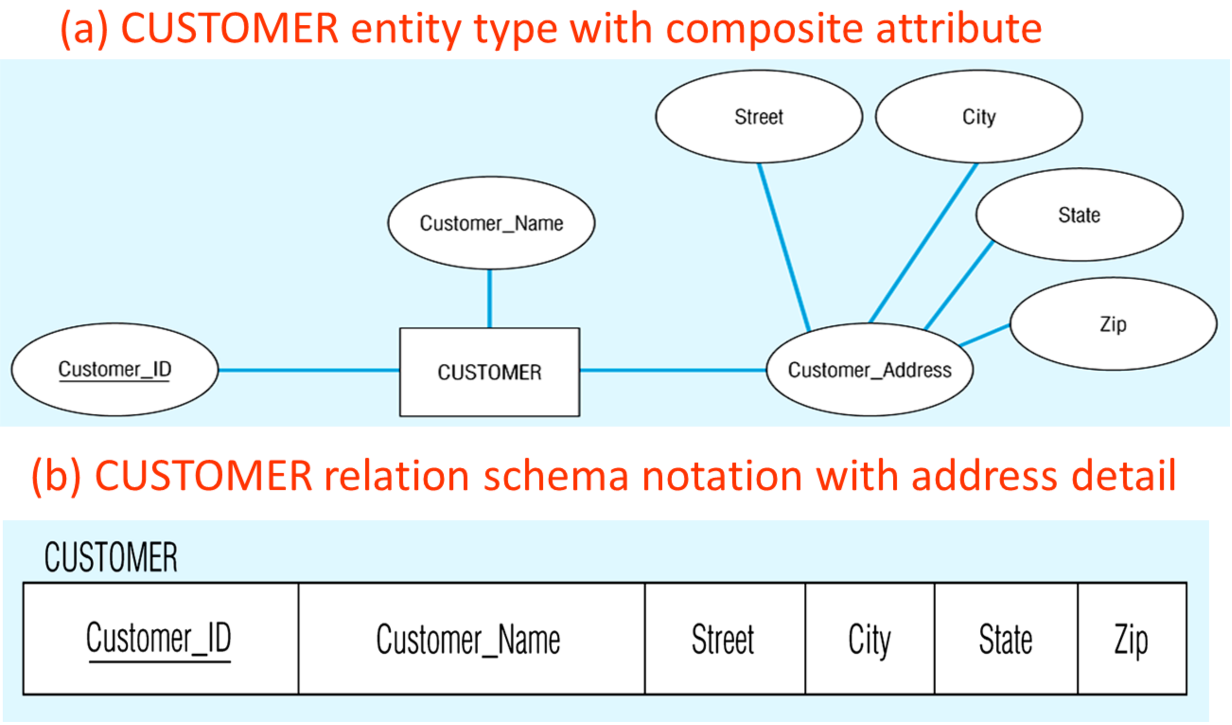
Mapping an ER diagram to a relational schema

Reference: Chapter 5: Logical Database Design and the Relational Model from ***Modern Database Management 9th Edition***. ***Jeffrey A. Hoffer, Mary B. Prescott, Heikki Topi***

Step 1: Mapping Regular (Strong) Entities to Relations

- Each regular entity type in an ER diagram is transformed into a relation.
- The name given to the relation is generally the same as the entity type.
- Each simple attribute of the type becomes an attribute of the relation.
- E-R attributes map directly onto the relation.
- The identifier of the entity type becomes the primary key of the corresponding relation.
- **Composite attributes**: Use only their simple, component attributes (Figure 1).


(a) CUSTOMER entity type with composite attribute

(b) CUSTOMER relation schema notation with address detail

Figure 1. Mapping a composite attribute

- **Multivalued Attribute**: Becomes a separate relation with a **foreign key** taken from the superior entity. Two new relations (rather than one) are created (Figure 2).
  - First relation contains all of the attributes of the entity type except the multivalued attribute.

- Second relation contains two attributes that form the primary key of the second relation.
- The first of these is the primary key for the first relation, which becomes a foreign key in the second relation. The second is the multivalued attribute.

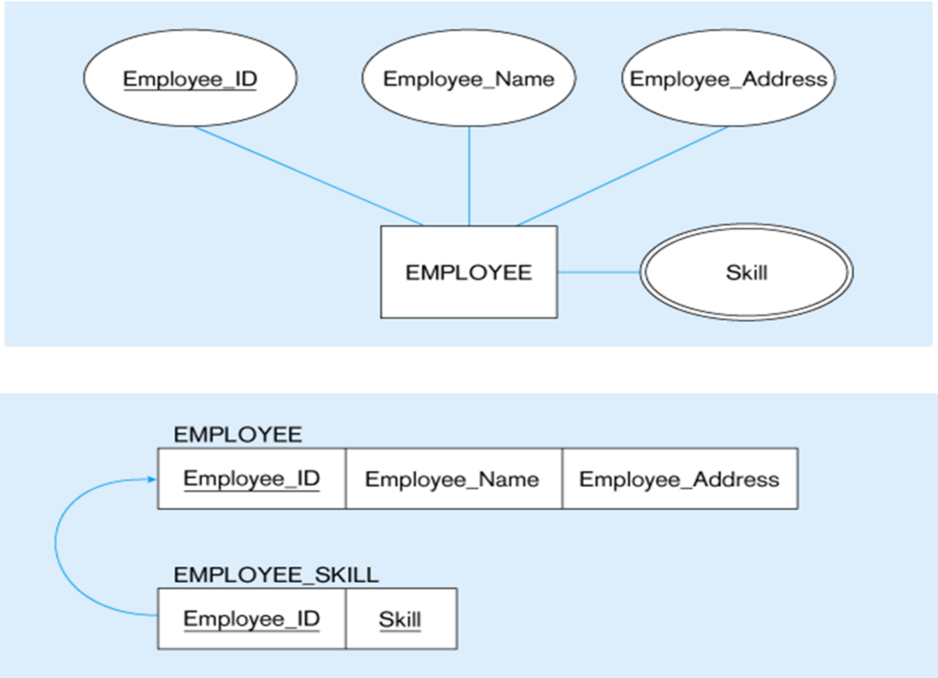


Figure 2. Mapping the multi-valued attribute “skills” into a new relation (EMPLOYEE_SKILL); One-to -many relationship between original entity and new relation

Step 2: Mapping Weak Entities (Figure 3)

- You must already have created a **relation** corresponding to the identifying type.
- For each weak entity type, create a new relation and include all of the simple attributes (or simple components of composite attributes) as attributes of this relation.
- Then include the primary key of the identifying relation as a **foreign key** attribute in this new relation.
- The primary key of the new relation is the combination of this **primary key** of the identifying and **the partial identifier** of the weak entity type.
- A weak entity becomes a separate relation with a **foreign** key taken from the superior (identifying) entity
    - Primary key composed of: a) Partial identifier of weak entity and b) Primary key of identifying relation (strong entity).

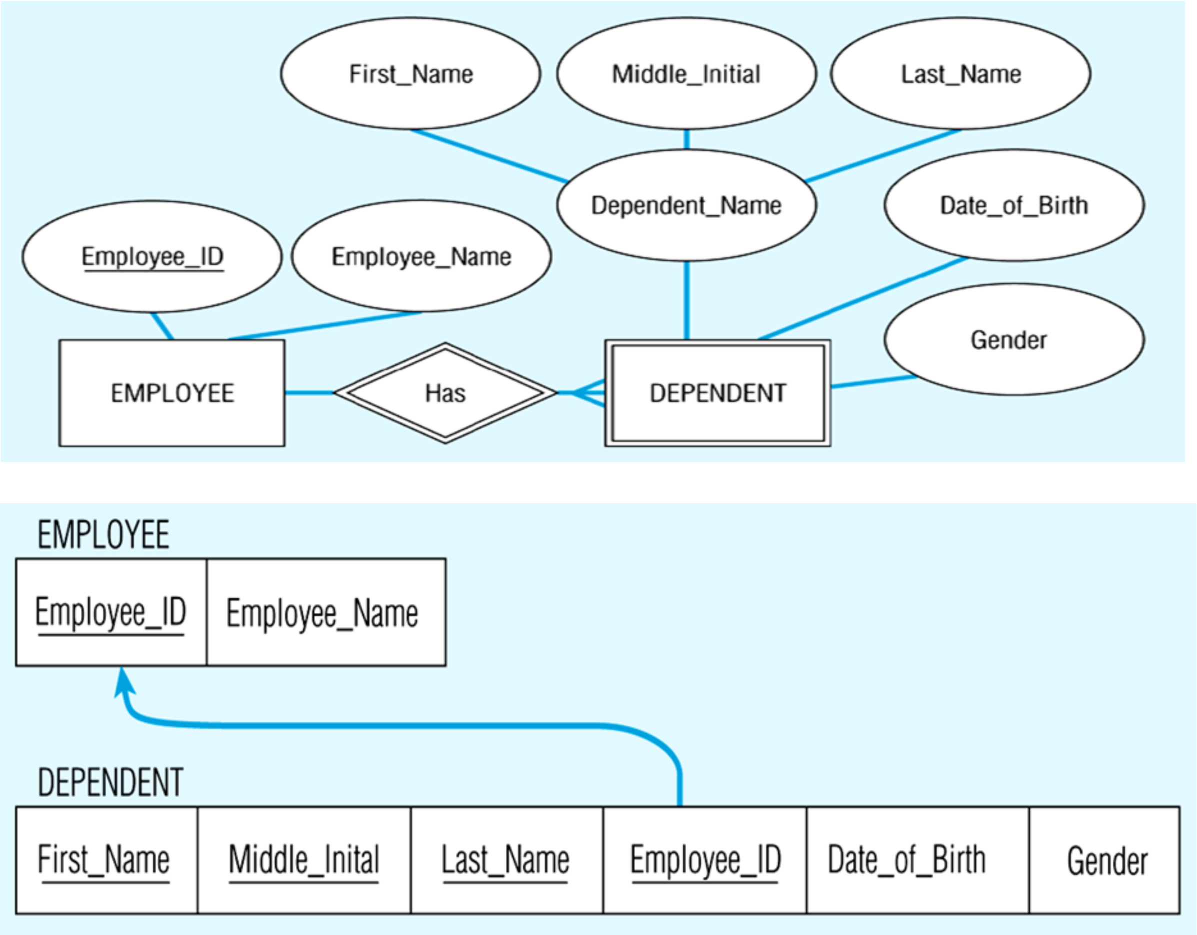
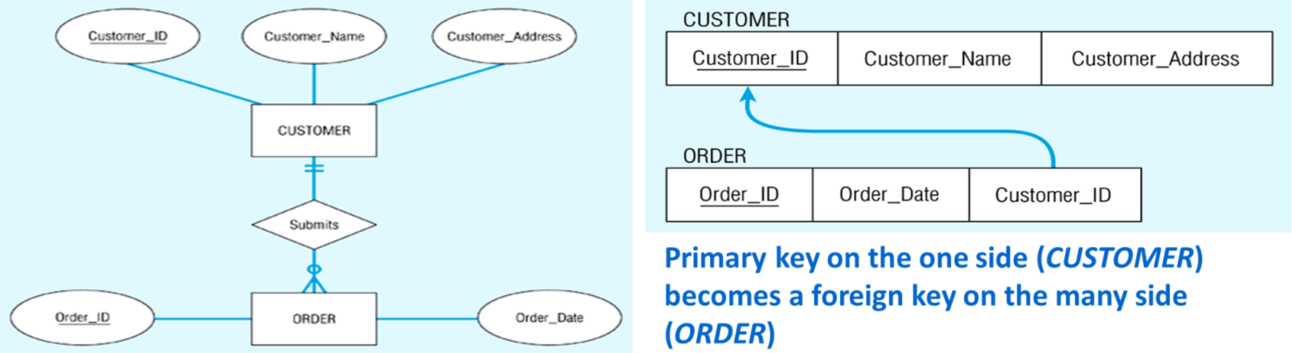


Figure 3. Map weak entity type

Step 3: Mapping Binary Relationships

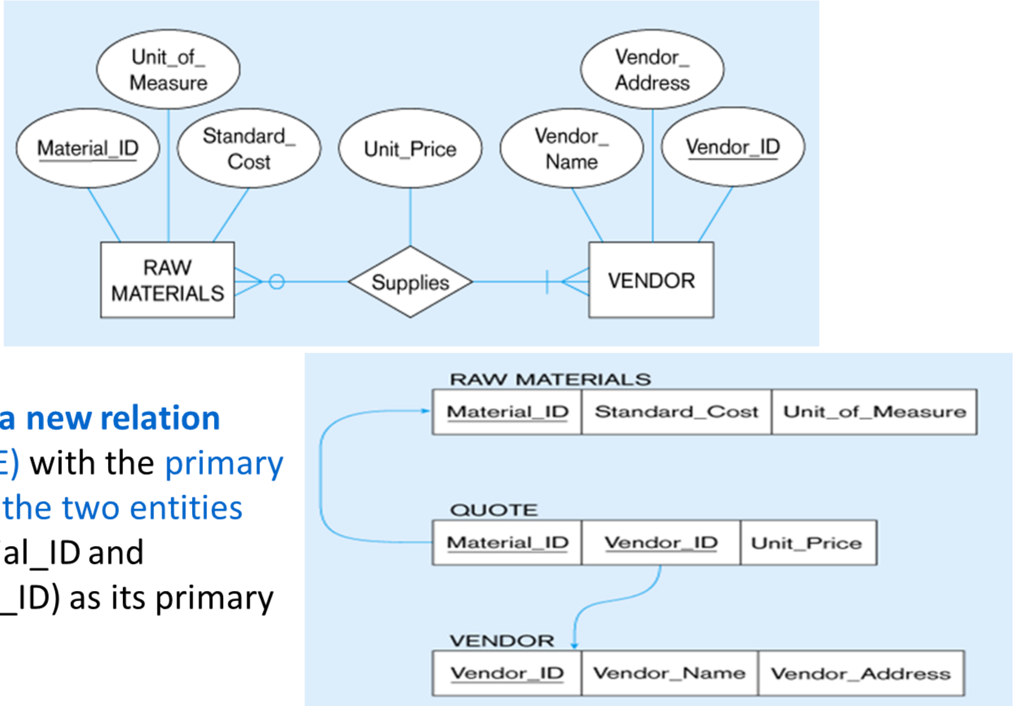
- The procedure for representing relationships depends on both the degree of the relationships (unary, binary, or ternary) and the cardinalities of the relationships.
- One-to-Many–Primary key on the one side becomes a **foreign** key on the many side (Figure 4).
    - First create a relation for each of the two entity types participating in the relationship.
    - Next include the **primary key** attribute(s) of the entity on the one-side as a **foreign key** in the relation that is on the many-side.



Primary key on the one side (*CUSTOMER*) becomes a foreign key on the many side (*ORDER*)

Figure 4. Map 1:N relationship

- Many-to-Many–Create a ***new relation*** with the **primary keys** of the two entities as its **primary key (**Figure 5**).**
    - If such a relationship exists between entity types A and B, we create a new relation C, then include as **foreign keys** in C the primary keys for A and B, then these attributes become the primary key of C.

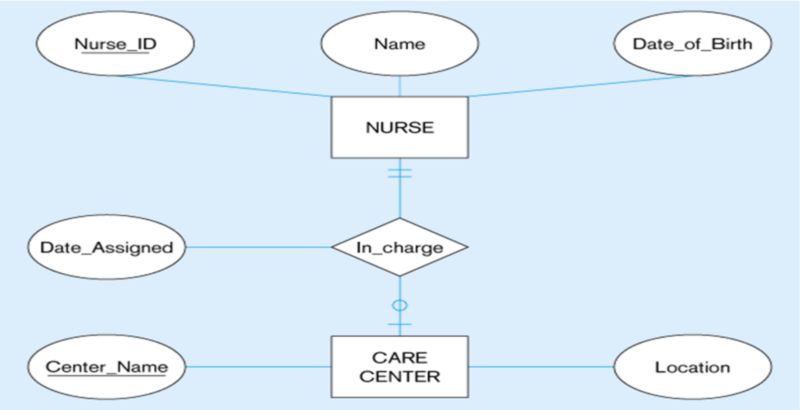


Create a new relation (QUTOE) with the primary keys of the two entities (Material_ID and Vendor_ID) as its primary key.

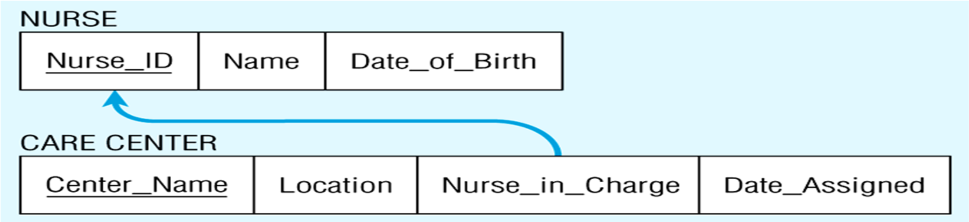
Figure 5. Map M:N relationship

- One-to-One–Primary key on the **partial participation side** becomes a **foreign key** on the **total participation side**. These can be viewed as a special case of one-to-many relationships.
    - Firstly, two relations are created, one for each of the participating entity types.

- Secondly, the **primary key** of one of the relations is included as a **foreign key** in the other relation.
- In a 1:1 relationship, the association in one direction is nearly always **optional** one, while the association in the other direction is **mandatory** one.
- You should include in the relation on the optional side of the relationship the **foreign key** of the entity type that has the **mandatory** participation in the 1:1 relationship.
- This approach avoids the need to store null values in the foreign key attribute.
- Any attributes associated with the relationship itself are also included in the same relation as the **foreign key**.



Primary key on the **partial participation side** becomes a **foreign key** on the **total participation side**. These can be viewed as a special case of 1:N relationships.



NURSE

Nurse_ID	Name	Date_of_Birth

CARE CENTER

Center_Name	Location	Nurse_in_Charge	Date_Assigned

Figure 6. Map binary 1:1 relationships

Primary keys are underlined.

Referential integrity

- Referential Integrity–rule states that any foreign key value (on the relation of the many side) MUST match a primary key value in the relation of the one side. (Or the foreign key can be null).
- Referential integrity constraints are drawn via arrows from dependent to parent table.

- Referential integrity constraints are implemented with foreign key to primary key references.