

# COP 4710: Database Systems Spring 2004

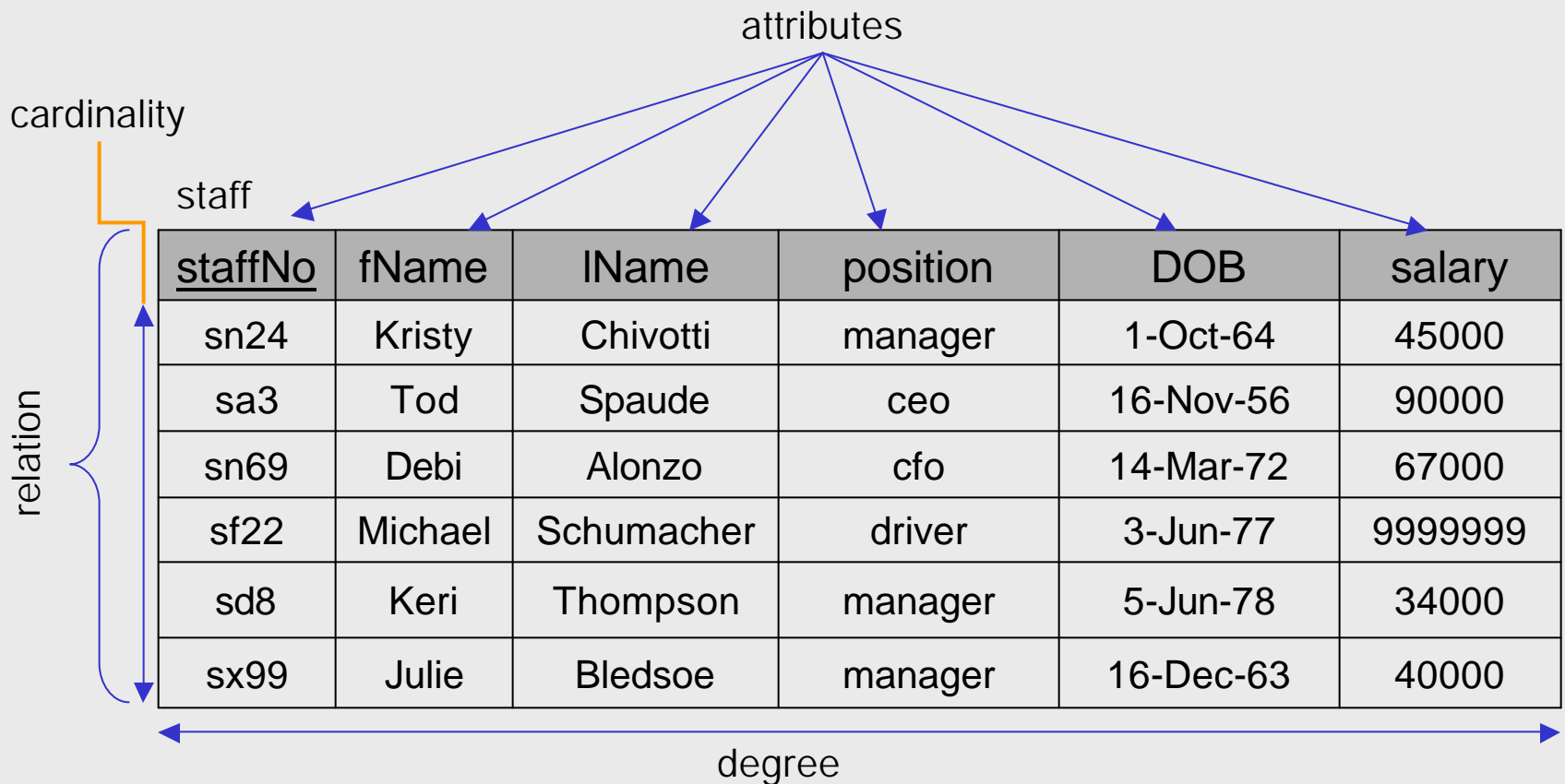
-Day 6 – January 26, 2004 –  
The Relational Data Model – Part 2

Instructor : Mark Llewellyn  
markl@cs.ucf.edu  
CC1 211, 823-2790  
<http://www.cs.ucf.edu/courses/cop4710/spr2004>

School of Electrical Engineering and Computer Science  
University of Central Florida



# An Example Relation



# Logical Design

- During logical design you transform the conceptual design into relational database schemas.
  - The inputs to the process are the E-R diagrams and the outputs are the relational schemas.
- Mapping the E-R diagrams to relations is a relatively straightforward process with a well-defined set of rules. In fact many CASE tools (Computer Aided Software Engineering tools) can automatically perform many of the conversion steps. However, it is important that you understand the steps in this process for three reasons:
  1. CASE tools often cannot model more complex data relationships such as ternary relationships and superclass/subclass relationships. These steps will need to be done manually.
  2. There are some legitimate alternatives where you must manually choose an alternative.
  3. You need to be prepared to perform a quality check on the results obtained with the CASE tool.



# Logical Design (cont.)

- In the steps that we'll need to follow to map E-R diagrams into relational schemas, it will be helpful to remember that we've defined three basic types of entities which are summarized below:
  - **Regular (strong) entities** are entities that have an independent existence and generally represent real-world objects such as persons or products. Represented in ERDs by rectangles with a single line.
  - **Weak entities** are entities that cannot exist except with an identifying relationship with an owner (strong) entity type. Weak entities are identified by a rectangle with a double line.
  - **Associative entities** (also sometimes called gerunds) are formed from many-to-many relationships between other entity types. Associative entities are represented by a rectangle with a single line that enclosed the diamond relationship symbol.



# Mapping E-R Diagrams to Relational Schemas

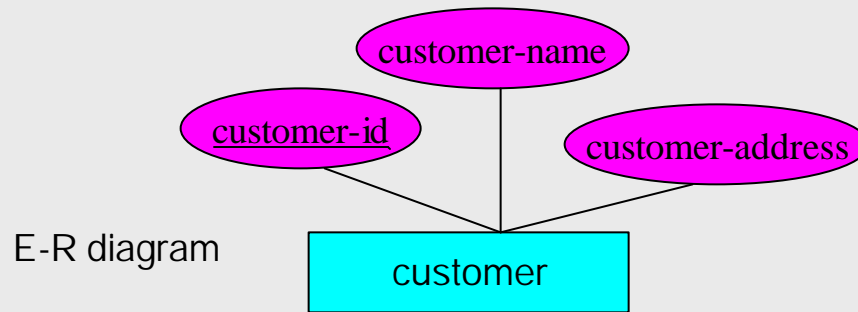
## STEP 1: Mapping Regular (Strong) Entities

- Each regular entity in an ERD is transformed into a relation schema.
- The name given to the relation is generally the same as the entity type.
- Each simple attribute of the entity type becomes an attribute of the relation schema.
- The identifier becomes the primary key of the corresponding relation.



# Mapping E-R Diagrams to Relational Schemas (cont.)

## STEP 1: Map Regular (Strong) Entities - EXAMPLE



customer

<u>customer-id</u>	customer-name	customer-address
--------------------	---------------	------------------

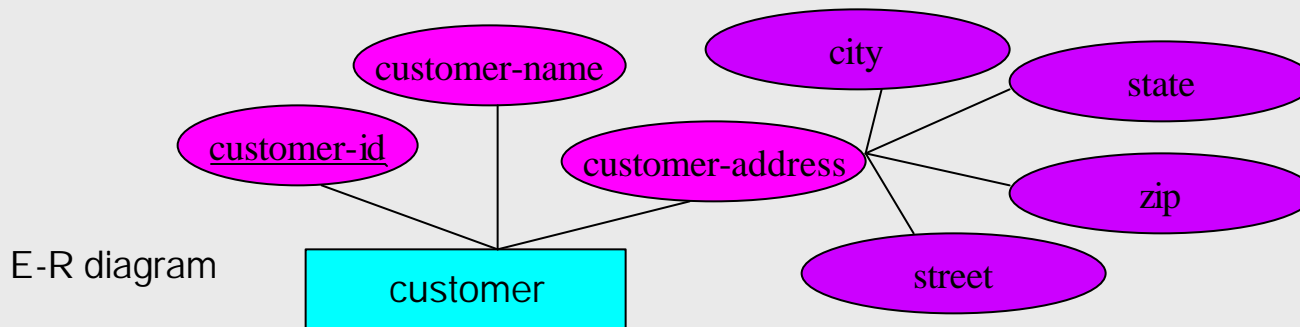
Customer relation



# Mapping E-R Diagrams to Relational Schemas (cont.)

## Composite Attributes:

- When a regular entity type has a composite attribute, only the simple component attributes of the composite attribute are included in the new relation schema.



customer

<u>customer-id</u>	customer-name	street	city	state	zip
--------------------	---------------	--------	------	-------	-----

Customer relation



# Mapping E-R Diagrams to Relational Schemas (cont.)

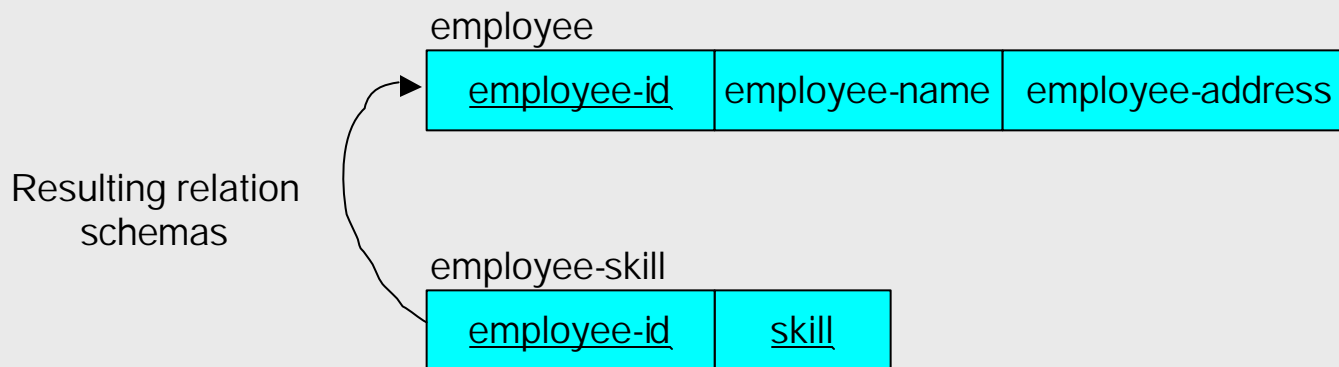
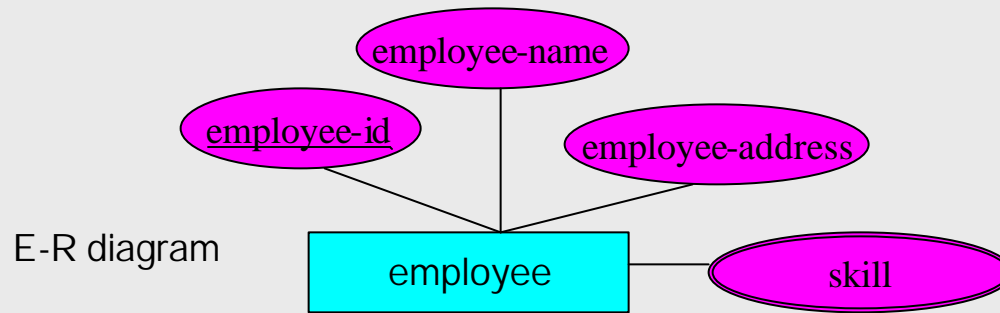
## Multi-valued Attributes:

- When a regular entity type contains a multi-valued attribute, two new relation schemas (rather than one) are created.
- The first relation schema contains all of the attributes of the entity type except the multi-valued attribute. The second relation schema contains two attributes that form the primary key of the second relation schema. The first of these attributes is the primary key of the first relation schema, which becomes a foreign key in the second relation. The second is the multi-valued attribute.
- The name of the second relation should capture the semantics of the multi-valued attribute.



# Mapping E-R Diagrams to Relational Schemas (cont.)

## Multi-valued Attributes Example:



# Mapping E-R Diagrams to Relational Schemas (cont.)

## Multi-valued Attributes:

- Notice in the previous relational schemas constructed due to the multi-valued attribute *skill*, that the resulting relation schema *employee-skill* has only key attributes.
- Each tuple simply records the fact that a given employee possesses a certain skill.
- This provides the database designer the opportunity to suggest to the users that new attributes can be added to this relation.
  - For example, the attributes *years-experience* and/or *certification-date* might be appropriate new values to add to this relation.



# Mapping E-R Diagrams to Relational Schemas

## STEP 2: Mapping Weak Entities

- Recall that a weak entity type does not have an independent existence, but exists only through an identifying relationship with another entity type called the owner.
- A weak entity does not have a complete identifier, but must have an attribute called a partial identifier that permits distinguishing the various occurrences of the weak entity for each owner entity instance.
- The following procedure assumes that you have already created a relation schema corresponding to the identifying entity type. If you have not done this – do it now before proceeding.



# Mapping E-R Diagrams to Relational Schemas

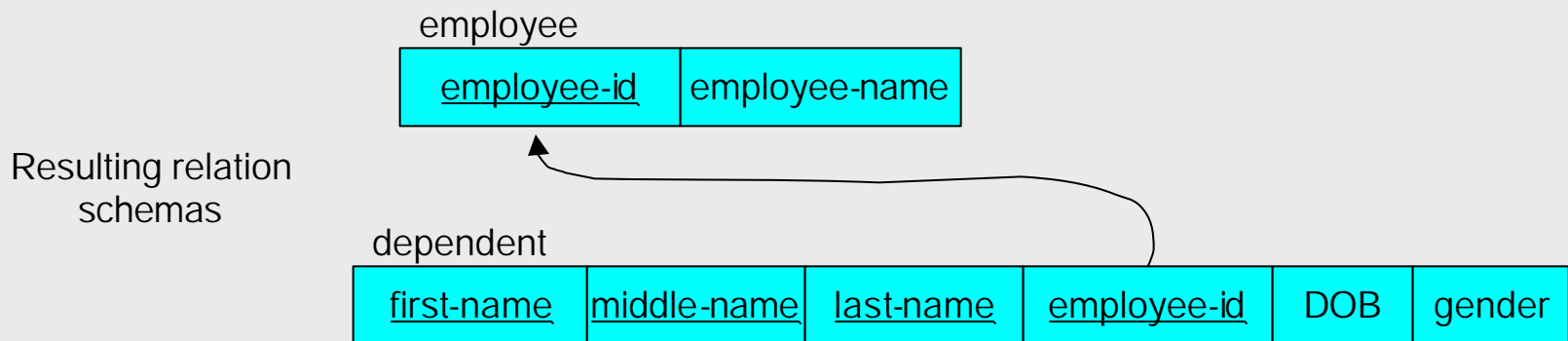
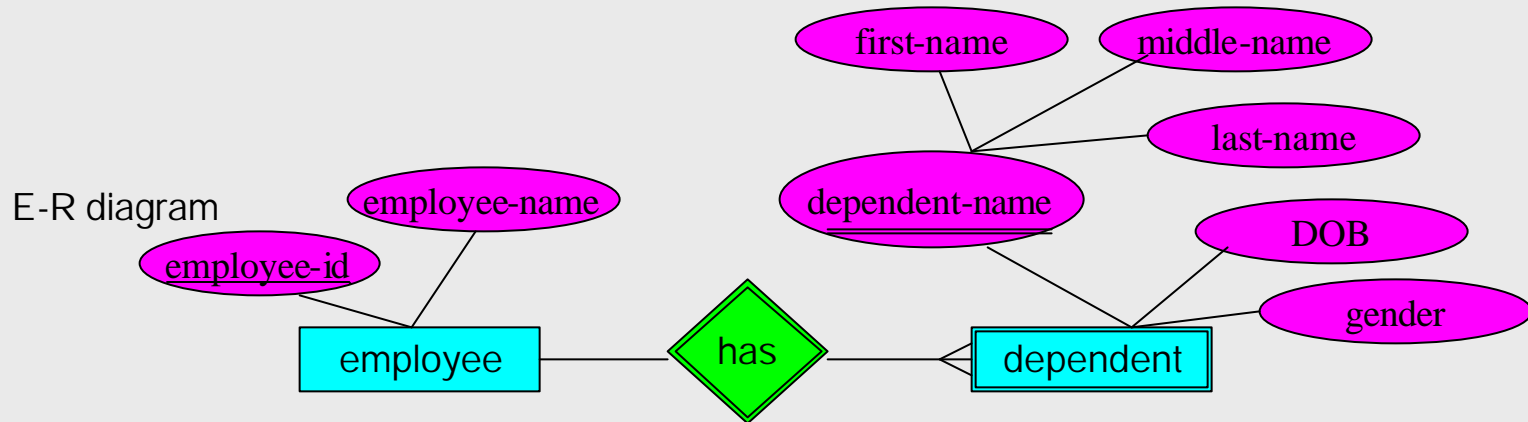
## STEP 2: Mapping Weak Entities - continued

- For each weak entity type, create a new relation schema and include all of the simple attributes (or simple components of composite attributes) as attributes of this relation schema.
- Then include the primary key of the identifying relation as a foreign key attribute in this new relation schema.
- The primary key of the new relation schema is the combination of this primary key of the identifying relation and the partial identifier of the weak entity type.



# Mapping E-R Diagrams to Relational Schemas

## STEP 2: Mapping Weak Entities - EXAMPLE



# Mapping E-R Diagrams to Relational Schemas

## STEP 3: Mapping Binary Relationships

- The procedure for mapping relationships into the relational model depends on both the degree of the relationship (unary, binary, ternary, etc.) and the cardinalities of the relationships.
- We'll look at the most common and important of these over the next several pages. Note that binary 1:M and binary M:1 relationships are symmetric.



# Mapping E-R Diagrams to Relational Schemas

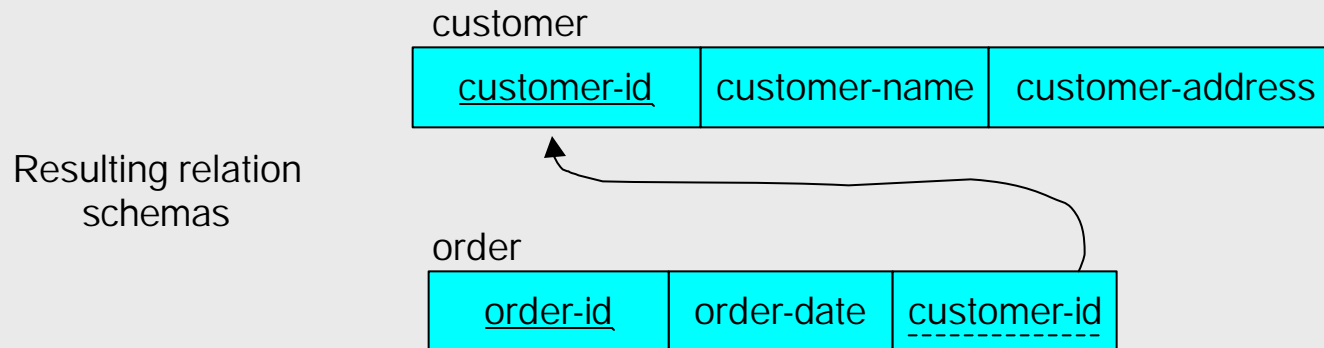
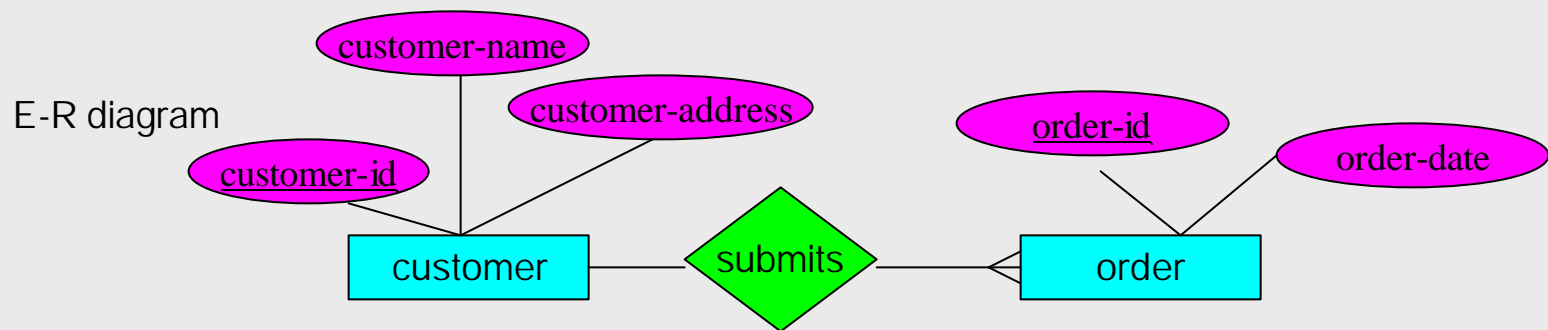
## STEP 3: Binary 1:M Relationships

- For each binary 1:M relationship, first create a relation schema for each of the two entity types participating in the relationship using the procedure from Step 1.
- Next, include the primary key attribute (or attributes) of the entity on the one-side of the relationship as a foreign key in the relation that is on the many-side of the relationship. (The primary key migrates to the many-side.)



# Mapping E-R Diagrams to Relational Schemas

## STEP 3: Binary 1:M - EXAMPLE



# Mapping E-R Diagrams to Relational Schemas

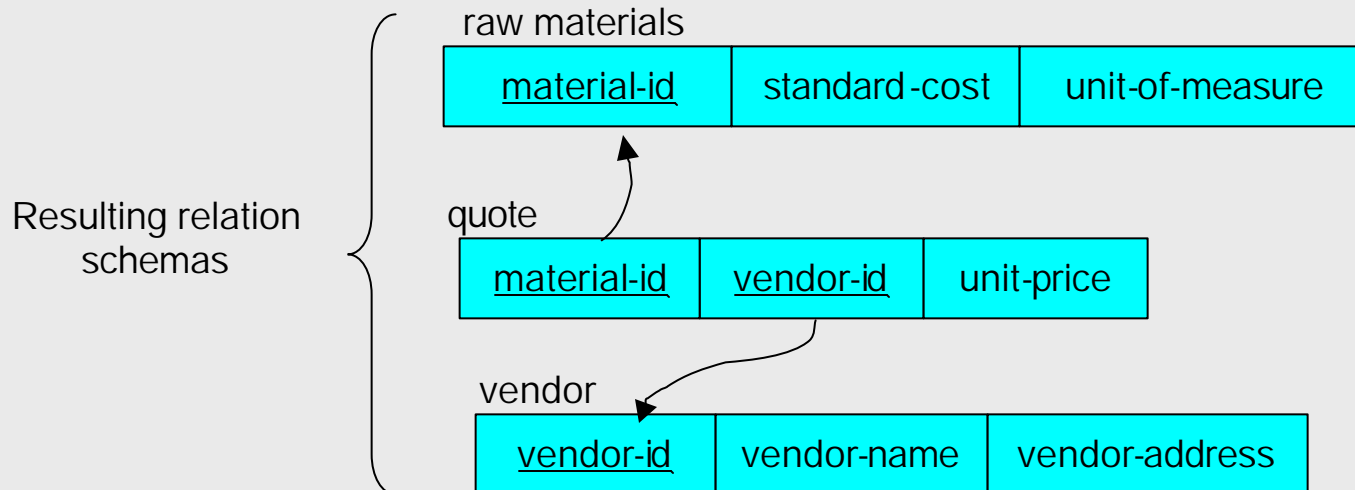
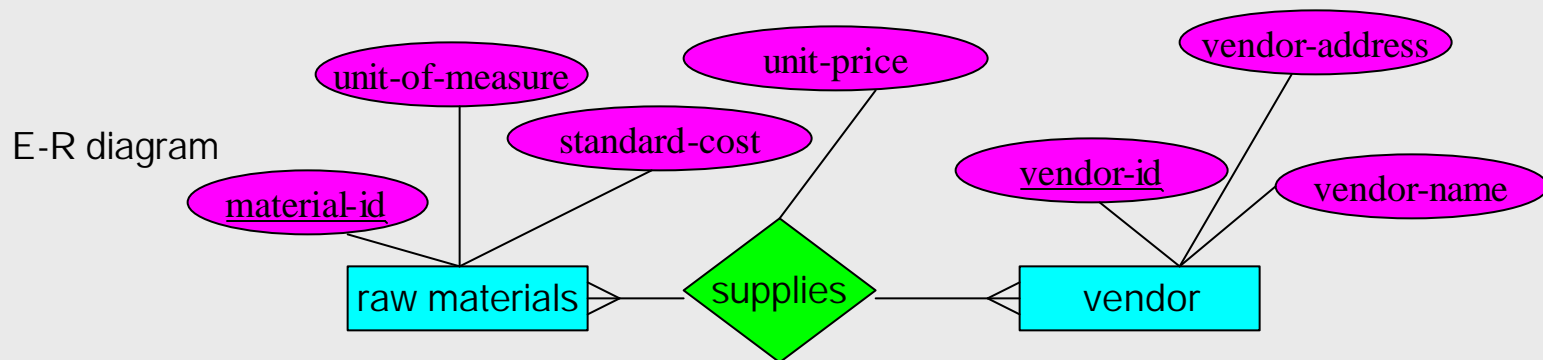
## STEP 3: Binary M:M Relationships

- For each binary M:N relationship between two entity types A and B, first create a new relation schema C.
- Include as foreign key attributes in C the primary key for each of the two participating entity types A and B. These attributes becomes the primary key of relation schema C.
- Any non-key attributes that are associated with the M:N relationship between A and B are included in the relation schema C.



# Mapping E-R Diagrams to Relational Schemas

## STEP 3: Binary M:N - EXAMPLE



# Mapping E-R Diagrams to Relational Schemas

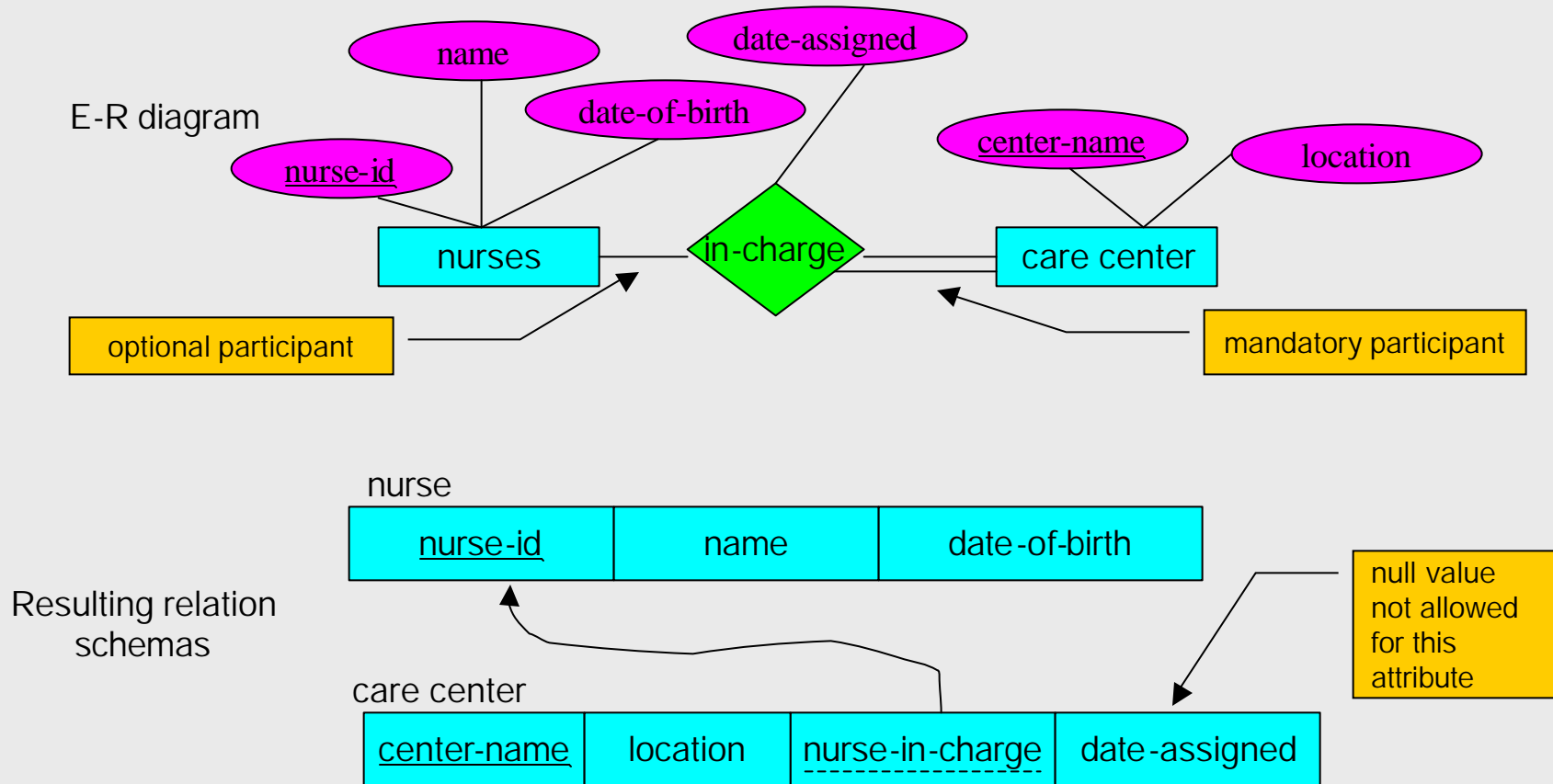
## STEP 3: Binary 1:1 Relationships

- The process of mapping such a relationship onto relation schemas requires two steps.
  1. Two relations are created, one for each of the participating entity types.
  2. 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 an optional one, while the association in the other direction is mandatory (recall participation constraints).
  - 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 will avoid 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.



# Mapping E-R Diagrams to Relational Schemas

## STEP 3: Binary 1:1 - EXAMPLE



# Mapping E-R Diagrams to Relational Schemas

## STEP 4: Mapping Associative Entities

- Mapping an associative entity to a relation schema is similar to the procedure followed for mapping a M:N relationship. Two steps are required:
  1. Create three relation schemas, one for each of the two participating entity types, and the third for the associative entity. The relation formed from the associative entity is called the *associative relation*.
  2. The actions in this step depend on whether or not the associative entity was assigned an identifier in the E-R diagram. Two cases exist:
    - An identifier was not assigned.
    - An identifier was assigned.

We'll examine each case separately.



# Mapping E-R Diagrams to Relational Schemas

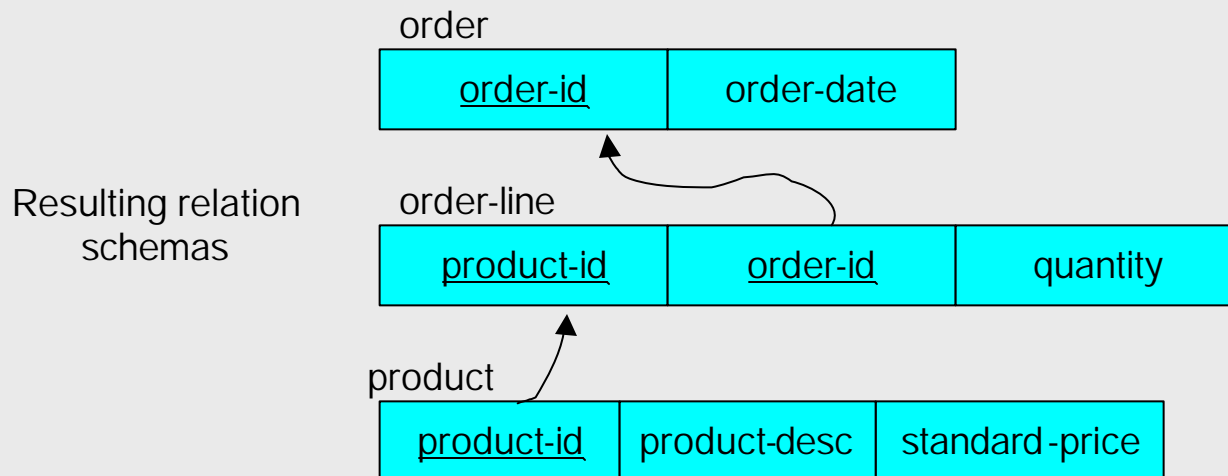
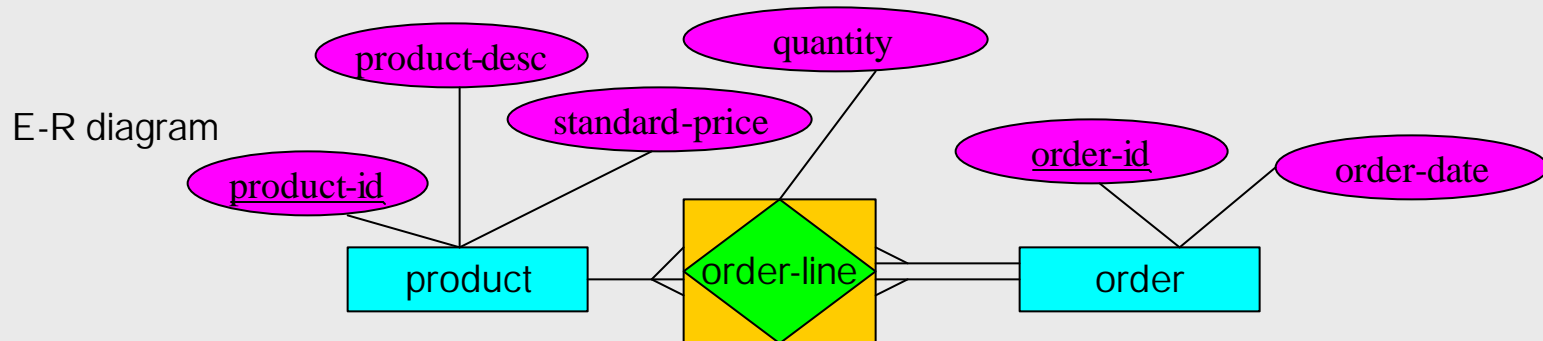
## STEP 4: Mapping Associative Entities – No Identifier Assigned

- If an identifier was not assigned, the default primary key for the associative relation consists of the two primary key attributes from the other two relations.
- These attributes are then foreign keys that reference the other two relations.
- An example of this is shown on the next page, but note the similarity of this example to that of the M:N relationship case.



# Mapping E-R Diagrams to Relational Schemas

## STEP 4: Associative Entity: EXAMPLE – no identifier assigned



# Mapping E-R Diagrams to Relational Schemas

## STEP 4: Mapping Associative Entities – Identifier Assigned

- Sometimes the data modeler will assign an identifier (called a surrogate identifier or key) to the associative entity type on the ERD.
- There are two basic reasons this may occur:
  1. The associative entity type has a natural identifier that is familiar to end users.
  2. The default identifier (consisting of the identifiers for each of the participating entity types) may not uniquely identify instances of the associative entity.
- In either case, the process for mapping the associative entity is modified as follows:



# Mapping E-R Diagrams to Relational Schemas

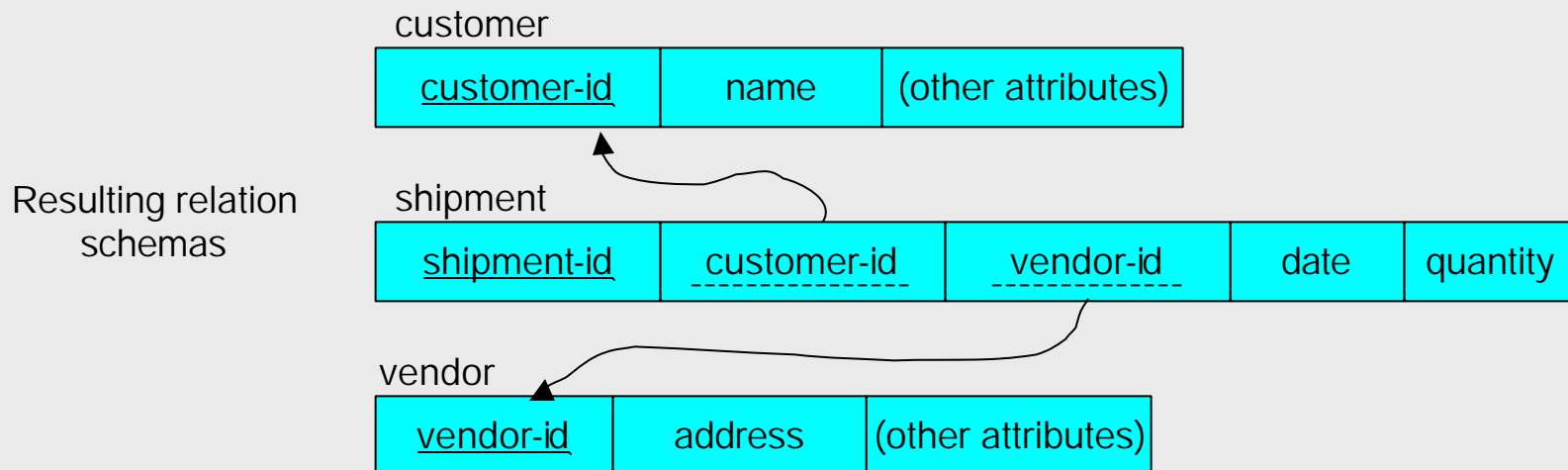
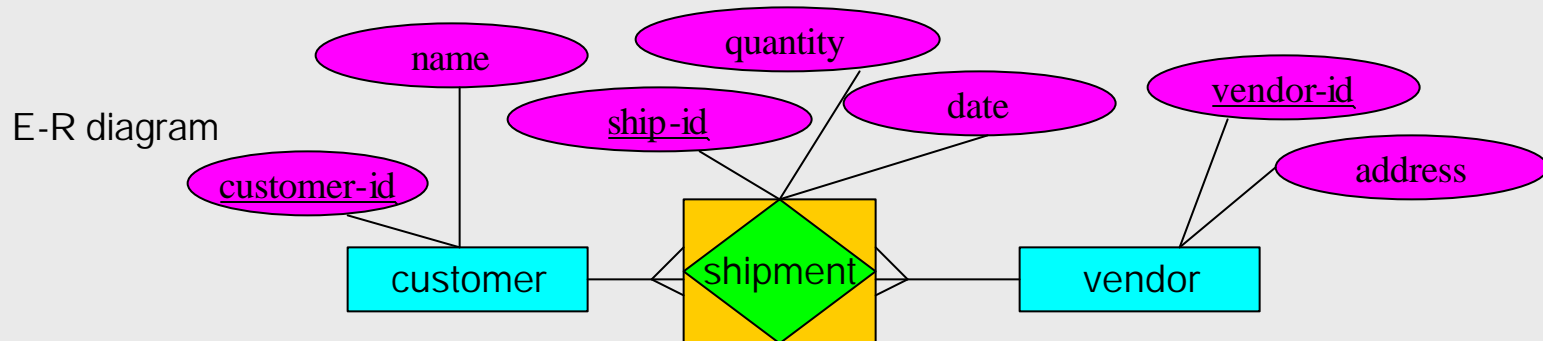
## STEP 4: Mapping Associative Entities – Identifier Assigned

- As before, a new associative relation is created to represent the associative entity.
- The primary key for the associative relation is the identifier assigned on the ERD (rather than the default key as in the previous case).
- The primary keys for the two participating entity types are then included as foreign keys in the associative relation.
- An example appears on the next page.



# Mapping E-R Diagrams to Relational Schemas

## STEP 4: Associative Entity: EXAMPLE – identifier assigned



# Mapping E-R Diagrams to Relational Schemas

## STEP 5: Mapping Unary (recursive) Relationships

- Recall that a recursive relationship is defined as a relationship between instances of a single entity type.
- The two most important cases of unary relationships are the 1:M and M:M cardinalities.
- We'll again look at these two cases separately as they are handled somewhat differently.



# Mapping E-R Diagrams to Relational Schemas

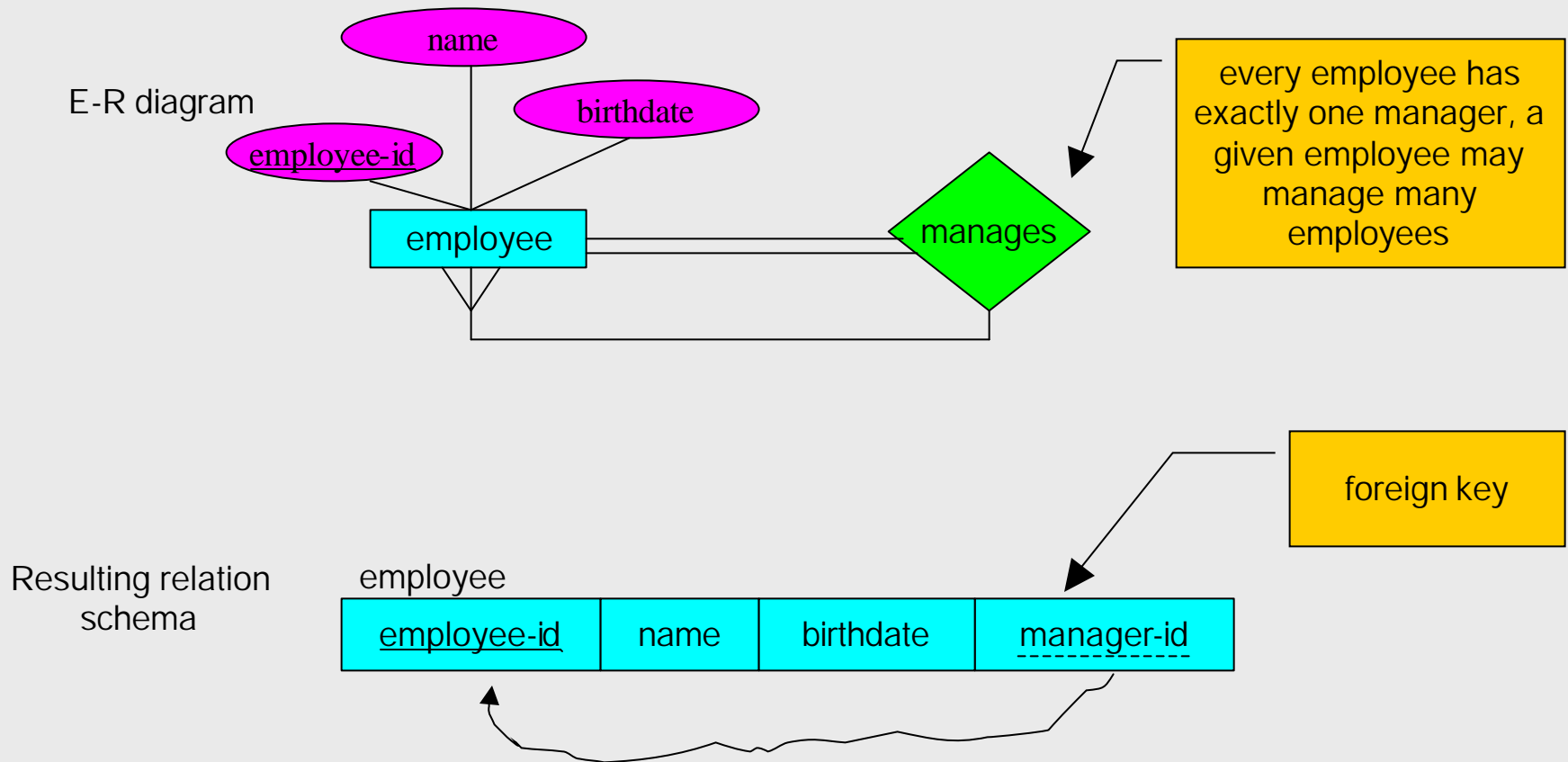
## STEP 5: Mapping Recursive Relationships – 1:M Case

- The entity type in the unary relationship is mapped onto a relation schema using the procedure described in Step 1.
- Next, a foreign key attribute is added *within* the same relation that references the primary key values (this foreign key must have the same domain as the primary key).
- A **recursive foreign key** is a foreign key in a relation that references the primary key values of that same relation.



# Mapping E-R Diagrams to Relational Schemas

## STEP 5: Mapping Recursive Relationships: EXAMPLE – 1:M



# Mapping E-R Diagrams to Relational Schemas

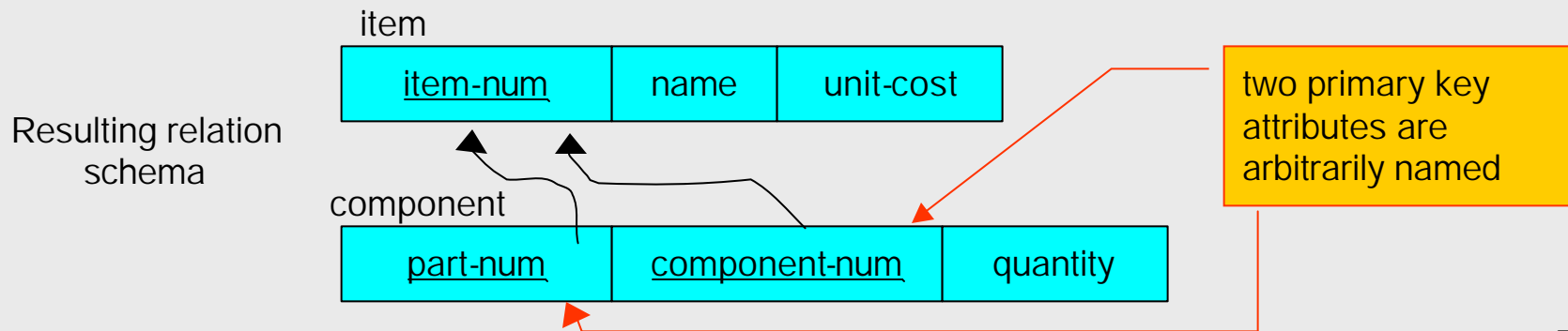
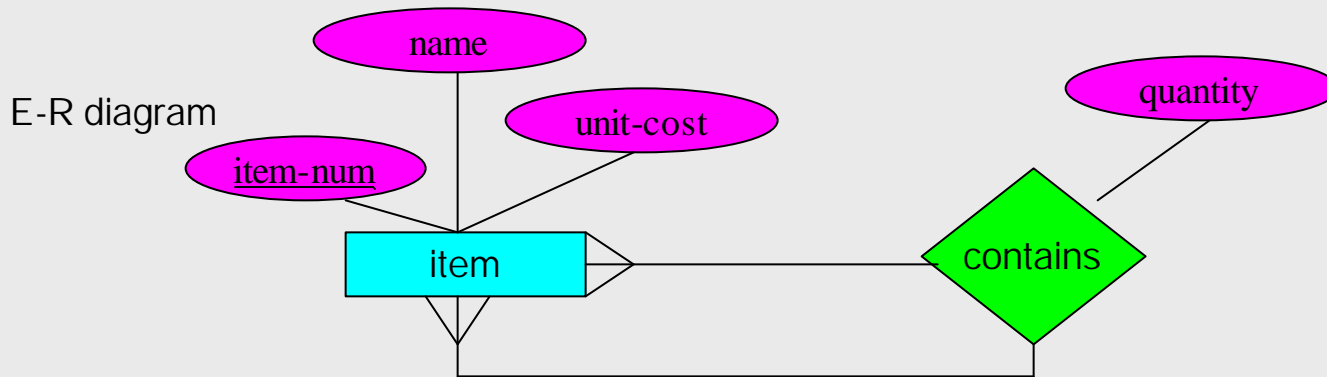
## STEP 5: Mapping Recursive Relationships – M:N Case

- With this type of recursive relationship, two relation schemas are created: one to represent the entity type and the other an associative relation to represent the M:N relationship itself.
- The primary key of the associative relation consists of two attributes. These attributes (which do not necessarily have the same name) both take their values from the primary keys of the other relation.
- Any non-key attribute of the relationship is included in the associative relation.
- The example on the next page illustrates such a case representing a bill of materials relationship among items that are assembled from other items or components.



# Mapping E-R Diagrams to Relational Schemas

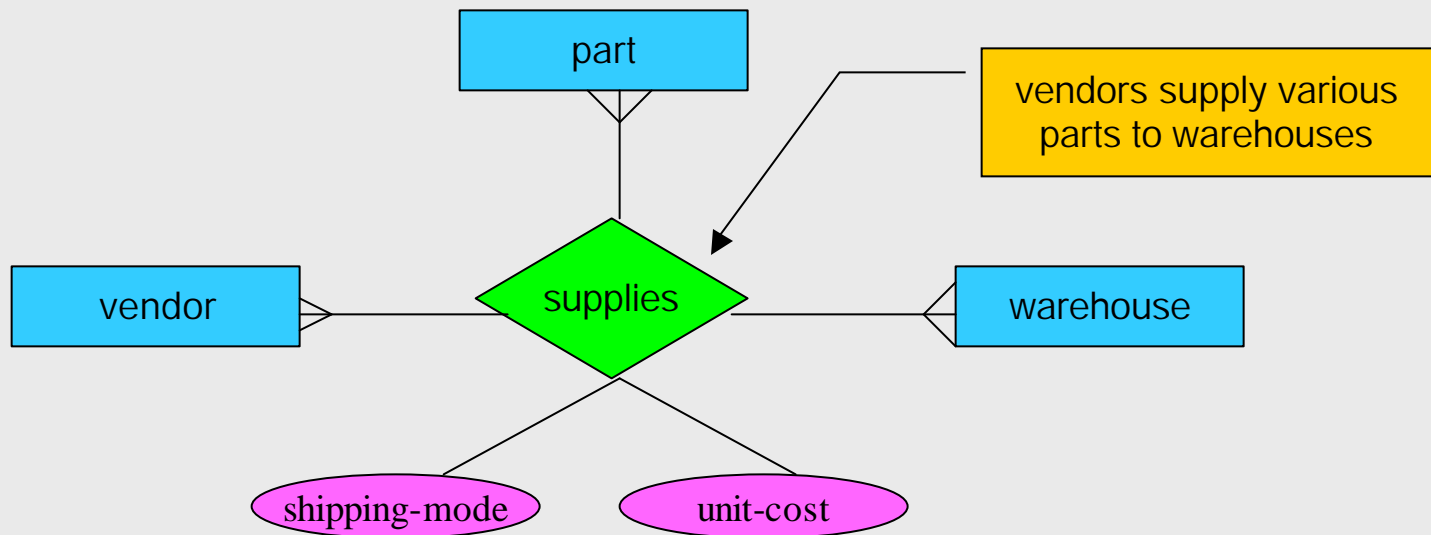
## STEP 5: Mapping Recursive Relationships: EXAMPLE – M:N



# Mapping E-R Diagrams to Relational Schemas

## STEP 6: Mapping Ternary (and n-ary) Relationships

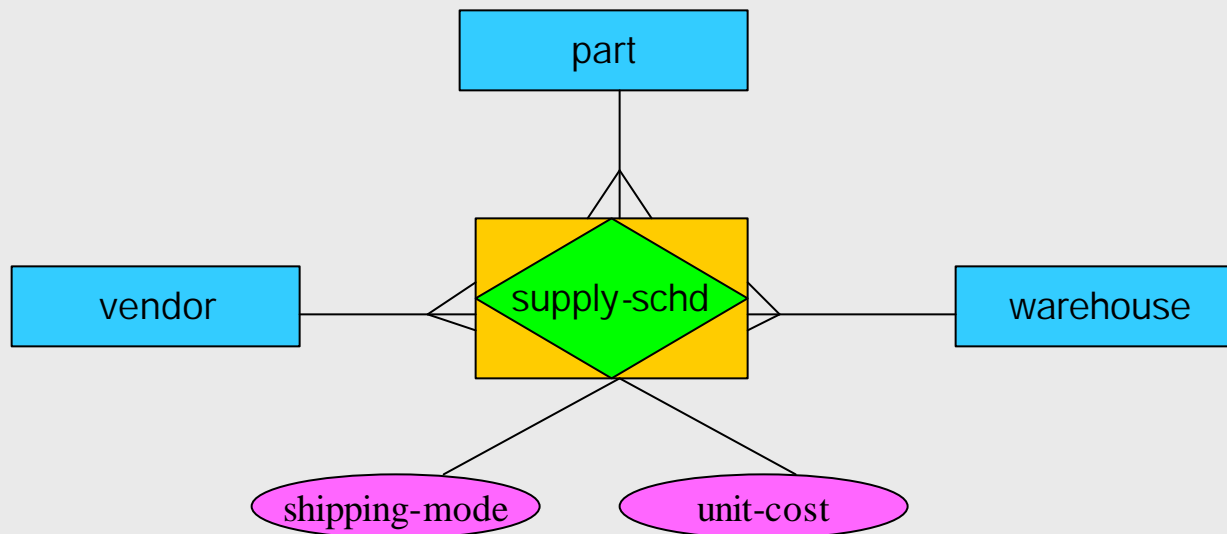
- Recall that a ternary relationship is defined as a relationship among three entity types as shown below.



# Mapping E-R Diagrams to Relational Schemas

## STEP 6: Mapping Ternary (and n-ary) Relationships

- It is strongly recommended that all ternary (or higher) relationships be converted associative entities before proceeding further. An example is shown below which converts the ERD from the previous page into one with an associative entity:



# Mapping E-R Diagrams to Relational Schemas

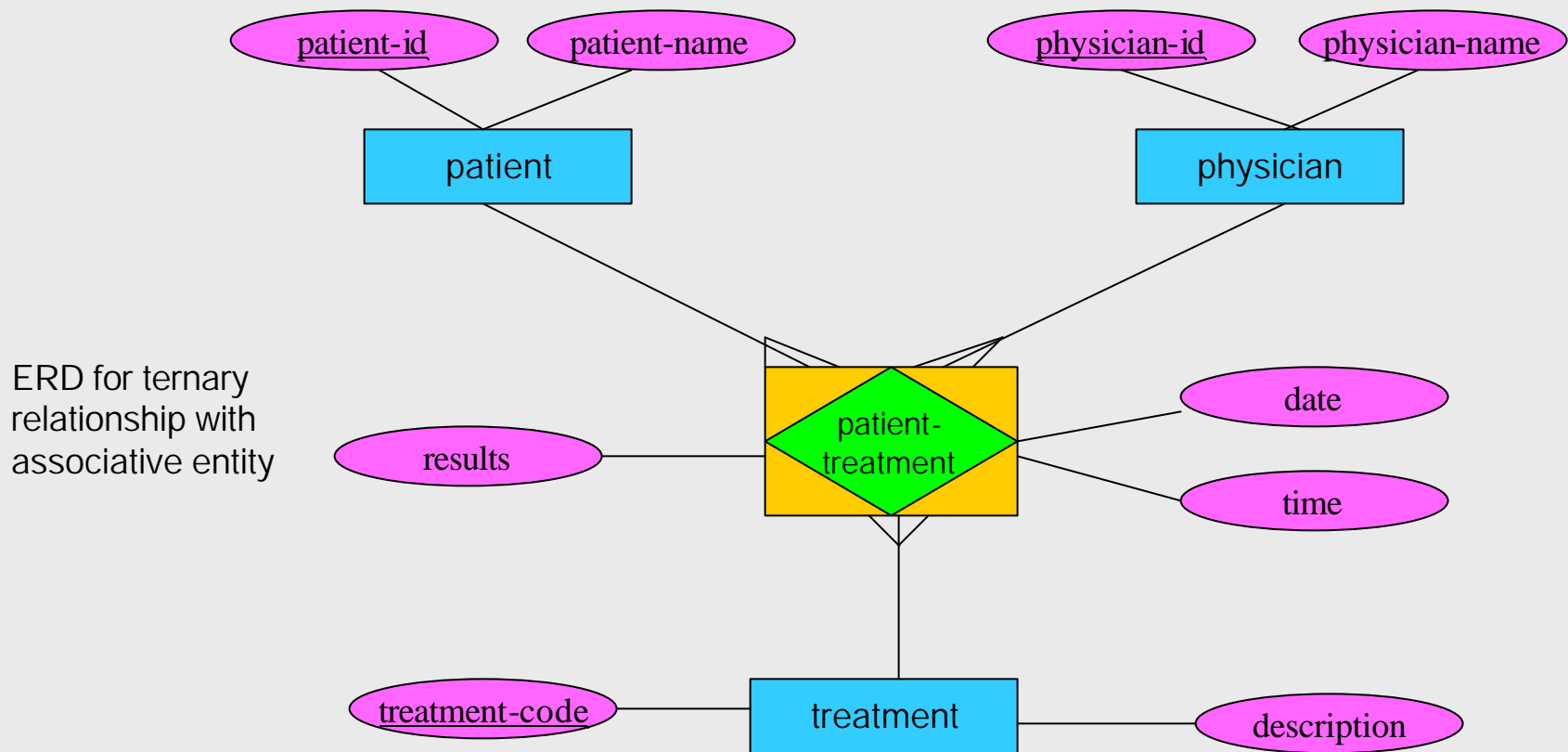
## STEP 6: Mapping Ternary (and n-ary) Relationships

- To map an associative entity type that links three regular entity types, create a new associative relation.
- The default primary key of this relation consists of the three primary key attributes for the participating entity types (in some cases additional attributes are required to form a unique primary key). These attributes then act in the role of foreign keys that reference the individual primary keys of the participating entity types.
- Any attributes of the associative entity type become attributes in the new associative relation.



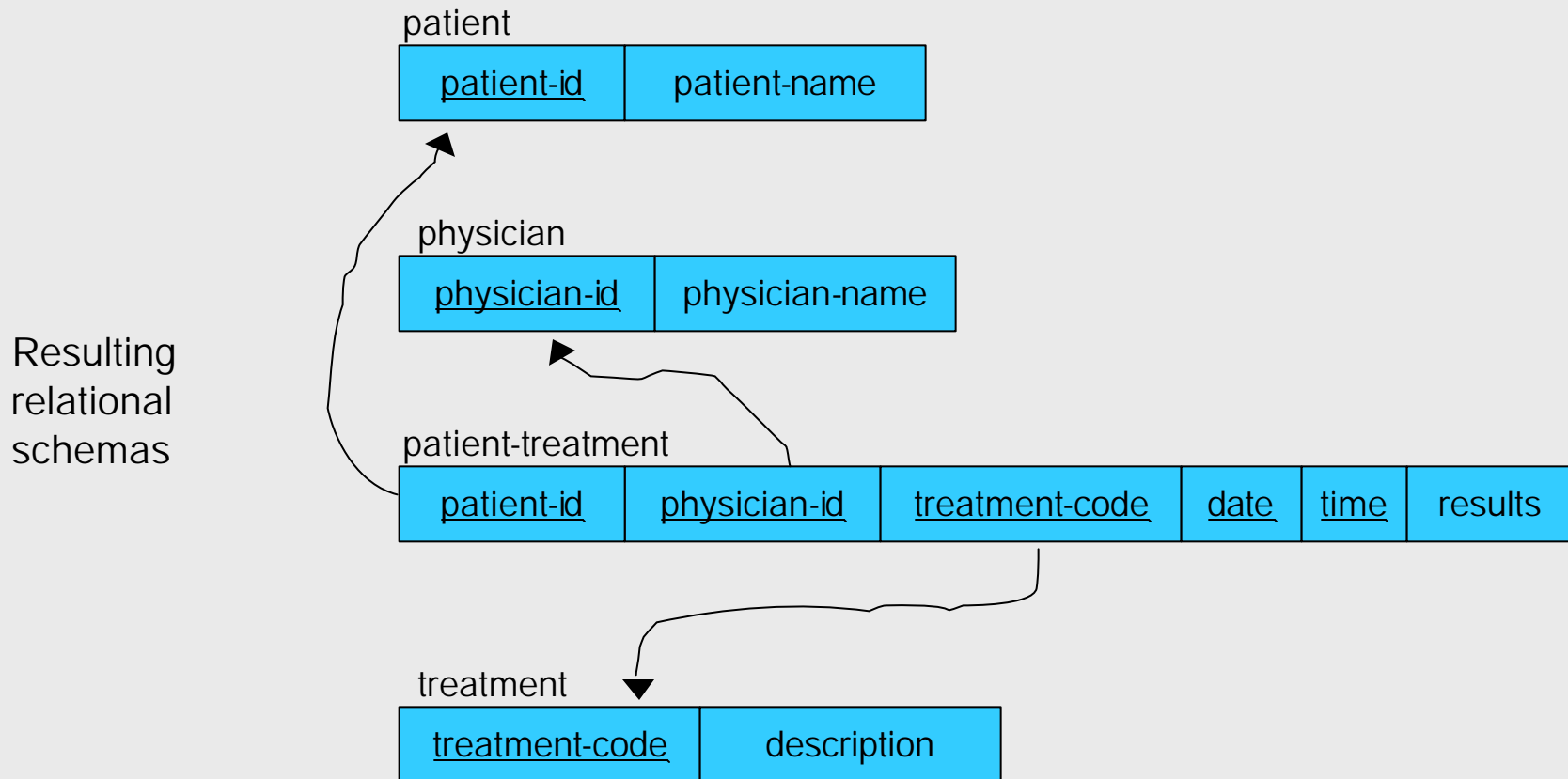
# Mapping E-R Diagrams to Relational Schemas

## STEP 6: Mapping Ternary (and n-ary) Relationships - EXAMPLE



# Mapping E-R Diagrams to Relational Schemas

## STEP 6: Mapping Ternary (and n-ary) Relationships - EXAMPLE



# Mapping E-R Diagrams to Relational Schemas

## STEP 7: Mapping Supertype/Subtype Relationships

- The relational data model does not directly support supertype/subtype relationships. Fortunately, there are various strategies that database designers can use to represent these relationships with the relational data model.
- We'll examine one of the more common techniques that has been used for modeling supertype/subtype relationships.



# Mapping E-R Diagrams to Relational Schemas (cont.)

## STEP 7: Mapping Supertype/Subtype Relationships

1. Create a separate relation schema for the supertype and for each of its subtypes.
2. Assign to the relation schema created for the supertype the attributes that are common to all members of the supertype, including the primary keys.
3. Assign to the relation schema for each subtype the primary key of the supertype, and only those attributes that are unique to that subtype.
4. Assign one (or more) attributes of the supertype to function as the subtype discriminator.



# Mapping E-R Diagrams to Relational Schemas (cont.)

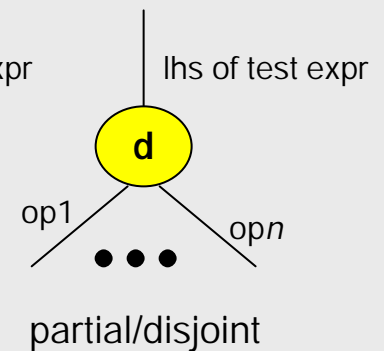
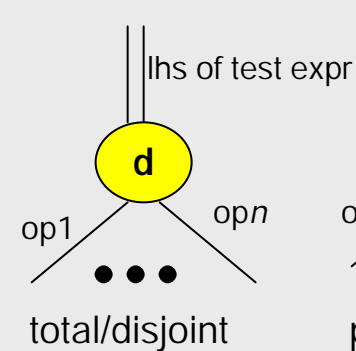
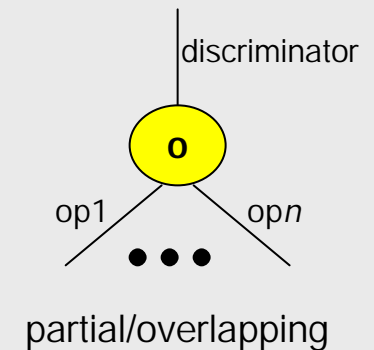
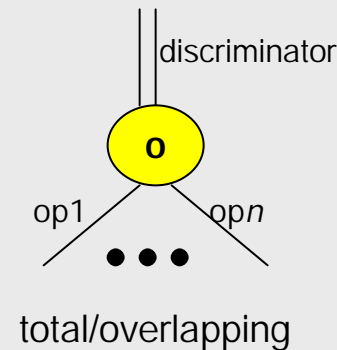
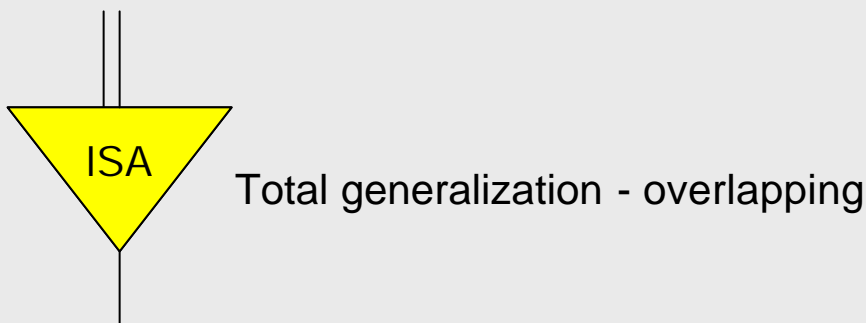
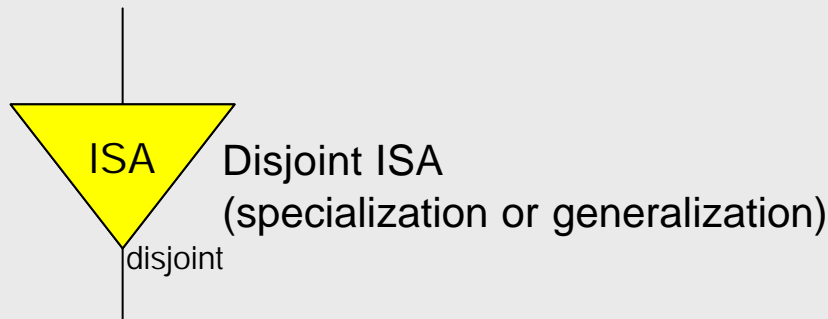
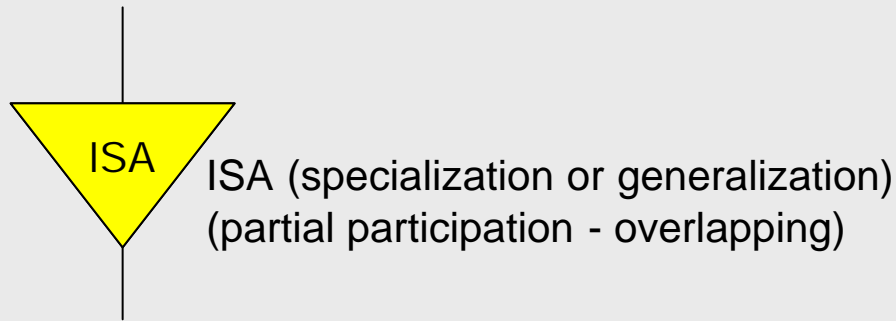
## STEP 7: Mapping Supertype/Subtype Relationships

### Defining subtype discriminators

- Given a supertype/subtype relationship, consider the problem of inserting a new instance of the supertype. Into which of the subtypes (if any) should this instance be inserted?
- A common approach uses a subtype discriminator. A **subtype discriminator** is an attribute of the supertype whose values determine the target subtype or subtypes (used when subclass membership is predicate based).
- Two cases arise: disjoint subtypes and overlapping subtypes.

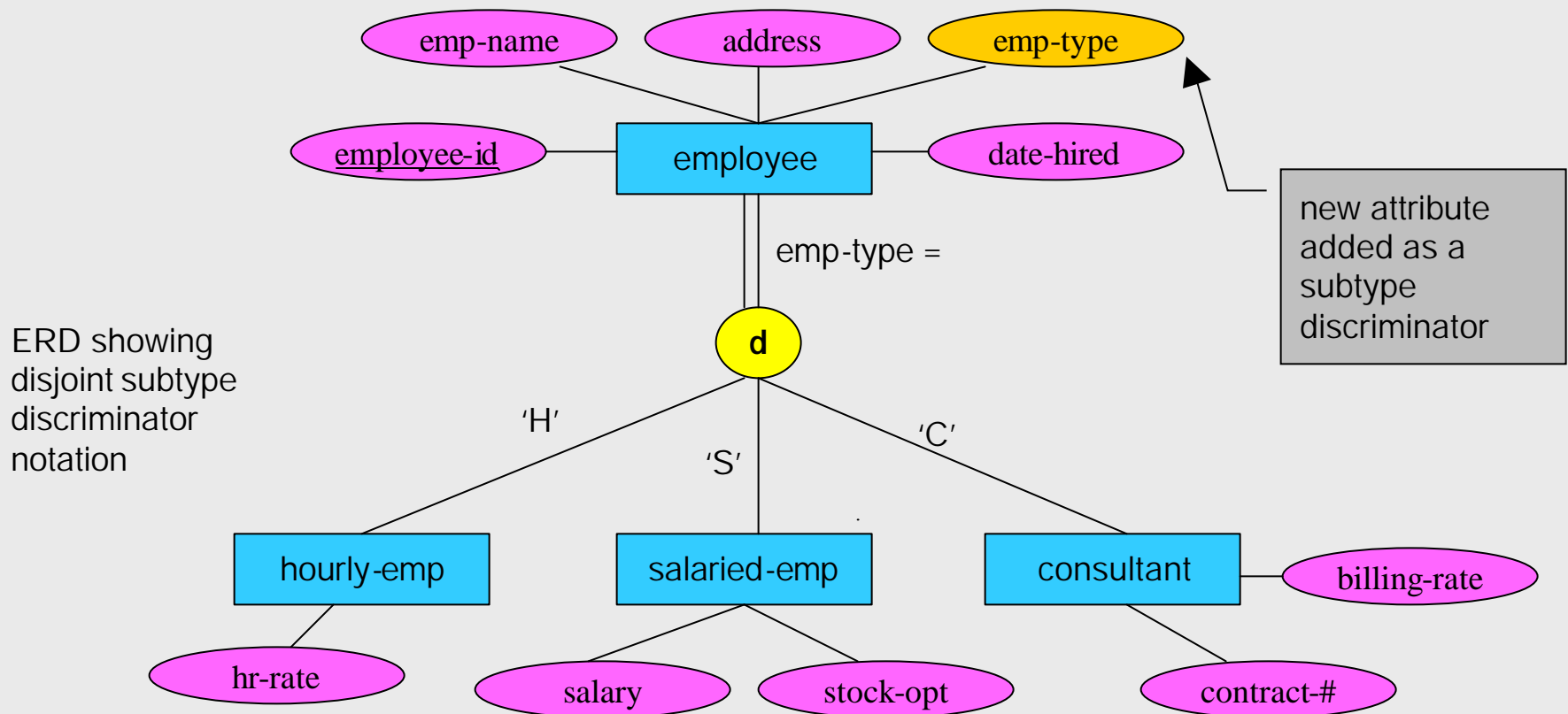


# Alternative Notation for Specialization



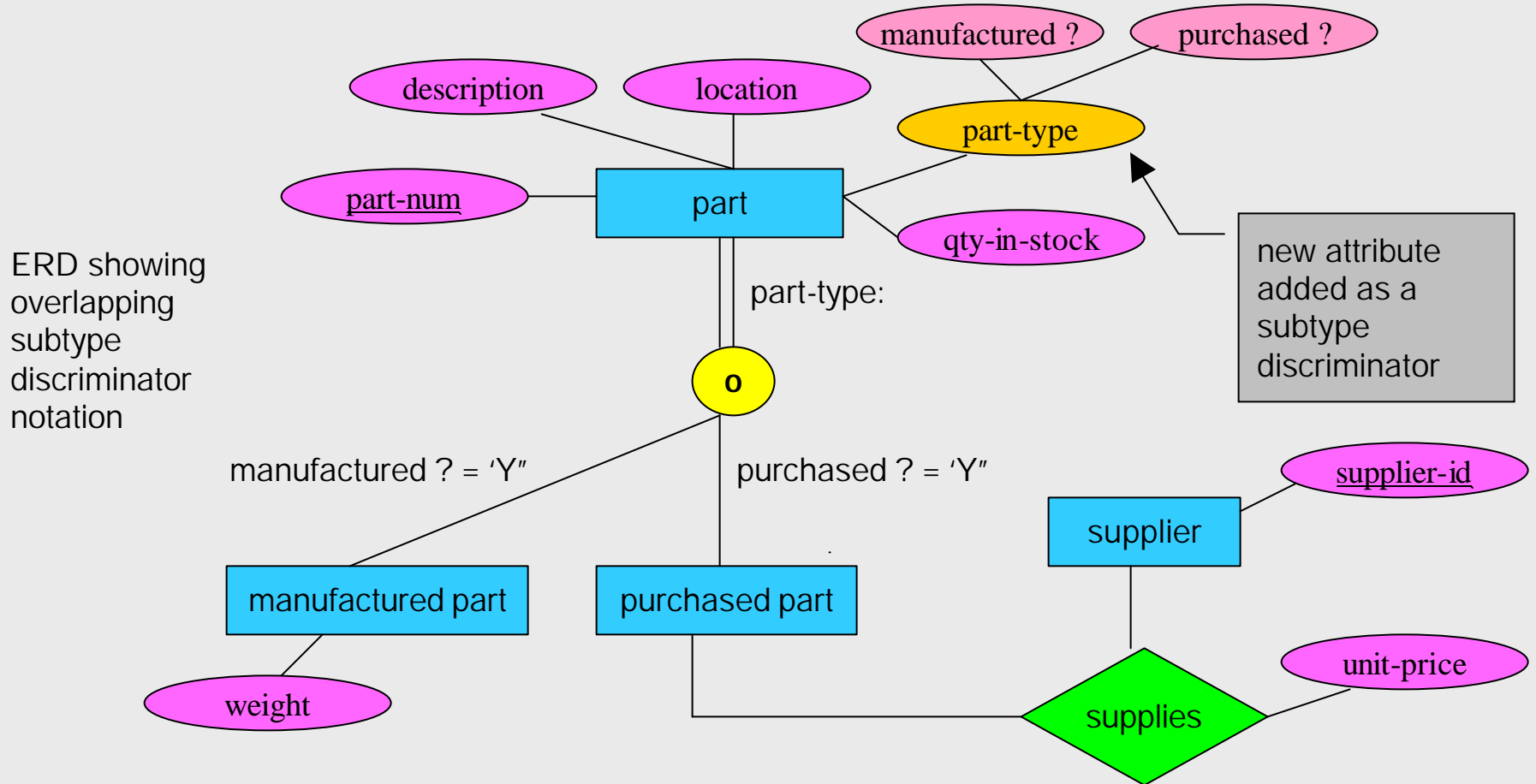
# Mapping E-R Diagrams to Relational Schemas

## EXAMPLE – Subtype discriminators in ERDs: disjoint subtypes



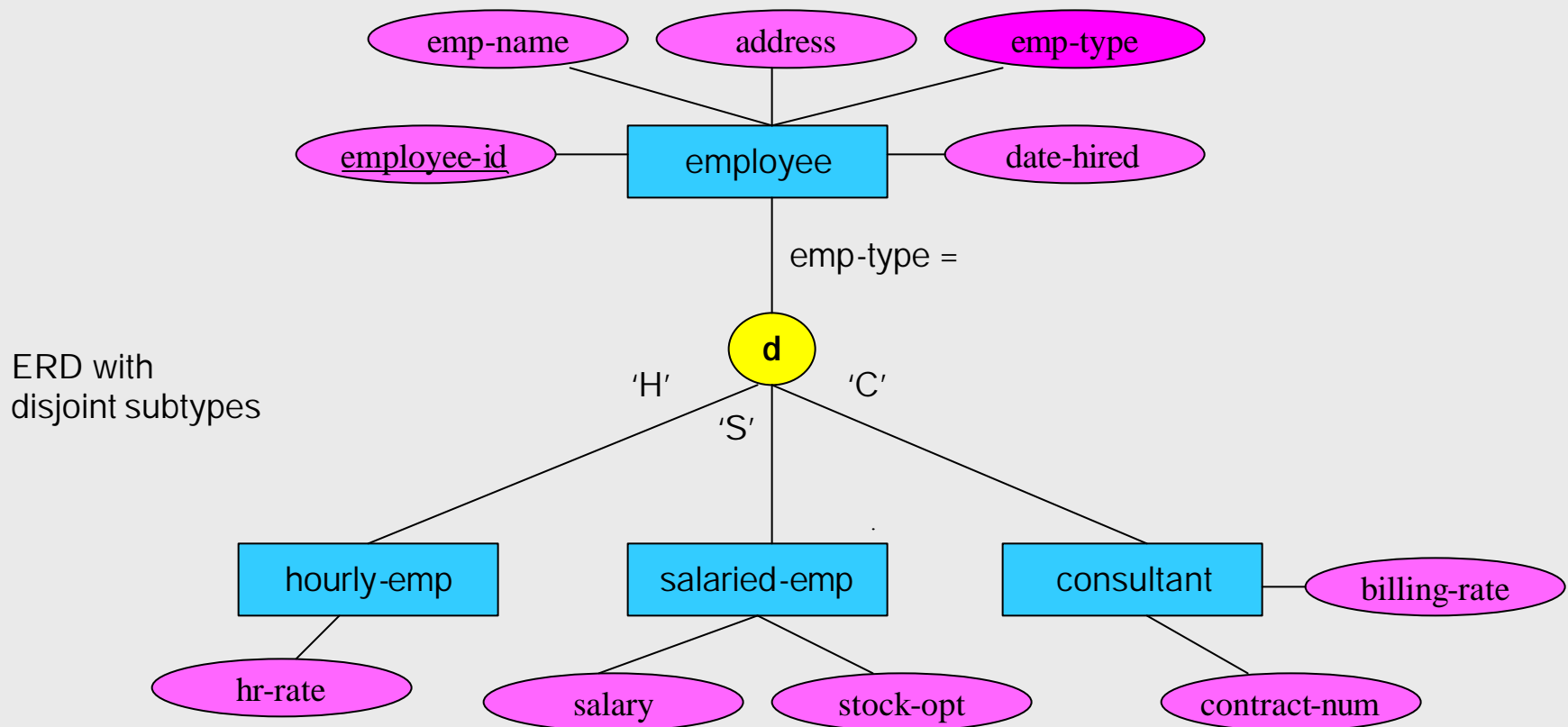
# Mapping E-R Diagrams to Relational Schemas

## EXAMPLE – Subtype discriminators in ERDs: overlapping subtypes



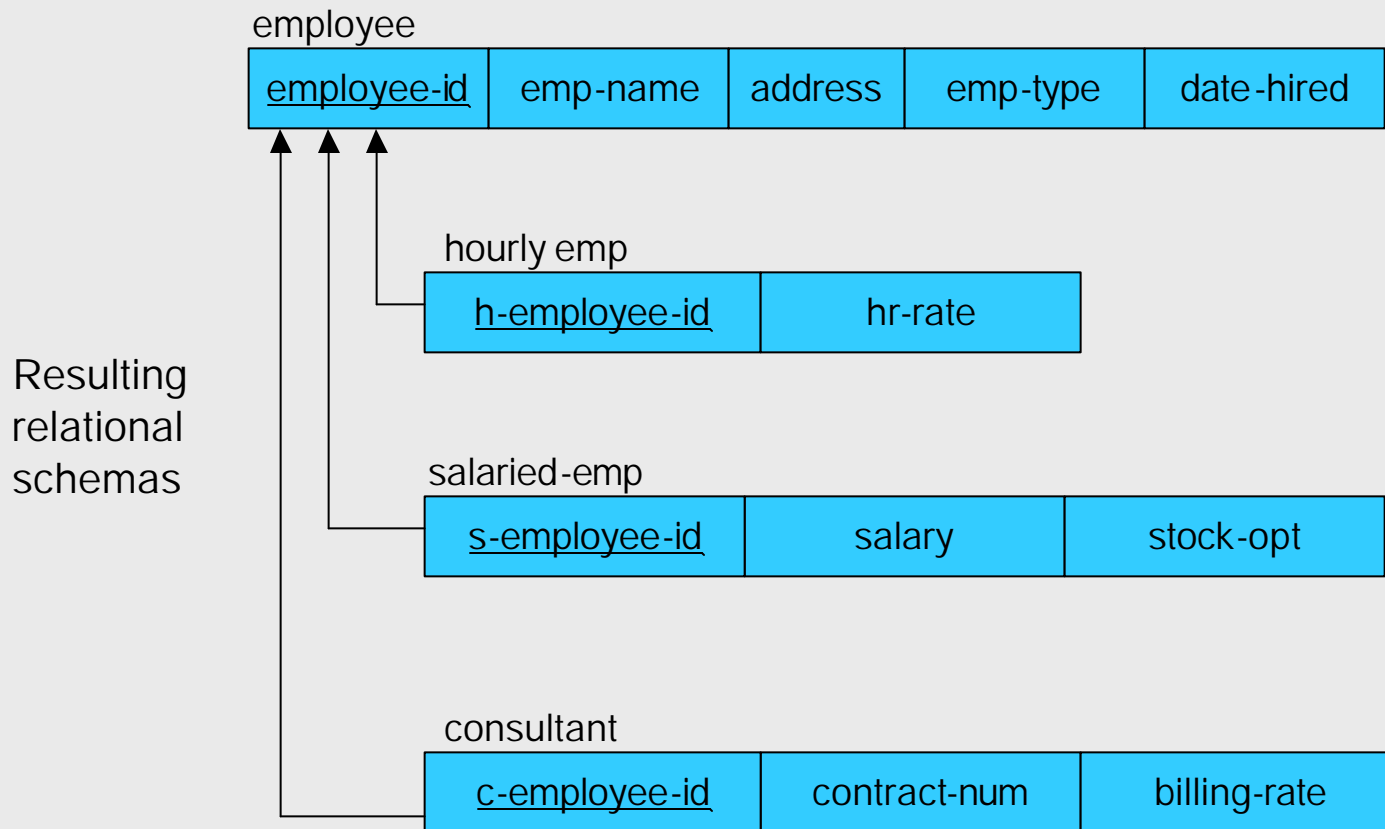
# Mapping E-R Diagrams to Relational Schemas

## STEP 7 – Mapping Supertype/Subtype Relationships - EXAMPLE



# Mapping E-R Diagrams to Relational Schemas

## STEP 7: Mapping Supertype/Subtype Relationships - EXAMPLE



# Mapping E-R Diagrams to Relational Schemas

## PRACTICE

