**Lesson 17**

**Objectives**

* Transformation of (E)ER Model into Relational Model
  + Attributes
    - Simple
    - Composite
    - Multivalued
  + Entities
    - Regular
    - Weak
    - Associative
  + Relationship
    - Unary
      * One-to-one
      * One-to-Many
      * Many-to-Many
    - Binary
      * One-to-one
        + Minimum/Maximum same on both
        + Optional-Mandatory
      * One-to-Many
      * Many-to-Many
    - Ternary
    - Quaternary
  + Super Type
  + Subtype

**Regular Entities**

Regular entities are mapped to relations.

**Example**

Student

Above entity “Student” will be transformed into Relation

Student (………..)

**Attributes**

**Simple attributes**

E-R attributes map directly onto the columns/fields of relation

**Example**

In the above example regular entity “Customer” mapped into relation and simple attributes mapped as fields/columns of relation.

**Composite attributes**

Use only their simple component (attributes). In other words simple components of composite will be mapped as fields/columns of relation.

**Example**

In the above example simple components (zip, state, city, street) of custome\_address becomes the columns of relation (cutomer).

**Multivalued Attribute**

Separate relation will be created for each multivalued attributes with a foreign key taken from the superior entity.

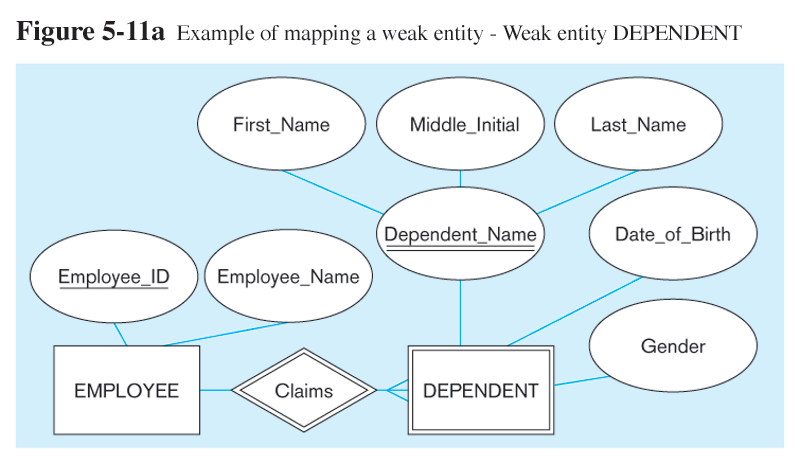
**Example**

In the above example a separate relation “Employee\_Skill” is created for multivalued “Skill” with foreign key “Employee\_ID” from Employee.

**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**

****

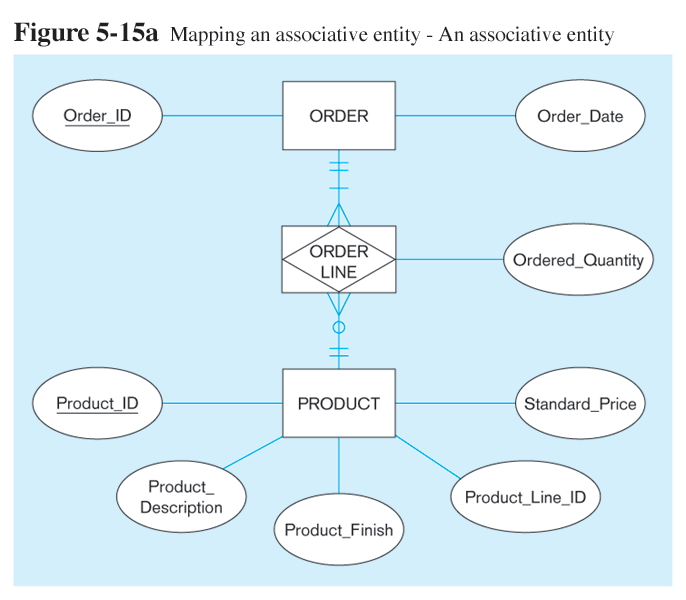
Above ER model having week entity “Dependent” will be transformed into relational as follow.

In the above example “Dependent” is a weak entity. It is transformed into relation with foreign key from Employee. Employee also have composite attribute, so simple components are becomes fields/column.

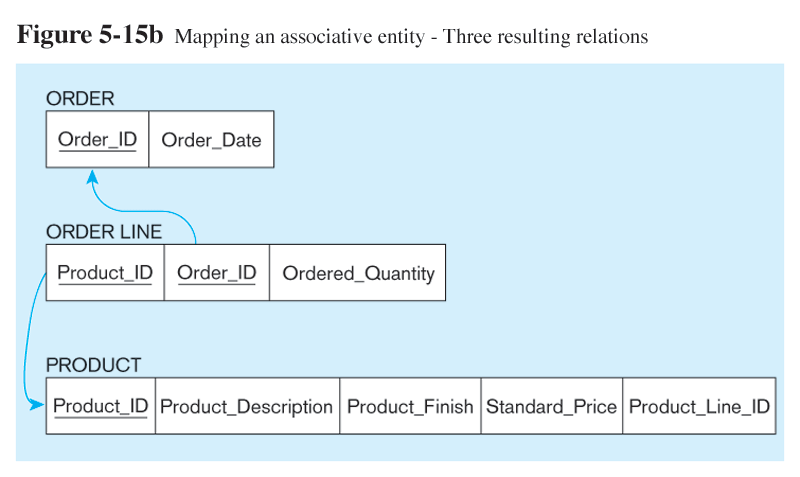
**Associative Entity**

* 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)

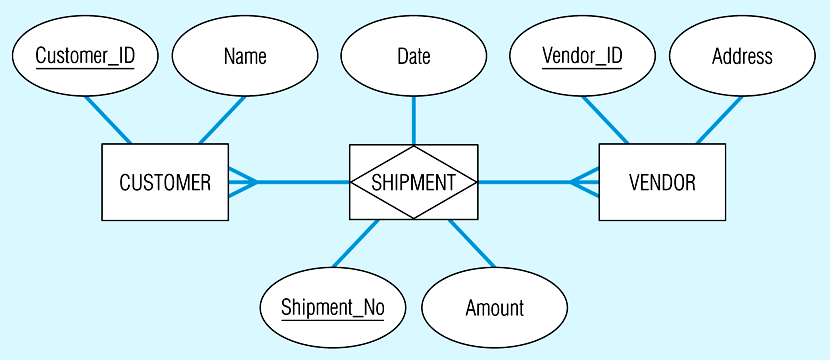
**Example**

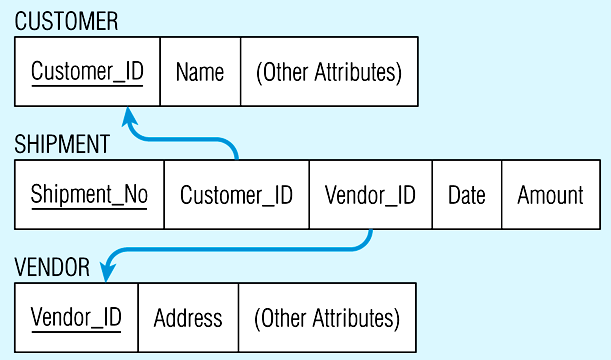
****

In the above example “Order Line” is an associative entity. A separate relation is created for Order line

****

* Identifier Assigned
  + It is natural and familiar to end-users
  + Default identifier may not be unique

****

****

**Relationship**

**Unary Relationship**

**One-to-Many Relationship**

Recursive foreign key in the same relation

**Example**

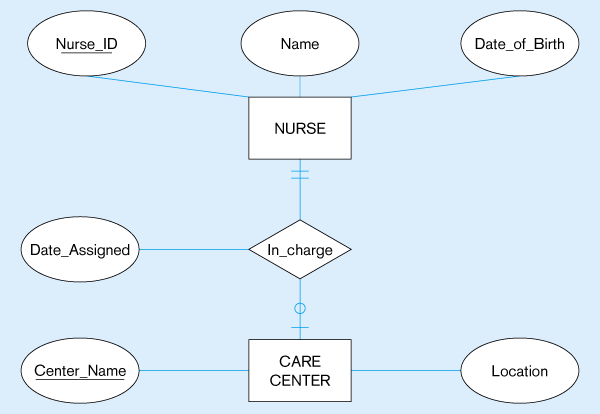
**Many-to-Many Relationship**

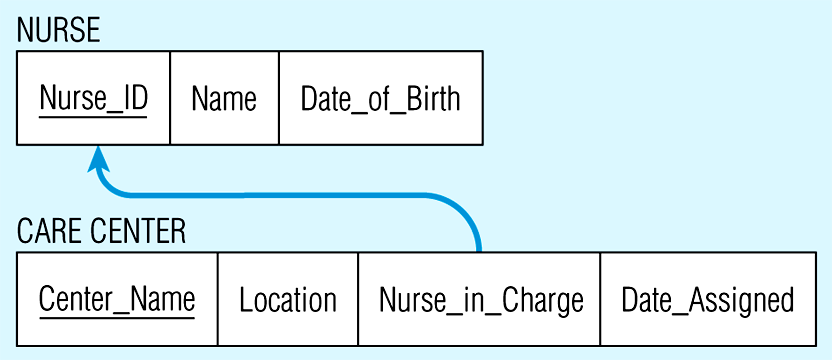
**Binary Relationship**

**One-to-One Relationship**

Primary key on the mandatory side becomes a foreign key on the optional side

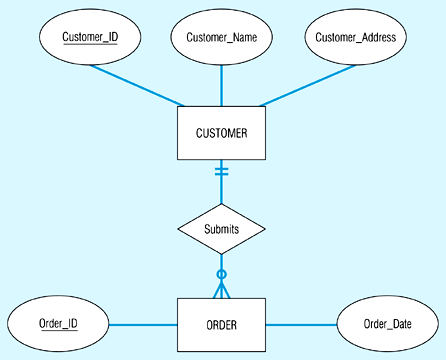
**Example**

****

****

**One-to-Many Relationship**

Primary key on the one side becomes a foreign key on the many side

****

**Many-to-Many Relationship**

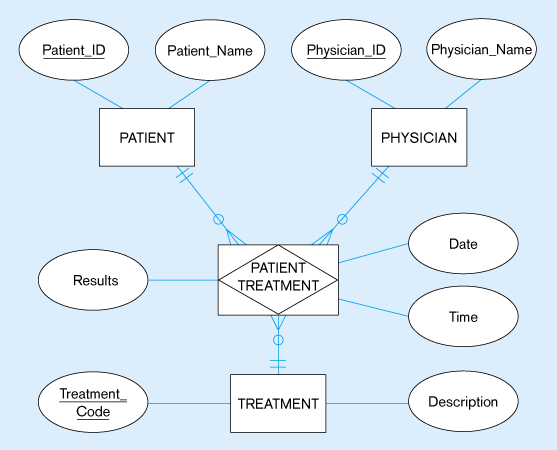
Create a *new relation* with the primary keys of the two entities as its primary key

**Example**

**Ternary/Quaternary Relationship**

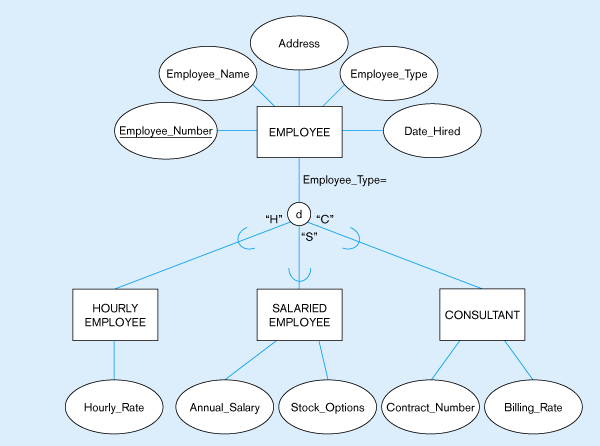
* One relation for each entity and one for the associative entity
* Associative entity has foreign keys to each entity in the relationship

**Example**

****

**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

****