

Database Design Using ER to Relational Mapping

Overview

- **ER-to-Relational Mapping Algorithm**
 - Step 1: Mapping of Regular Entity Types
 - Step 2: Mapping of Weak Entity Types
 - Step 3: Mapping of Binary 1:1 Relation Types
 - Step 4: Mapping of Binary 1:N Relationship Types.
 - Step 5: Mapping of Binary M:N Relationship Types.
 - Step 6: Mapping of Multivalued attributes.
 - Step 7: Mapping of N-ary Relationship Types.

Mapping regular entity types

- Relations (tables) correspond with *entity types* and with *many-to-many relationship types*
- Rows correspond with *entity instances* and with *many-to-many relationship instances*
- Columns correspond with *attributes*

Mapping regular entity types

- For each regular entity type E in ER, create a relation R that includes all the simple attributes of E
- Choose one of the key attributes of E as primary key of R
- If the chosen key is a composite, then the set of simple attributes that form it will together form pk of R

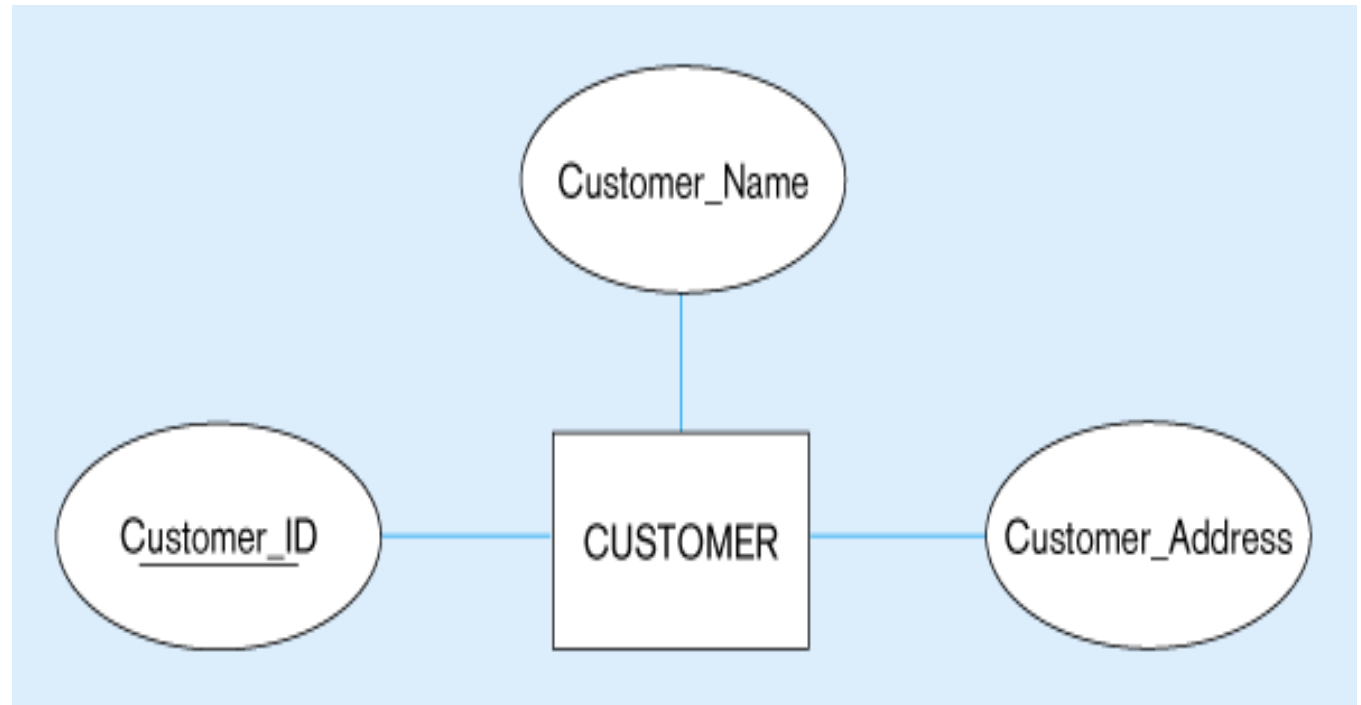
Mapping regular entity types

Mapping Regular Entities to Relations

1. **Simple attributes:** E-R attributes map directly onto the relation
2. **Composite attributes:** Use only their simple, component attributes
3. **Multi-valued Attribute:** Becomes a separate relation with a foreign key taken from the superior entity

Mapping regular entity types

(a) CUSTOMER entity type with simple attributes

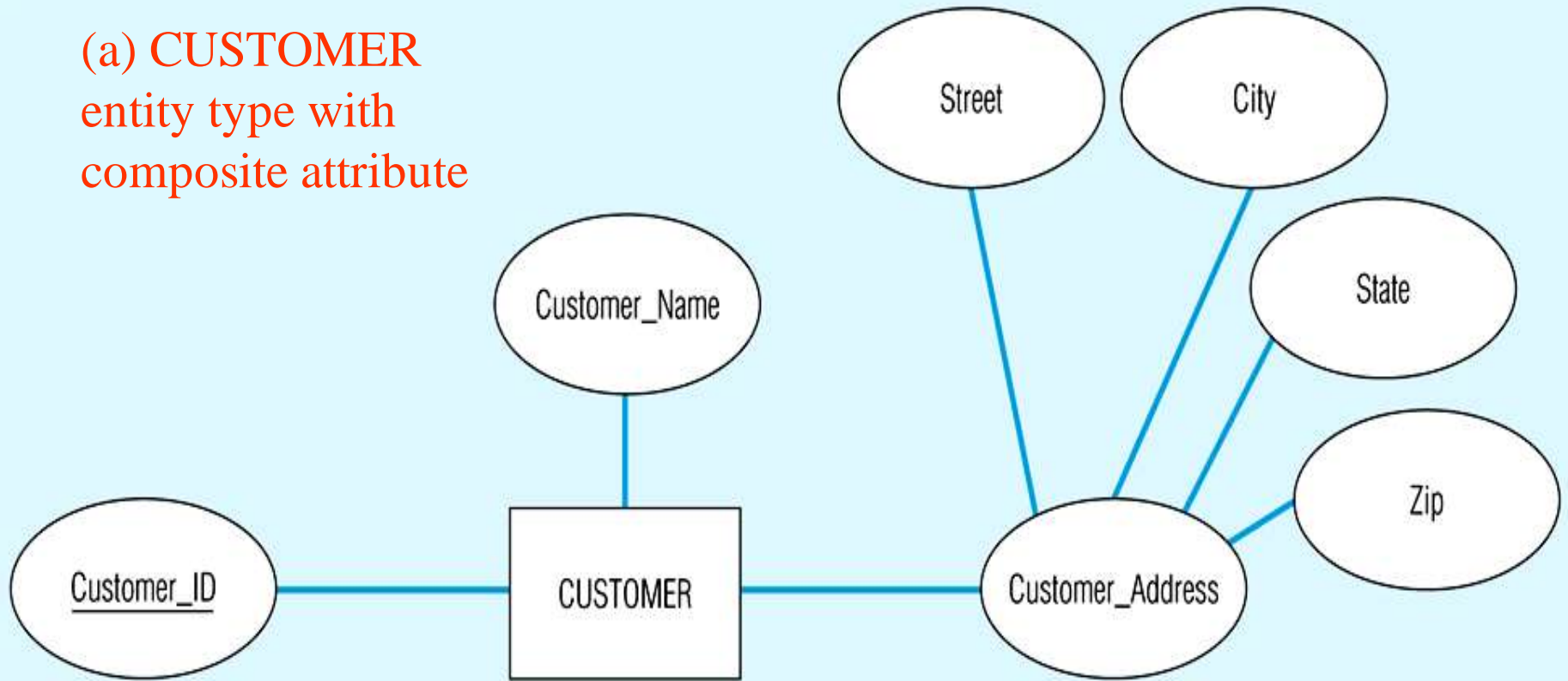


(b) CUSTOMER relation

CUSTOMER		
<u>Customer_ID</u>	Customer_Name	Customer_Address

Mapping regular entity types

(a) CUSTOMER
entity type with
composite attribute

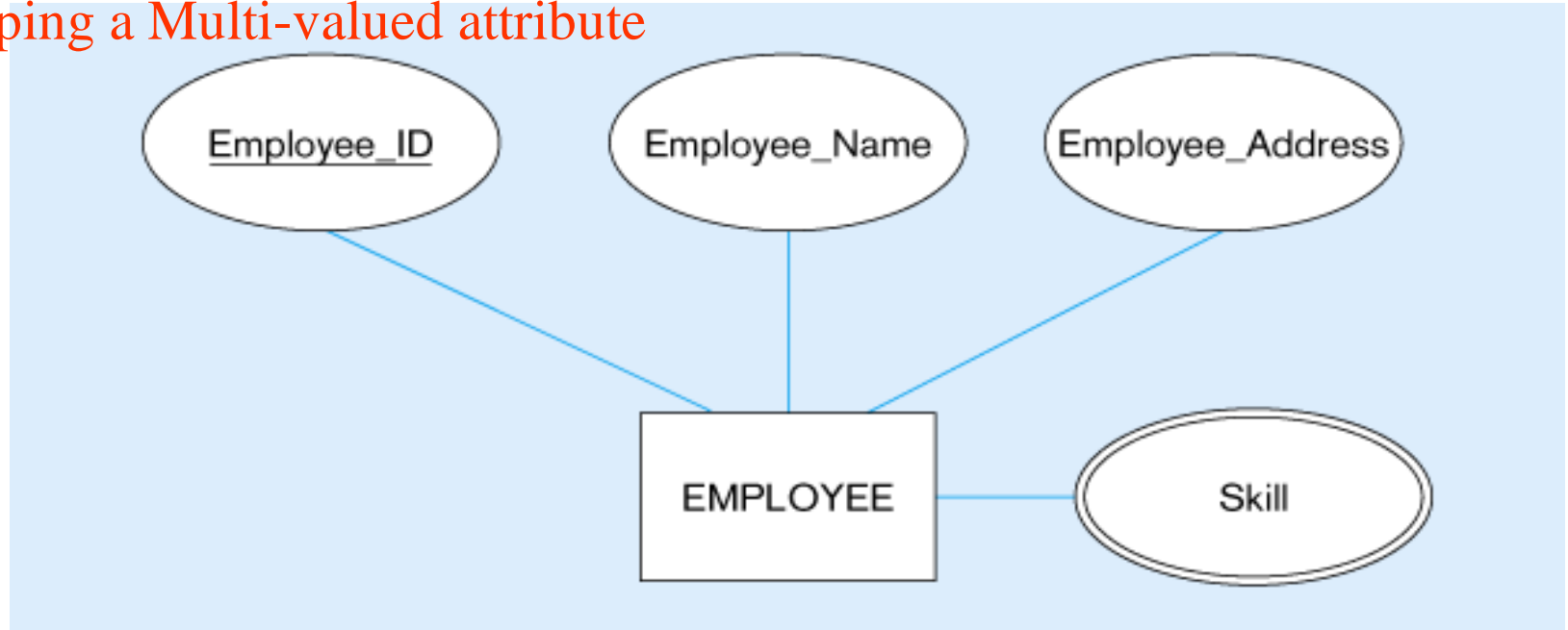


(b) CUSTOMER relation with address detail

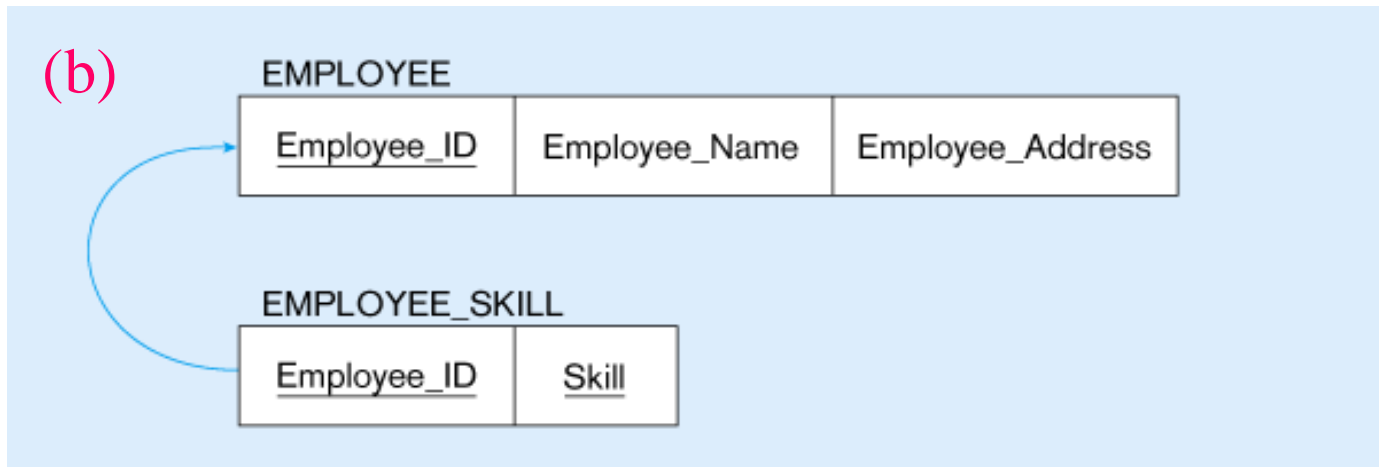
CUSTOMER

<u>Customer_ID</u>	Customer_Name	Street	City	State	Zip
--------------------	---------------	--------	------	-------	-----

Mapping a Multi-valued attribute



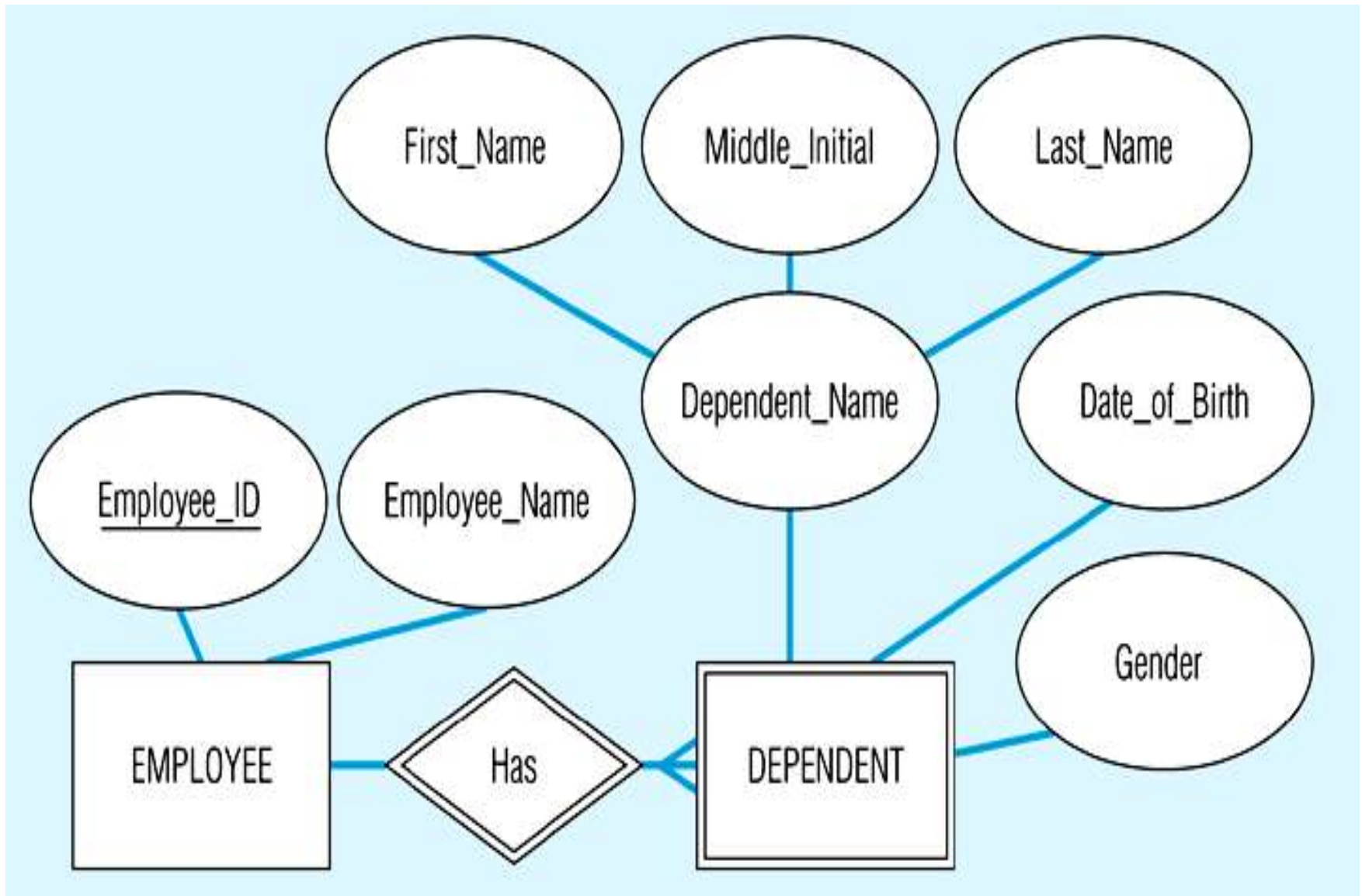
Multivalued attribute becomes a separate relation with foreign key



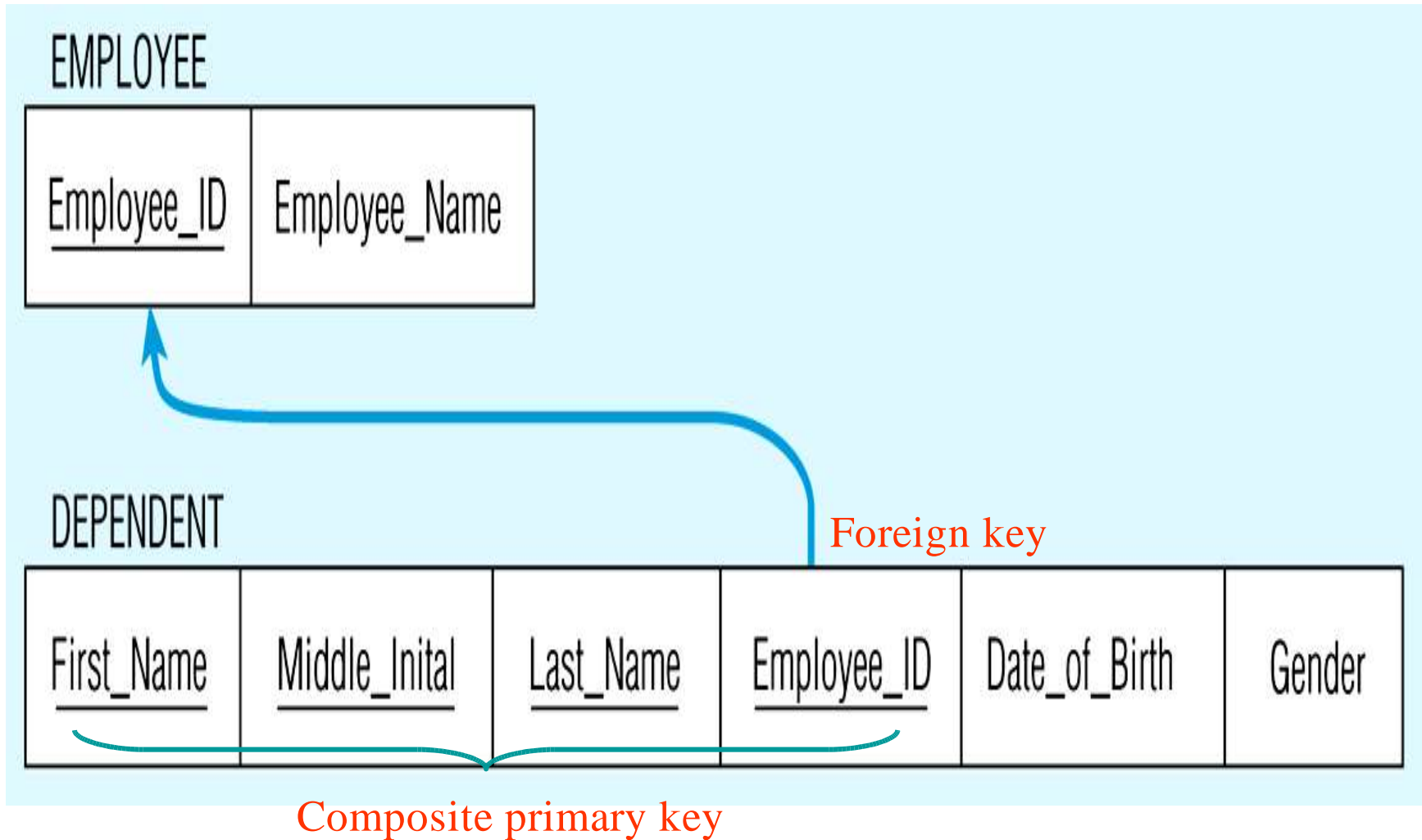
Mapping weak entity types

- Weak entity type becomes a separate relation with a foreign key taken from the superior (owner) entity type
 - Primary key composed of:
 - Partial identifier of weak entity
 - Primary key of identifying relation (strong entity)

Mapping weak entity types



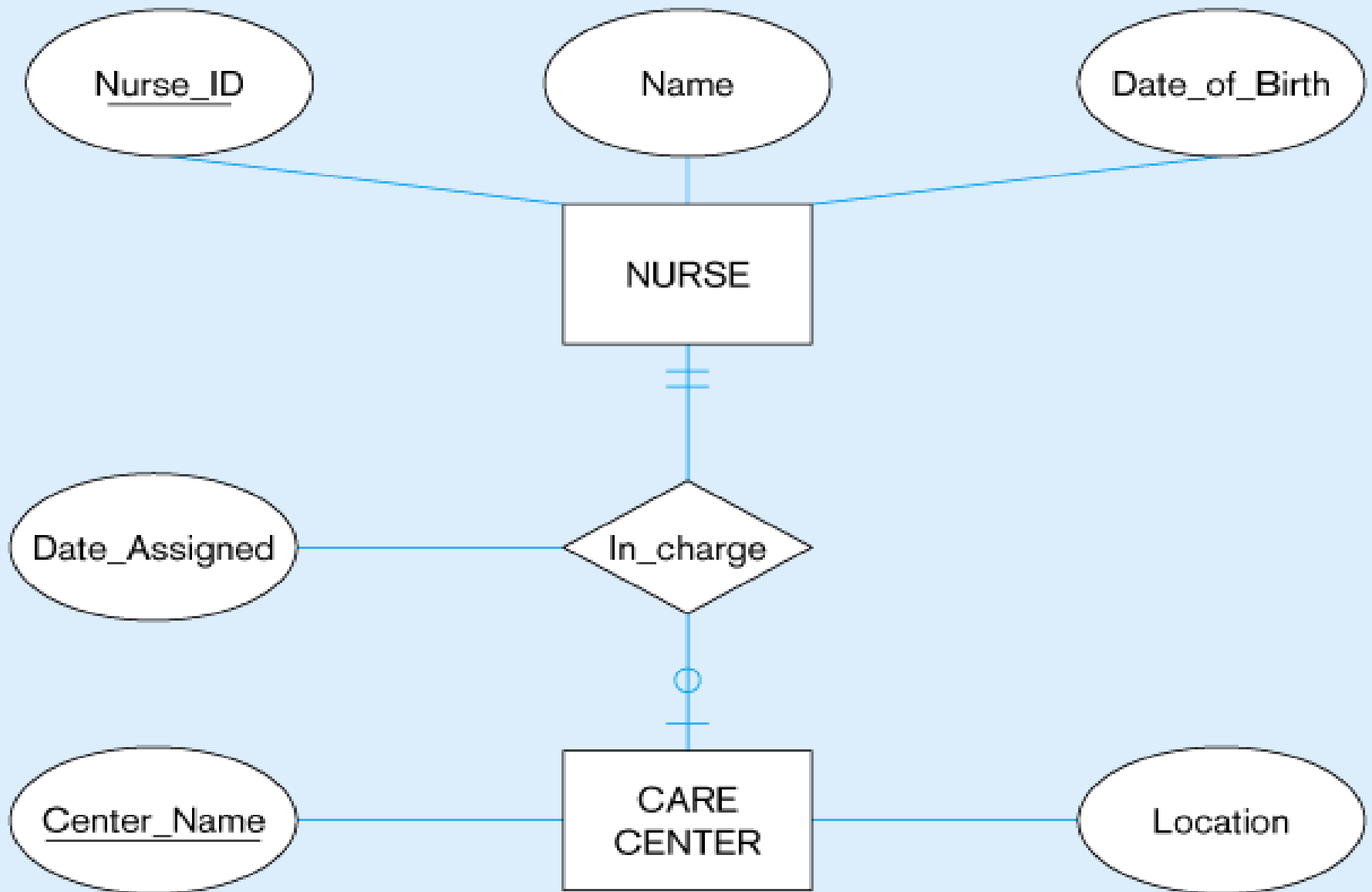
Mapping weak entity types



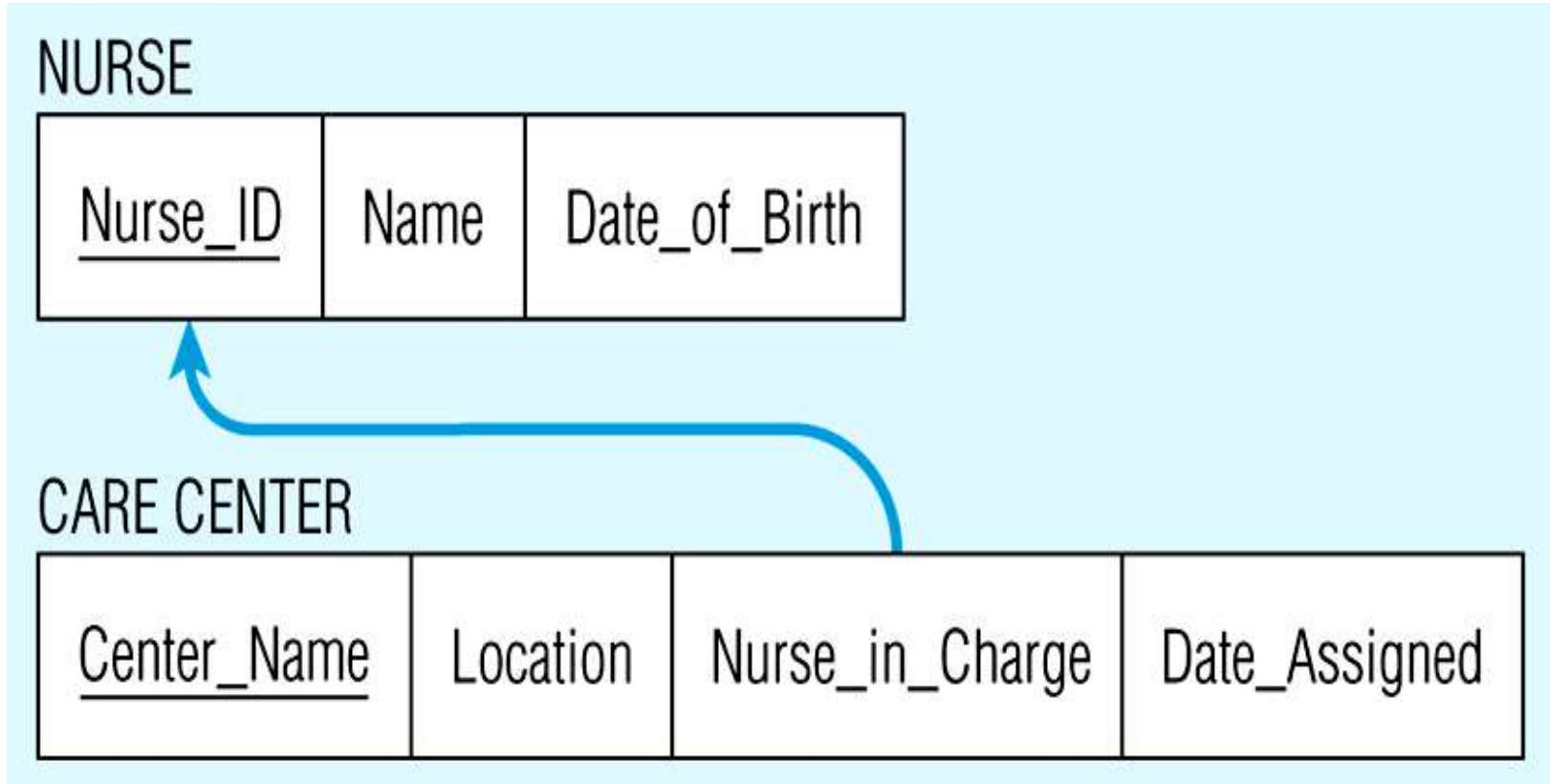
Mapping 1:1 relationship types

- Foreign key approach
 - Primary key of optional (partial) side becomes foreign key in mandatory (total) side
- Merged relation approach
 - Merge two entity types and the relationship into a single relation when *both participations are total*

Mapping 1:1 relationship types

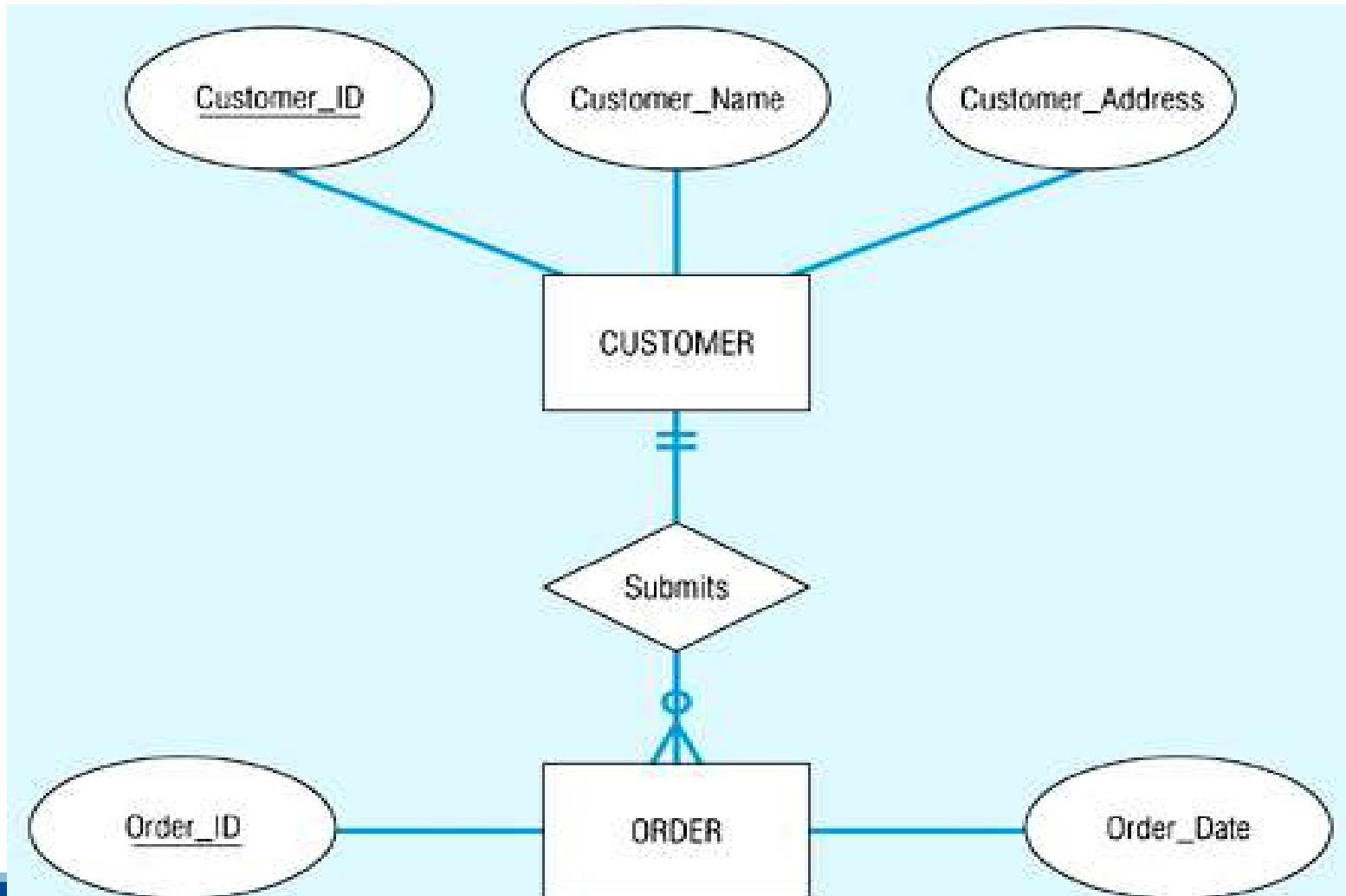


Mapping 1:1 relationship types

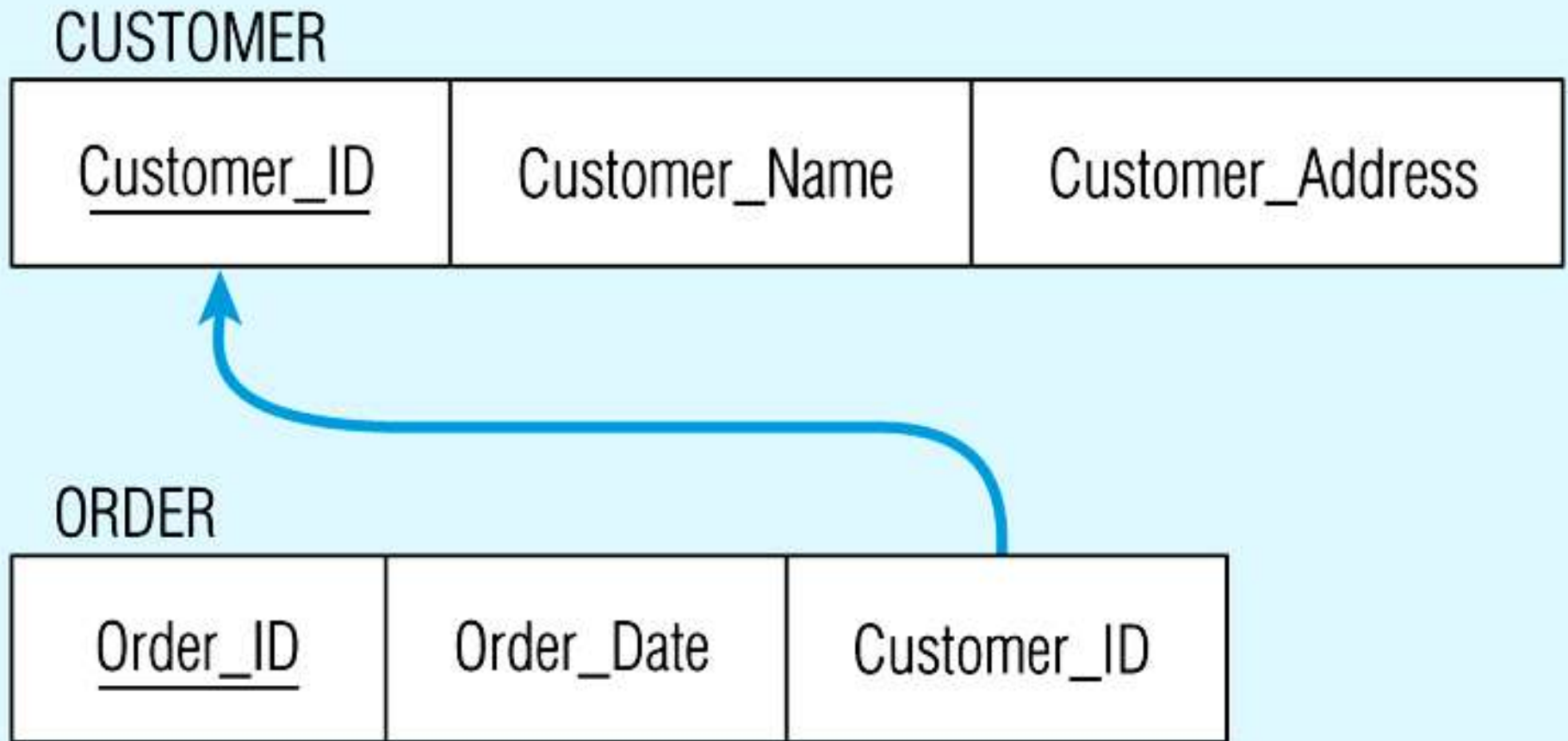


Mapping 1:N relationship types

- Primary key on the 1- one side becomes a foreign key on the N - many side



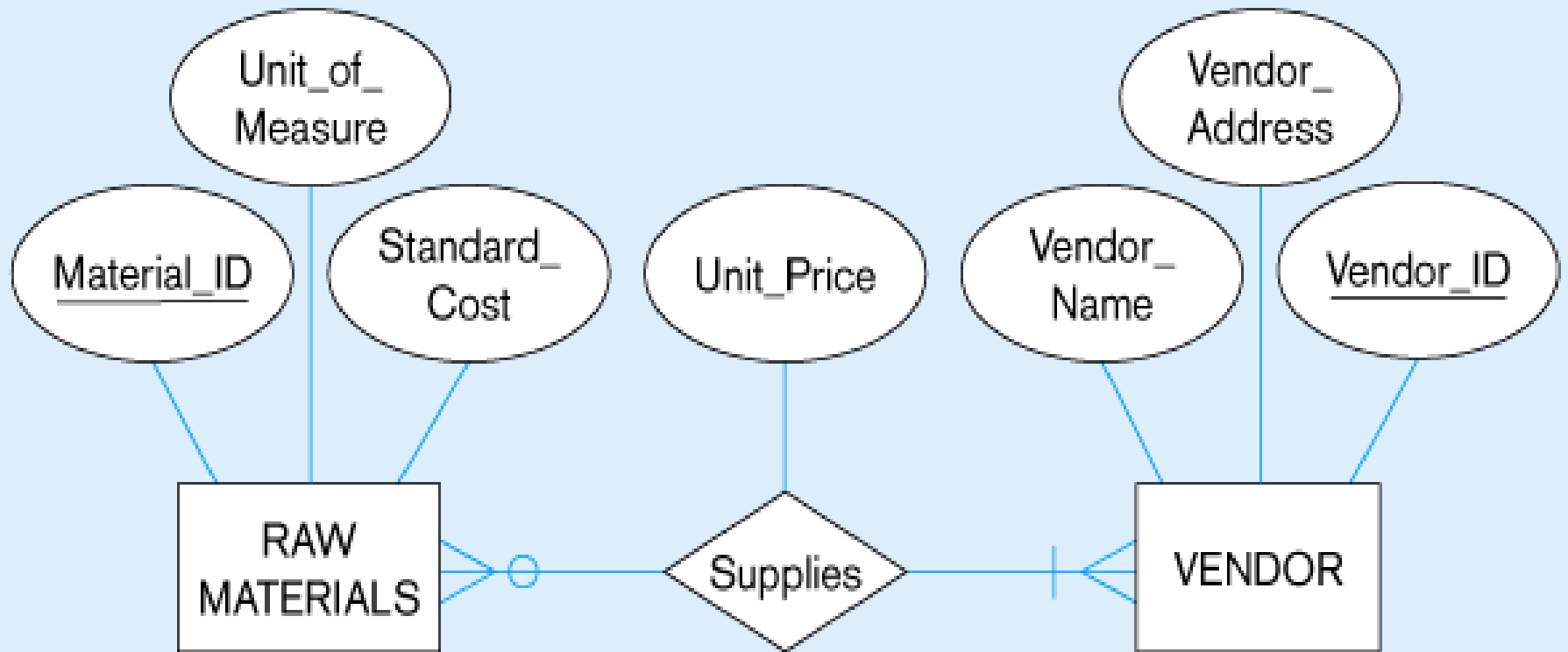
Mapping 1:N relationship types



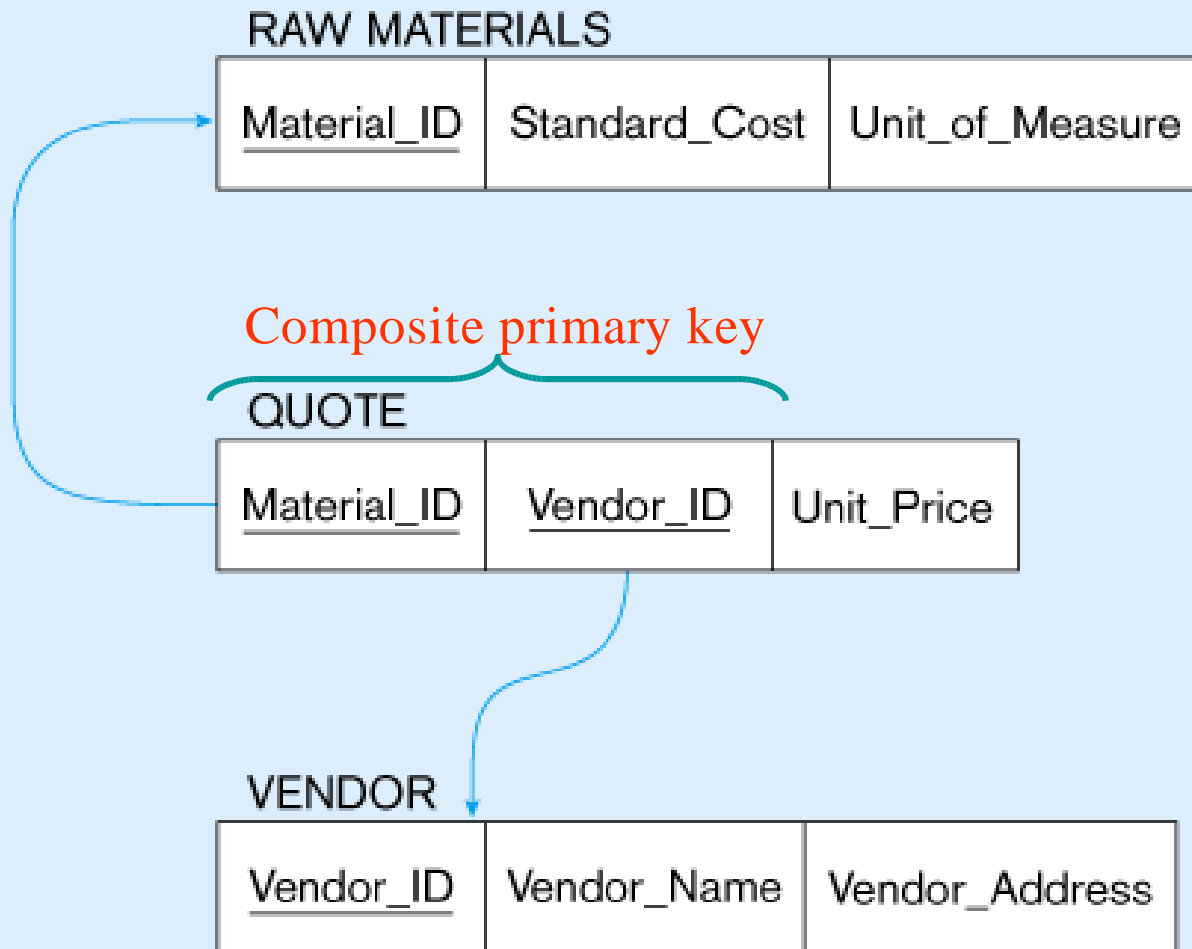
Mapping M:N relationship types

- Create a *new relation* with primary key of both the entity types as its primary key
 - combination of primary key of participating entity types
- Also include any of the simple attribute(s) of M:N relationship type as attributes of new relation

Mapping M:N relationship types



Mapping M:N relationship types

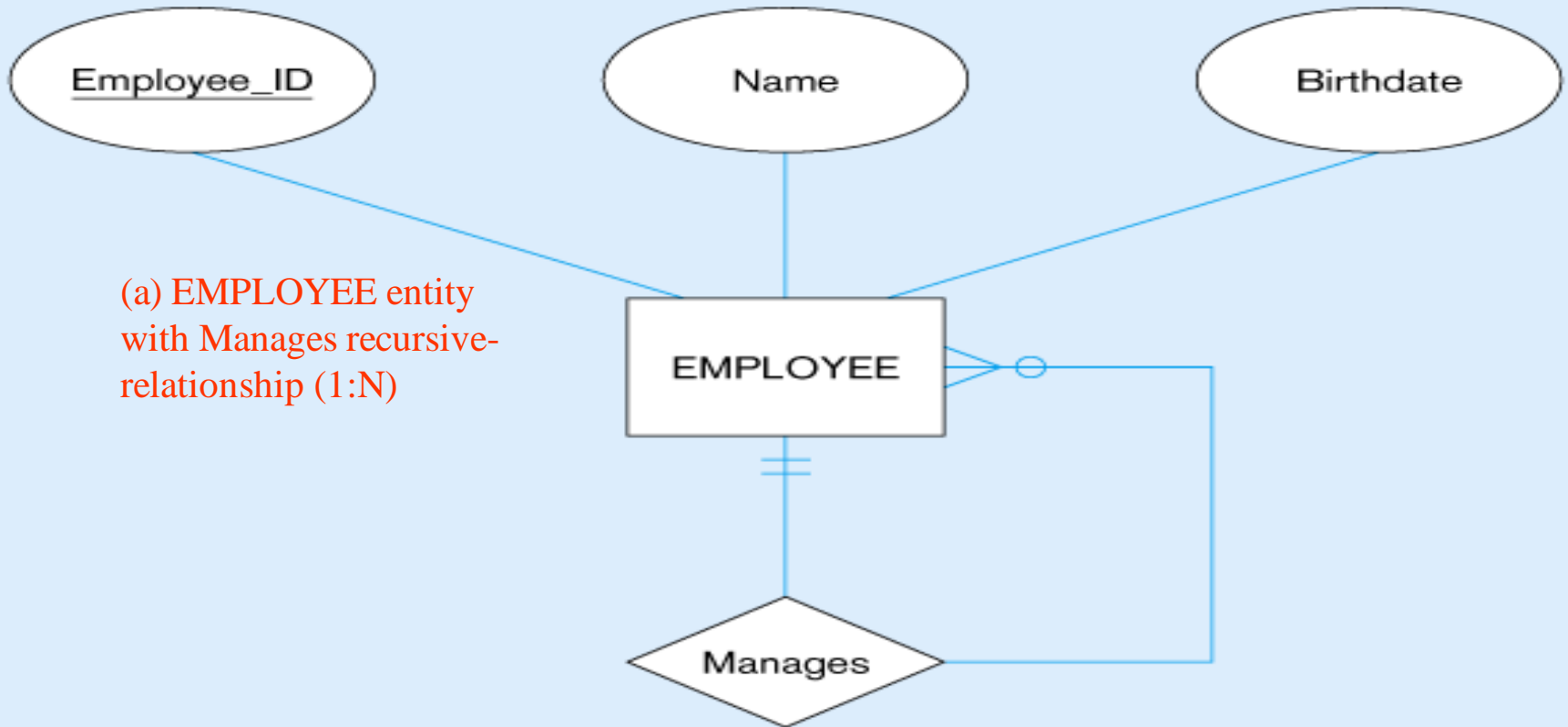


Mapping unary relationship type

- **One-to-Many** - Recursive foreign key in the same relation
- **Many-to-Many** - Two relations:
 - One for the entity type
 - One for an associative relation in which the primary key has two attributes, both taken from the primary key of the entity

Mapping unary relationship type

(a) EMPLOYEE entity with Manages recursive-relationship (1:N)



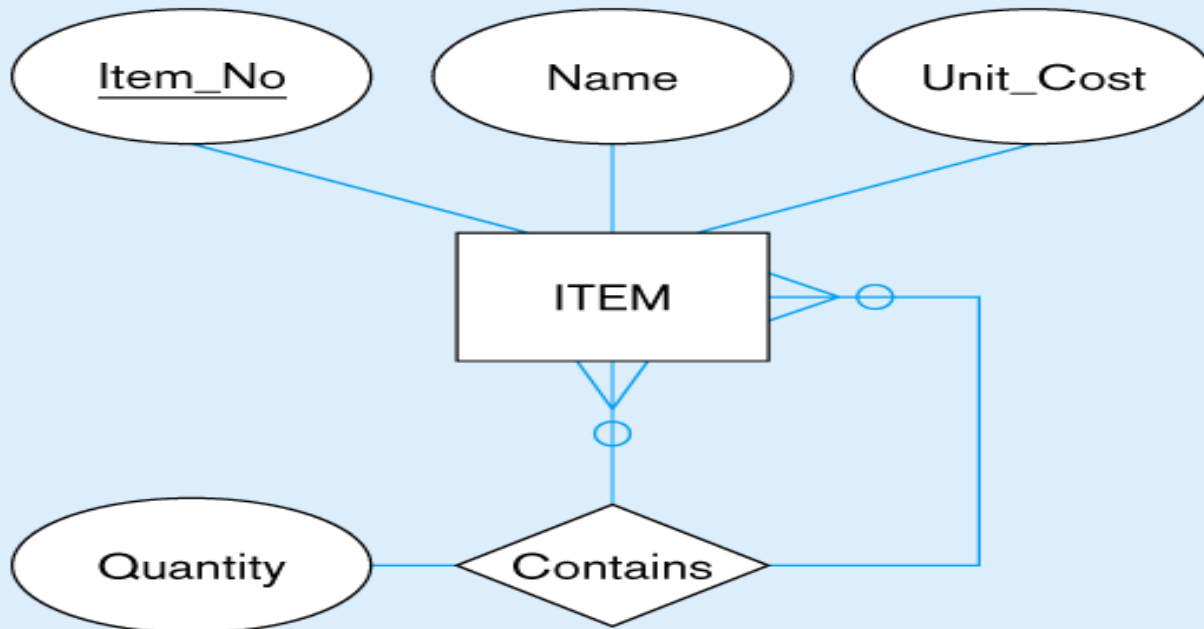
(b) EMPLOYEE relation with recursive foreign key

EMPLOYEE

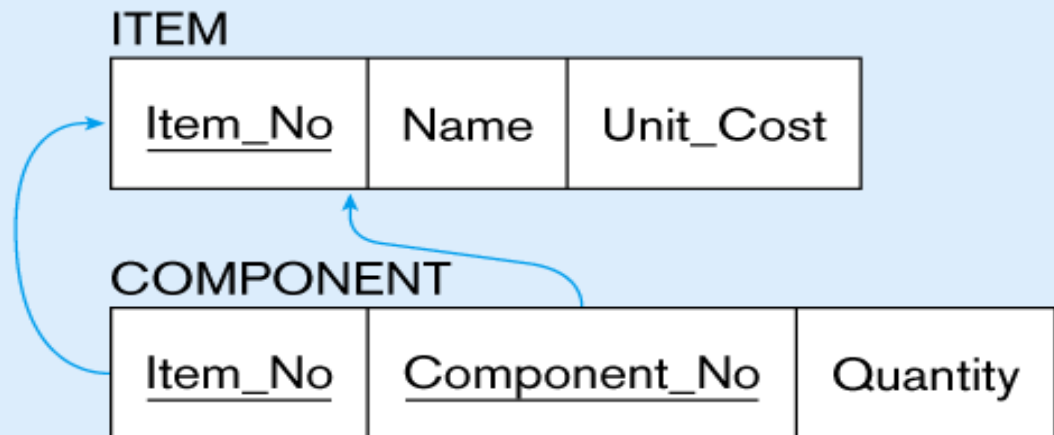
<u>Employee_ID</u>	Name	Birthdate	<u>Manager_ID</u>
--------------------	------	-----------	-------------------

Mapping unary relationship type

(a) Bill-of-materials relationships (M:N)



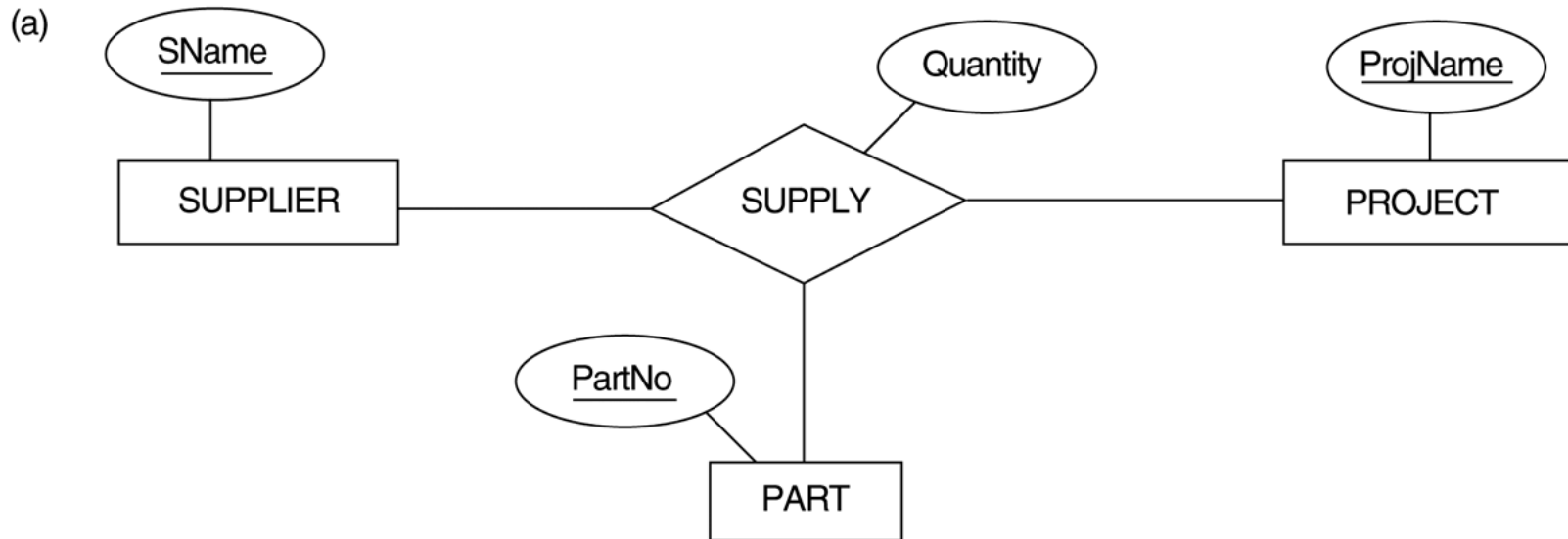
(b) ITEM and COMPONENT relations



Mapping n-ary relationship types

- 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 attributes of the n-ary relationship type as attributes of S.
- **Example:** The relationship type SUPPY in the ER on the next slide.
 - This can be mapped to SUPPLY, whose primary key is the combination of the three foreign keys {SNAME, PARTNO, PROJNAME}

Mapping n-ary relationship types



Mapping n-ary relationship types

SUPPLIER

<u>SNAME</u>	...
--------------	-----

PROJECT

<u>PROJNAME</u>	...
-----------------	-----

PART

<u>PARTNO</u>	...
---------------	-----

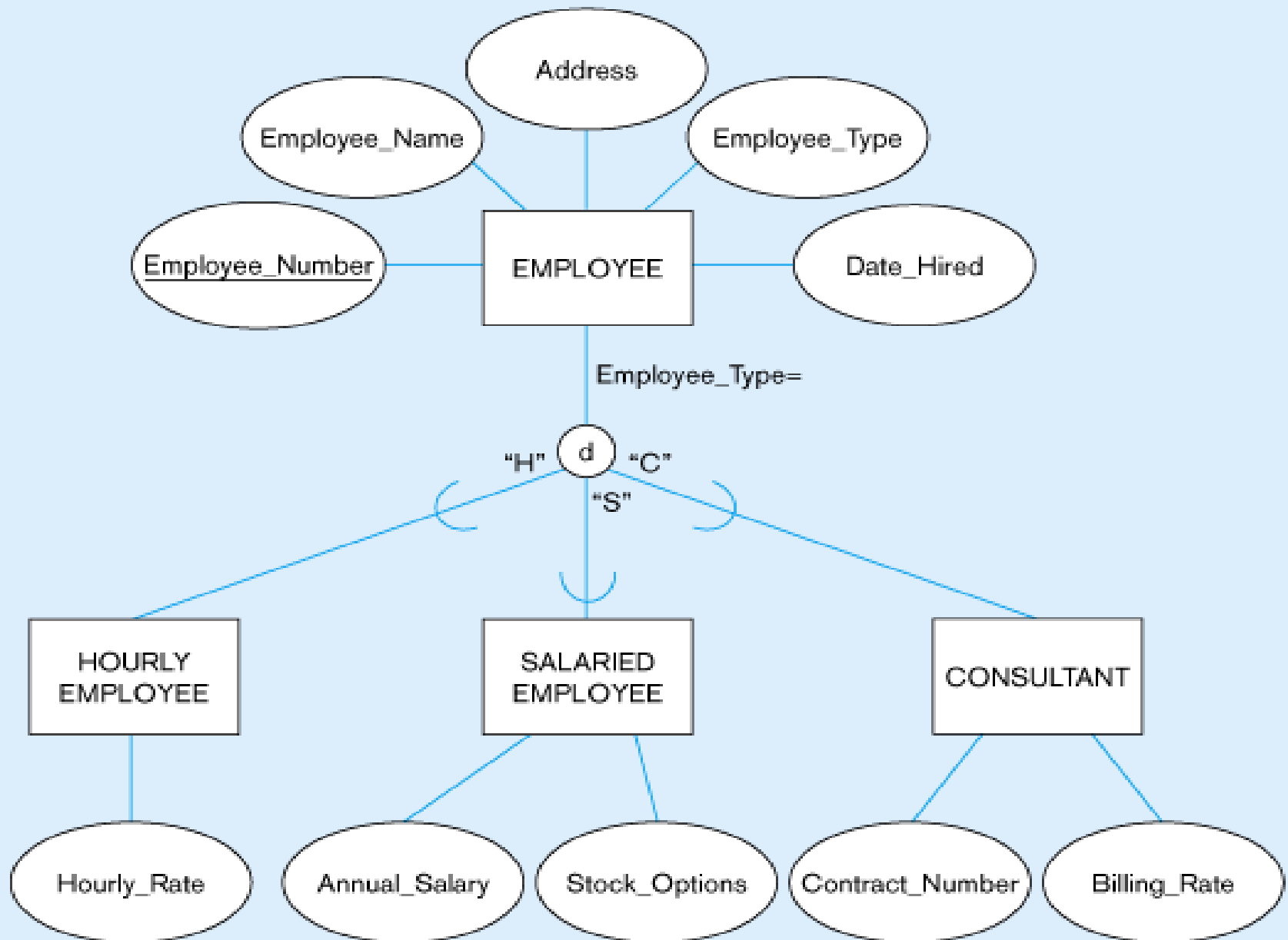
SUPPLY

<u>SNAME</u>	PROJNAME	<u>PARTNO</u>	QUANTITY
--------------	----------	---------------	----------

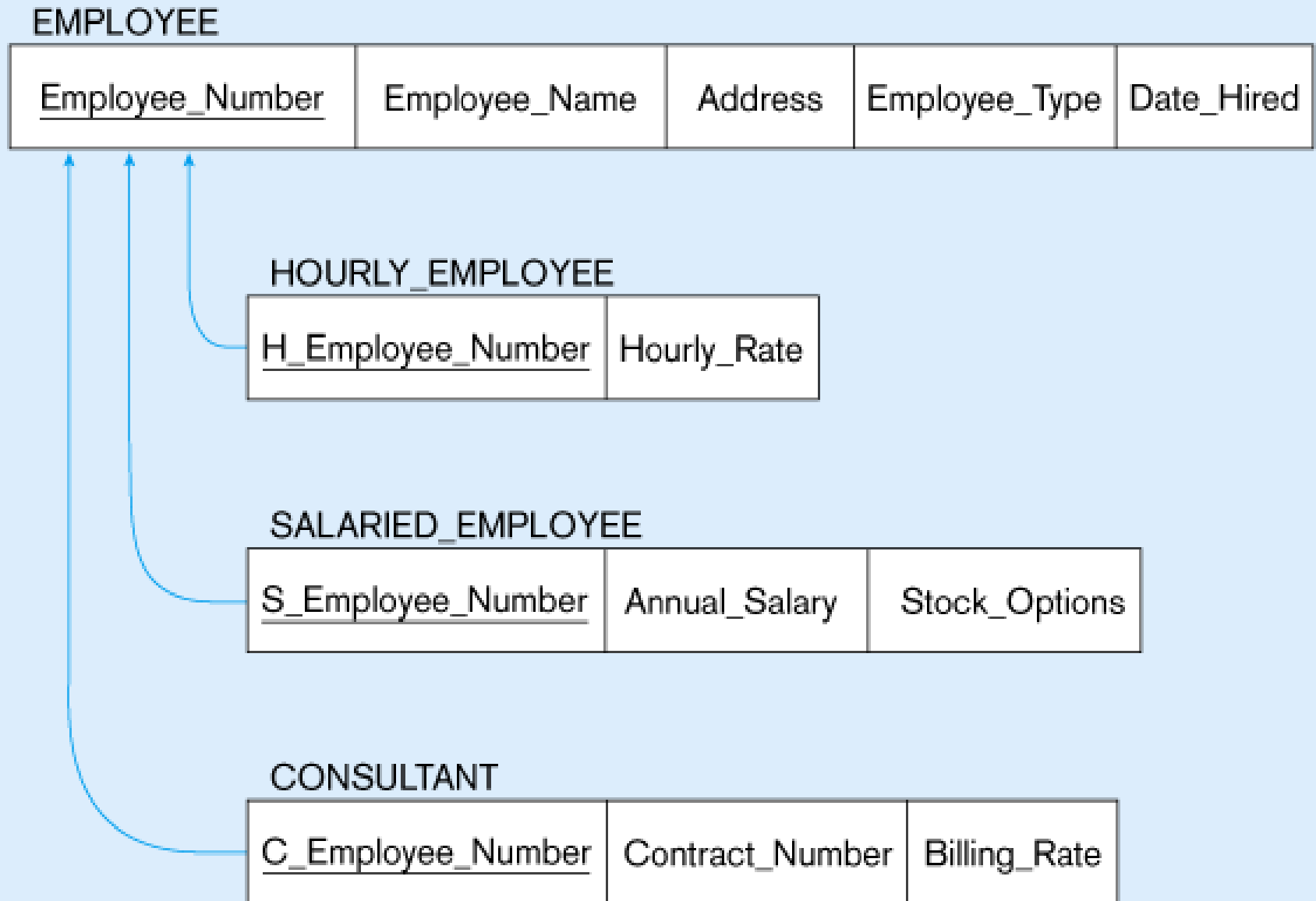
Mapping supertype/subtype relationships

- One relation for supertype and for each subtype
- Supertype attributes (including identifier and subtype discriminator) go into supertype relation
- Subtype attributes go into each subtype; primary key of supertype relation also becomes primary key of subtype relation
- 1:1 relationship established between supertype and each subtype, with supertype as primary table

Mapping supertype/subtype relationships



Mapping supertype/subtype relationships



**THANK
YOU!**