributes.

  Primary key: Chose one of the key attributes of E as primary key for R. If the chosen key of R is composite, the set of simple attributes of the composite key form the primary key of R.

1.1 Mapping regular entity types (2)

Example

EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY

DEPARTMENT

DNAME	DNUMBER

PROJECT

PNAME	PNUMBER	PLOCATION

1.2 Mapping weak entity types (1)

- **Step 2:**

  For each *weak entity type W* in the ER schema with owner entity type E, create a relation R and include all simple (single-valued) attributes (or simple components of composite attributes) of W as attributes of R. In addition, include as foreign key attributes of R the primary key attribute(s) of the owner entity type(s).

  Primary key: The primary key of R is the combination of the primary key(s) of the owner entity type(s), and the partial key of the weak entity type W (if any).

1.2 Mapping weak entity types (2)

Example

EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY

DEPARTMENT

DNAME	DNUMBER

PROJECT

PNAME	PNUMBER	PLOCATION

DEPENDENT

ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP

1.3 Mapping 1:1 binary relationship types (1)

- **Step 3:**

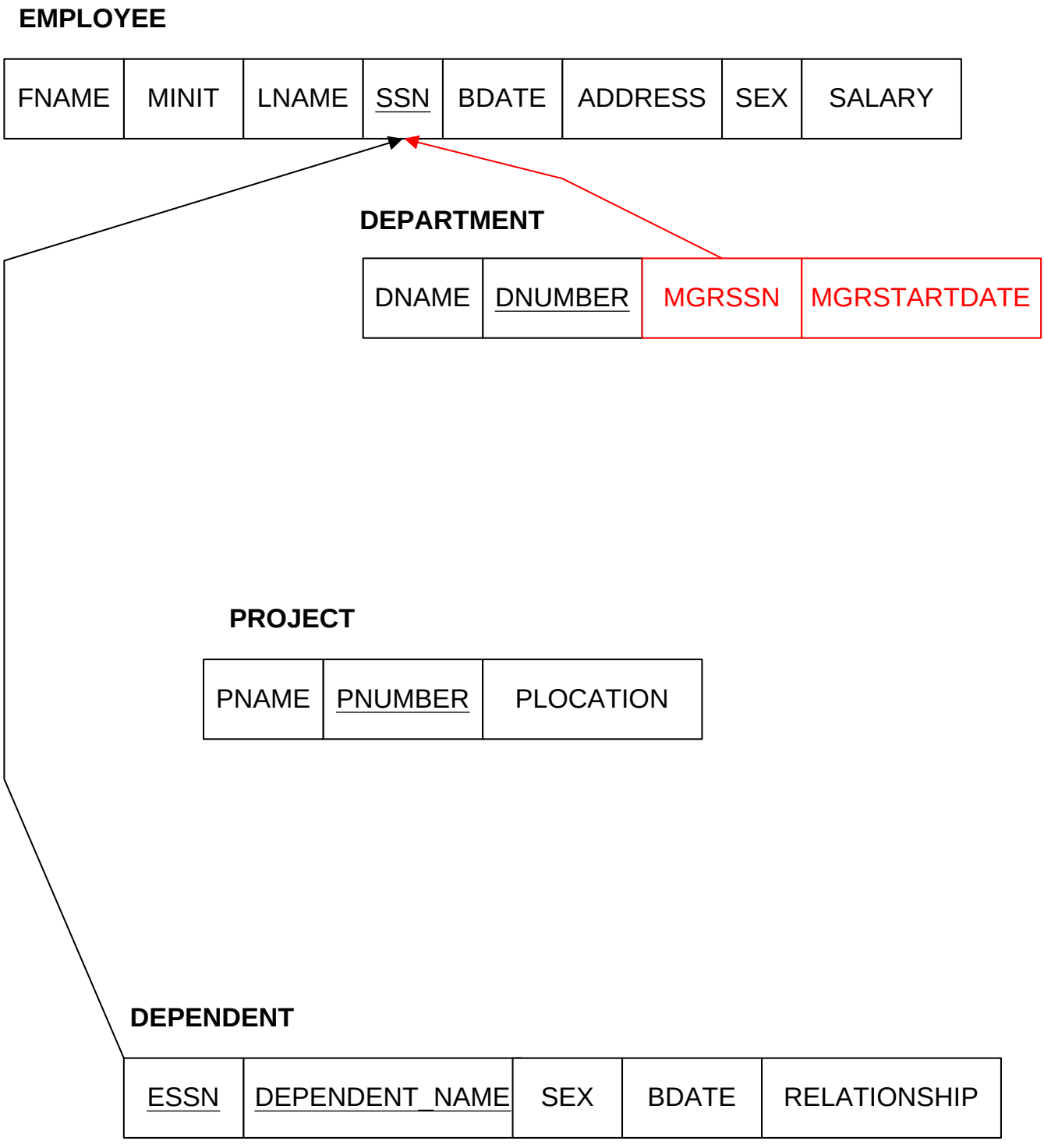
  For each *binary 1:1 relationship type R* in the ER schema, identify the relations S and T that correspond to the entity types participating in R. Choose one of the relations – say S – and include as foreign key in S the primary key of T. Include all the simple (single-valued) attributes (or simple components of composite attributes) of the 1:1 relationship type as attributes of S.

  Remark: It is better to chose an entity type with *total participation in R* in the role of relation S.

  Alternative: Merge the two relations that correspond to the participating entity types into one relation. This solution is appropriate when *both participations are total*.

1.3 Mapping 1:1 binary relationship types (2)

Example



1.4 Mapping 1:N binary relationship types (1)

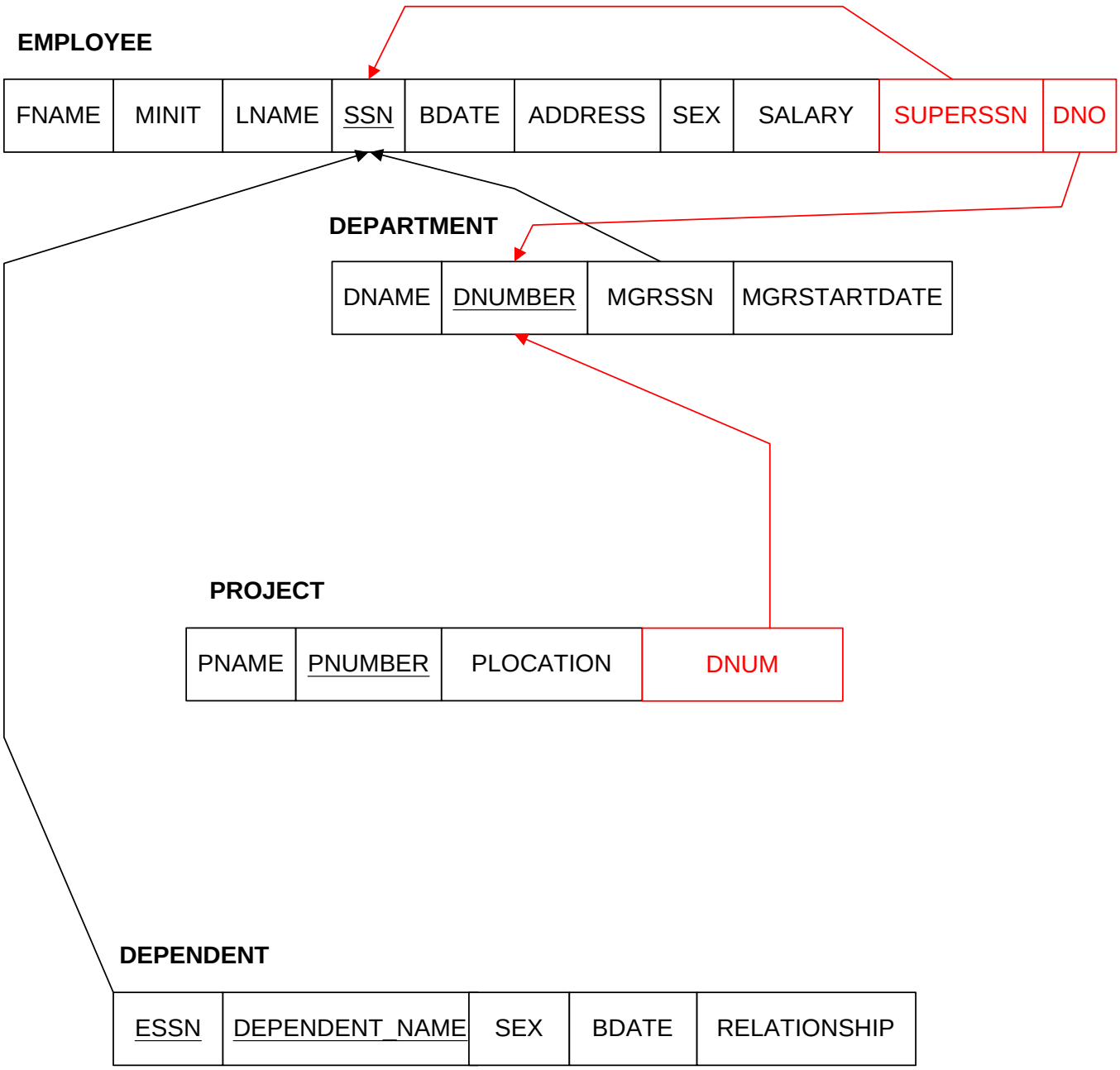
- **Step 4:**

  For each *regular binary 1:N relationship type R*, identify the relation S that represents the participating entity type at the N-side of the relationship type. Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R. Include any simple (single-valued) attributes (or simple components of composite attributes) of the 1:N relationship type as attributes of S.

  Remark: The entity type at the N-side of the relationship type is chosen because each entity instance of this entity type is related to at most one entity instance of the entity type in the 1-side of the relationship type.

1.4 Mapping 1:N binary relationship types (2)

Example



1.5 Mapping M:N binary relationship types (1)

- **Step 5:**

  For each *binary M:N relationship type R*, 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. Also include as attribute of S any simple (single-valued) attribute (or simple component of composite attribute) of the M:N relationship type

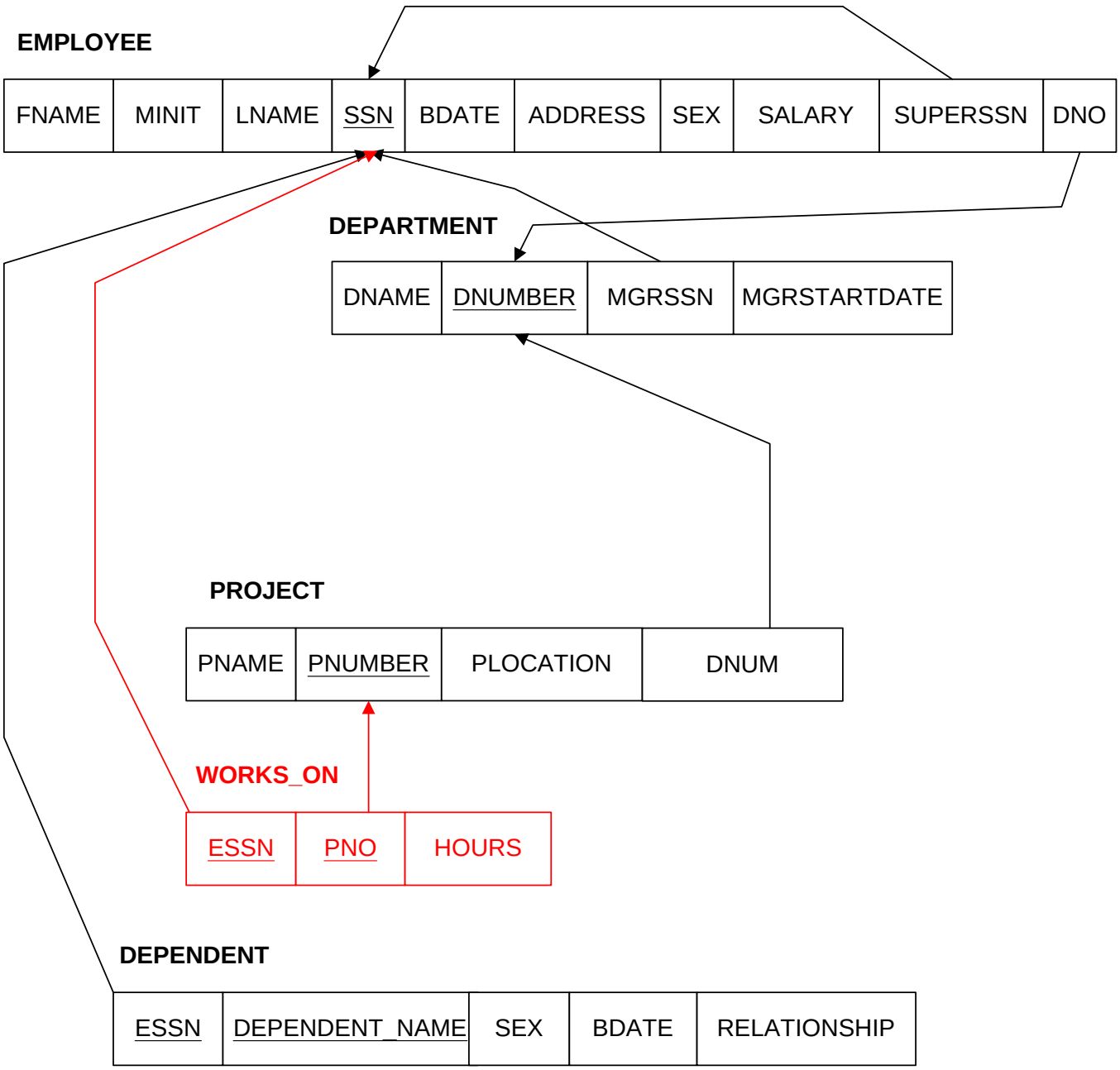
  Primary key: The combination of the key attributes of the relations that correspond to the participating entity types will form the primary key of S.

  Remark: We cannot represent an M:N relationship type by a single foreign key attribute in one of the participating relations because of the M:N cardinality ratio.

  Remark: We can always map 1:N and 1:1 relationships in a manner similar to M:N relationships.

1.5 Mapping M:N binary relationship types (2)

Example



1.6 Mapping multivalued attributes (1)

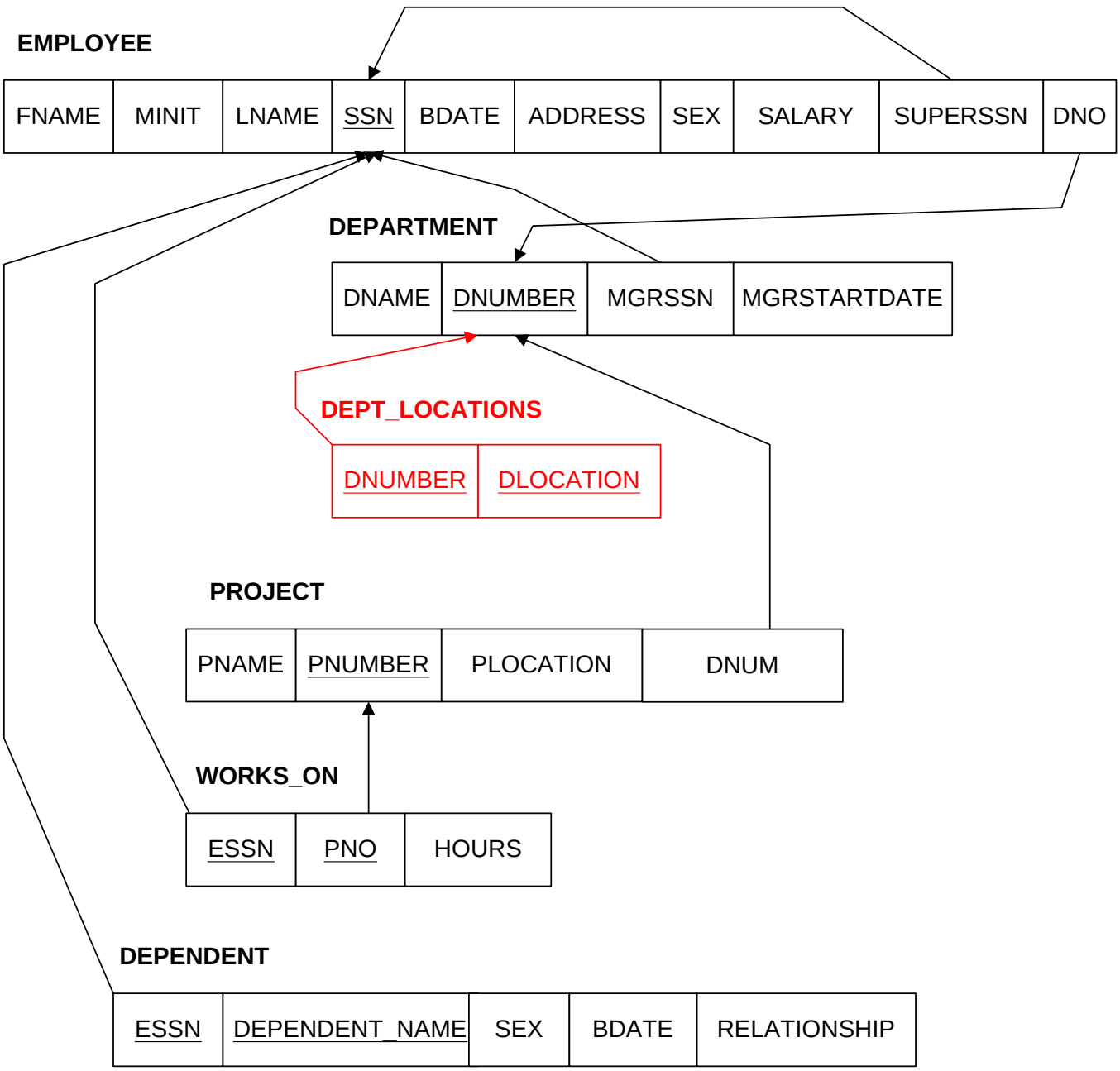
- **Step 6:**

  For each *multivalued attribute* A create a new relation R. Relation R will include an attribute corresponding to A (or its simple component attributes if A is composite), plus the primary key attribute K -- as a foreign key in R-- of the relation corresponding to the entity type or relationship type that has A as an attribute.

  Primary key: The primary key of R is the combination of A and K.

1.6 Mapping multivalued attributes (2)

Example



1.7 Mapping regular n-ary relationship types (1)

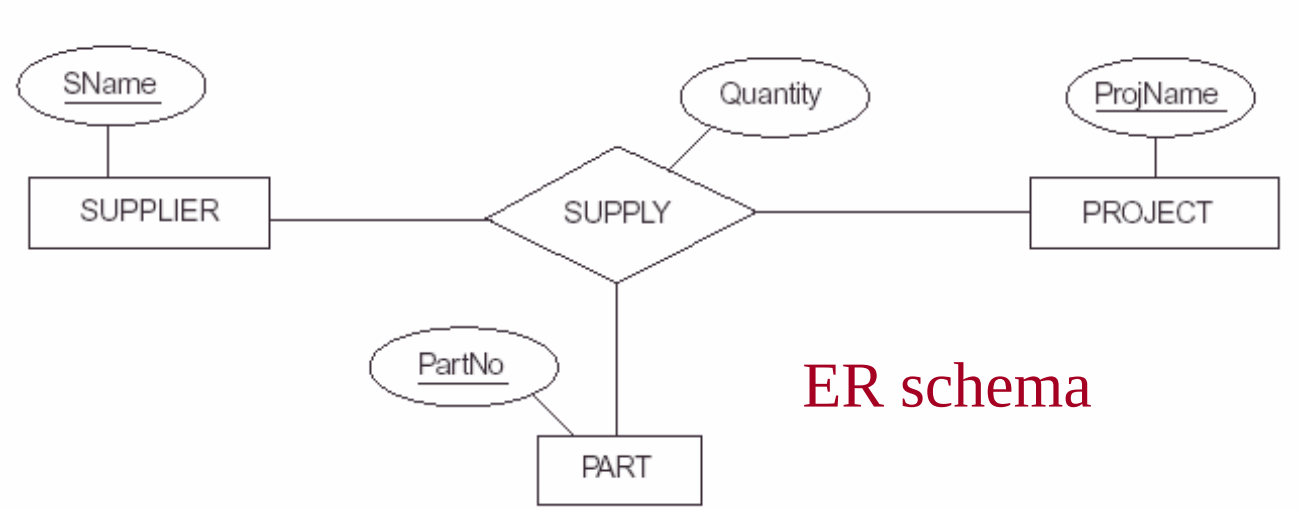
- **Step 7:**

  For each *n-ary relationship type* R, 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. Also include any simple (single-valued) attributes (or simple component attributes of a composite attribute) of the n-ary relationship type as attributes of S.

  Primary key: The primary key of S is usually the combination of all the foreign keys that reference the relations representing the participating entity types.

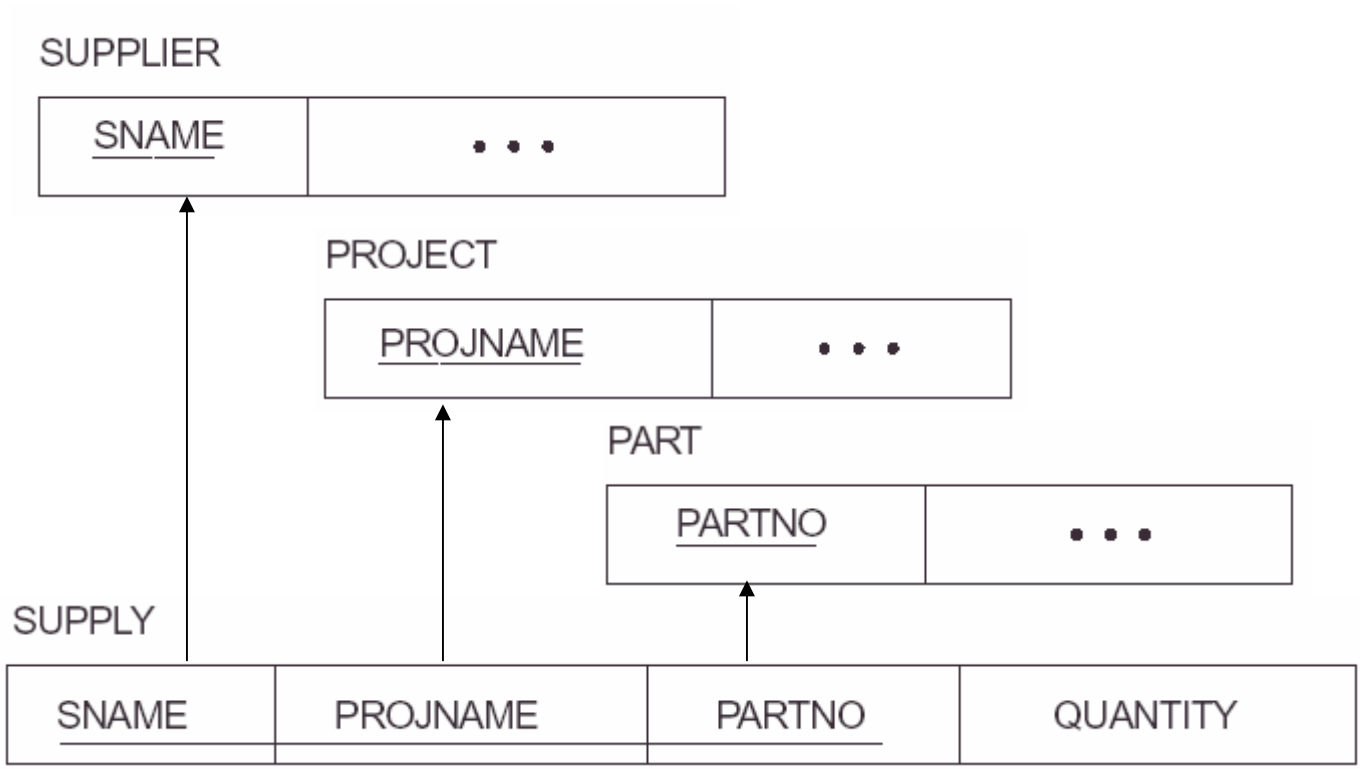
1.7 Mapping regular n-ary relationship types (2)

Example



ER schema

Relational schema



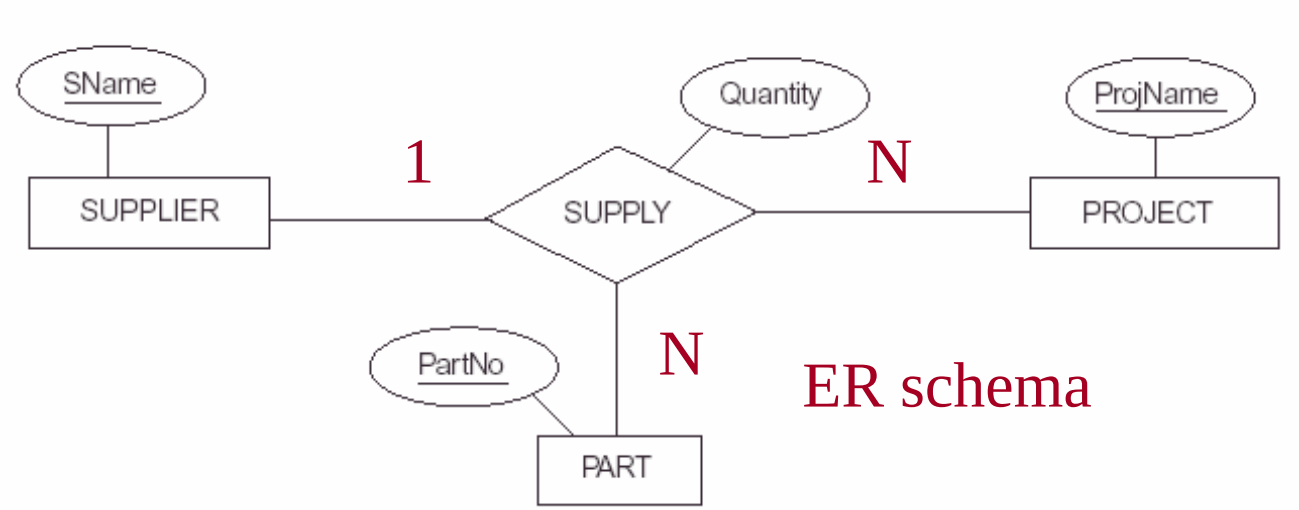
1.7 Mapping regular n-ary relationship types (3)

- Remark 1

  If the cardinality constraint of any of the entity types E participating in R is 1, then the primary key of S should not include the foreign key attribute that references the relation corresponding to E.

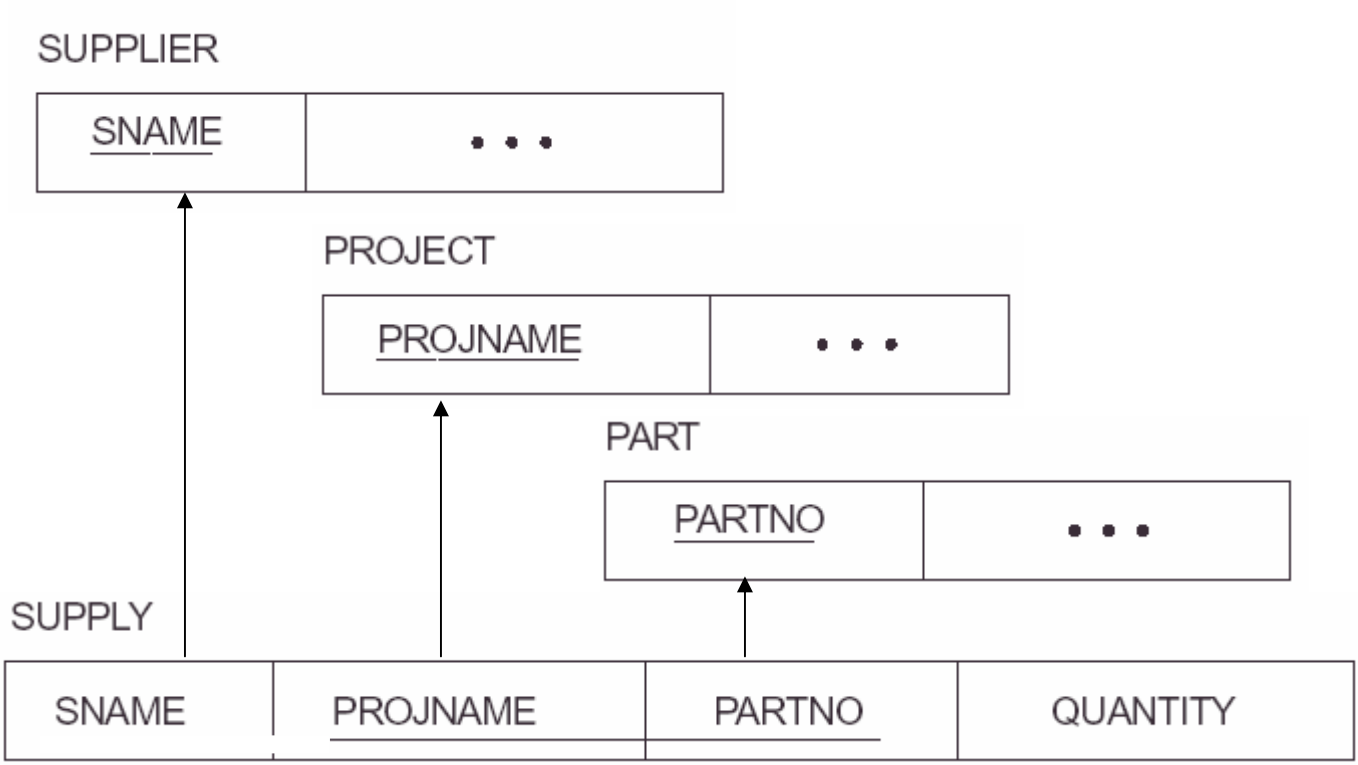
1.7 Mapping regular n-ary relationship types (4)

Example



ER schema

Relational schema



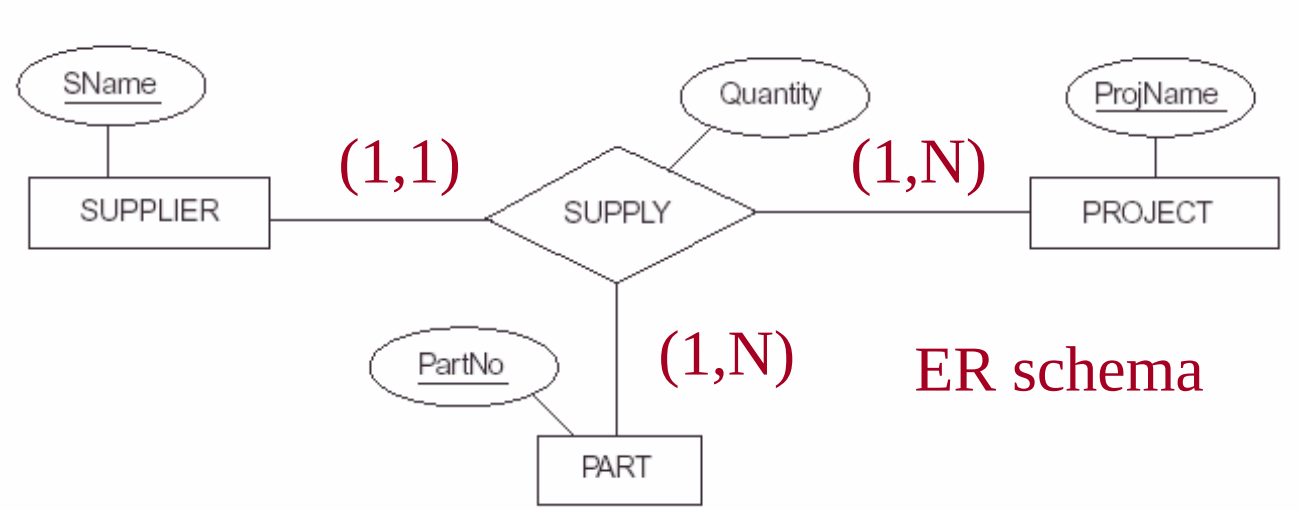
1.7 Mapping regular n-ary relationship types (5)

- Remark 2

  If max = 1 for an entity type participating in R, then the primary key of S should not include the foreign key attributes that reference the relations corresponding to the other participating entity types.

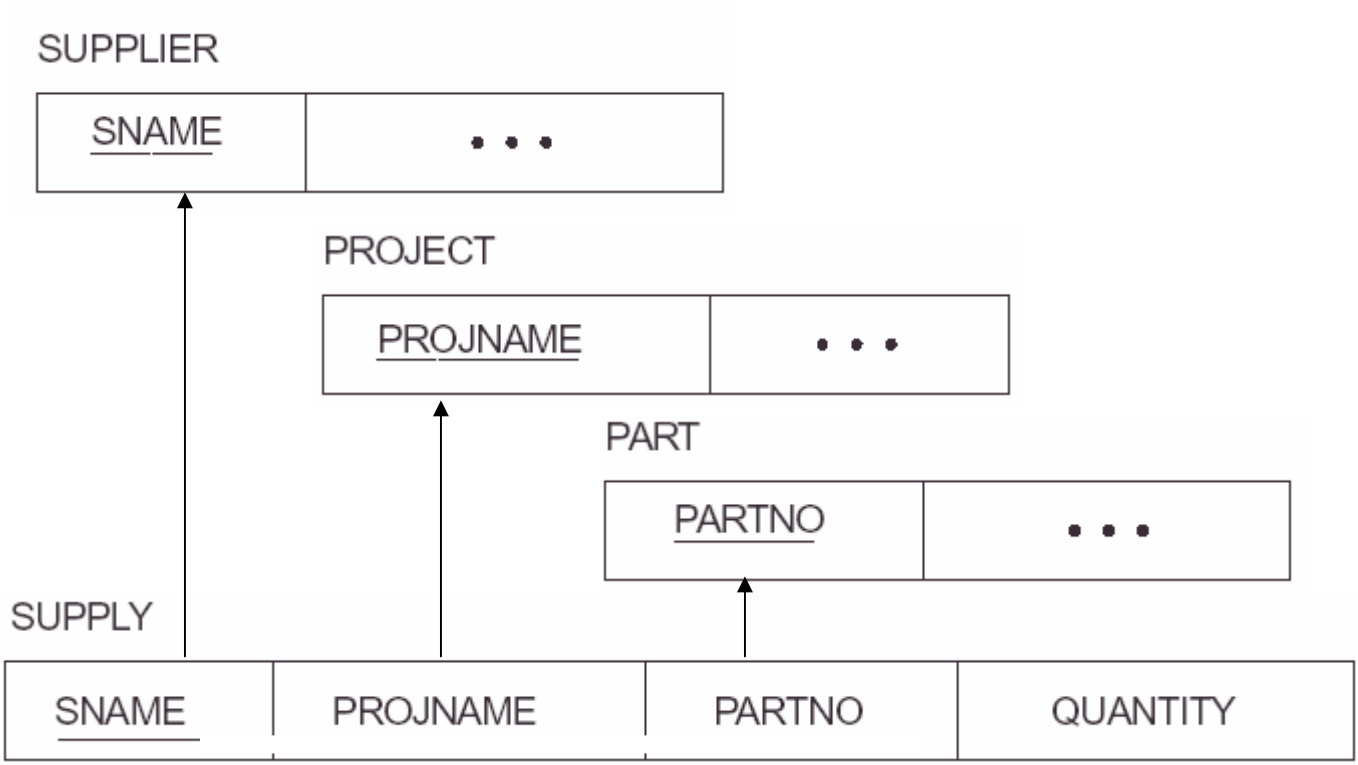
1.7 Mapping regular n-ary relationship types (6)

Example



ER schema

Relational schema



1.8 Mapping Superclass/Subclass Relationships and Specialization (1)

- **Step 8:**

  We examine 4 different (the most common) options.

  Attrs(R) denote the attributes of relation R
  PK(R) denote the primary key of R

  We assume a superclass C with m subclasses $S_1, \ldots, S_m$.
  C has attributes $K, A_1, \ldots, A_n$.
  K is the key of C.

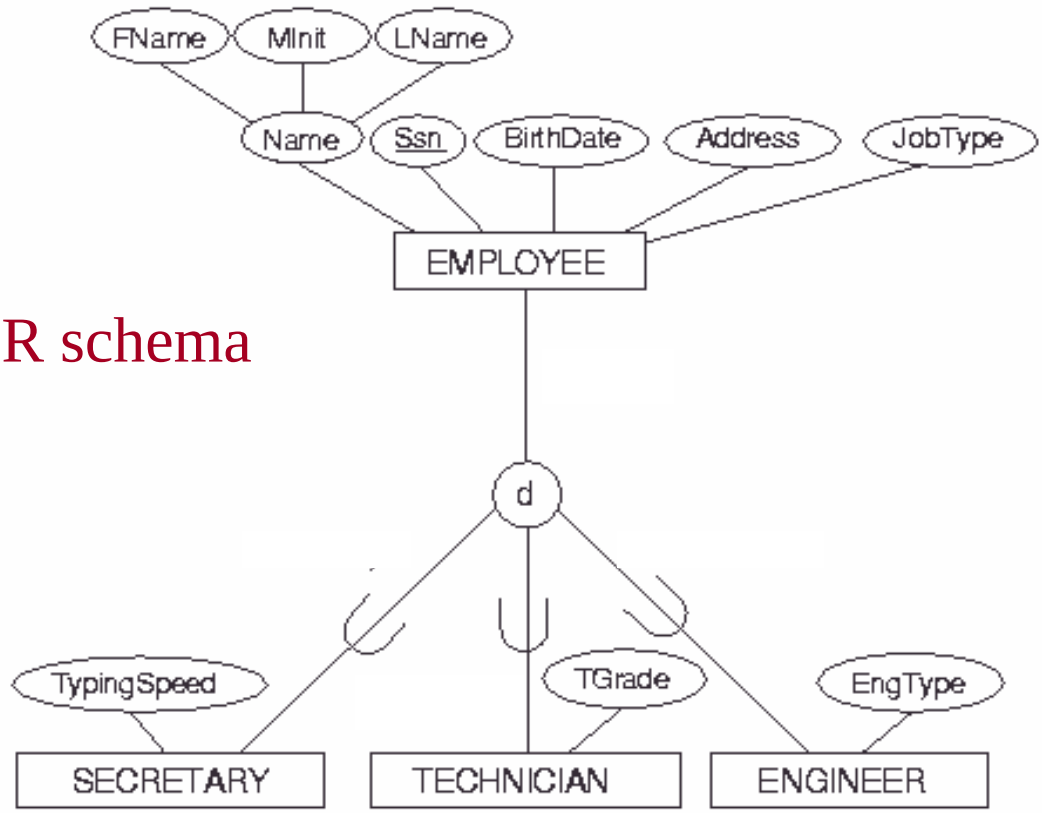
- **Option A**

  Create a relation L for C with attributes $Atts(L) = \{K, A_1, \ldots, A_n\}$, and PK(L) = K.

  Create a relation L_i for each subclass S_i, $1 \leq i \leq m$ such that $Atts(L_i) = \{K\} \cup \{\text{attributes of } S_i\}$, $PK(L_i) = K$, and K is a foreign key to L.

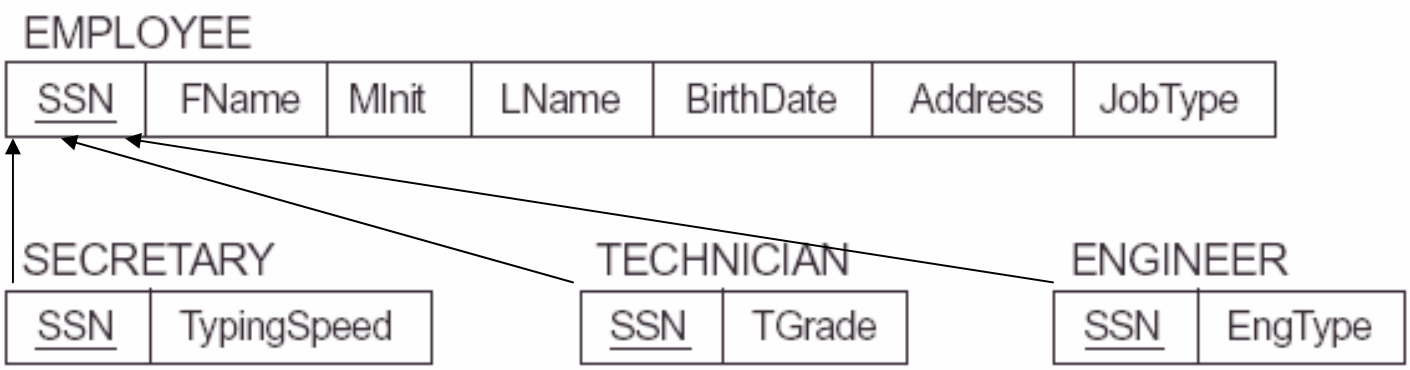
1.8 Mapping Superclass/Subclass Relationships and Specialization (2)

- **Example** for option A:



ER schema

Relational schema



1.8 Mapping Superclass/Subclass Relationships and Specialization (3)

- Remarks for Option A:

  Option A works for any constraint on specialization: disjoint or overlapping, total or partial.

1.8 Mapping Superclass/Subclass Relationships and Specialization (4)

- We assume a superclass C with m subclasses $S_1, \ldots, S_m$.
  C has attributes $K, A_1, \ldots, A_n$.
  K is the key of C.

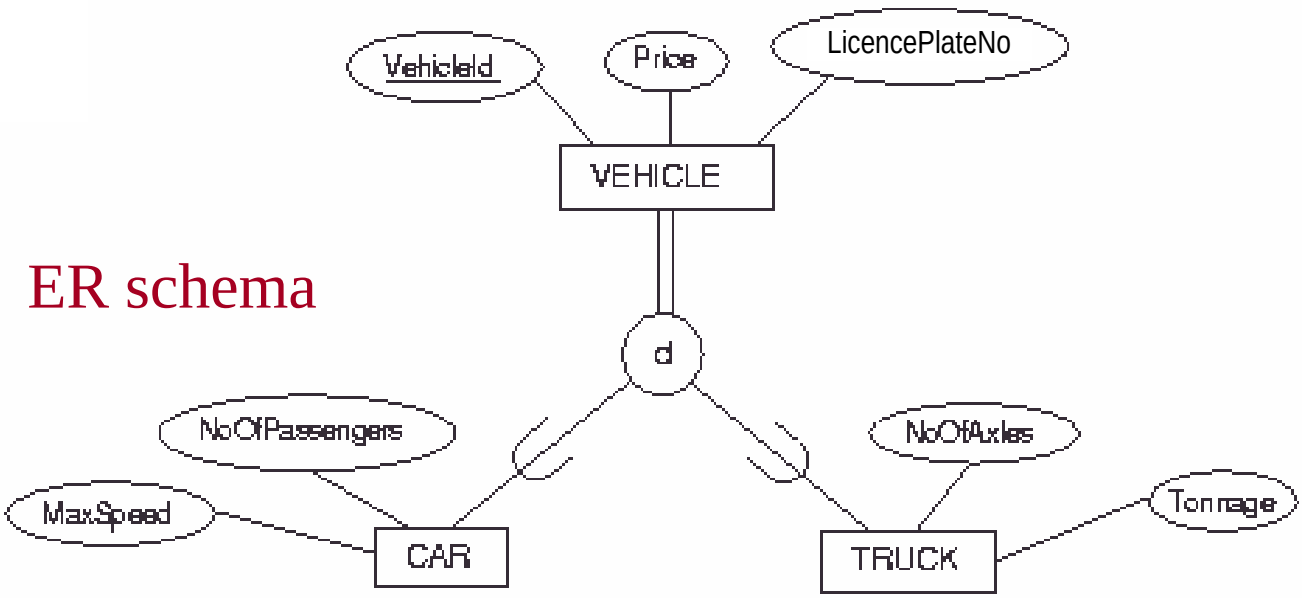
- **Option B**

  Create a relation L_i for each subclass S_i, $1 \leq i \leq m$ such that
  $Atts(L_i) = \{\text{attributes of } S_i\} \cup \{K, A_1, \ldots, A_n\}$ and
  $PK(L_i) = K$.

1.8 Mapping Superclass/Subclass Relationships and Specialization (5)

- **Example** for option B:

ER schema



CAR

VehicleId	LicensePlateNo	Price	MaxSpeed	NoOfPassengers

TRUCK

VehicleId	LicensePlateNo	Price	NoOfAxles	Tonnage

1.8 Mapping Superclass/Subclass Relationships and Specialization (6)

- Remarks for Option B:

  Option B works well only when *both the disjoint and total constraints hold*.

1.8 Mapping Superclass/Subclass Relationships and Specialization (7)

We assume a superclass C with m subclasses $S_1, \ldots, S_m$.
C has attributes $K, A_1, \ldots, A_n$.
K is the key of C.

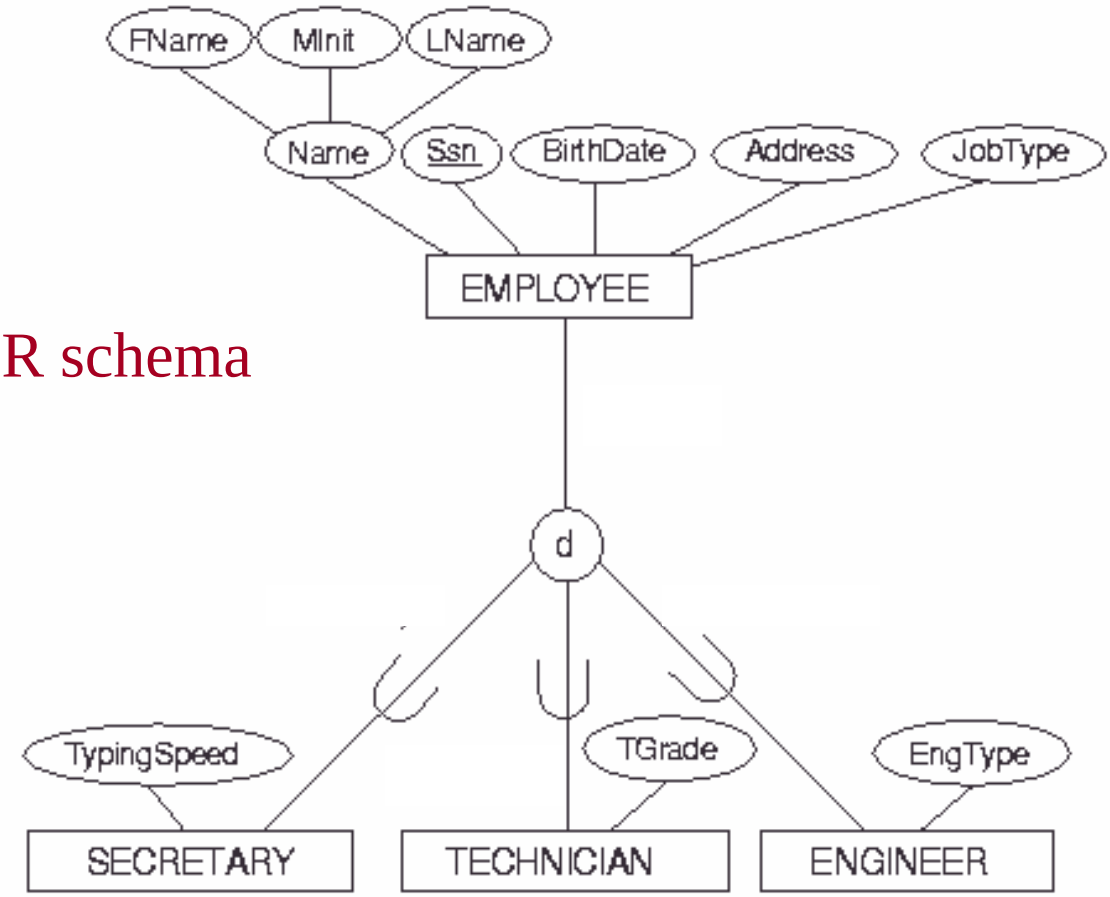
- **Option C**

Create a single relation L with attributes
$Atts(L) = \{K, A_1, \ldots, A_n\} \cup \{\text{attributes of } S_1\} \cup \ldots \cup \{\text{attributes of } S_m\} \cup \{T\}$, and
$PK(L) = K$.

T is a **type attribute** that indicates the subclass to which each tuple belongs, if any.

1.8 Mapping Superclass/Subclass Relationships and Specialization (8)

- **Example** for option C:



ER schema

Relational schema

EMPLOYEE

SSN	FName	MInit	LName	BirthDate	Address	JobType	TypingSpeed	TGrade	EngType

JobType is the type attribute.

1.8 Mapping Superclass/Subclass Relationships and Specialization (9)

- Remarks for Option C:

  Option C is for a specialization whose subclasses are *disjoint*.

  An entity will have null values for all the specific (local) attributes of the subclasses in which it does not belong.

  Therefore, it is not recommended if the subclasses have many local attributes.

1.8 Mapping Superclass/Subclass Relationships and Specialization (10)

We assume a superclass C with m subclasses $S_1, \ldots, S_m$.
C has attributes $K, A_1, \ldots, A_n$.
K is the key of C.

- **Option D**

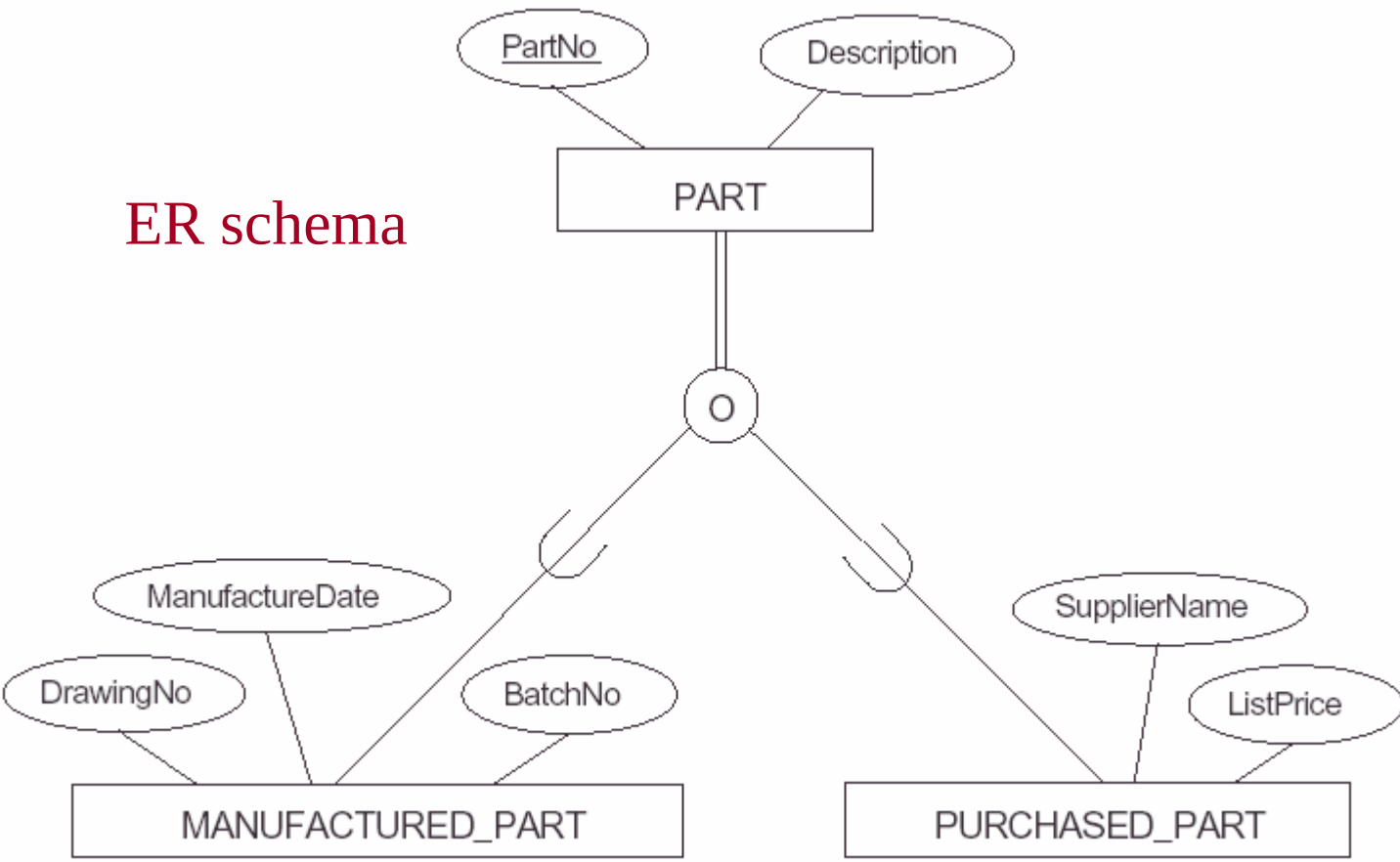
Create a single relation L with attributes
$Atts(L) = \{K, A_1, \ldots, A_n\} \cup \{\text{attributes of } S_1\} \cup \ldots \cup \{\text{attributes of } S_n\} \cup \{T_1, \ldots, T_m\}$, and
$PK(L) = K$.

Each T_i is a Boolean attribute indicating whether a tuple belongs to subclass S_i.

1.8 Mapping Superclass/Subclass Relationships and Specialization (11)

- **Example** for option D:

ER schema



Relational schema

PART

PartNo	Description	MFlag	DrawingNo	ManufactureDate	BatchNo	PFlag	SupplierName	ListPrice

MFlag and PFlag are Boolean type attributes.

1.8 Mapping Superclass/Subclass Relationships and Specialization (12)

- Remarks for Option D:

  Option D works for any constraint on specialization: disjoint or overlapping, total or partial.

  An entity that does not belong to some of the subclasses will have null values for all specific (local) attributes of these subclasses.

  Therefore, Option D (like option C) is not recommended if the subclasses have many local attributes.

1.8 Mapping Superclass/Subclass Relationships and Specialization (13)

- Remark

  When we have a multilevel specialization (or generalization) hierarchy or lattice, we do not have to follow the same mapping option for all the specializations.

  One mapping option can be used for part of the hierarchy or lattice and other options for other parts.