COP 4710: Database Systems
Spring 2006
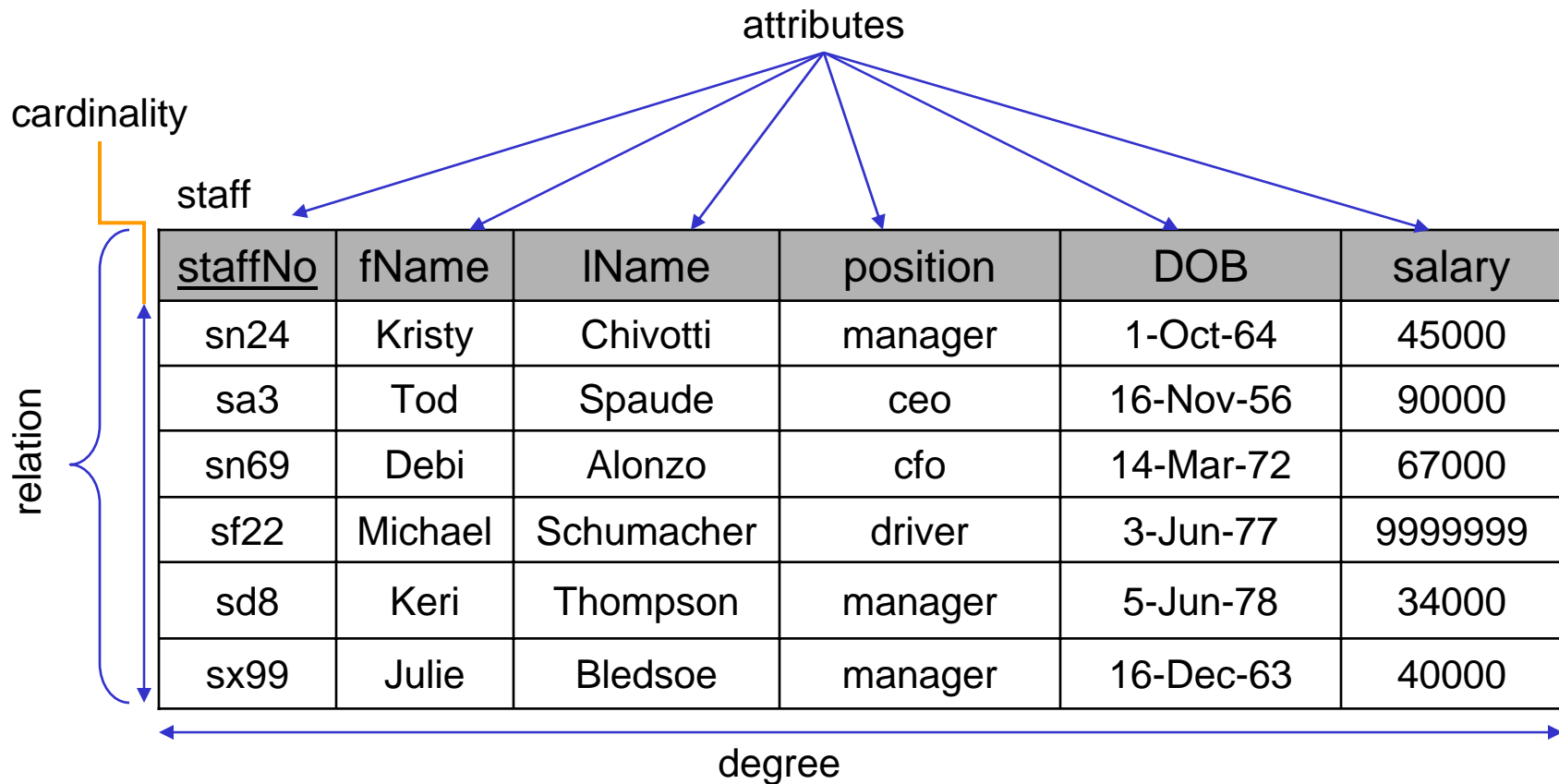
Chapter 3 – The Relational Data Model – Part 1

Instructor :   Mark Llewellyn
                    markl@cs.ucf.edu
                    CSB 242, 823-2790
                    http://www.cs.ucf.edu/courses/cop4710/spr2006

School of Electrical Engineering and Computer Science
University of Central Florida

An Example Relation

attributes

cardinality

staff

staffNo	fName	lName	position	DOB	salary
sn24	Kristy	Chivotti	manager	1-Oct-64	45000
sa3	Tod	Spaude	ceo	16-Nov-56	90000
sn69	Debi	Alonzo	cfo	14-Mar-72	67000
sf22	Michael	Schumacher	driver	3-Jun-77	9999999
sd8	Keri	Thompson	manager	5-Jun-78	34000
sx99	Julie	Bledsoe	manager	16-Dec-63	40000

relation

degree

Example Domain Definitions

Attribute	Domain Name	Meaning	Domain Definition
staffNo	staffnumbers	set of all possible staff numbers	character: size 4, must begin with letter s.
fName, lName	name	set of all possible person names	character: size 20
DOB	date	date person was born	date: range from 1-Jan-20, format: dd-mmm-yy
salary	salaries	possible values of staff salaries	monetary: 7 digits, range 10,000-9,999,999
position	alljobs	set of all possible positions	select one from set: {ceo, cfo, coo,manager, asst. manager, driver, secretary}

Alternate Terminology for Relational Model

Formal Term	Alternative 1	Alternative 2
relation	table	file
tuple	row	record
attribute	column	field

What is a Relation

- To understand the true meaning of the term relation, we need to review some basic math concepts:

- Given two sets D_1 and D_2 where

$$D_1 = \{2, 4\} \text{ and } D_2 = \{1, 3, 5\}$$

- The Cartesian product of these two sets, written $D_1 \times D_2$, is the set of all ordered pairs such that the first element is a member of D_1 and the second element is a member of D_2.

  - $D_1 \times D_2 = \{(2, 1), (2, 3), (2, 5), (4, 1), (4, 3), (4, 5)\}$

- Any subset of this Cartesian product is a relation.

  - Thus, we could produce relation R such that: $R = \{(2, 3), (4, 3)\}$

- We can specify some condition which will select elements from $D_1 \times D_2$ to be included in R, such as:

  - $R = \{(x, y) \mid x \in D_1, y \in D_2, \text{ and } y = 3\}$

What is a Relation (cont.)

- Given three sets D_1, D_2, and D_3 where

$$D_1 = \{2, 4\}, D_2 = \{1, 3\}, \text{ and } D_3 = \{3, 6\}$$

- The Cartesian product of three sets, written $D_1 \times D_2 \times D_3$, is the set of all ordered triples such that the first element is a member of D_1, the second element is a member of D_2, and the third element is a member of D_3.

  - $D_1 \times D_2 \times D_3 = \{(2, 1, 3), (2, 1, 6), (2, 3, 3), (2, 3, 6)$

    $(4, 1, 3), (4, 1, 6), (4, 3, 3), (4, 3, 6)\}$

- Any subset of this Cartesian product is a relation.

- In general, if D_1, D_2, ... D_n are n sets. Their Cartesian product is defined as: $D_1 \times D_2 \times \cdots \times D_n = \left\{ (d_1, d_2, \cdots, d_n) \,\middle|\, d_1 \in D_1, d_2 \in D_2, \cdots, d_n \in D_n \right\}$

  and generally written as:

$$\overset{n}{\underset{i=1}{X}} D_i$$

What is a Relation (cont.)

- A relational schema is a named relation defined by a set of attribute and domain name pairs.

  - $R_i = \{A_1{:}d_1, A_2{:}d_2, ..., A_n{:}d_n \mid d_1 \in D_1, d_2 \in D_2, ..., d_n \in D_n\}$

- A relational database schema is a set of relation schemas, each with a distinct name.

  - $R = \{R_1, R_2, ..., R_n\}$

What is a Relation (cont.)

A relation has the following properties:

1. The relation has a name that is distinct from all other relation names in the relational schema.

2. Each cell (attribute) contains exactly one atomic value.

3. Each attribute has a distinct name.

4. The values of an attribute are all from the same domain.

5. Each tuple is distinct; there are no duplicate tuples.

6. The order of the attributes has no significance.

7. The order of the tuples has not significance, theoretically. (However, in practice, the order may affect the efficiency of accessing tuples. Much more on this later.)

Relation Schemas vs. Relation Instances

- There is an important distinction to be made between a relation schema and a relation instance.

- The schema is the name and attributes for the relation and is relatively immutable.

- An instance is a set of tuples for that relation, and the instance may change frequently. Indeed most updates and certainly every insert and deletion will change the instance.

  - A snapshot database models the current "state" of the real world which is captured in the database. At any given moment in time it is modeling the current "instance" of the real world. If the real world state changes, so too must the database to maintain the representation of the current real world instance.

Equivalent Relations

A	B	C
1	2	3
3	2	1
4	4	1
2	1	3

a relation instance

B	C	A
2	3	1
2	1	3
4	1	4
1	3	2

a relation instance

equivalent relation instances

A	B	C
4	4	1
3	2	1
1	2	3
2	1	3

a relation instance

A	B	C
4	4	1
3	2	1
1	2	4
2	1	3

a relation instance

this relation instance is not equivalent to any of the other three

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 chose 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.
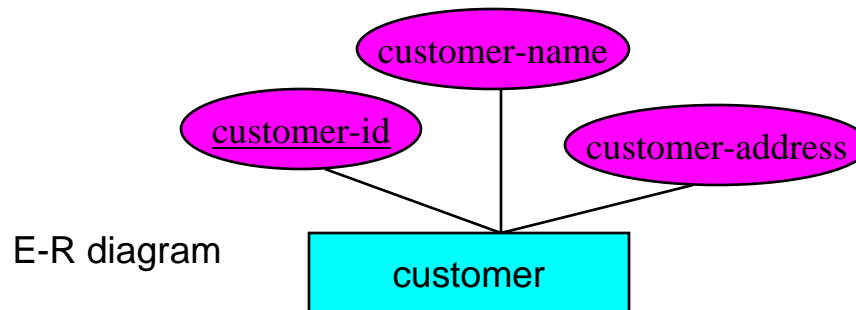
Mapping E-R Diagrams to Relational Schemas

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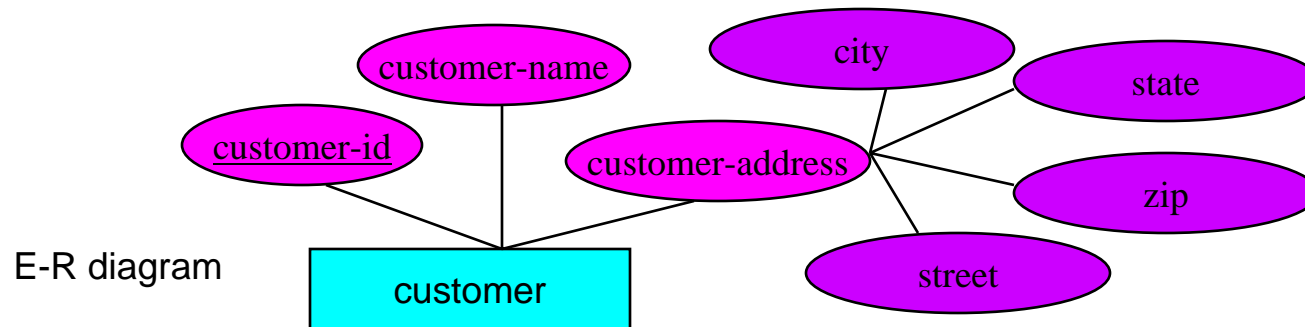
Mapping E-R Diagrams to Relational Schemas (cont.)

STEP 1: Map Regular (Strong) Entities - EXAMPLE

E-R diagram

customer-name

customer-id

customer-address

customer

customer

customer-id	customer-name	customer-address
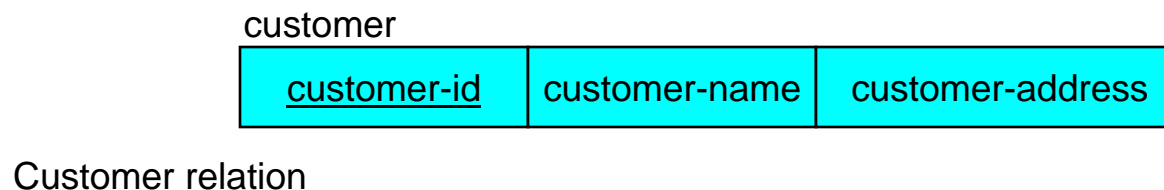
Customer relation

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.

E-R diagram

customer

customer					
customer-id	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:

E-R diagram

employee-name

employee-id

employee-address

employee

skill

Resulting relation schemas

employee

employee-id	employee-name	employee-address

employee-skill
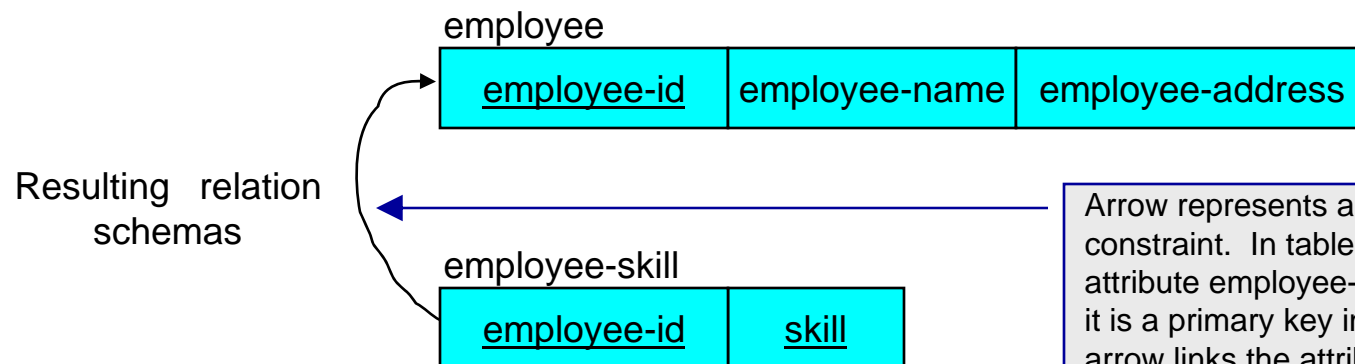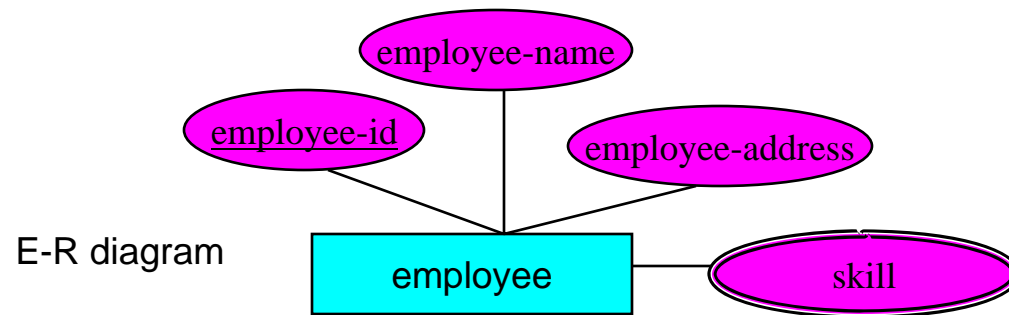
employee-id	skill

Arrow represents a referential integrity constraint. In table employee-skill the attribute employee-id is a foreign key, i.e., it is a primary key in another table. The arrow links the attribute to the table where it is the primary key.

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.

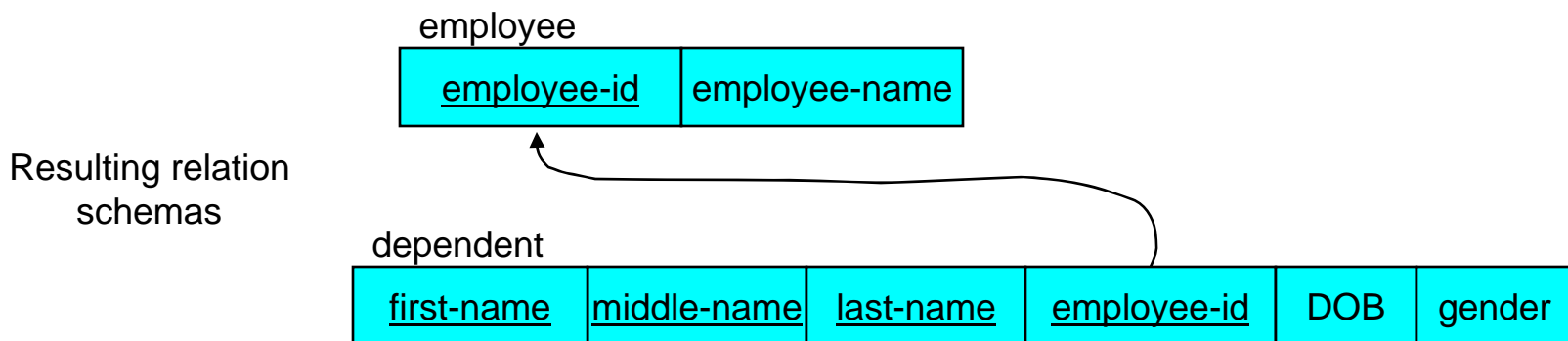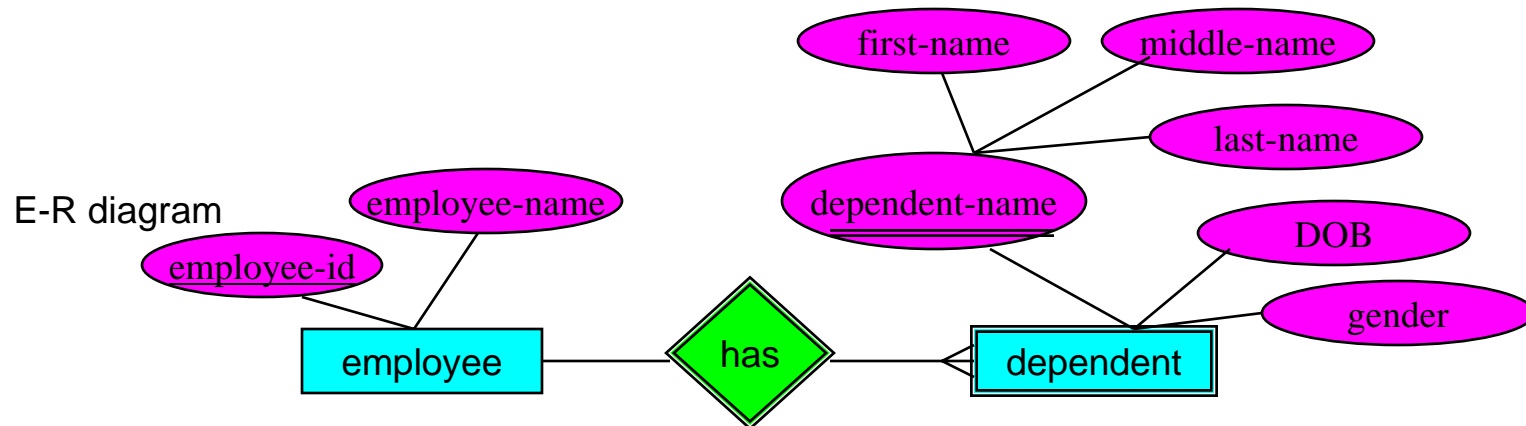# Mapping E-R Diagrams to Relational Schemas

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

E-R diagram

first-name   middle-name

last-name

employee-name   dependent-name

employee-id   DOB

gender

employee   has   dependent

employee

employee-id	employee-name

Resulting relation
schemas

dependent

first-name	middle-name	last-name	employee-id	DOB	gender

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.
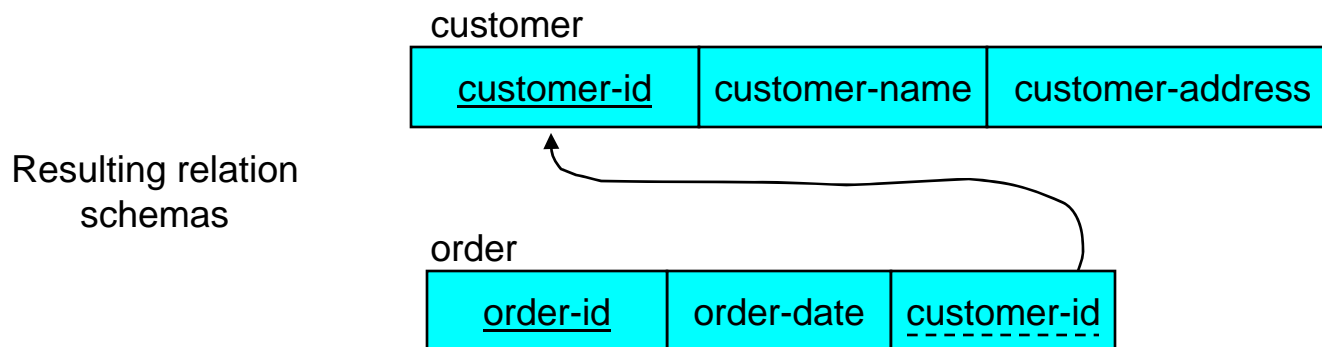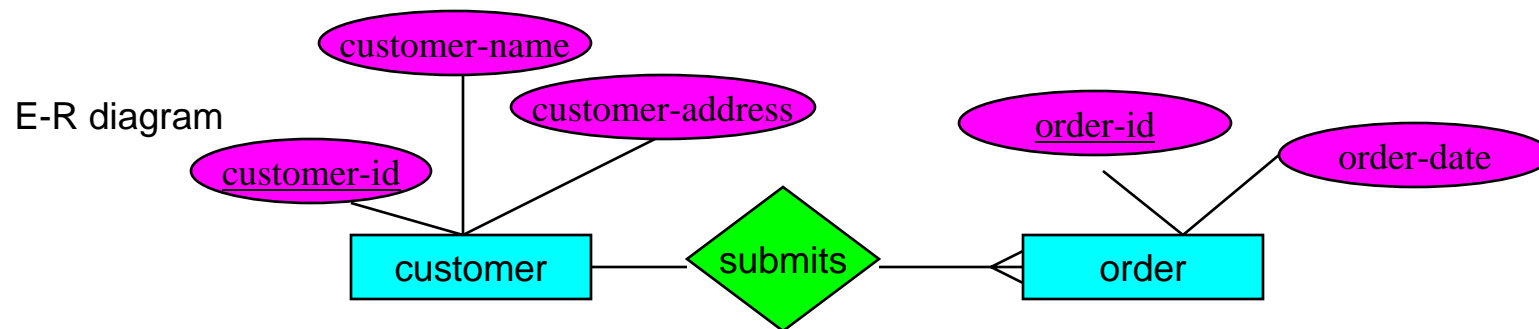
Mapping E-R Diagrams to Relational Schemas

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

E-R diagram

Resulting relation schemas

customer

customer-id	customer-name	customer-address

order

| order-id	order-date	customer-id

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
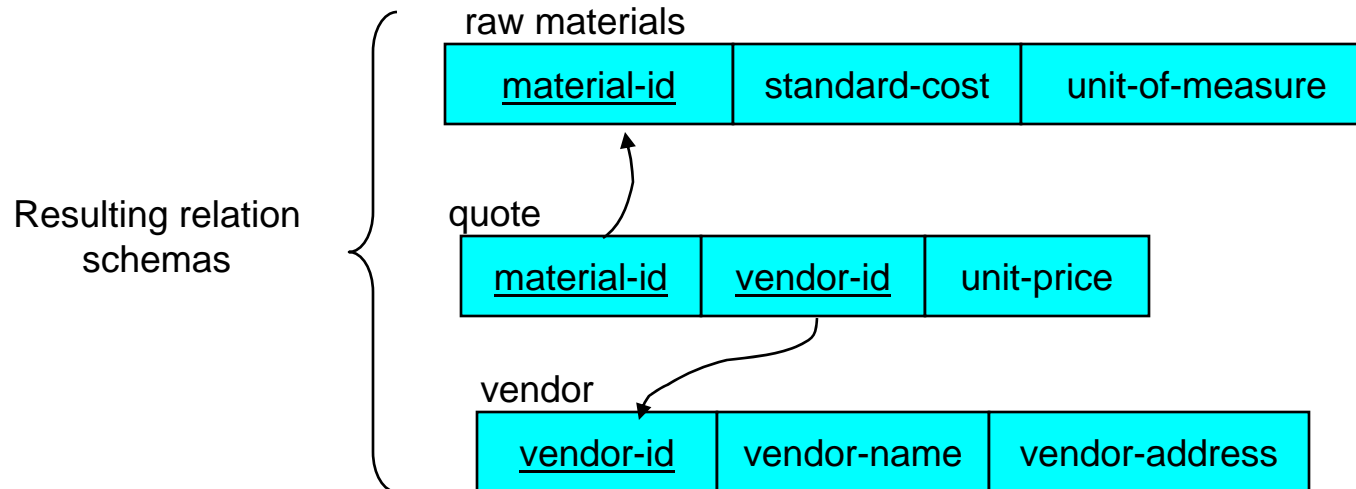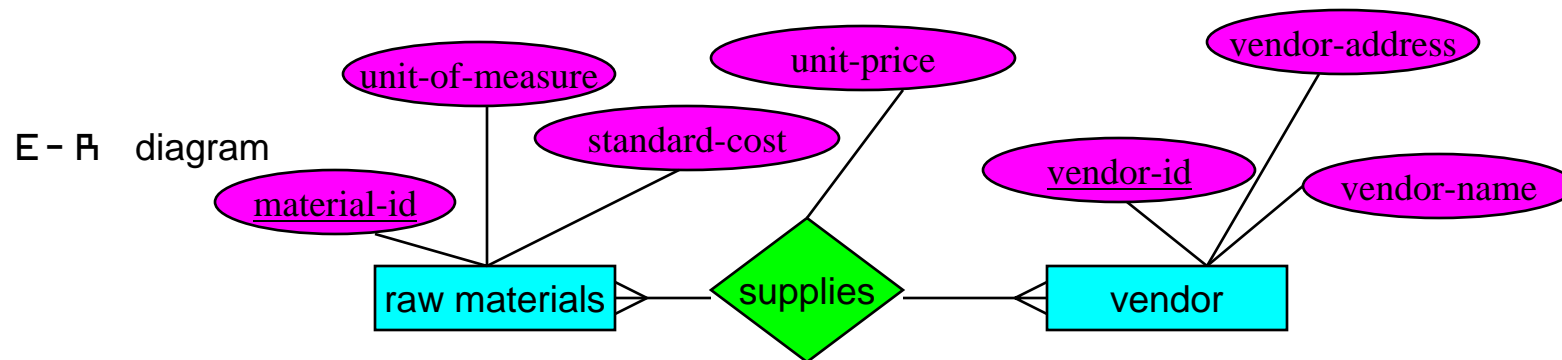
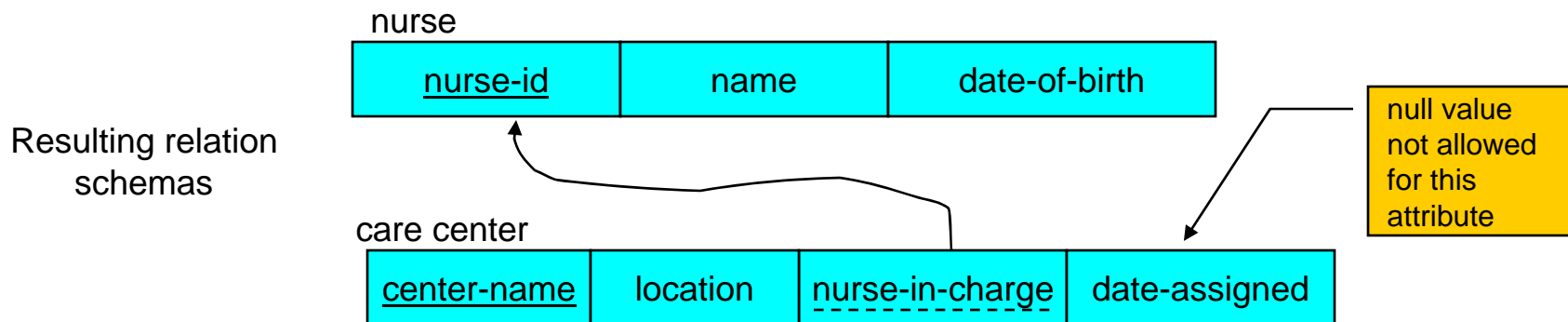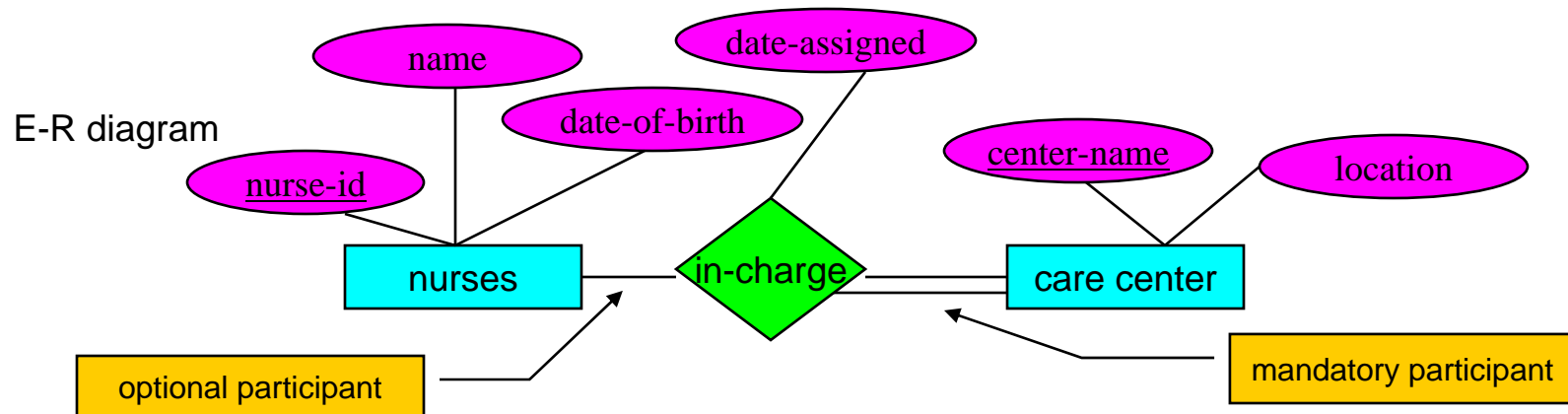# Mapping E-R Diagrams to Relational Schemas

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

E-R diagram

name
date-assigned
date-of-birth
nurse-id
center-name
location

nurses — in-charge — care center

optional participant

mandatory participant

Resulting relation schemas

nurse

nurse-id	name	date-of-birth

care center

center-name	location	nurse-in-charge	date-assigned

null value not allowed for this attribute

An Aside On Associative Entities

- The presence of one or more attributes on a relationship suggests to the designer that the relationship should perhaps instead be represented as an entity type.

- As associative entity is an entity type that associates the instances of one or more entity types and contains attributes that are peculiar to the relationship between those entity instances.

- For example, (see ER diagram on next page) consider an organization that wishes to record the date (month and year) when an employee completes each certification course. The date completed cannot be associated with either entity sets EMPLOYEE or COURSE, because Date_Completed is a property of the relationship Completes.
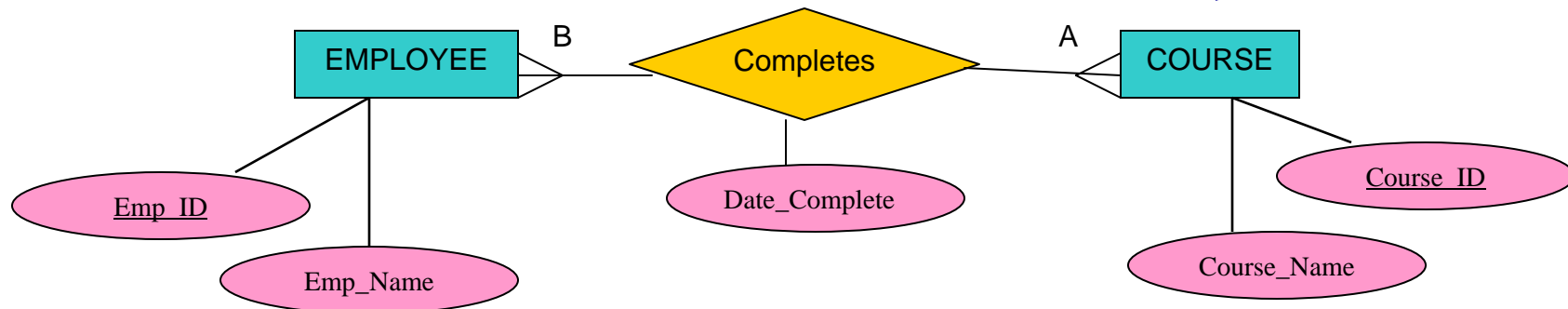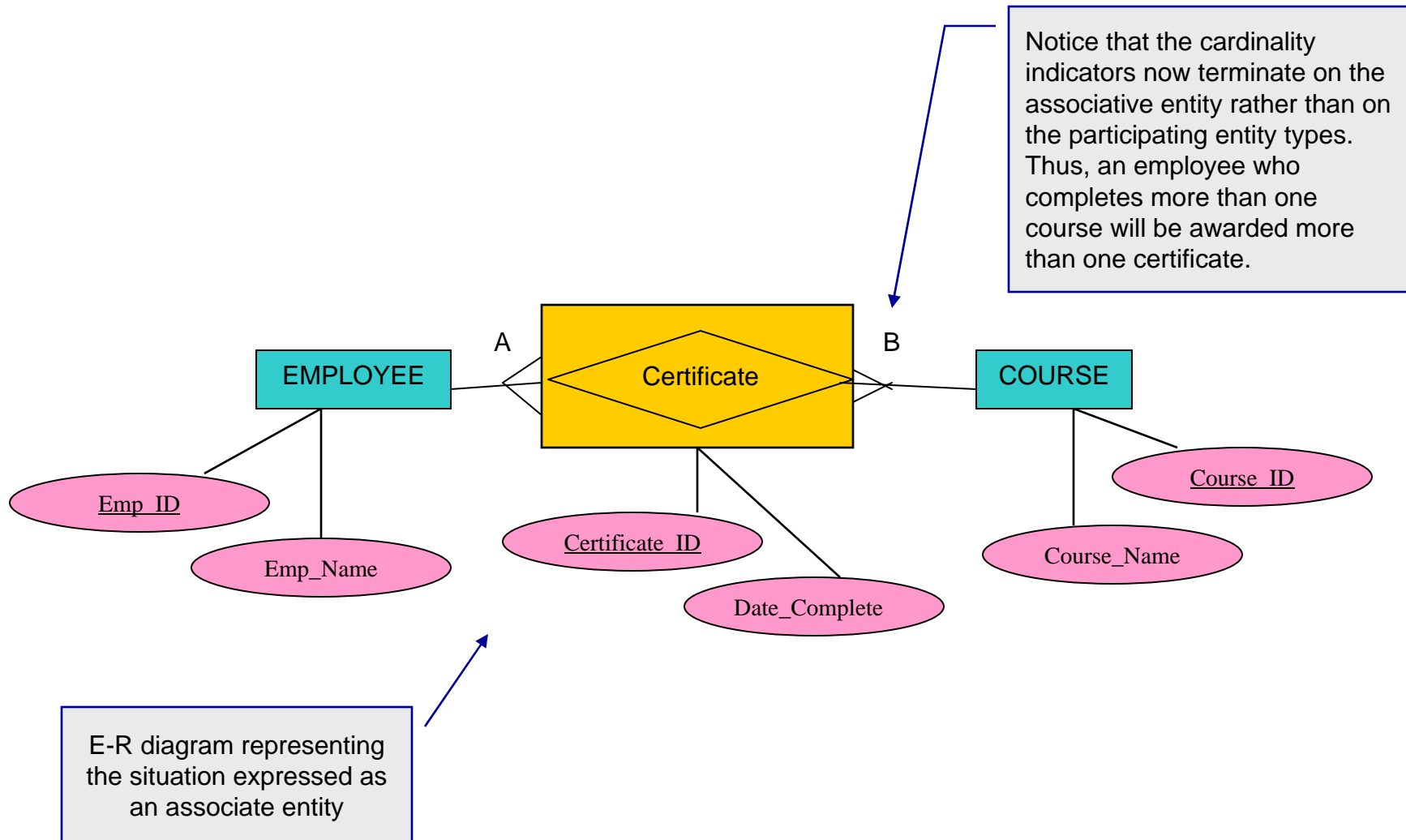
An Aside On Associative Entities (cont.)

Employee_Name	Course_Title	Date_Completed
Kristi	C++	6/2005
Kristi	Java	12/2005
Debi	SQL	11/2005
Angela	SQL	10/2005
Angela	Perl	1/2006

Some sample data

E-R diagram representing the situation

B EMPLOYEE Completes A COURSE

Emp_ID

Emp_Name

Date_Complete

Course_ID

Course_Name

An Aside On Associative Entities (cont.)

Notice that the cardinality indicators now terminate on the associative entity rather than on the participating entity types. Thus, an employee who completes more than one course will be awarded more than one certificate.

A

EMPLOYEE

Certificate

B

COURSE

Emp_ID

Emp_Name

Certificate_ID

Date_Complete

Course_ID

Course_Name

E-R diagram representing the situation expressed as an associate entity

An Aside On Associative Entities (cont.)

- How do you know whether to convert a relationship into an associative entity type?

- There are four conditions that should exist:

1. All of the relationships for the participating entity types are "many" relationships.

2. The resulting associative entity type has independent meaning to end users, and preferably can be identified with a single-attribute identifier.

3. The associative entity has one or more attributes, in addition to the identifier.

4. The associative entity participates in one or more relationships independent of the entities related in the associated relationship.

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.
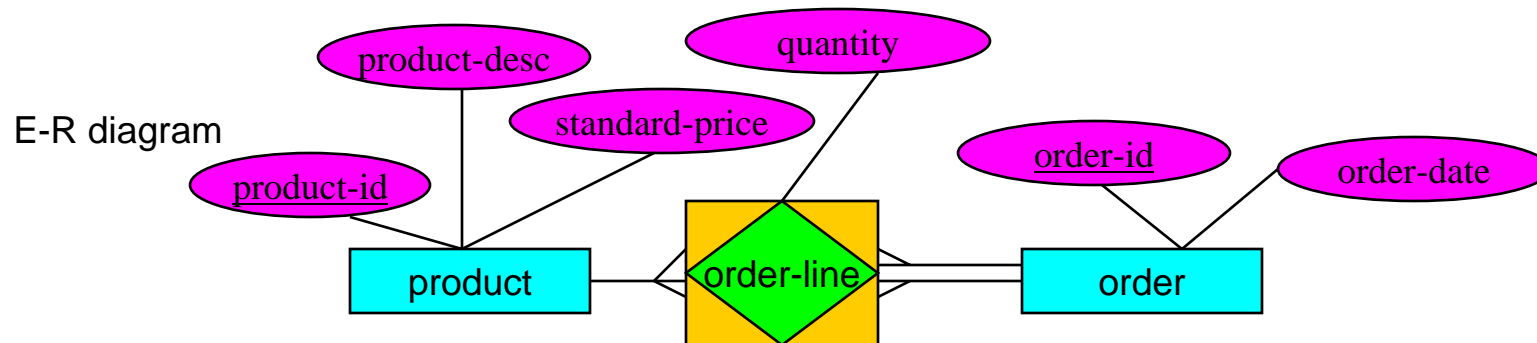
Mapping E-R Diagrams to Relational Schemas

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
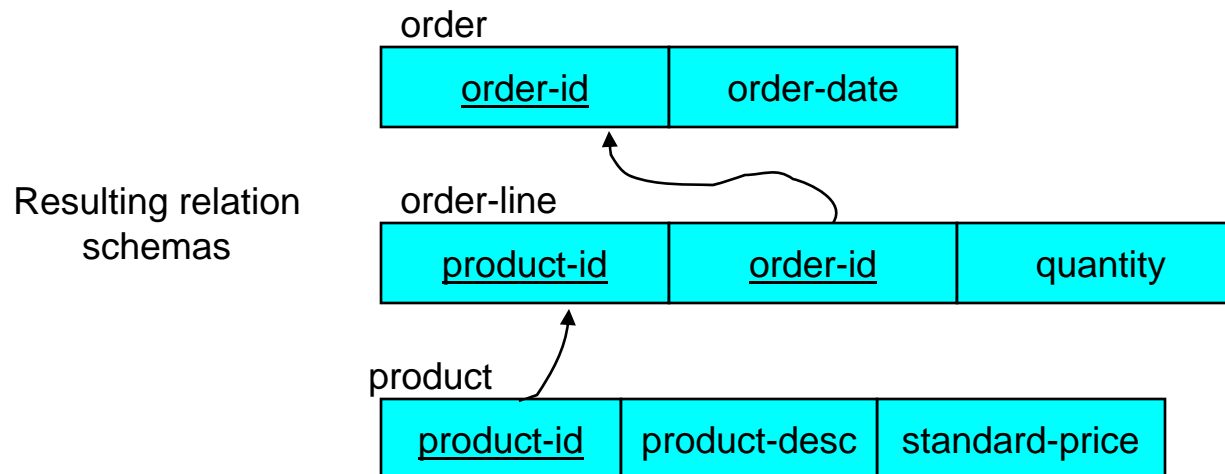
E-R diagram

Resulting relation schemas

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:

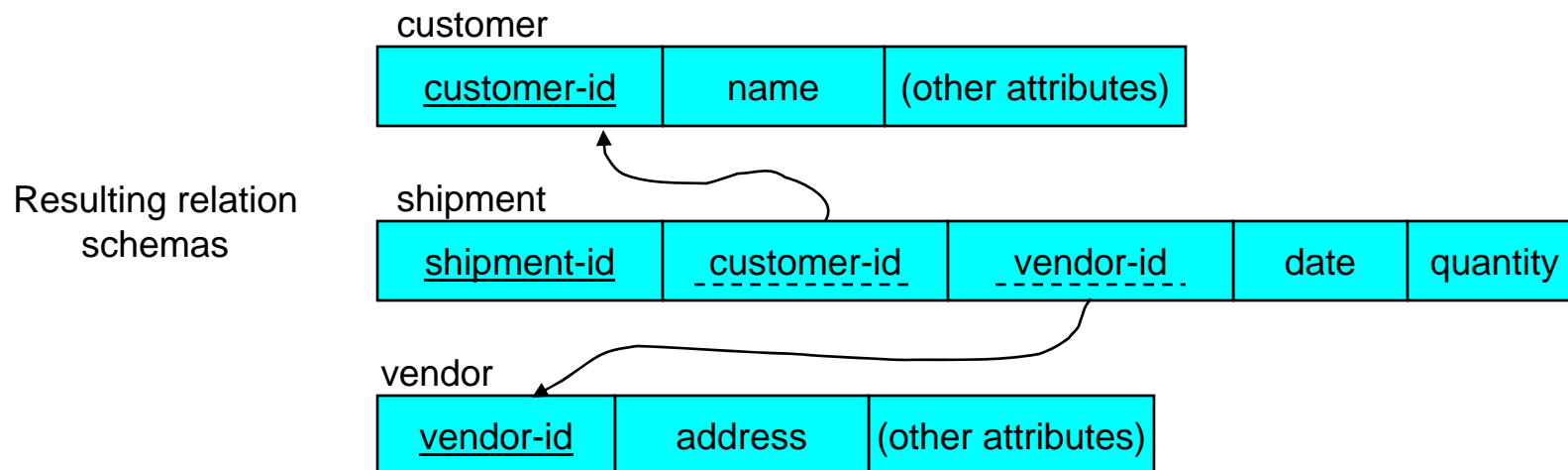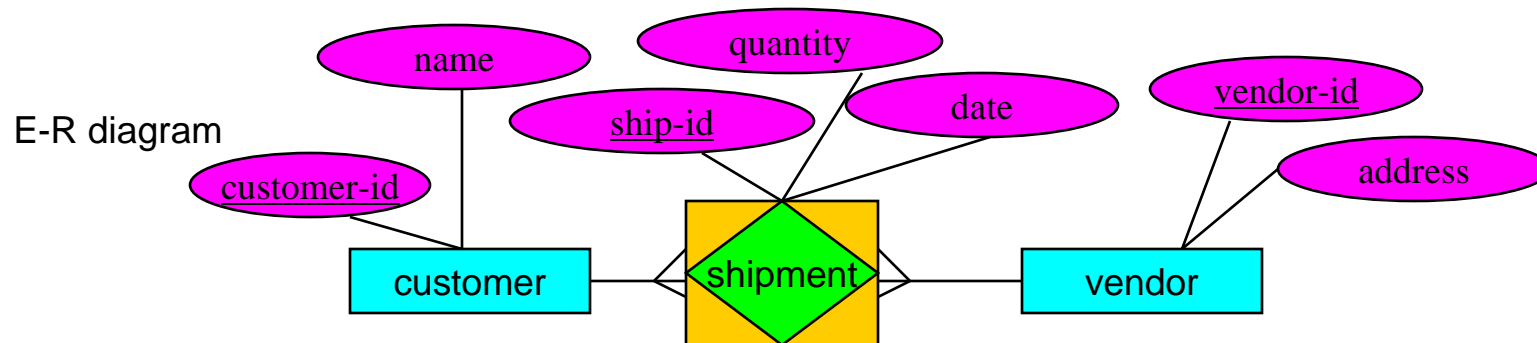# Mapping E-R Diagrams to Relational Schemas

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

E-R diagram

Resulting relation schemas

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.
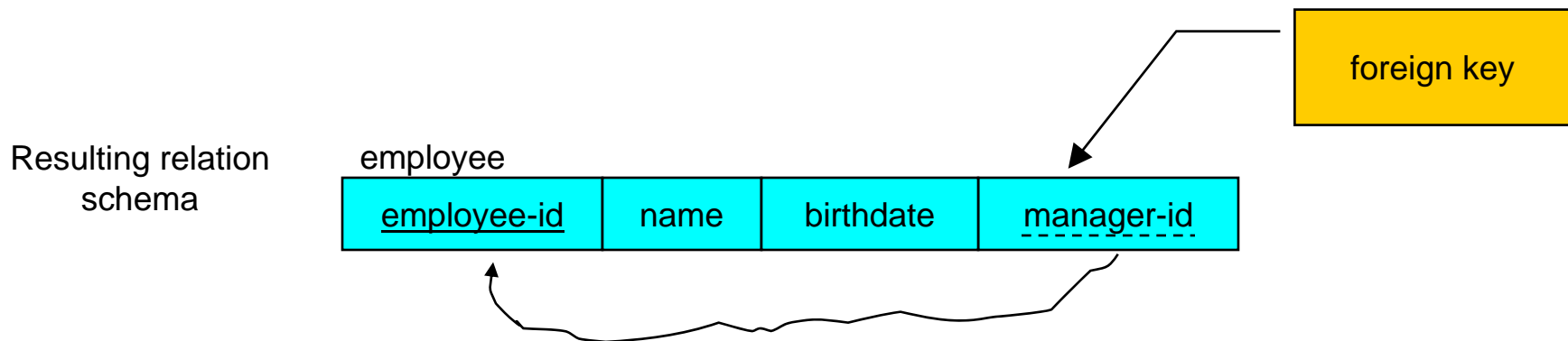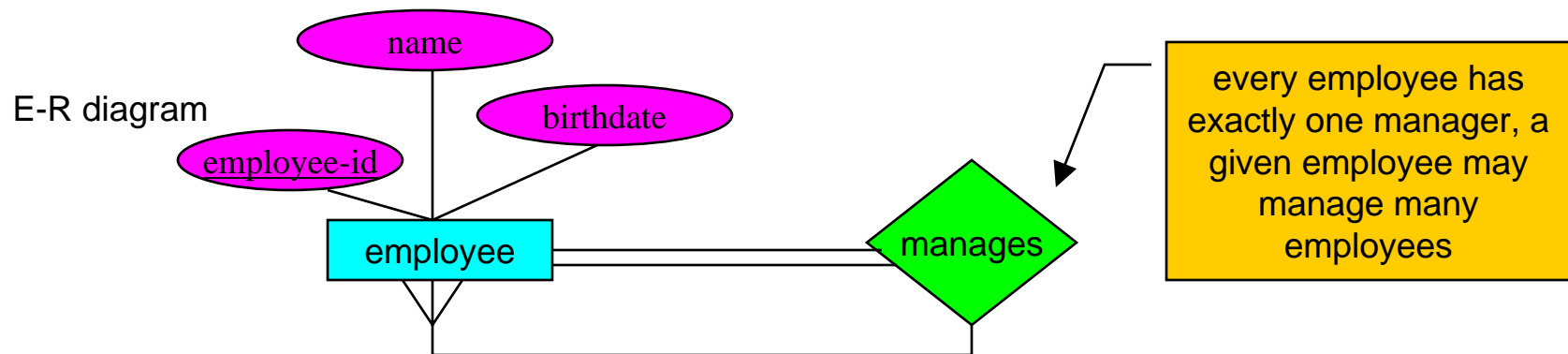
Mapping E-R Diagrams to Relational Schemas

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

E-R diagram

name

birthdate

employee-id

employee

manages

every employee has exactly one manager, a given employee may manage many employees

foreign key

Resulting relation schema

employee

employee-id	name	birthdate	manager-id

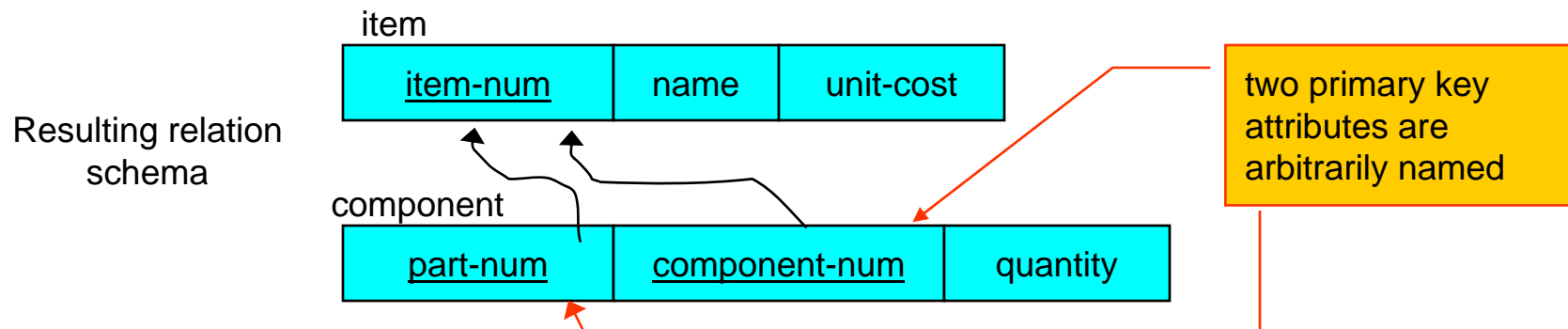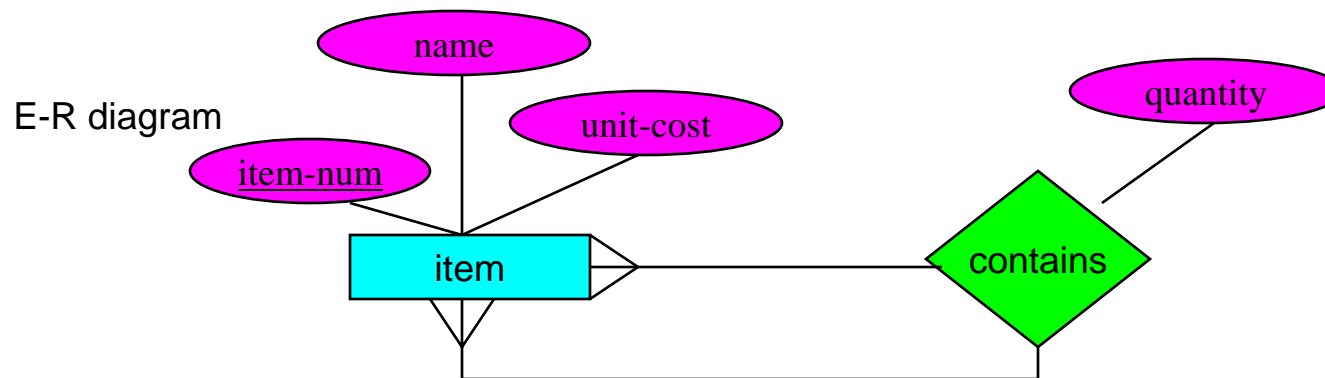# Mapping E-R Diagrams to Relational Schemas

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.
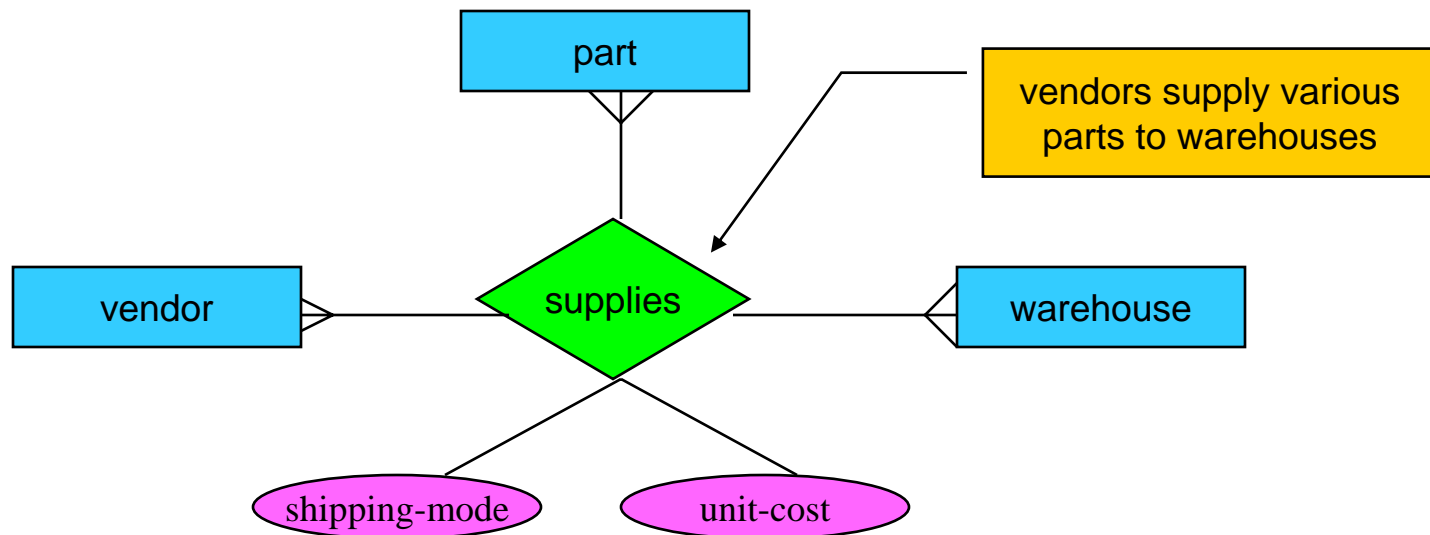
Mapping E-R Diagrams to Relational Schemas

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

E-R diagram

name

unit-cost

item-num

quantity

item

contains

Resulting relation schema

item

item-num	name	unit-cost

two primary key attributes are arbitrarily named

component

part-num	component-num	quantity

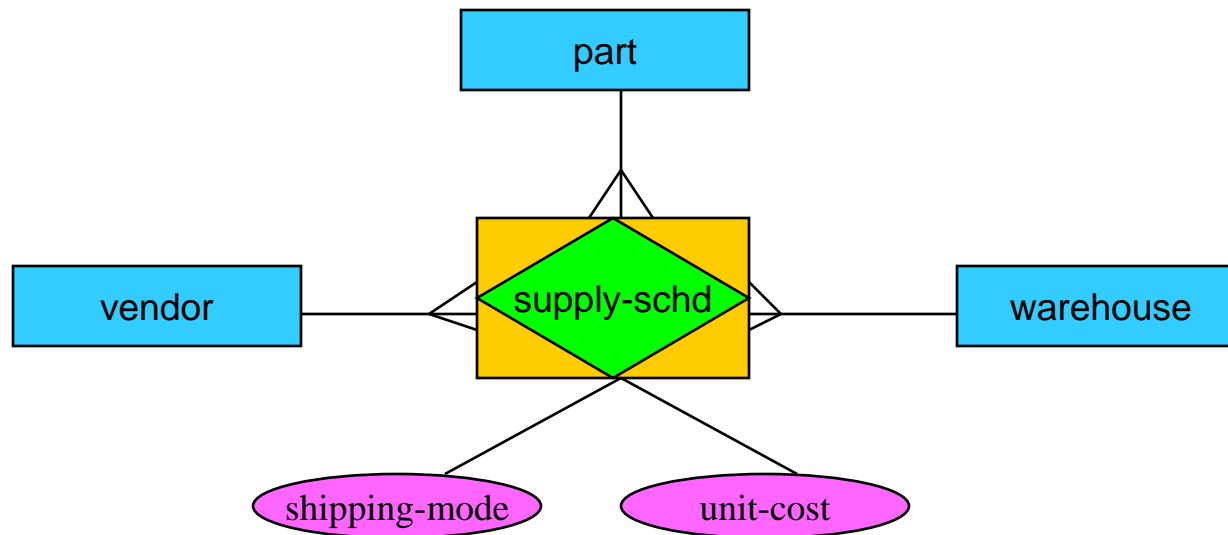# Mapping E-R Diagrams to Relational Schemas

- 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:
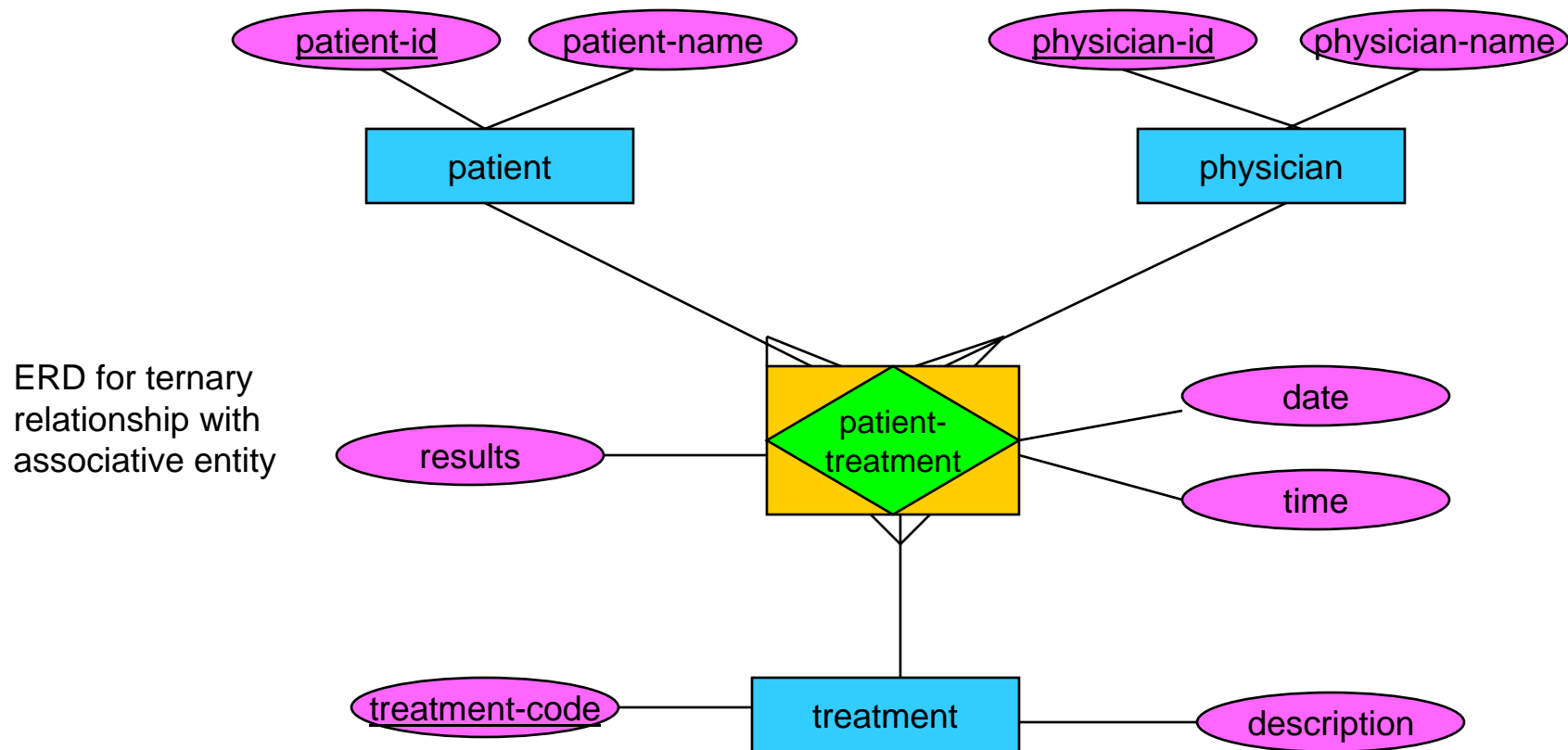
Mapping E-R Diagrams to Relational Schemas

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.

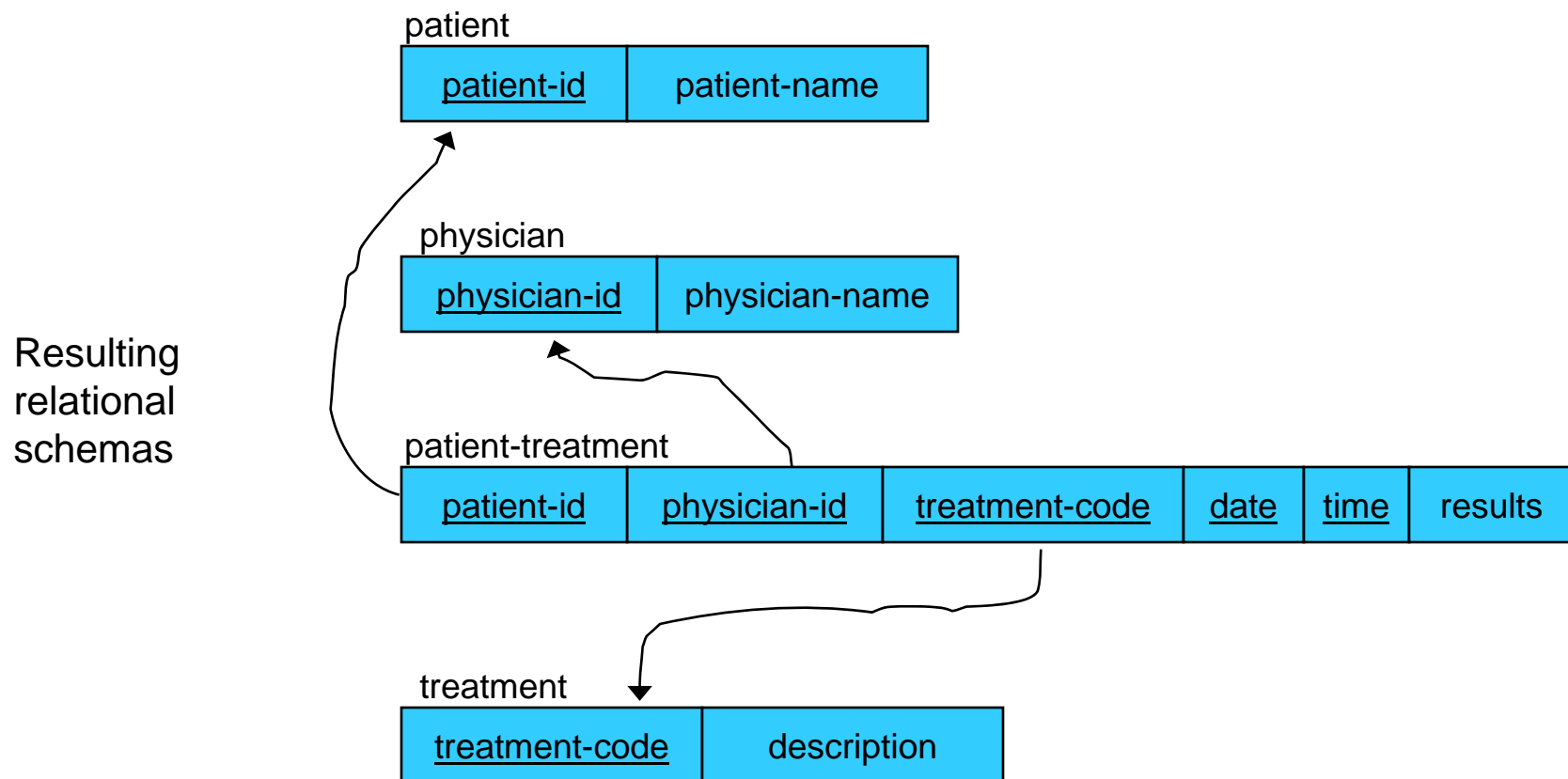- Any attributes of the associative entity type become attributes in the new associative relation.

Mapping E-R Diagrams to Relational Schemas

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

ERD for ternary
relationship with
associative entity

Mapping E-R Diagrams to Relational Schemas

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

Resulting relational schemas

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

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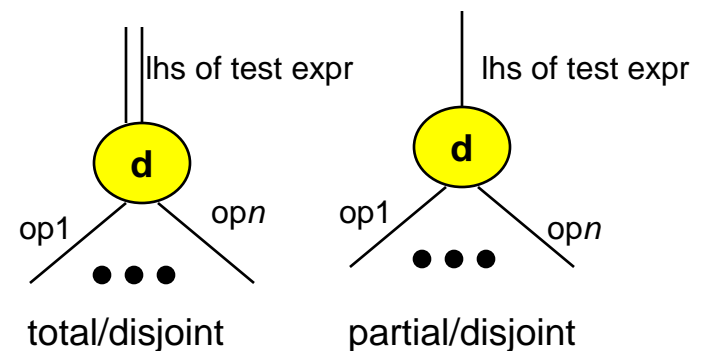
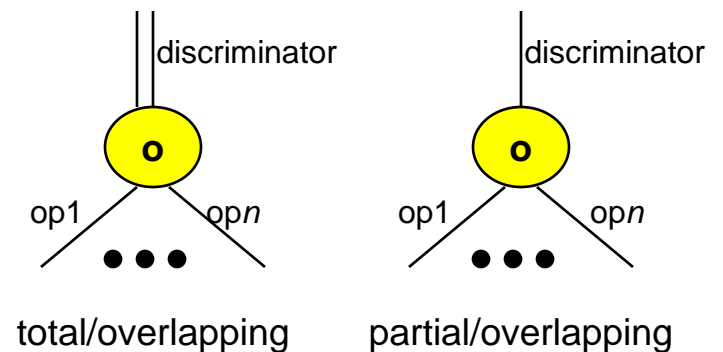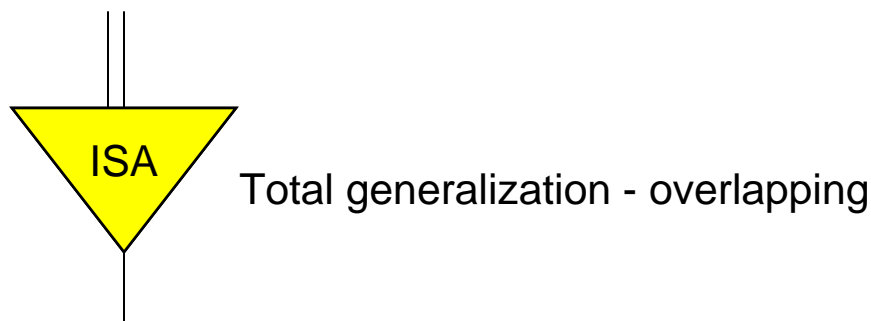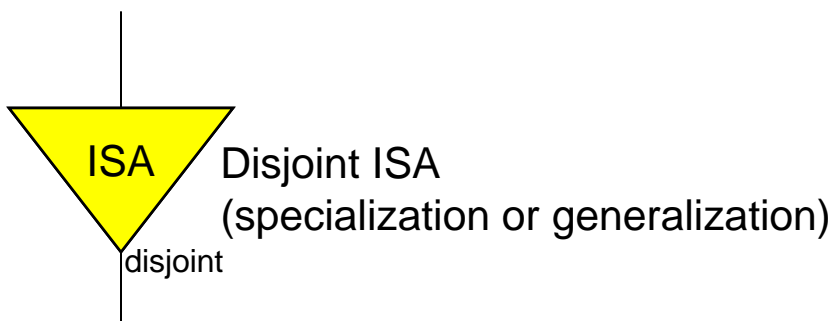# Mapping E-R Diagrams to Relational Schemas (cont.)

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

ISA (specialization or generalization)
(partial participation - overlapping)

Disjoint ISA
(specialization or generalization)

disjoint

Total generalization - overlapping

discriminator

o

op1   op*n*

● ● ●

total/overlapping

discriminator

o

op1   op*n*

● ● ●

partial/overlapping

lhs of test expr

d

op1   op*n*

● ● ●

total/disjoint

lhs of test expr

d

op1   op*n*

● ● ●

partial/disjoint
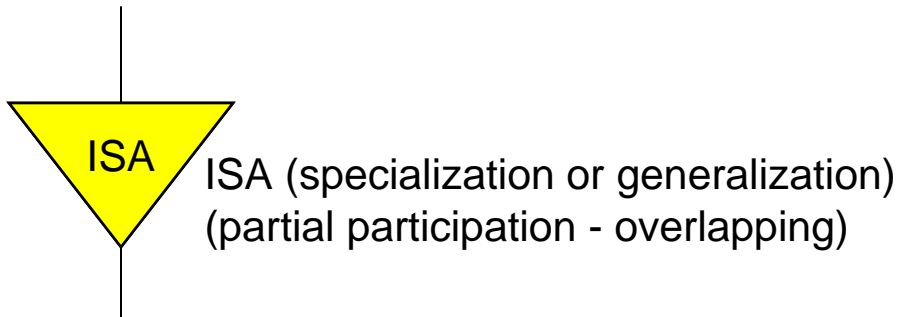
Mapping E-R Diagrams to Relational Schemas

EXAMPLE – Subtype discriminators in ERDs: disjoint subtypes

emp-name   address   emp-type

employee-id   employee   date-hired

new attribute added as a subtype discriminator

emp-type =

ERD showing disjoint subtype discriminator notation

d

'H'   'S'   'C'

hourly-emp   salaried-emp   consultant   billing-rate

hr-rate   salary   stock-opt   contract-#
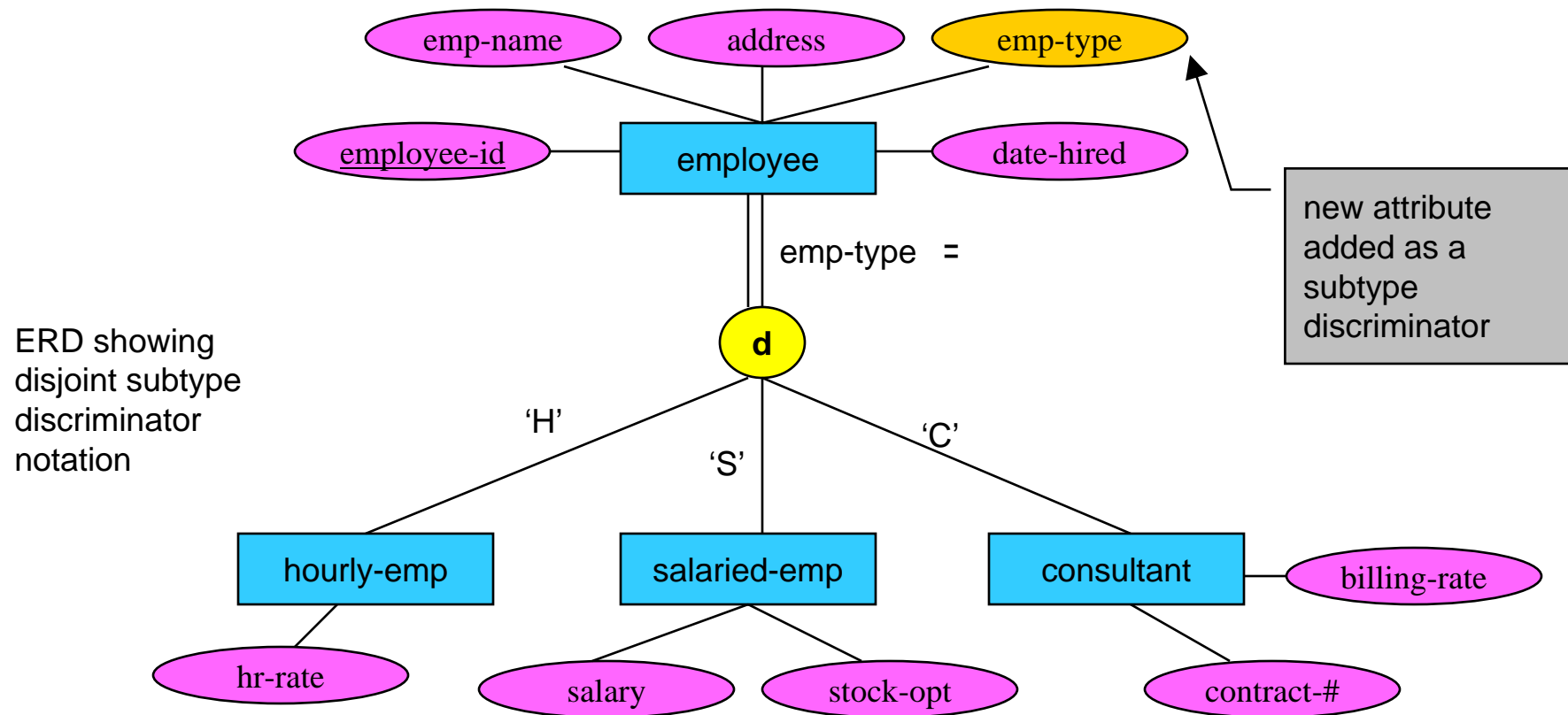
Mapping E-R Diagrams to Relational Schemas

EXAMPLE – Subtype discriminators in ERDs: overlapping subtypes
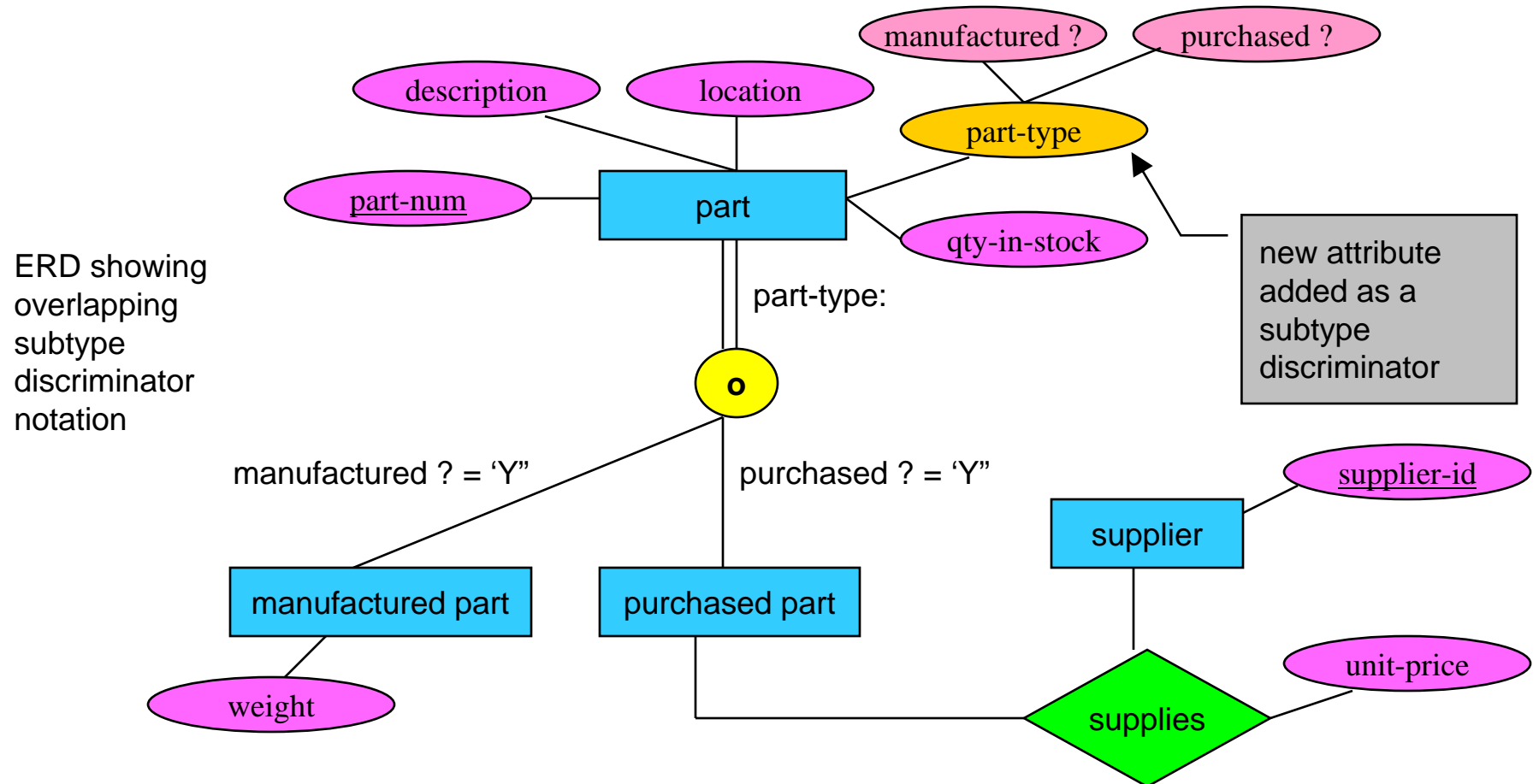
ERD showing overlapping subtype discriminator notation

manufactured ?   purchased ?

description   location   part-type

part-num   part   qty-in-stock

new attribute added as a subtype discriminator

part-type:

o

manufactured ? = 'Y"   purchased ? = 'Y"

supplier-id

supplier

manufactured part   purchased part

weight

unit-price

supplies

Mapping E-R Diagrams to Relational Schemas

STEP 7 – Mapping Supertype/Subtype Relationships - EXAMPLE

ERD with
disjoint subtypes

Mapping E-R Diagrams to Relational Schemas

STEP 7: Mapping Supertype/Subtype Relationships - EXAMPLE

employee

employee-id	emp-name	address	emp-type	date-hired

Resulting relational schemas

hourly emp

h-employee-id	hr-rate

salaried-emp

s-employee-id	salary	stock-opt

consultant

c-employee-id	contract-num	billing-rate
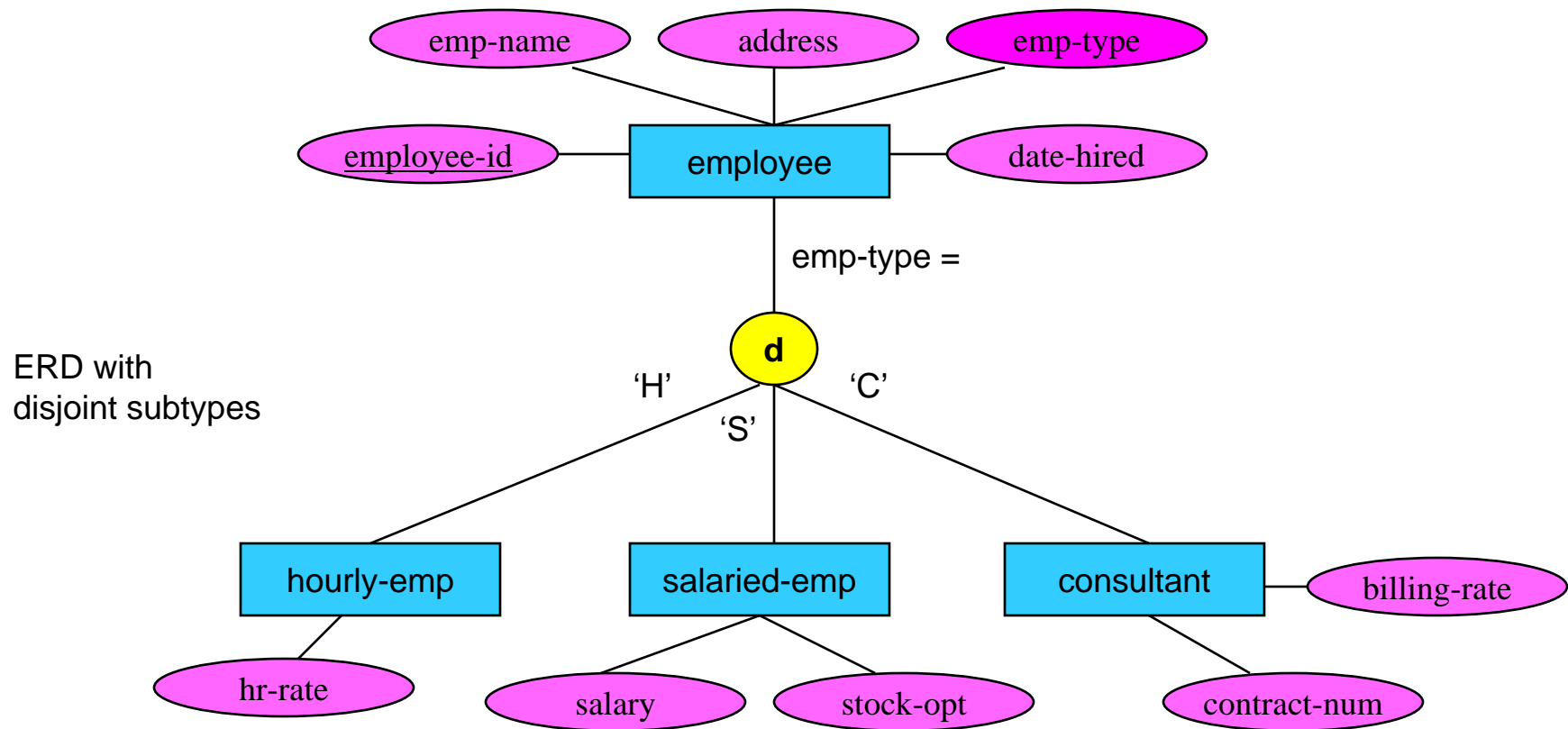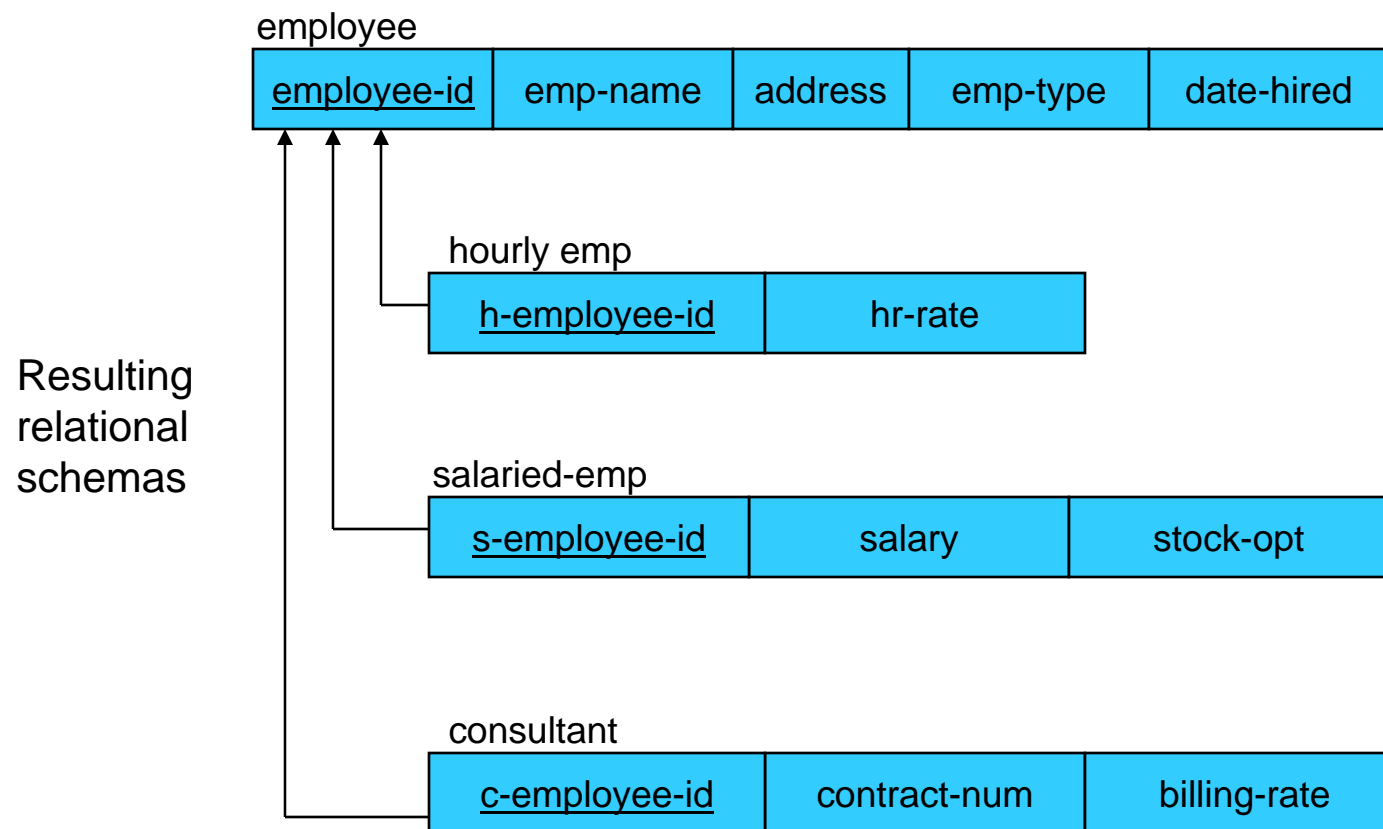
Mapping E-R Diagrams to Relational Schemas

PRACTICE