Mapping Rules

From ER (or EER) to Relational Schema

Inputs :ER Outputs: Data Model

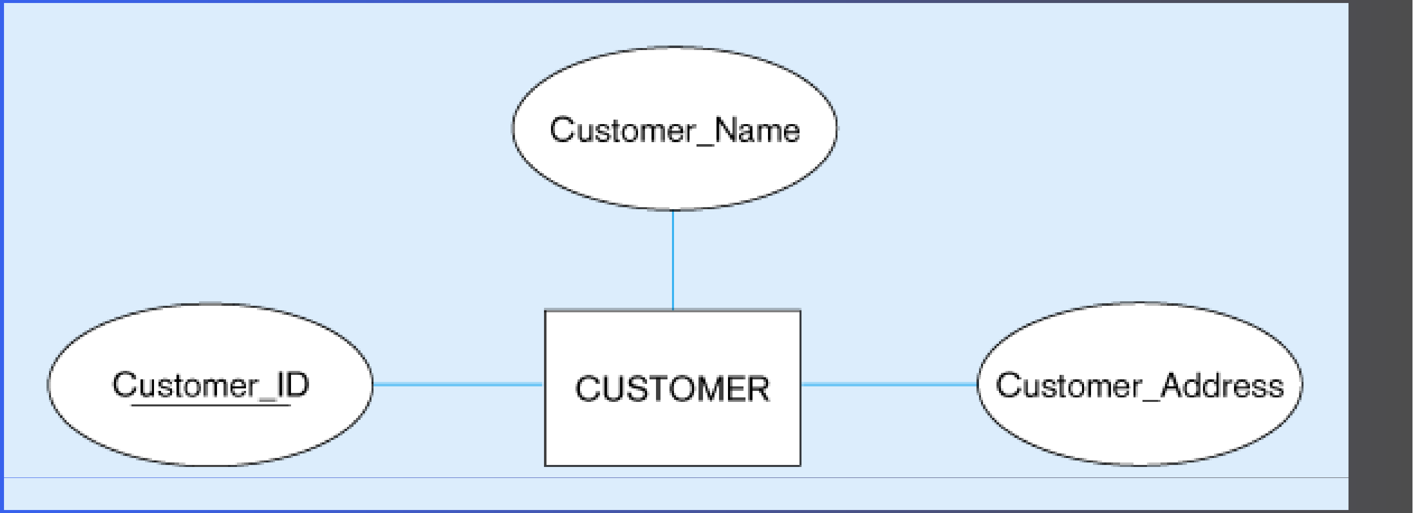
Transforming EER Diagrams into Relations

Mapping Regular Entities to Relations

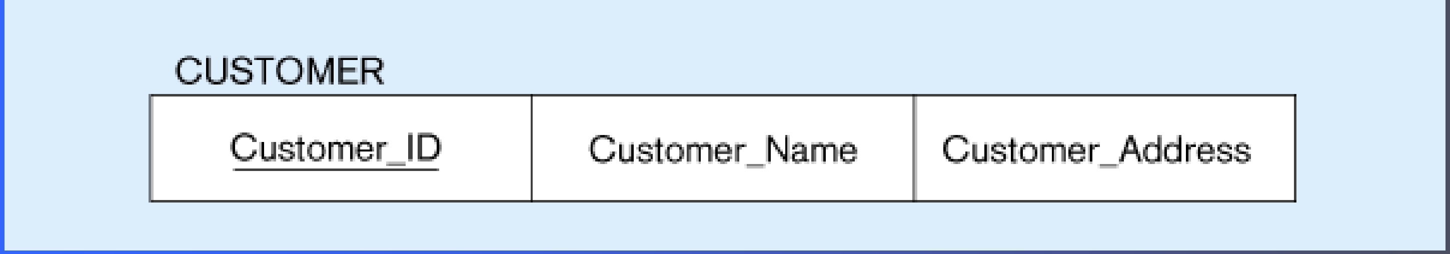
1. Simple attributes: E-R attributes map directly onto the relation
2. Composite attributes: Use only their simple, component attributes
3. Multi-valued Attribute - Becomes a separate relation with a foreign key taken from the superior entity

Mapping a regular entity

(a) CUSTOMER entity type with simple attributes



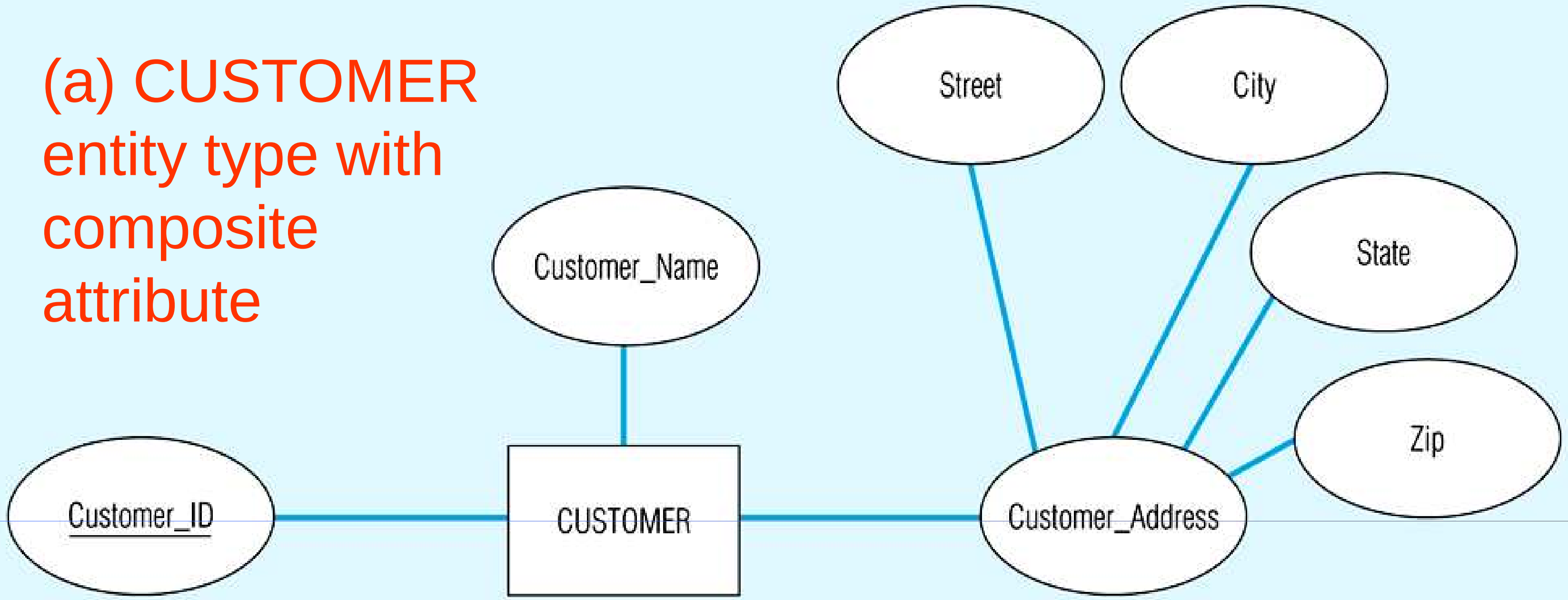
(b) CUSTOMER relation



CUSTOMER

Customer_ID	Customer_Name	Customer_Address

Mapping a composite attribute

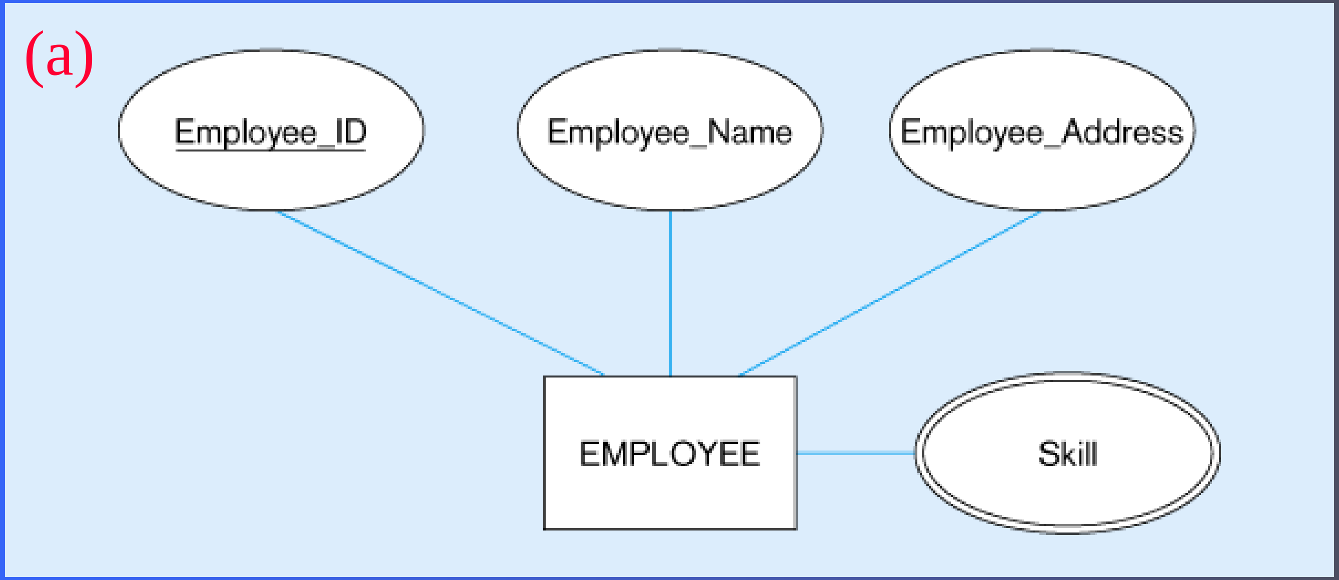


(a) CUSTOMER entity type with composite attribute

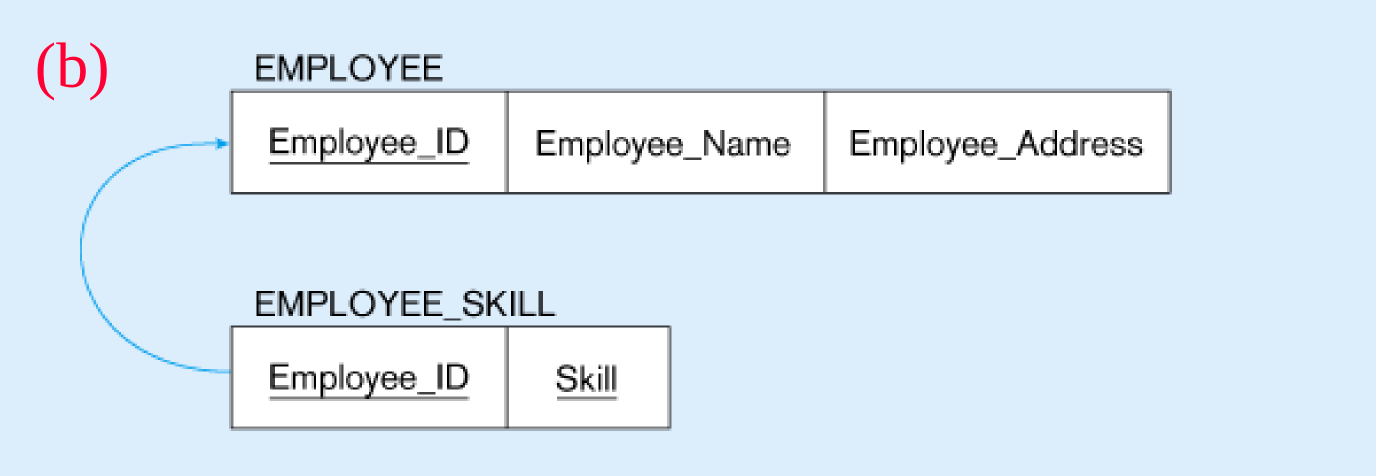
CUSTOMER (b) CUSTOMER relation with address detail

Customer_ID	Customer_Name	Street	City	State	Zip

Mapping a multivalued attribute



Multivalued attribute becomes a separate relation with foreign key



EMPLOYEE

Employee_ID	Employee_Name	Employee_Address

EMPLOYEE_SKILL

Employee_ID	Skill

1 – to – many relationship between original entity and new relation

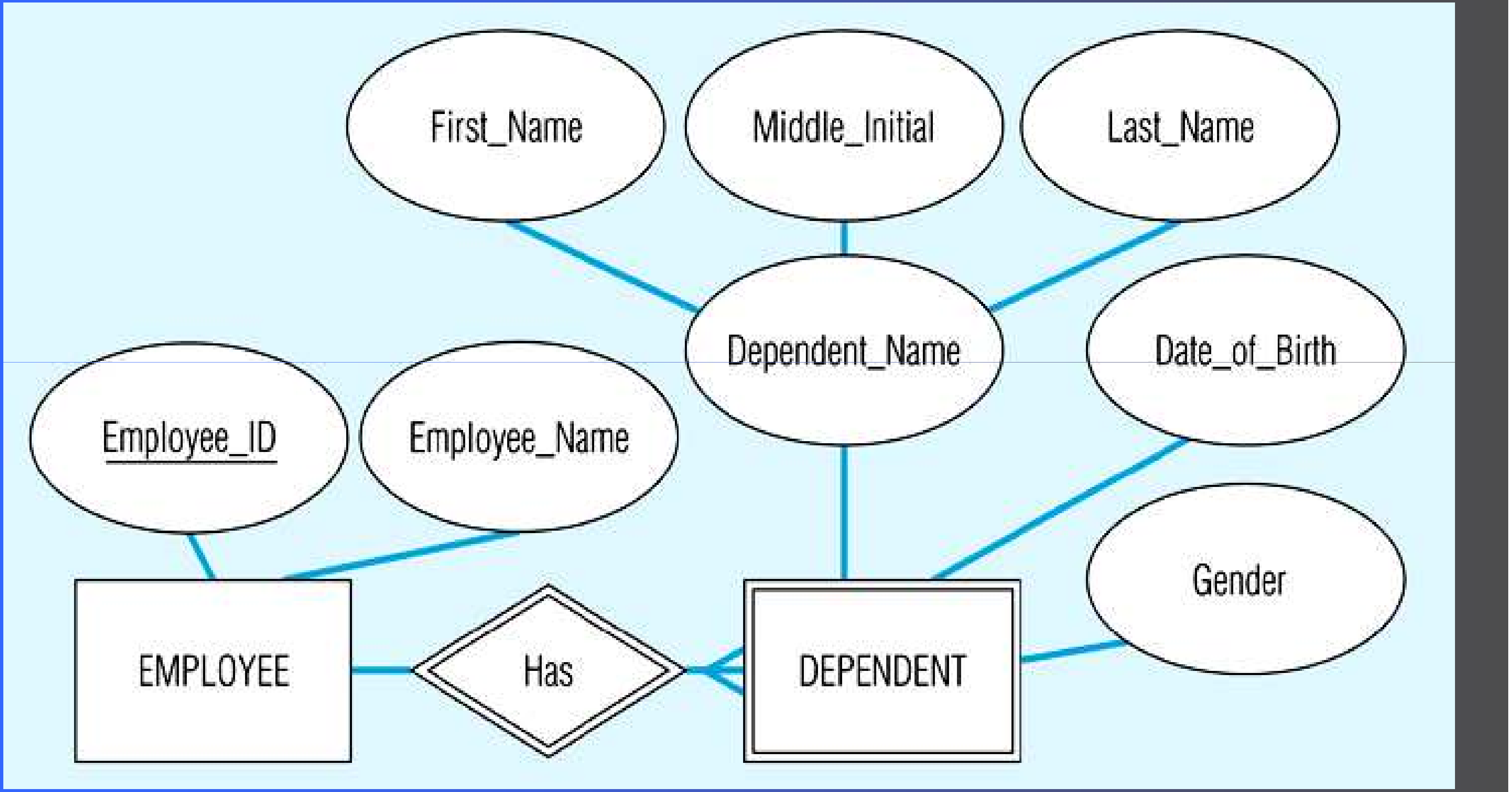
Transforming EER Diagrams into Relations

Mapping Weak Entities

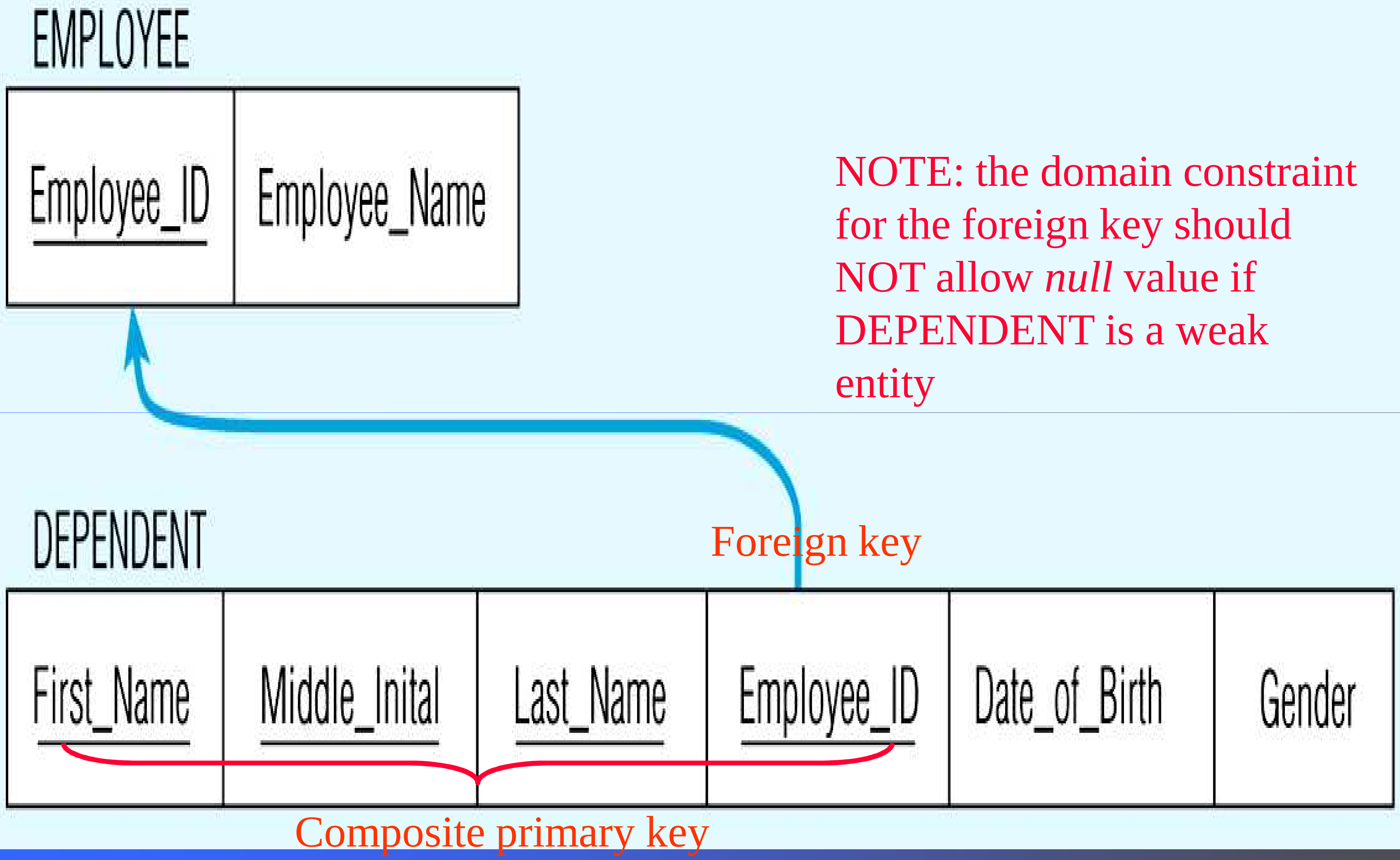
- Becomes a separate relation with a foreign key taken from the superior entity
- Primary key composed of:
  - Partial identifier of weak entity
  - Primary key of identifying relation (strong entity)

Example of mapping a weak entity

(a) Weak entity DEPENDENT



Relations resulting from weak entity



EMPLOYEE

Employee_ID	Employee_Name

DEPENDENT

First_Name	Middle_Inital	Last_Name	Employee_ID	Date_of_Birth	Gender

Foreign key

Composite primary key

NOTE: the domain constraint for the foreign key should NOT allow *null* value if DEPENDENT is a weak entity

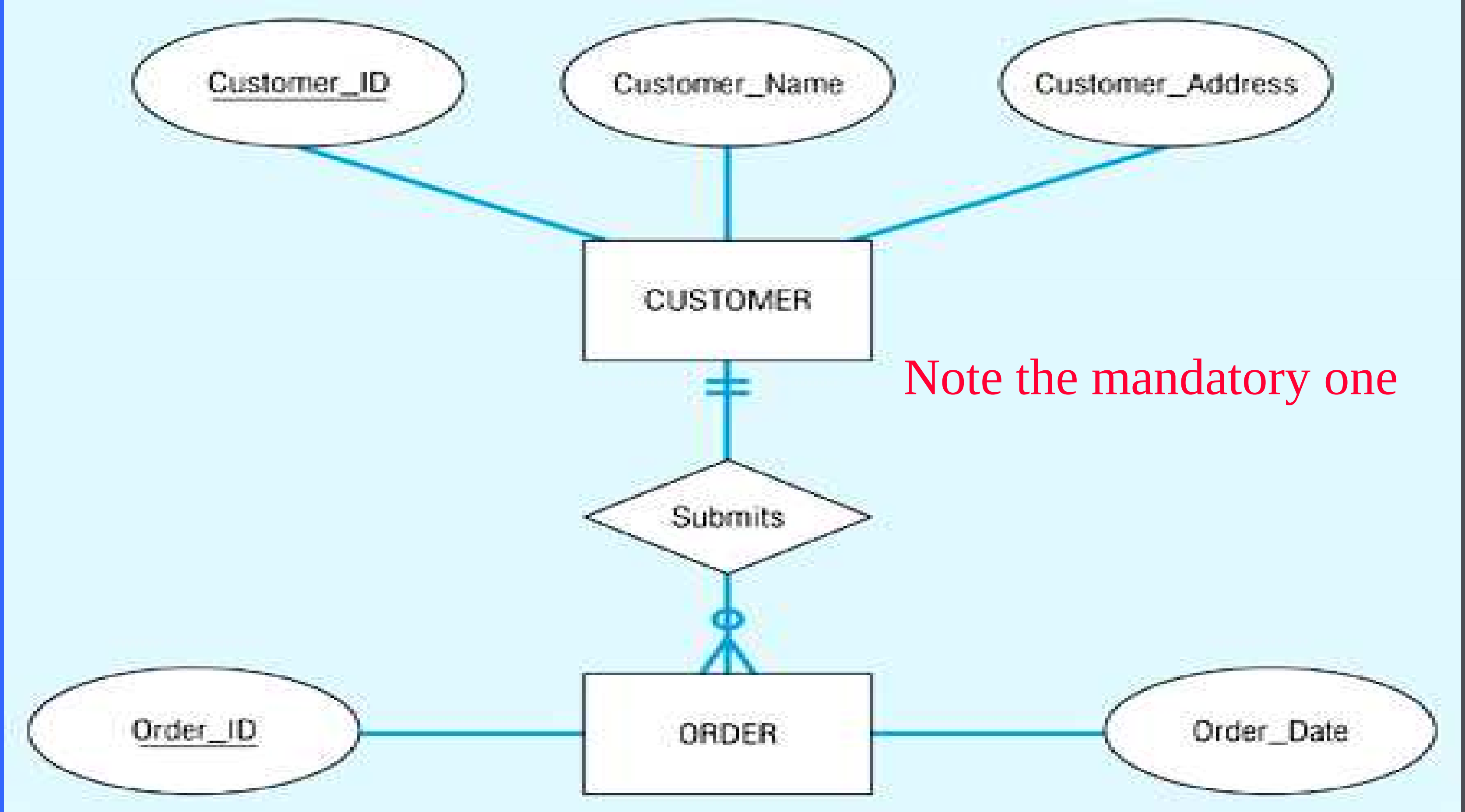
Transforming EER Diagrams into Relations

Mapping Binary Relationships

- One-to-Many - Primary key on the one side becomes a foreign key on the many side
- Many-to-Many - Create a ***new relation*** with the primary keys of the two entities as its primary key
- One-to-One - Primary key on the mandatory side becomes a foreign key on the optional side

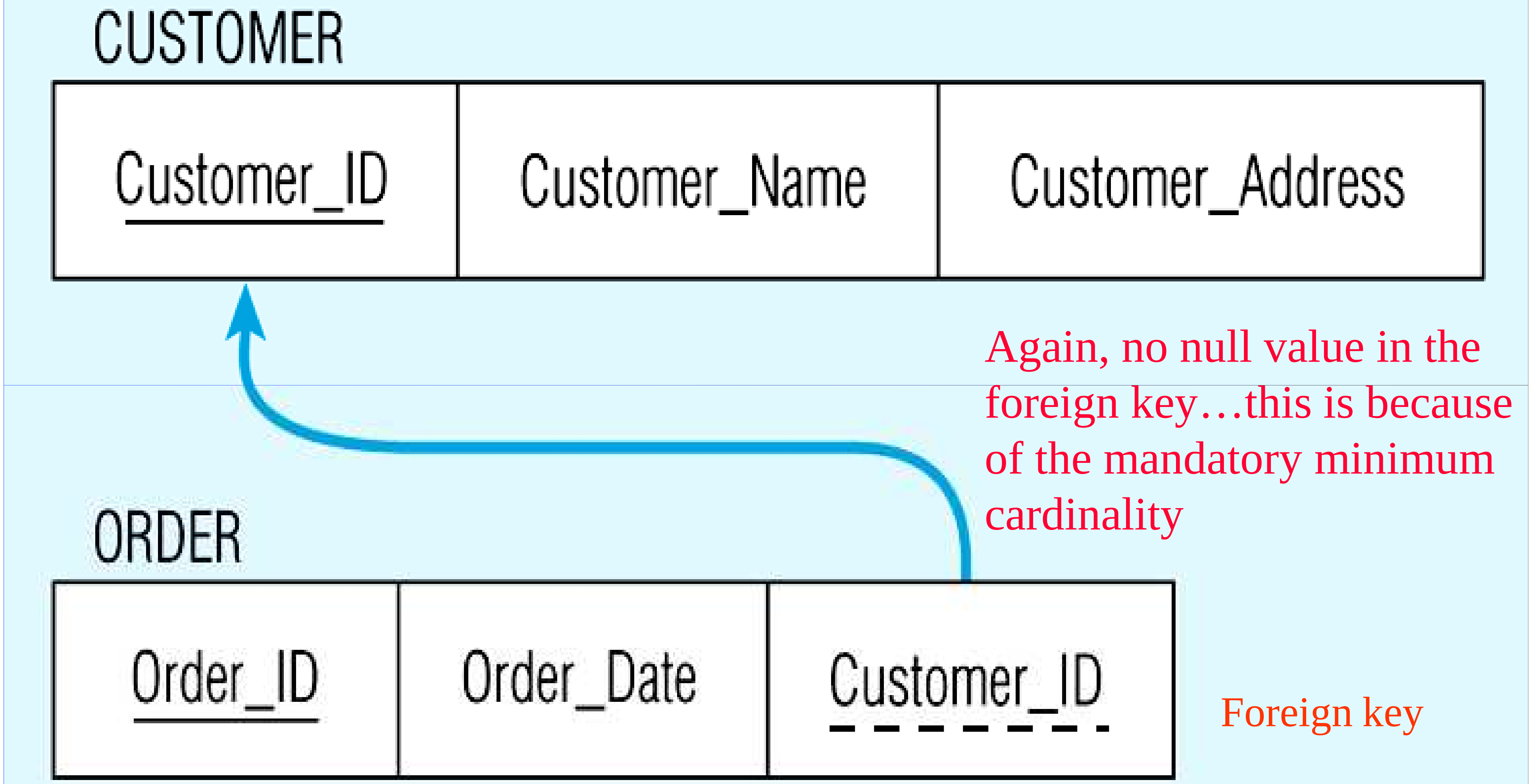
Example of mapping a 1:M relationship

(a) Relationship between customers and orders



Note the mandatory one

Mapping the relationship



CUSTOMER

Customer_ID | Customer_Name | Customer_Address

ORDER

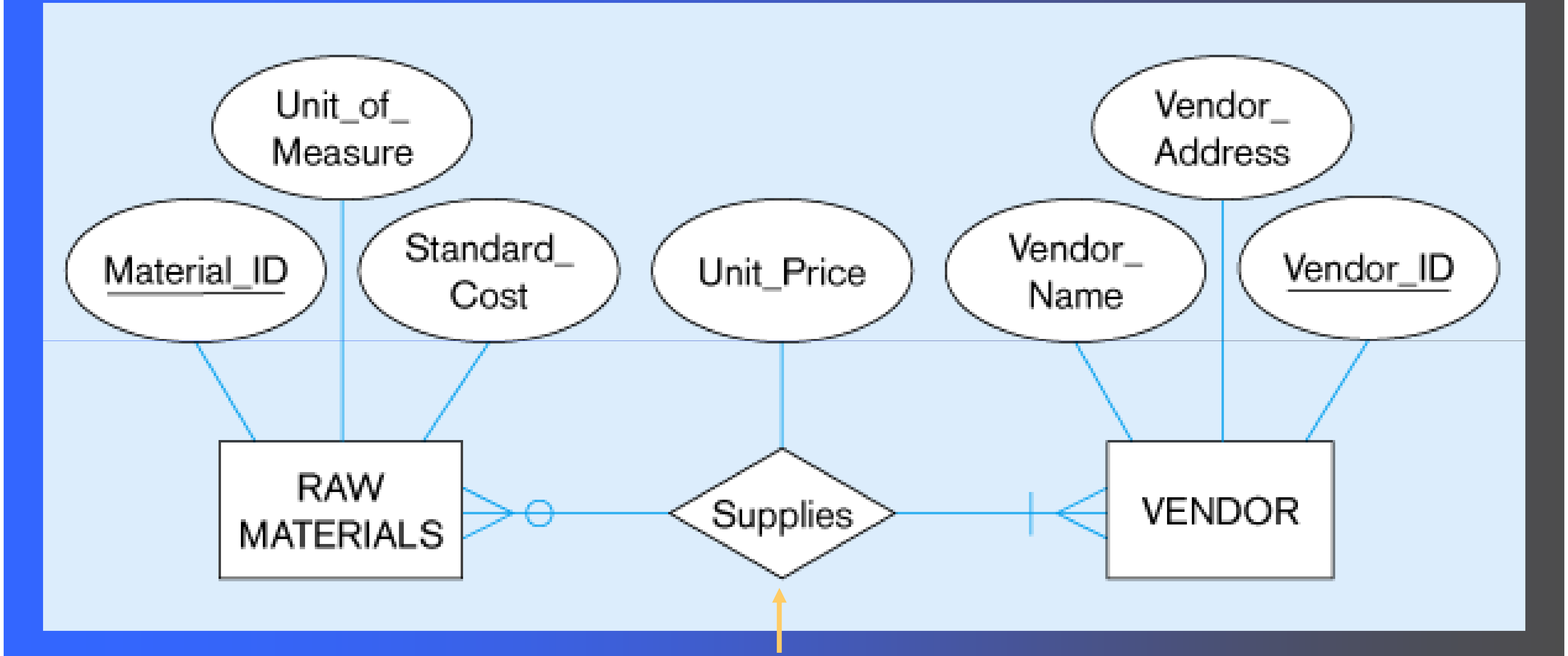
Order_ID | Order_Date | Customer_ID

Again, no null value in the foreign key…this is because of the mandatory minimum cardinality

Foreign key

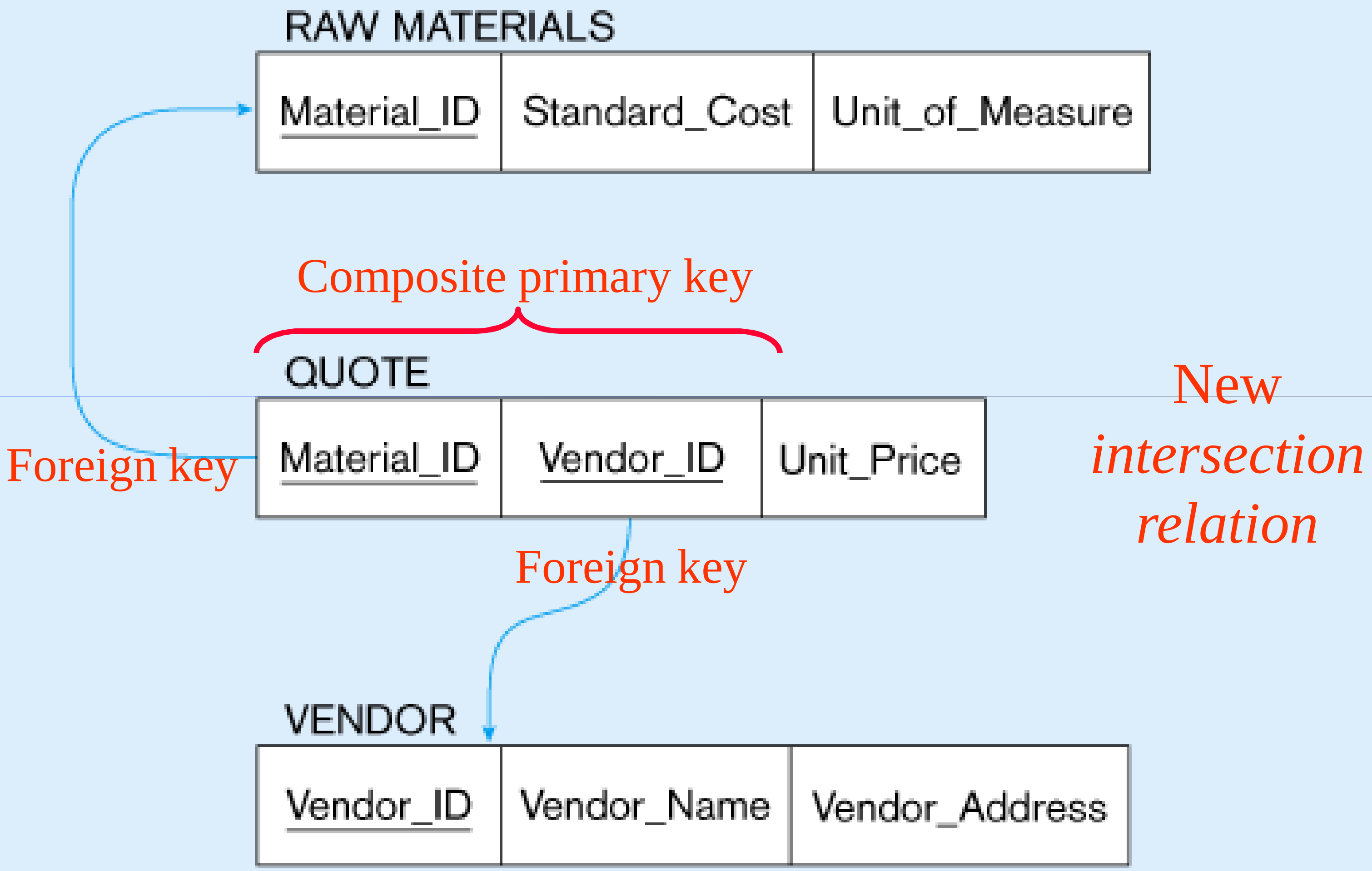
Example of mapping an M:N relationship

(a) ER diagram (M:N)



The *Supplies* relationship will need to become a separate relation

Three resulting relations



RAW MATERIALS

Material_ID	Standard_Cost	Unit_of_Measure

Composite primary key

QUOTE

Material_ID	Vendor_ID	Unit_Price

Foreign key

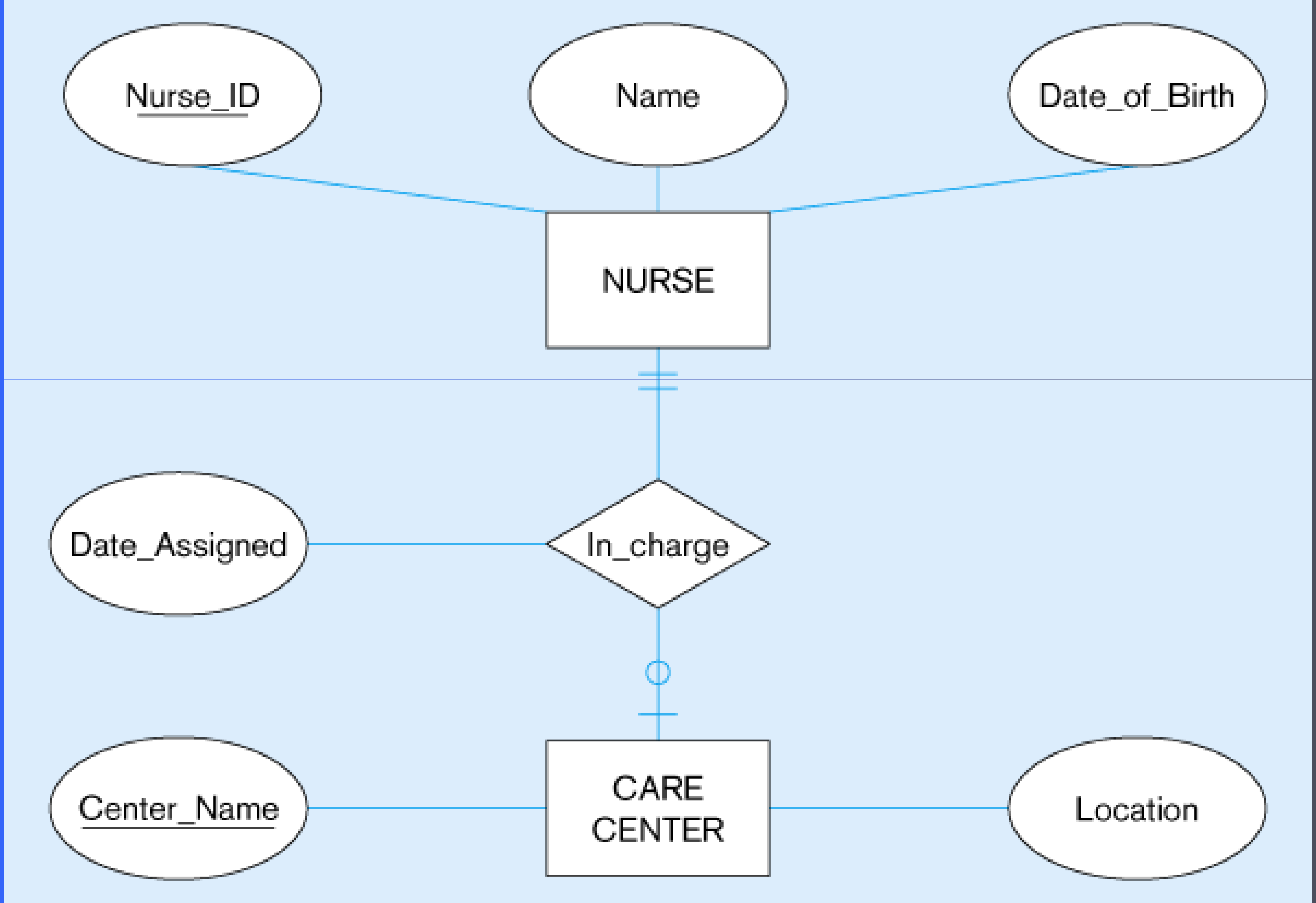
Foreign key

New *intersection relation*

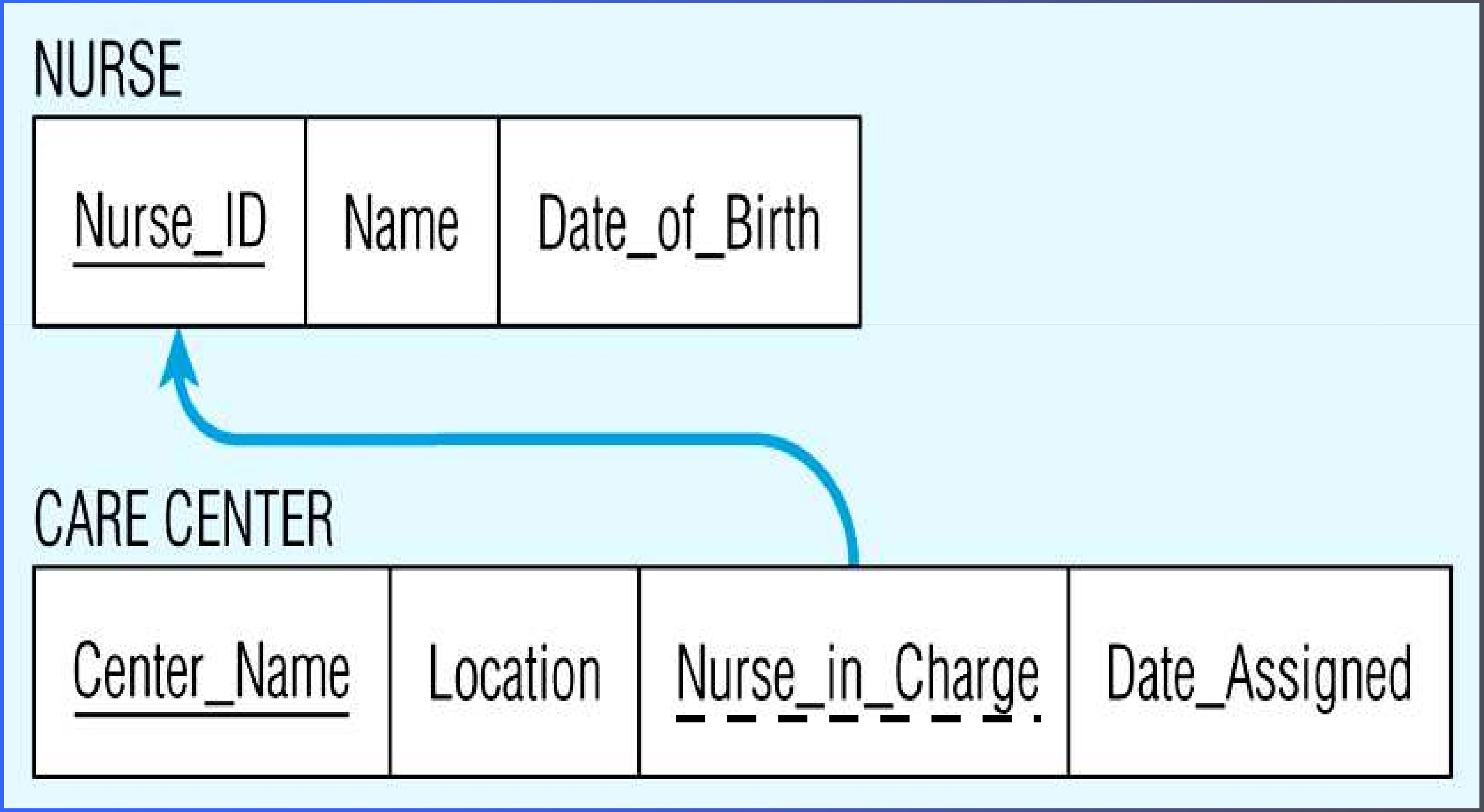
VENDOR

Vendor_ID	Vendor_Name	Vendor_Address

Mapping a binary 1:1 relationship



Resulting relations



NURSE

Nurse_ID	Name	Date_of_Birth

CARE CENTER

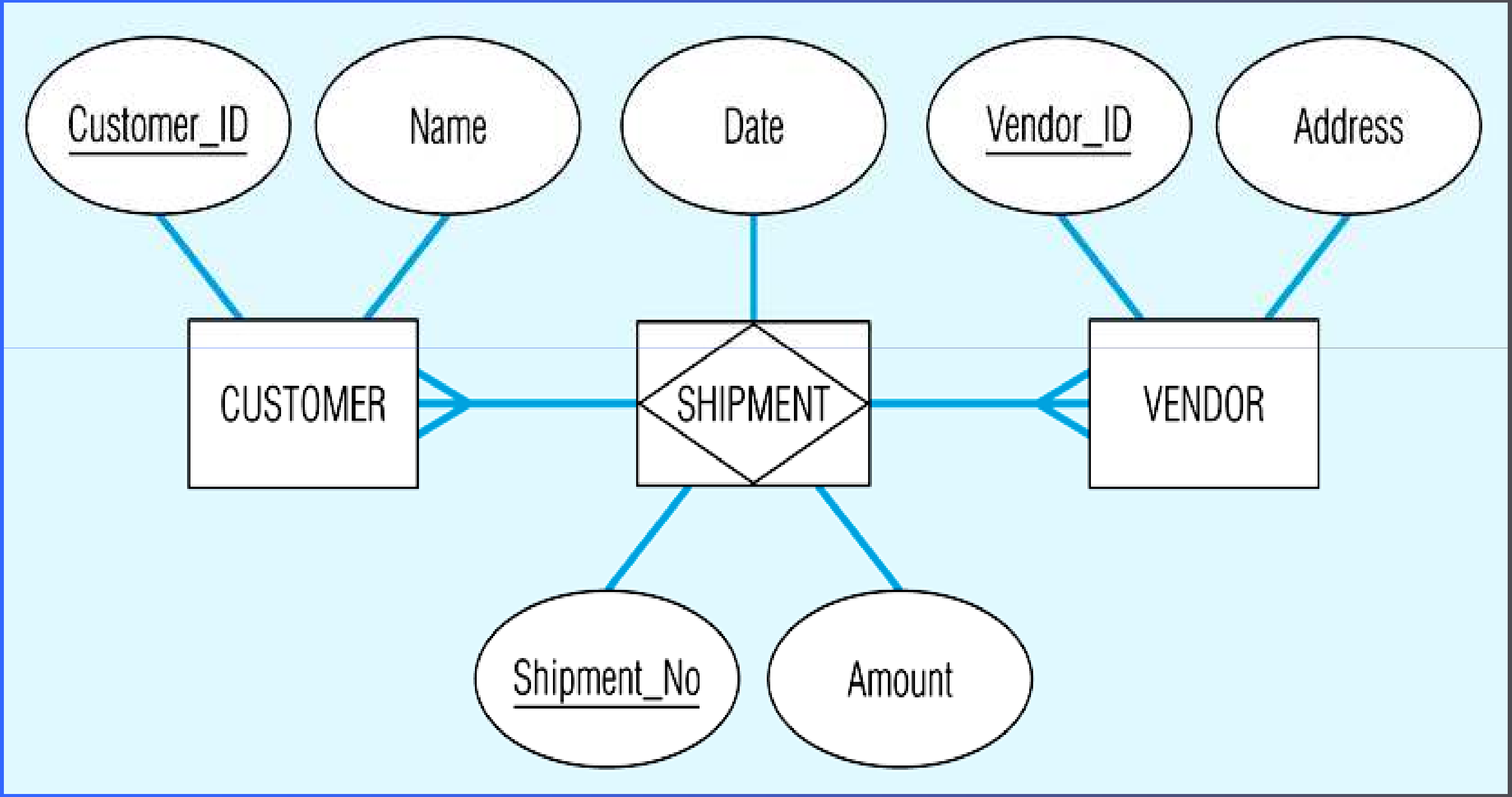
Center_Name	Location	Nurse_in_Charge	Date_Assigned

Transforming EER Diagrams into Relations

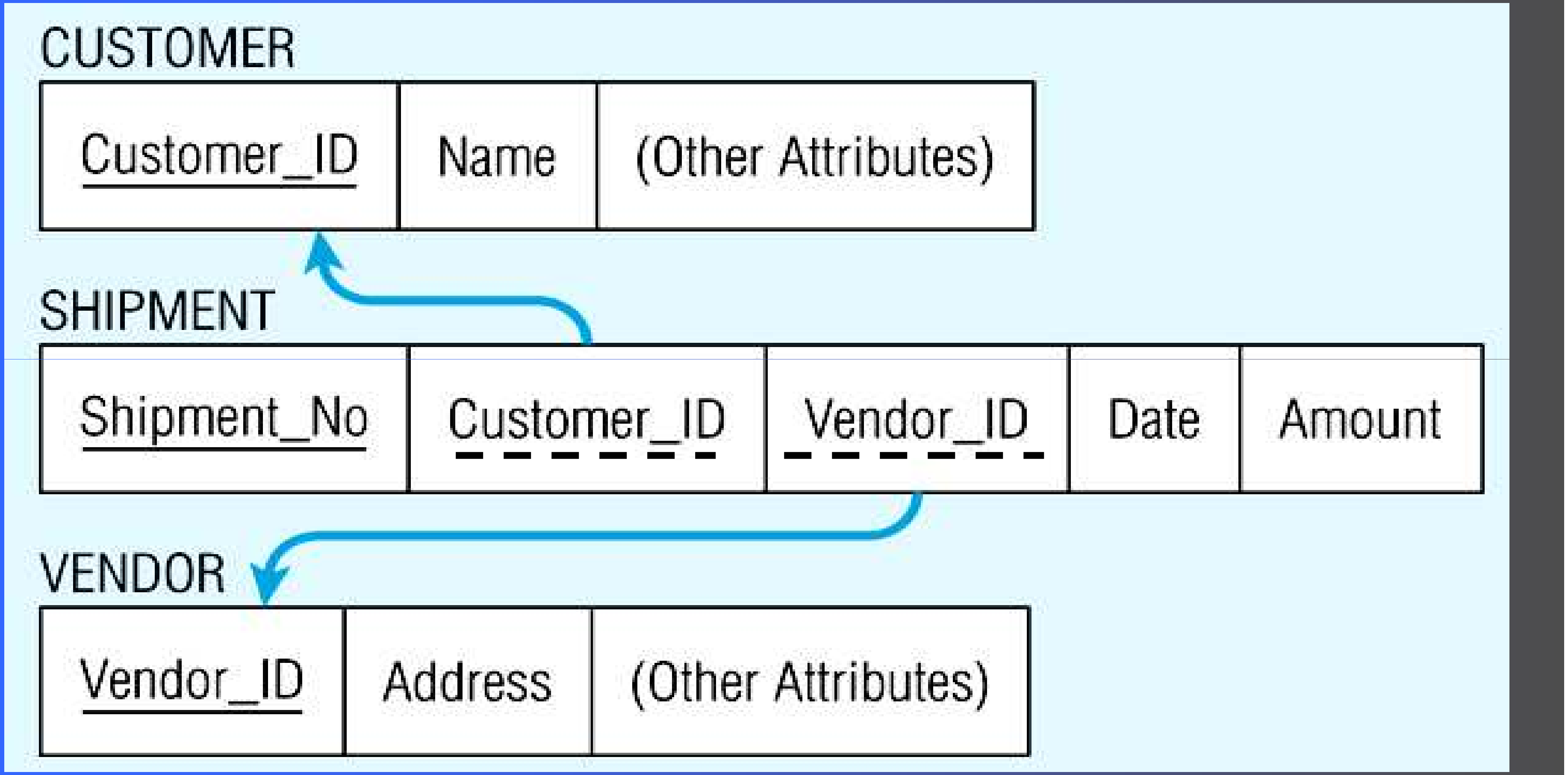
Mapping Associative Entities

- Identifier Not Assigned
  - Default primary key for the association relation is composed of the primary keys of the two entities (as in M:N relationship)
- Identifier Assigned
  - It is natural and familiar to end-users
  - Default identifier may not be unique

Mapping an associative entity



Three resulting relations



CUSTOMER

Customer_ID	Name	(Other Attributes)

SHIPMENT

Shipment_No	Customer_ID	Vendor_ID	Date	Amount

VENDOR

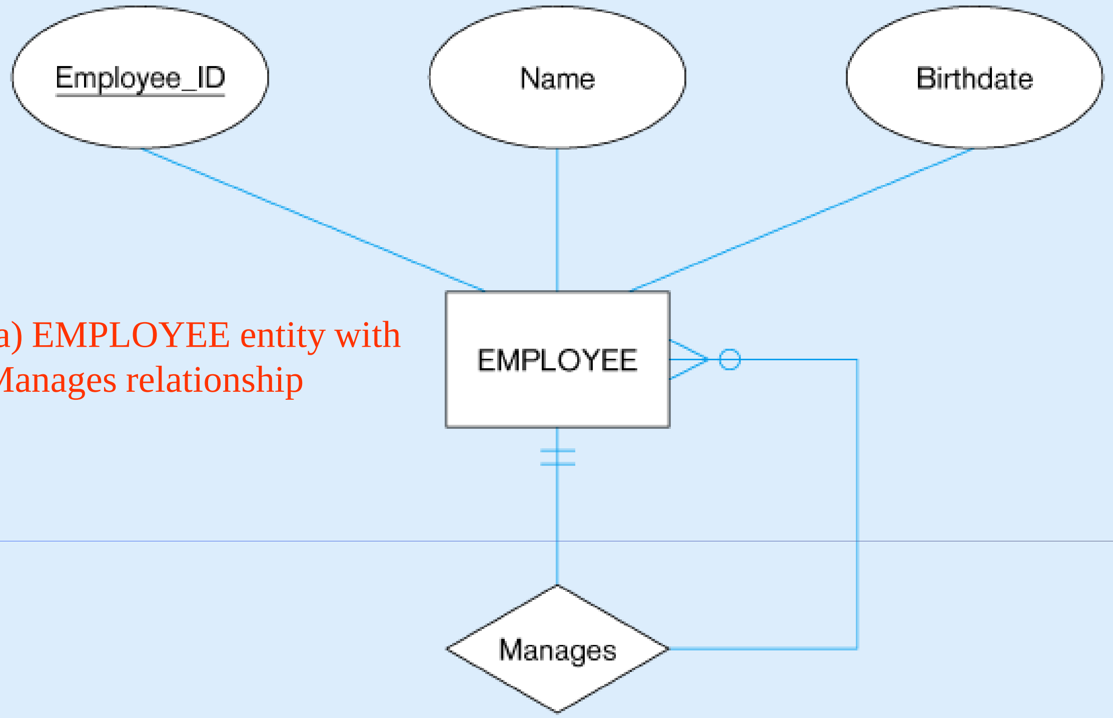
Vendor_ID	Address	(Other Attributes)

Transforming EER Diagrams into Relations

Mapping Unary Relationships

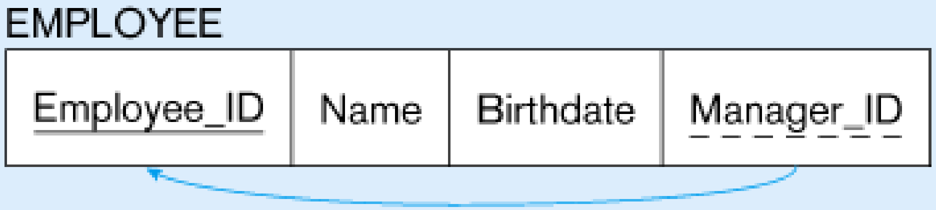
- One-to-Many - Recursive foreign key in the same relation
- Many-to-Many - Two relations:
  - One for the entity type
  - One for an associative relation in which the primary key has two attributes, both taken from the primary key of the entity

Mapping a unary 1:N relationship



(a) EMPLOYEE entity with Manages relationship

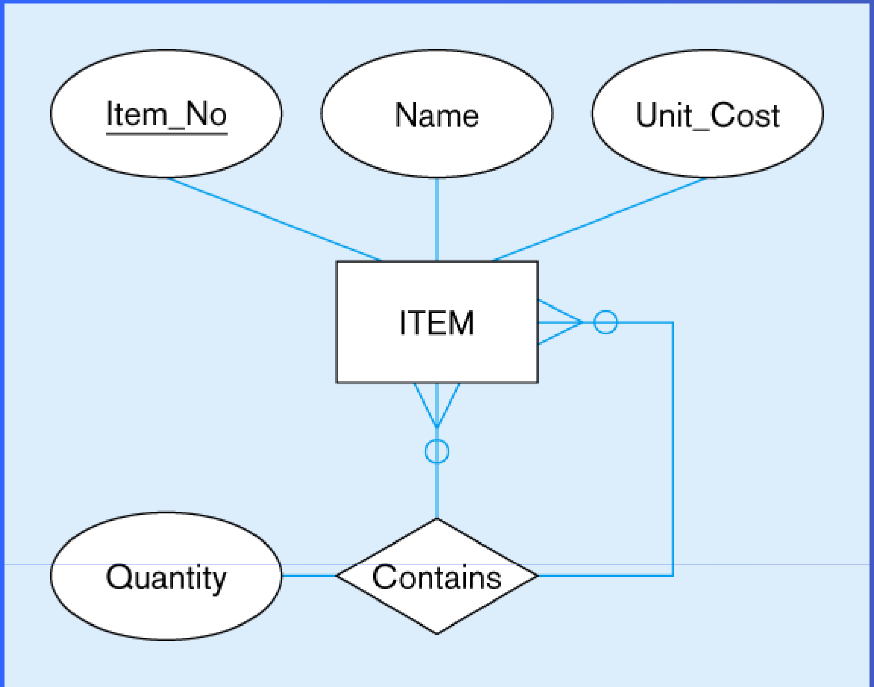
(b) EMPLOYEE relation with recursive foreign key



EMPLOYEE

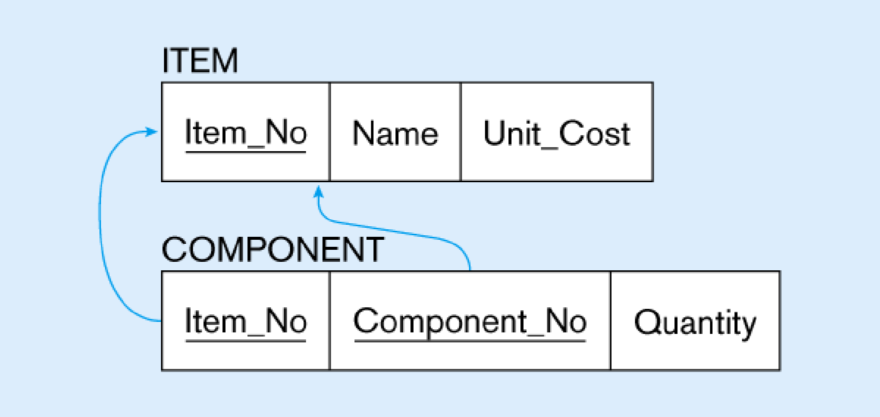
Employee_ID	Name	Birthdate	Manager_ID

Mapping a unary M:N relationship



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

(b) ITEM and COMPONENT relations



ITEM

Item_No	Name	Unit_Cost

COMPONENT

Item_No	Component_No	Quantity

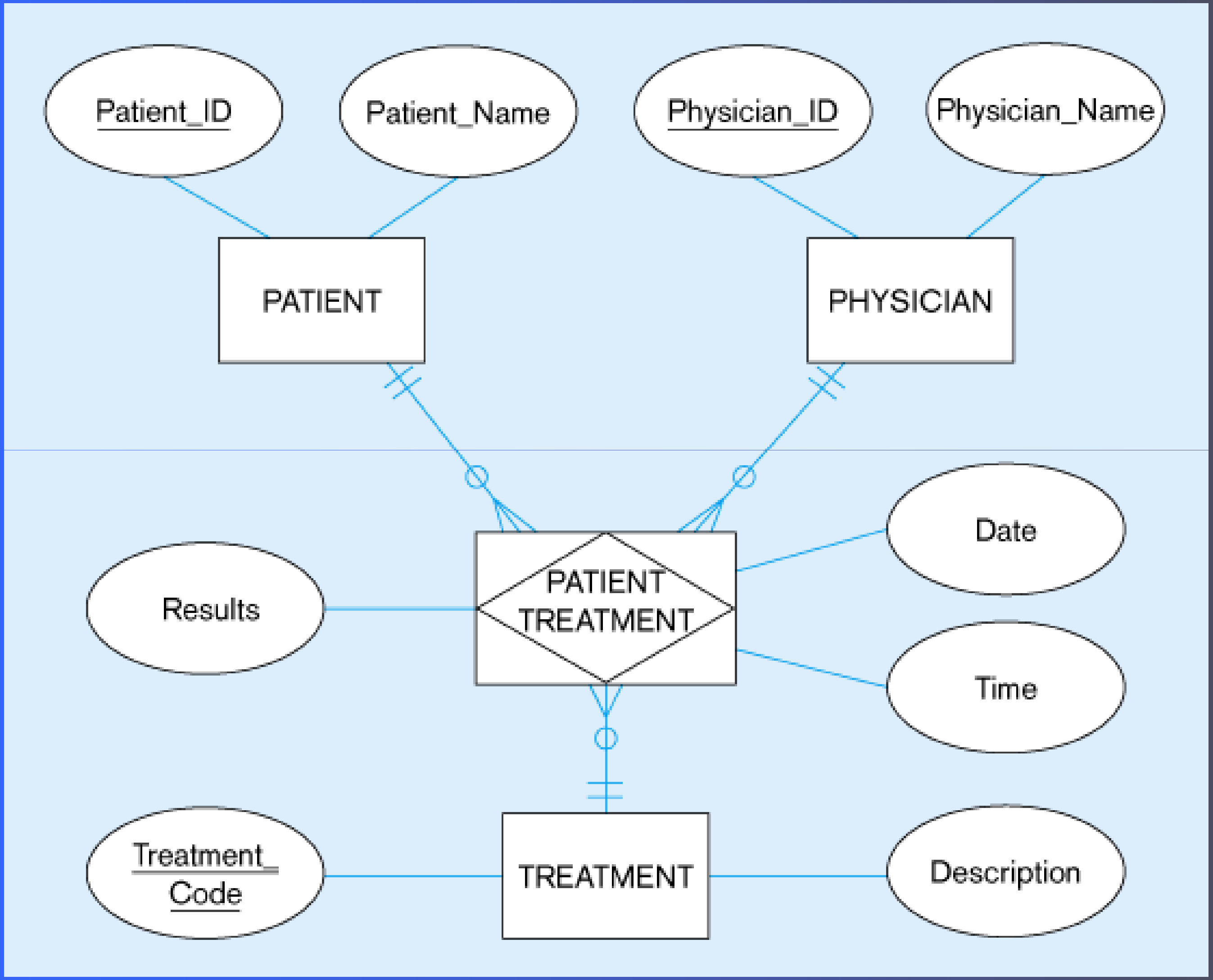
Transforming EER Diagrams into Relations

Mapping Ternary (and n-ary) Relationships

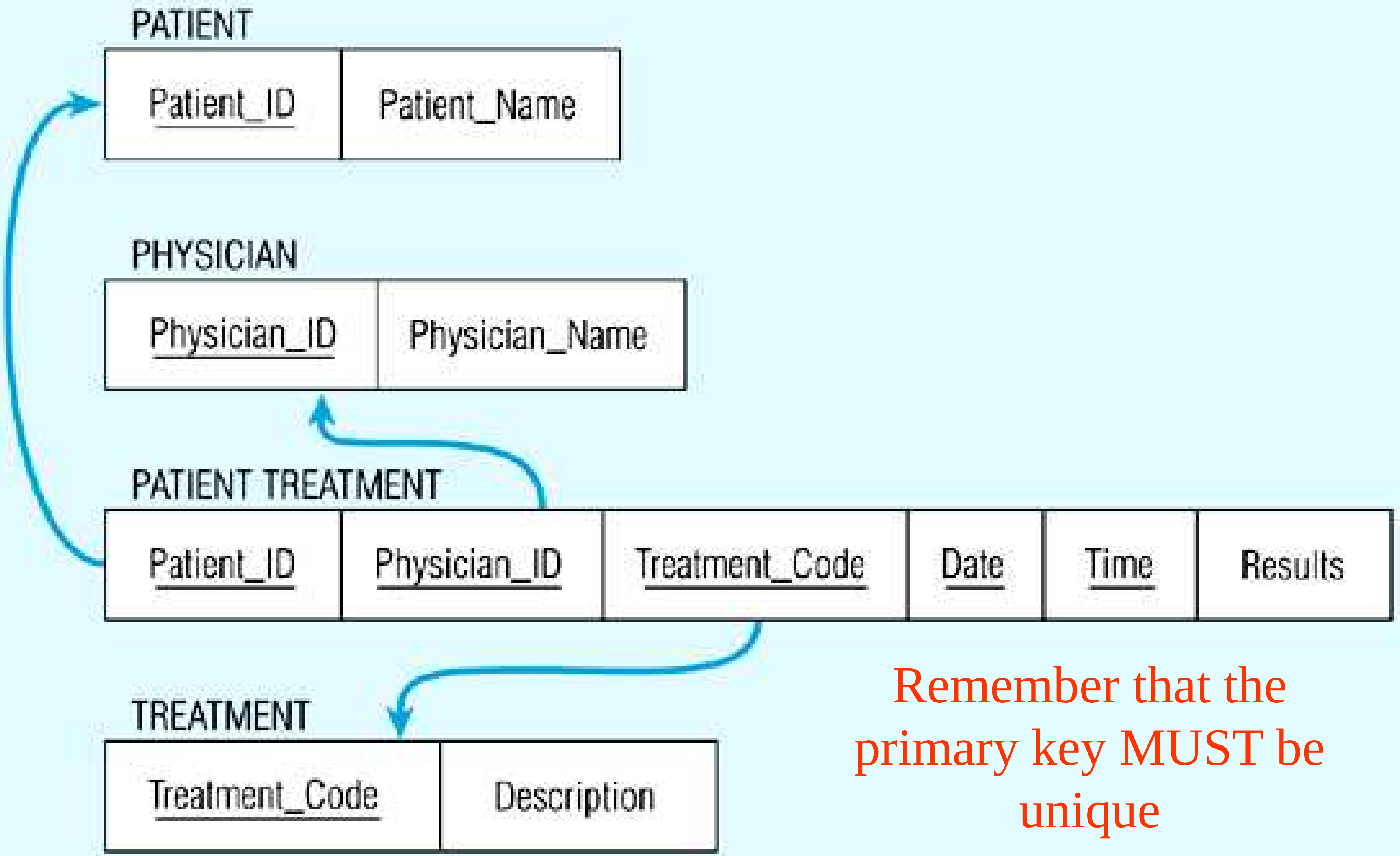
- One relation for each entity and one for the associative entity
- Associative entity has foreign keys to each entity in the relationship

Mapping a ternary relationship

(a) Ternary relationship with associative entity



Mapping the ternary relationship



PATIENT

Patient_ID	Patient_Name

PHYSICIAN

Physician_ID	Physician_Name

PATIENT TREATMENT

Patient_ID	Physician_ID	Treatment_Code	Date	Time	Results

TREATMENT

Treatment_Code	Description

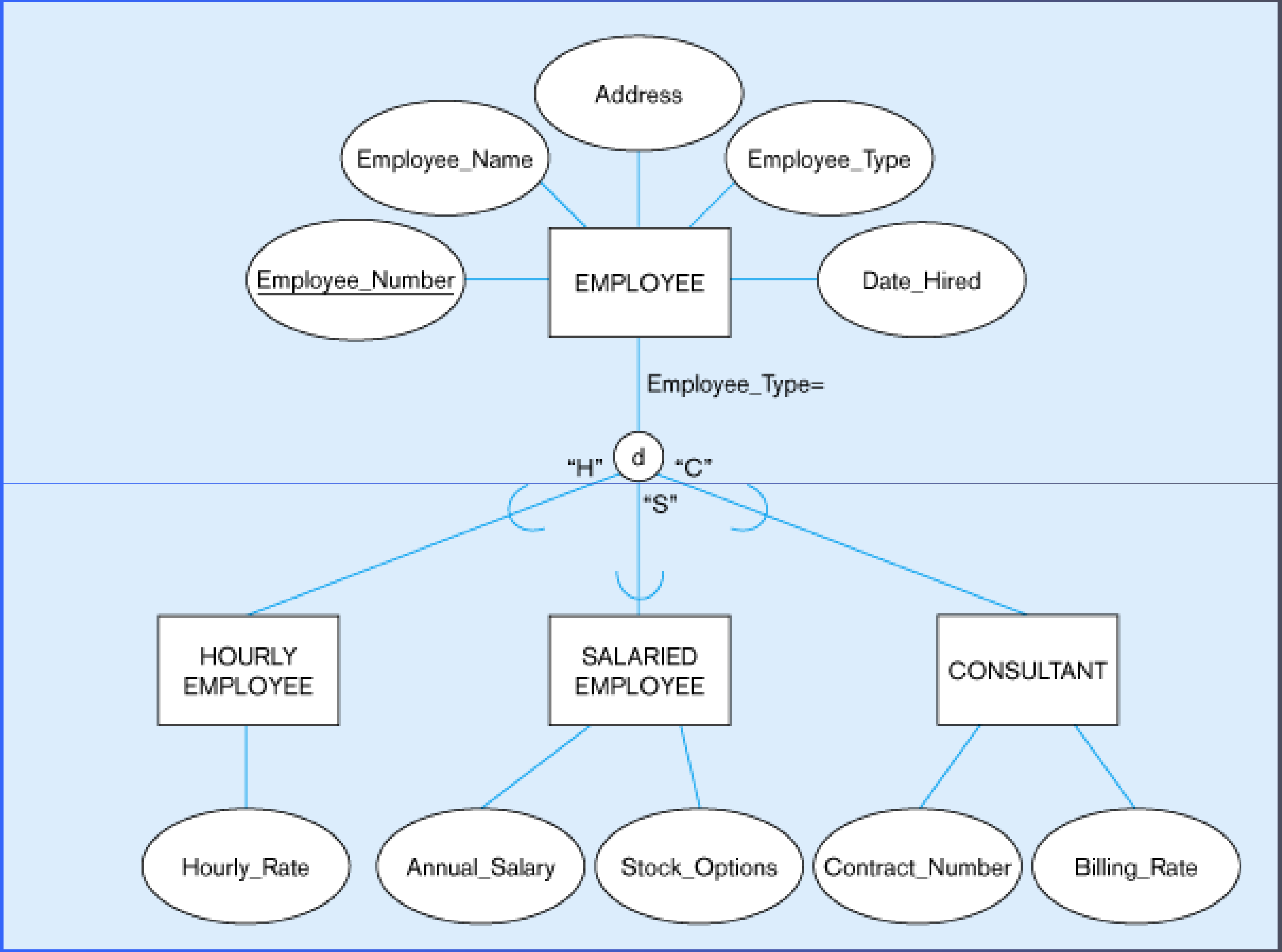
Remember that the primary key MUST be unique

Transforming EER Diagrams into Relations

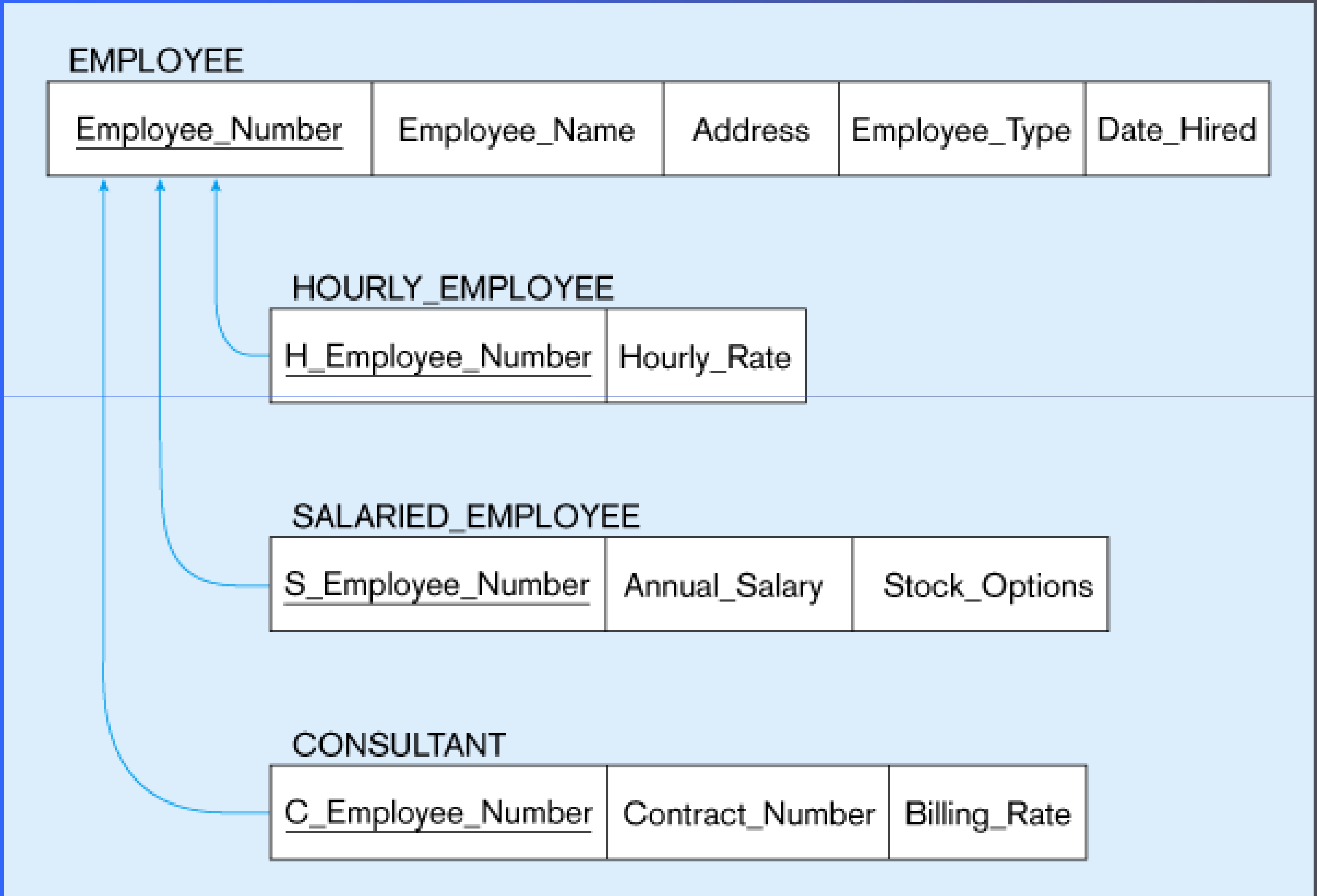
Mapping Supertype/Subtype Relationships

- 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

Supertype / subtype elationships



Mapping Supertype/subtype relationships to relations



EMPLOYEE

Employee_Number	Employee_Name	Address	Employee_Type	Date_Hired

HOURLY_EMPLOYEE

H_Employee_Number	Hourly_Rate

SALARIED_EMPLOYEE

S_Employee_Number	Annual_Salary	Stock_Options

CONSULTANT

C_Employee_Number	Contract_Number	Billing_Rate

What constitutes a Well structured Relation :

-Minimal Redundancy

- Allows users to insert , update and delete rows without inconsistencies.

……………….Next Lecture