CGS 2545: Database Concepts
Spring 2014

Chapter 3 – The Enhanced ER Model And Business Rules

Instructor : Dr. Mark Llewellyn
markl@cs.ucf.edu
HEC 236, 407-823-2790
http://www.cs.ucf.edu/courses/cgs2545/spr2014

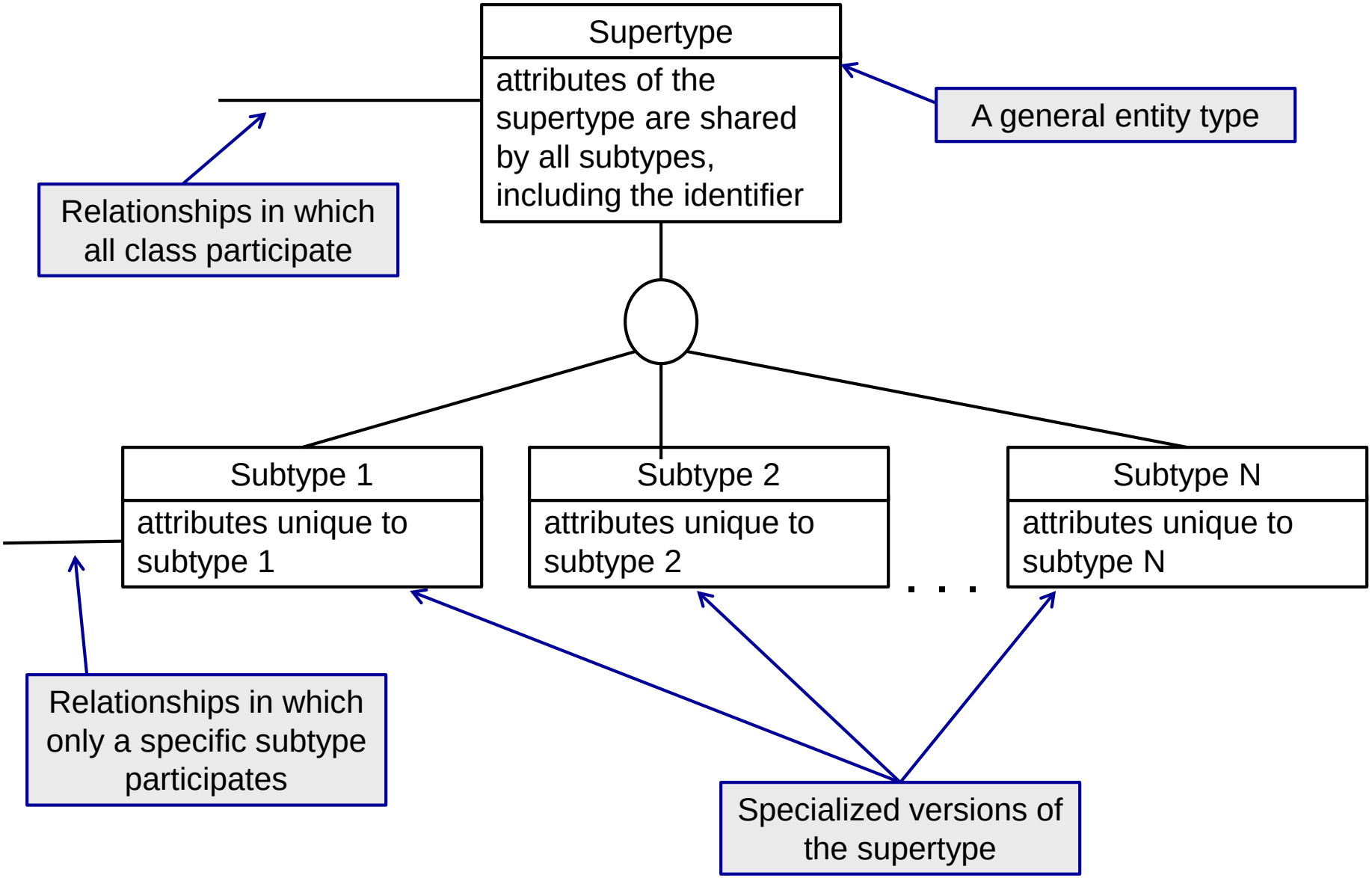
Department of Electrical Engineering and Computer Science
Computer Science Division
University of Central Florida

Introduction

- The basic ER model was introduce in the 1970s.
- It has been widely accepted as a suitable model for most business situations.
- However, the business environment has changed substantially in the 30 plus years that have elapsed since the introduction of the model.
- Business relationships are now more complex, and as a result, business data is much more complex.
- To cope with these changes the ER model has been enhanced in many areas since it was first introduced so that it can more accurately represent the complex data encountered in today's business environment.
- The term enhanced entity-relationship (EER) model is used to identify the model that has results from extending the original ER model with new modeling constructs.
- Two main extensions to the ER model incorporated in the EER model are: (1) supertype/subtype relationships, and (2) entity clustering techniques.

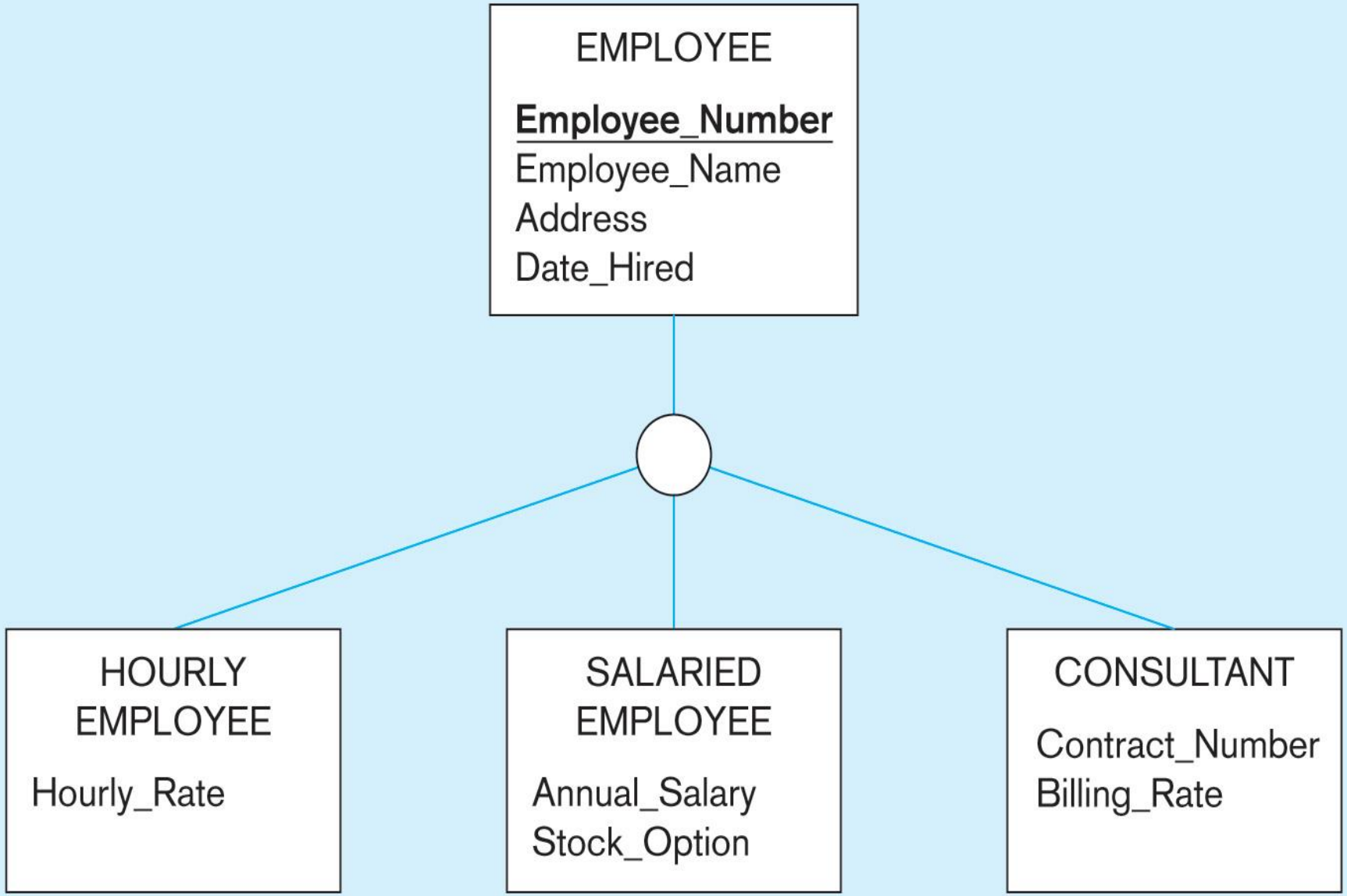
Supertypes and Subtypes

- Supertype: An generic entity type that has a relationship with one or more subtypes.
- Subtype: A subgrouping of the entities in an entity type which has attributes that are distinct from those in other subgroupings.
- Attribute Inheritance:
  - Subtype entities inherit values of all attributes of the supertype.
  - An instance of a subtype is also an instance of the supertype.
- The next page illustrates the supertype/subtype hierarchy.


Supertype
attributes of the supertype are shared by all subtypes, including the identifier
A general entity type
Relationships in which all class participate
Subtype 1
attributes unique to subtype 1
Subtype 2
attributes unique to subtype 2
Subtype N
attributes unique to subtype N
Relationships in which only a specific subtype participates
Specialized versions of the supertype

Relationships and Subtypes

- Relationships at the **supertype** level indicate that all subtypes will participate in the relationship.
- The instances of a **subtype** may participate in a relationship unique to that subtype. In this situation, the relationship is shown at the subtype level.
- An example is shown on the next page.


EMPLOYEE
Employee_Number
Employee_Name
Address
Date_Hired
HOURLY EMPLOYEE
Hourly_Rate
SALARIED EMPLOYEE
Annual_Salary
Stock_Option
CONSULTANT
Contract_Number
Billing_Rate

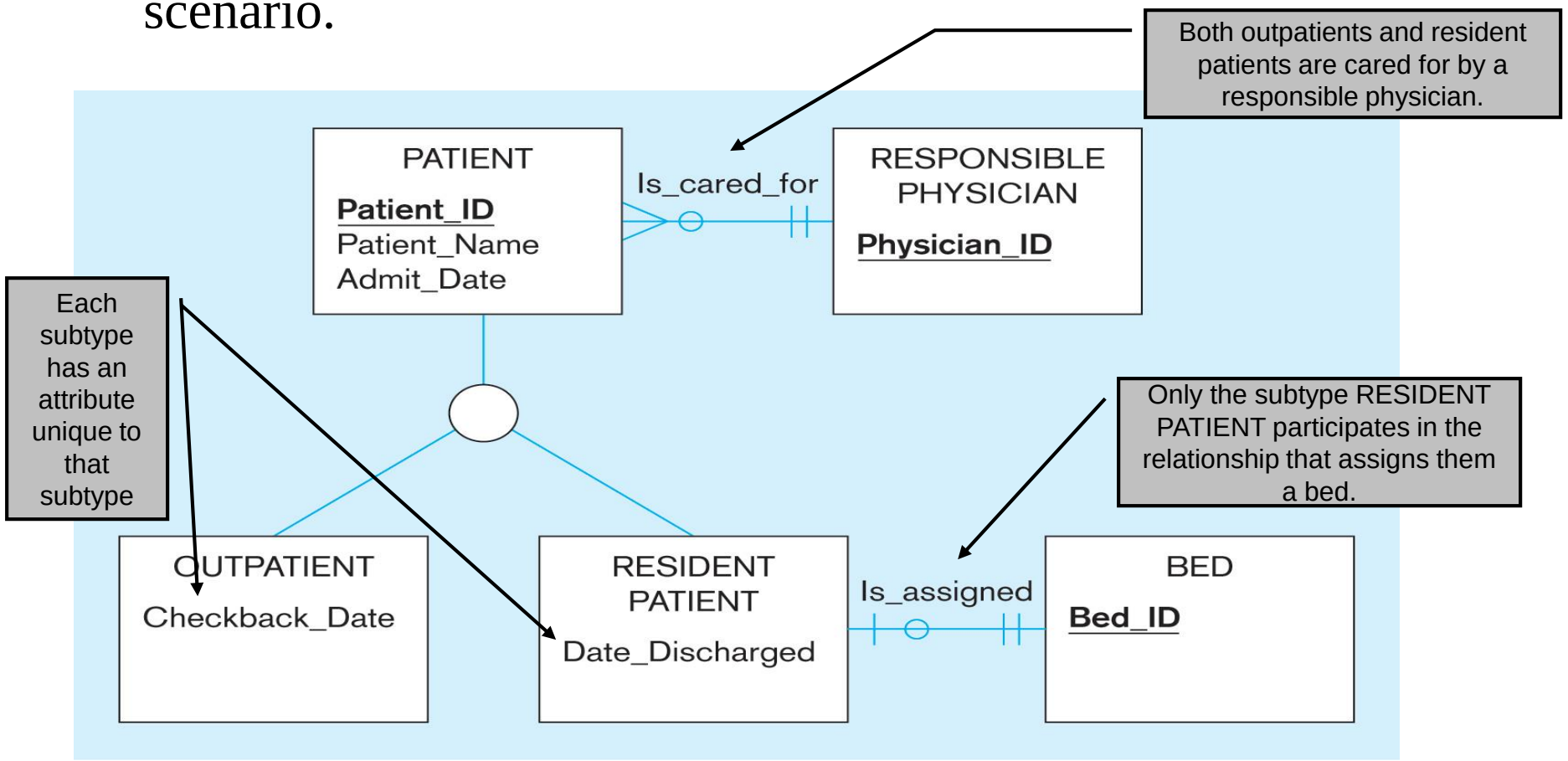
When To Use Supertype/Subtype Relationships

- Whether to use supertype/subtype relationships is a decision the data modeler must make in each situation.
- You should consider using subtypes when either (or both) of the following conditions are present:

1. There are attributes that apply to some (but not all) instances of an entity type. See the example on the previous page.
2. The instances of a subtype participate in a relationship that is unique to the subtype, i.e., other subtypes do not participate in the relationship.

When To Use Supertype/Subtype Relationships

- As an example of when to use subtypes, consider the following scenario.

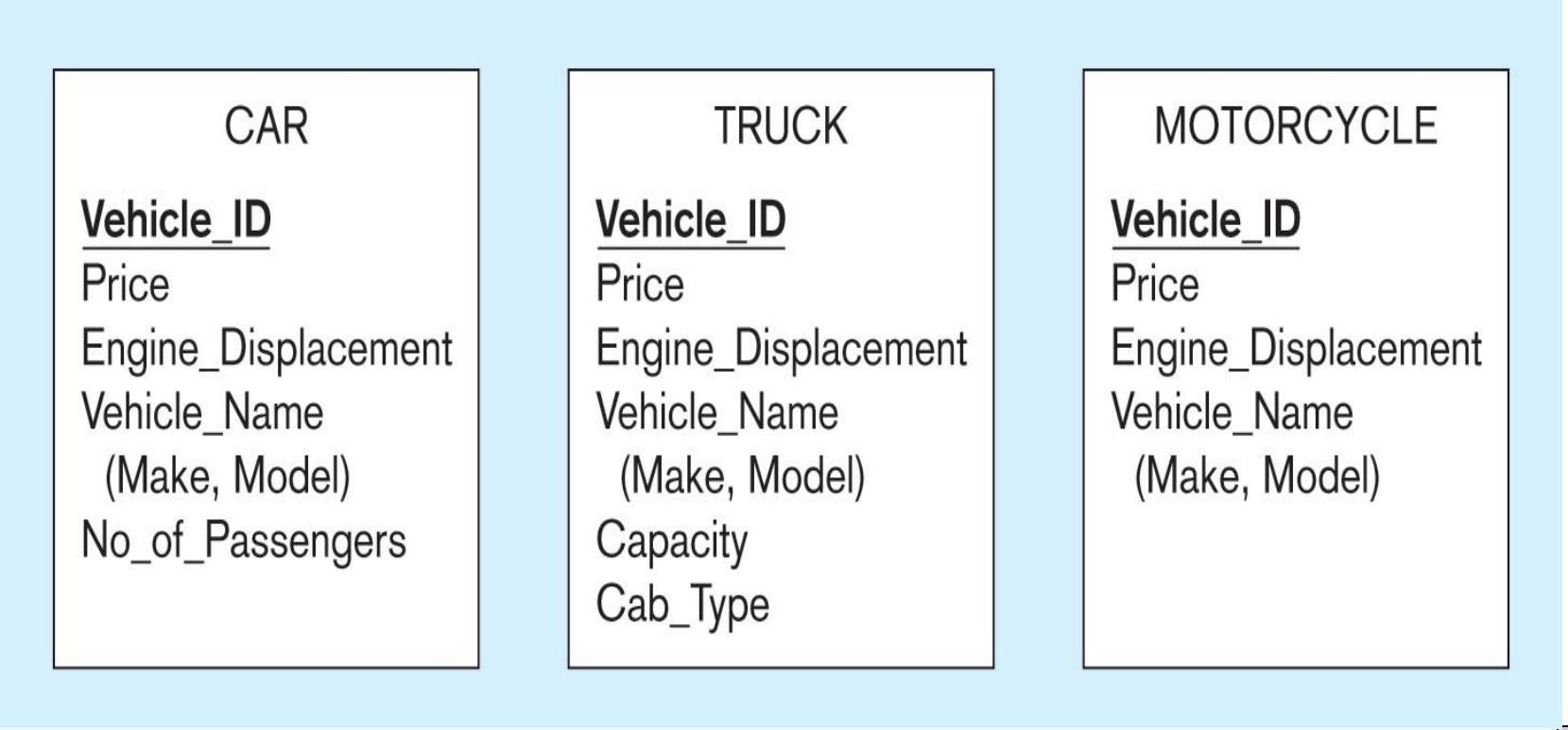


Generalization and Specialization

- **Generalization:** The process of defining a more general entity type from a set of more specialized entity types.
  - This is a BOTTOM-UP approach to design.
- **Specialization:** The process of defining one or more subtypes of the supertype, and forming supertype/subtype relationships.
  - This is a TOP-DOWN approach to design.

Generalization

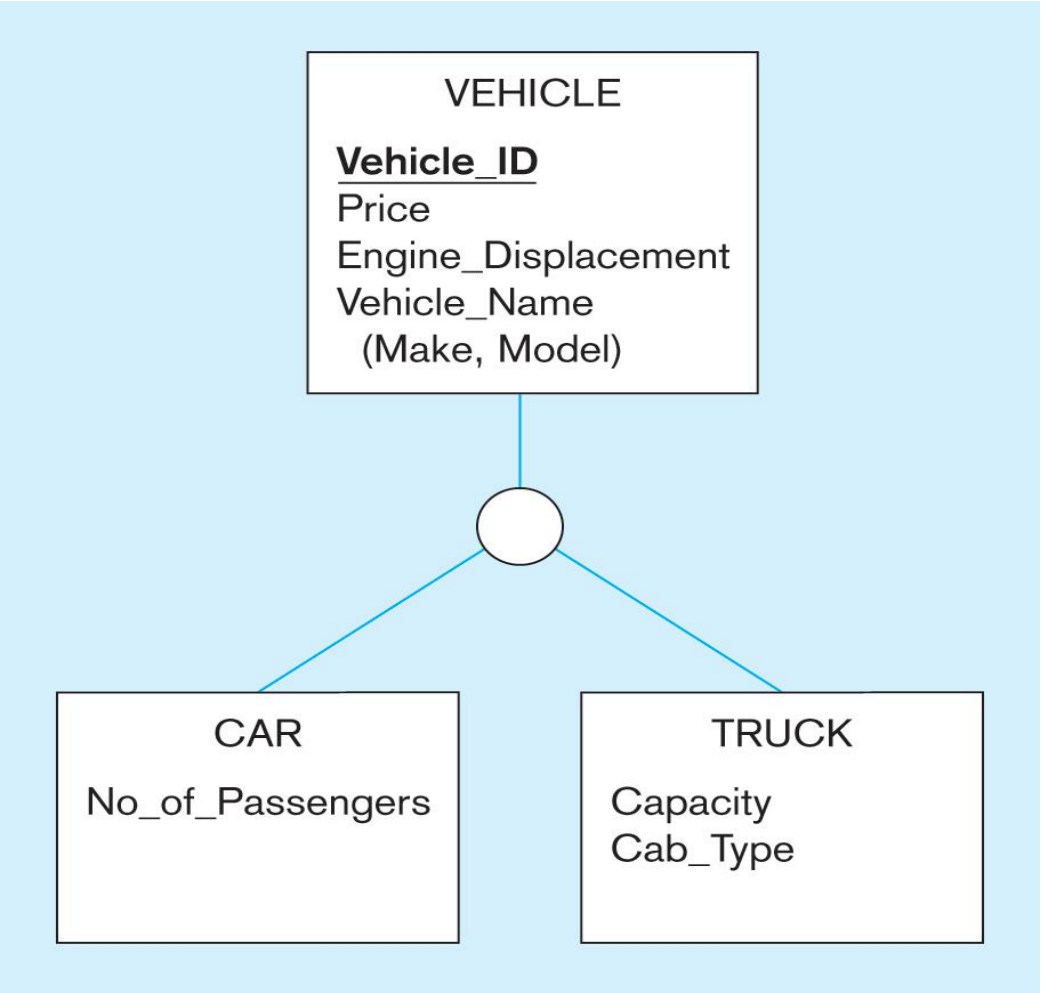
- The data modeler has identified three entity types.
- Notice the similarities and differences amongst these types.



Generalization

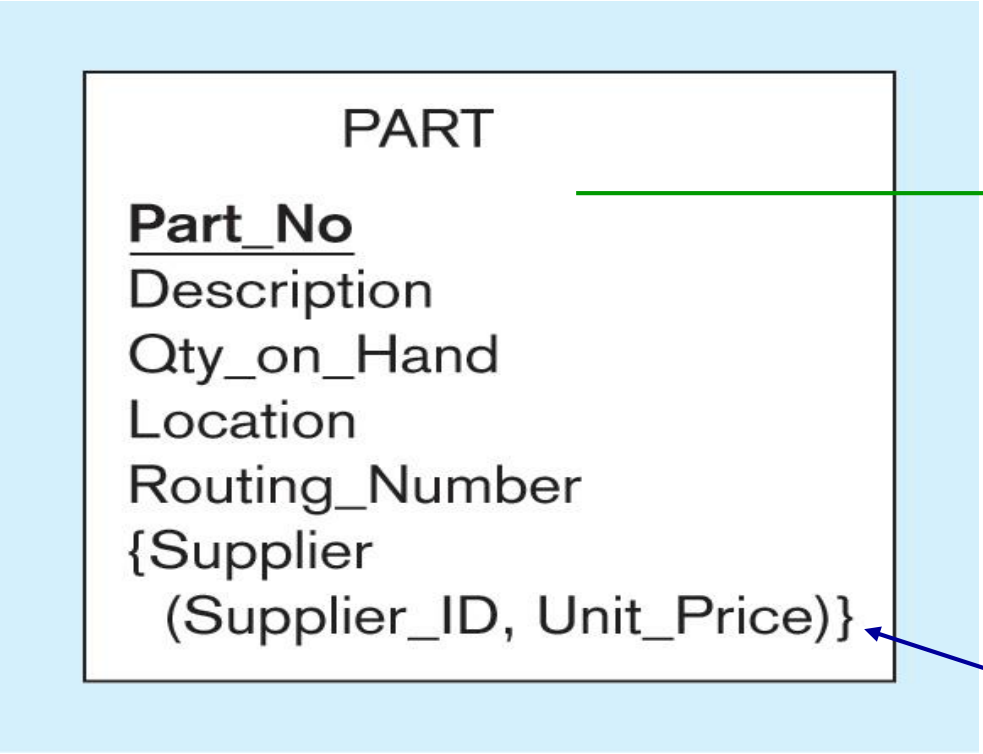
Question: What happened to the motorcycle entity type?

Answer: Since the class does not satisfy the conditions for developing a subtype. The type has no unique attributes and does not participate in any unique relationships. Therefore, motorcycles are simply vehicles without any specialization.

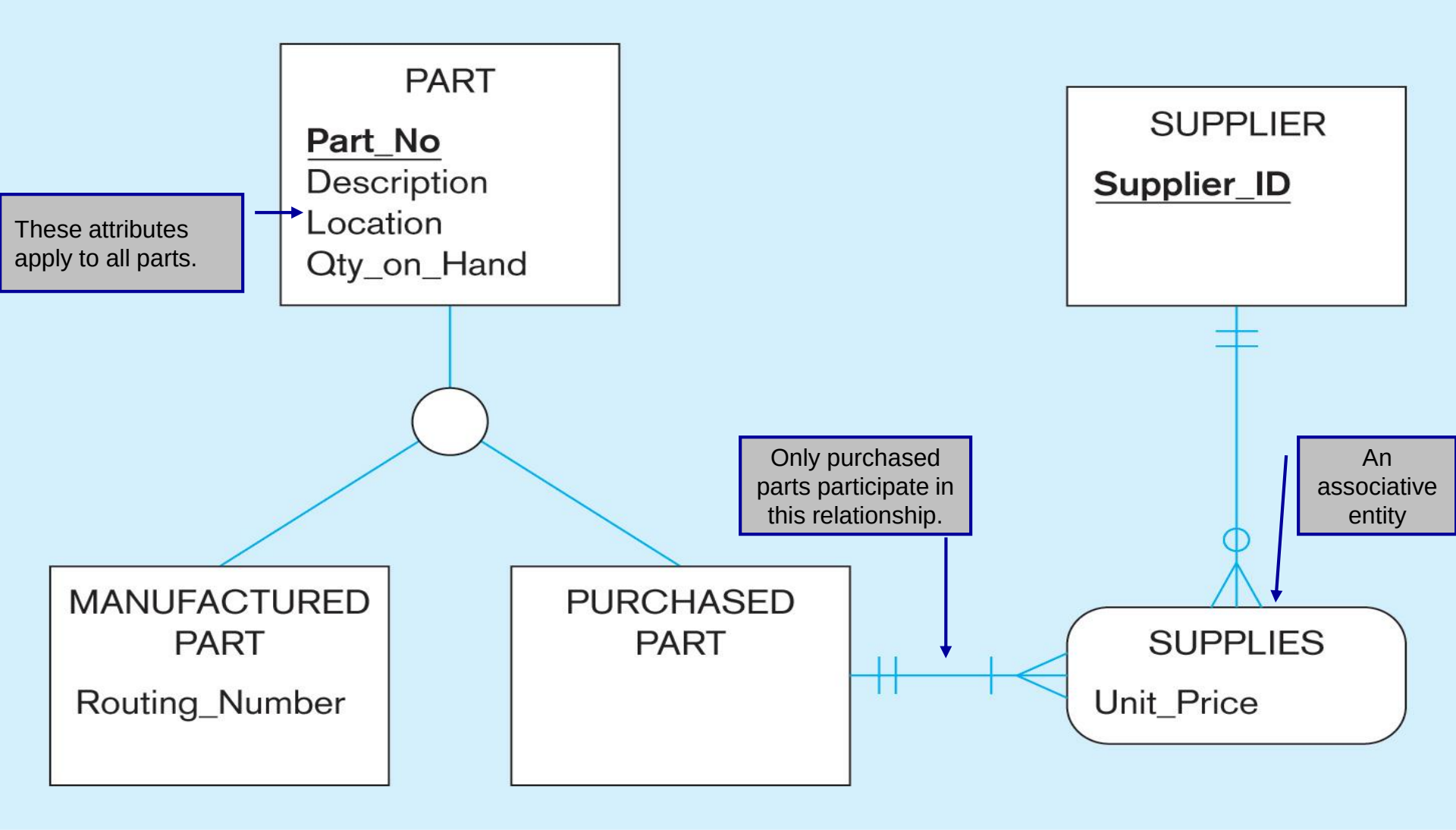


Specialization

• The data modeler has identified an entity type that contains a multi-valued attribute. Some of the attributes apply to all parts regardless of the source, while some of the attributes depend on the source.



Specialization



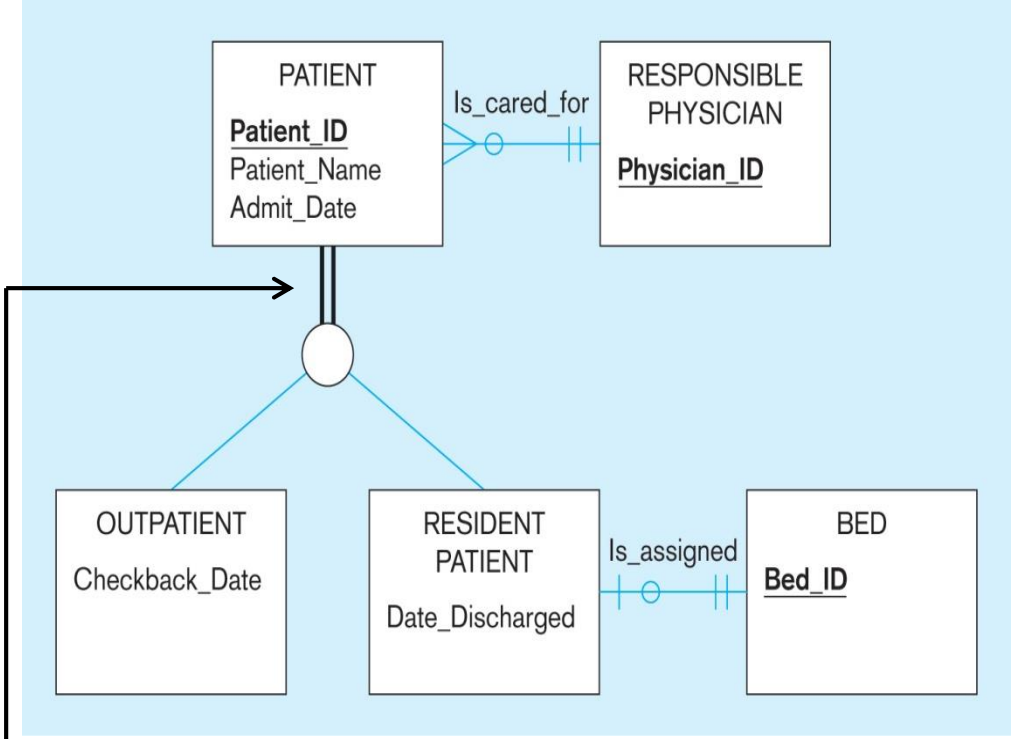
Constraints in Supertype/Subtype Relationships

Completeness Constraints

- A completeness constraint specifies whether an instance of a supertype ***must*** also be a member of at least one subtype. There are two possible cases:
  - Total Specialization Rule
    - All instances in the supertype must also be a member of at least one subtype. Represented by a double line from the supertype to the subclass split (see next page).
  - Partial Specialization Rule
    - Some instances in the supertype may not be members of any subtype. Represented by a single line (see next page).

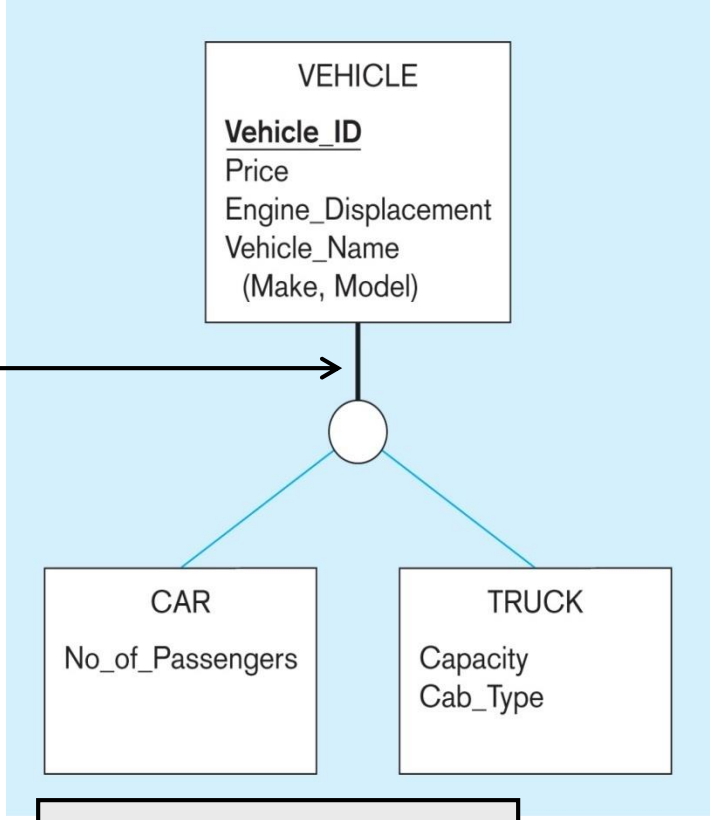
Constraints in Supertype/Subtype Relationships

Completeness Constraints



Total specialization

A patient must either be an outpatient or a resident patient.



Partial specialization

A vehicle may be either a car or a truck or neither.

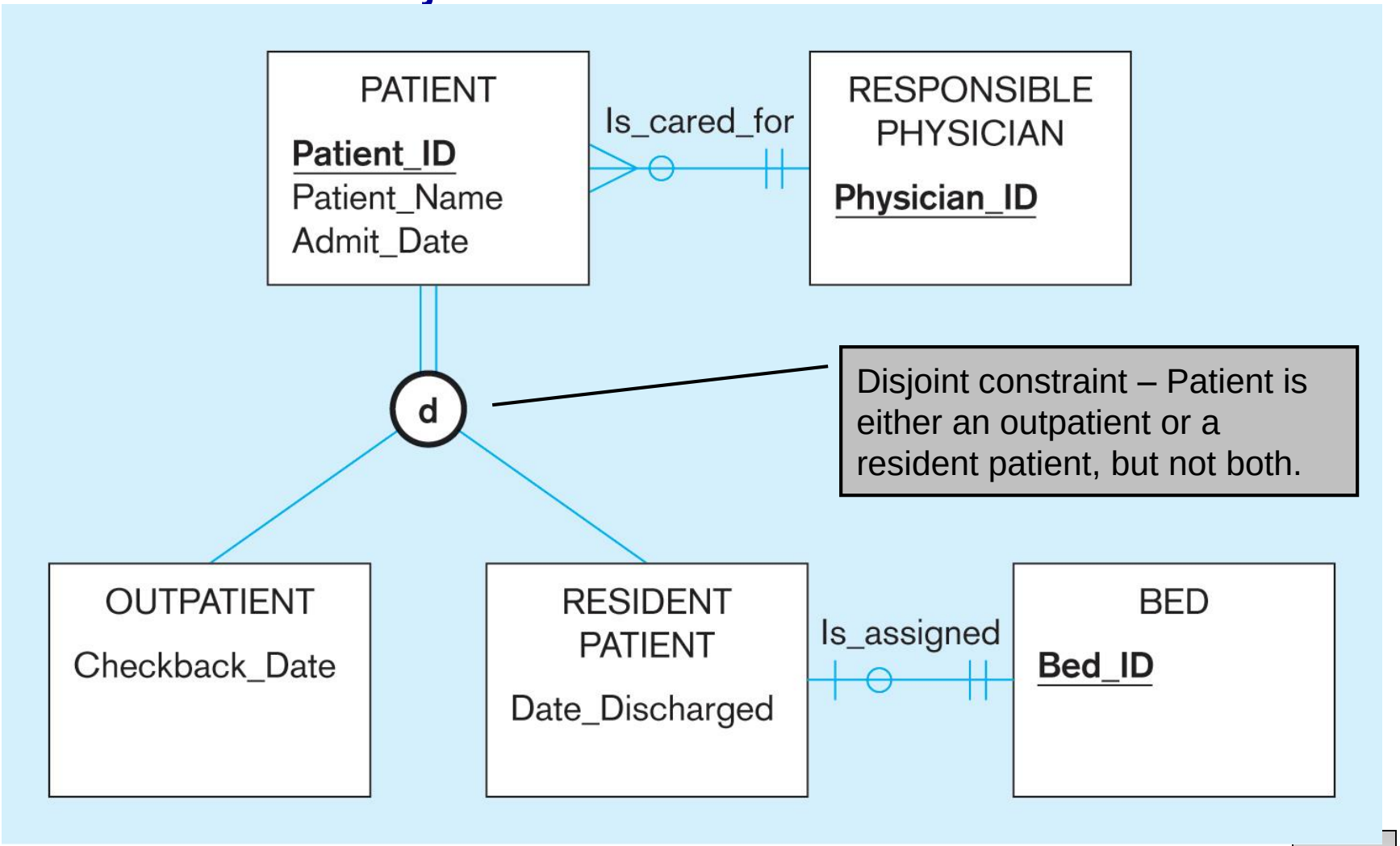
Constraints in Supertype/Subtype Relationships

Disjointness Constraints

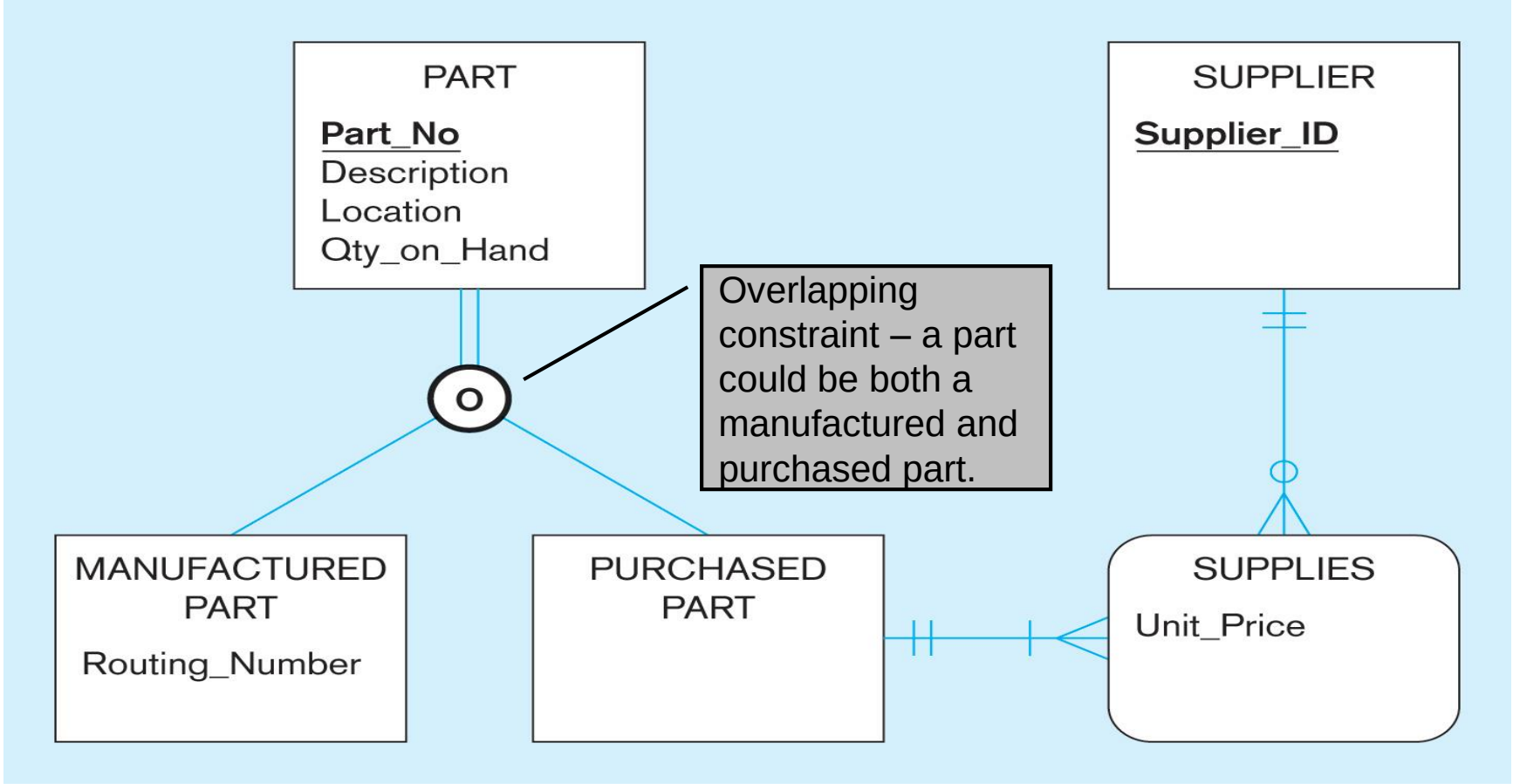
- Whether an instance of a supertype may *simultaneously* be a member of two (or more) subtypes. Again, two rules apply:
  - Disjoint Rule
    - An instance of the supertype can be only ONE of the subtypes. The letter "D" is placed in the category circle.
  - Overlap Rule
    - An instance of the supertype could be more than one of the subtypes. The letter "O" is placed in the category circle.

Constraints in Supertype/Subtype Relationships

Disjointness Constraints



Constraints in Supertype/Subtype Relationships
Disjointness Constraints

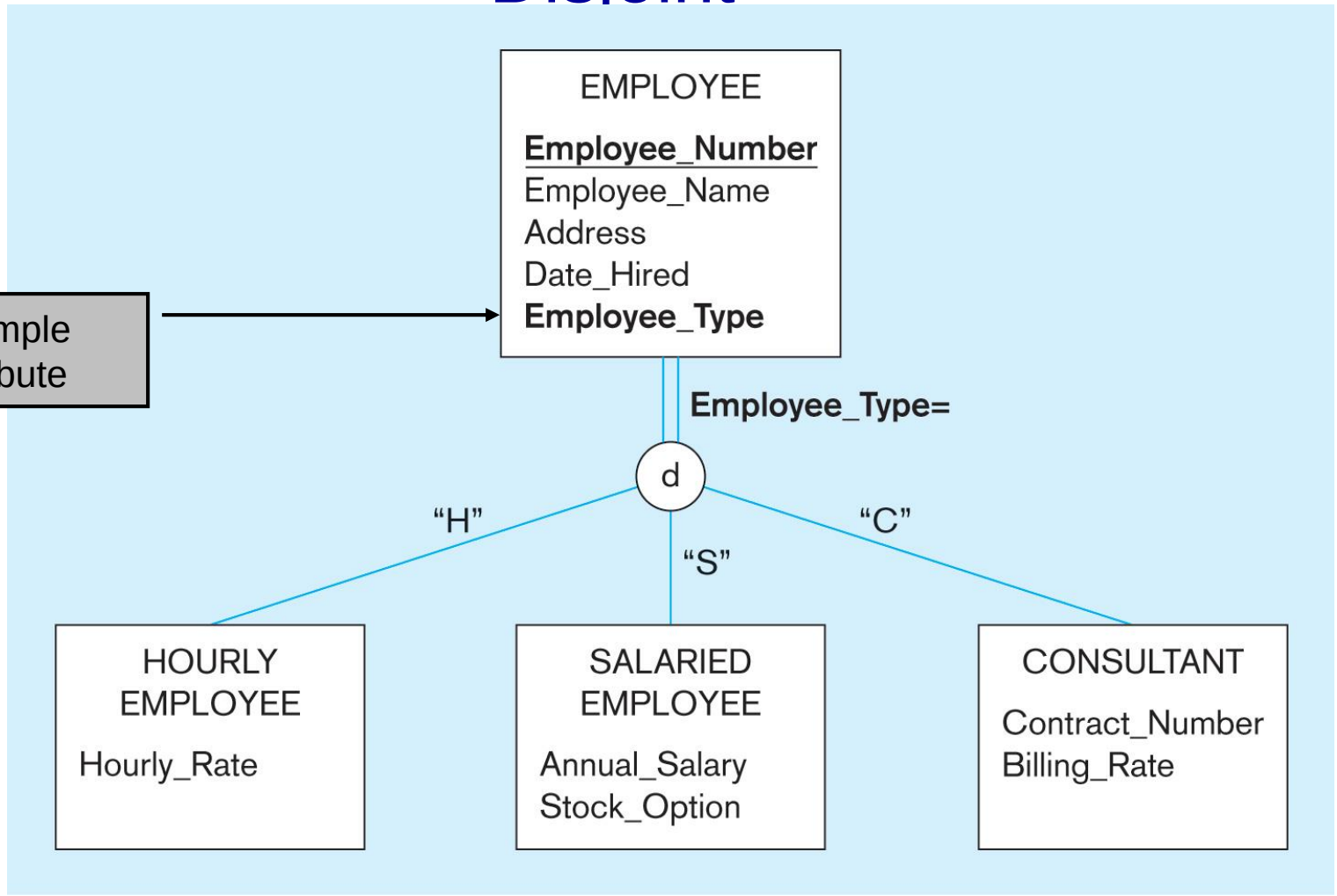


Defining Subtype Discriminators

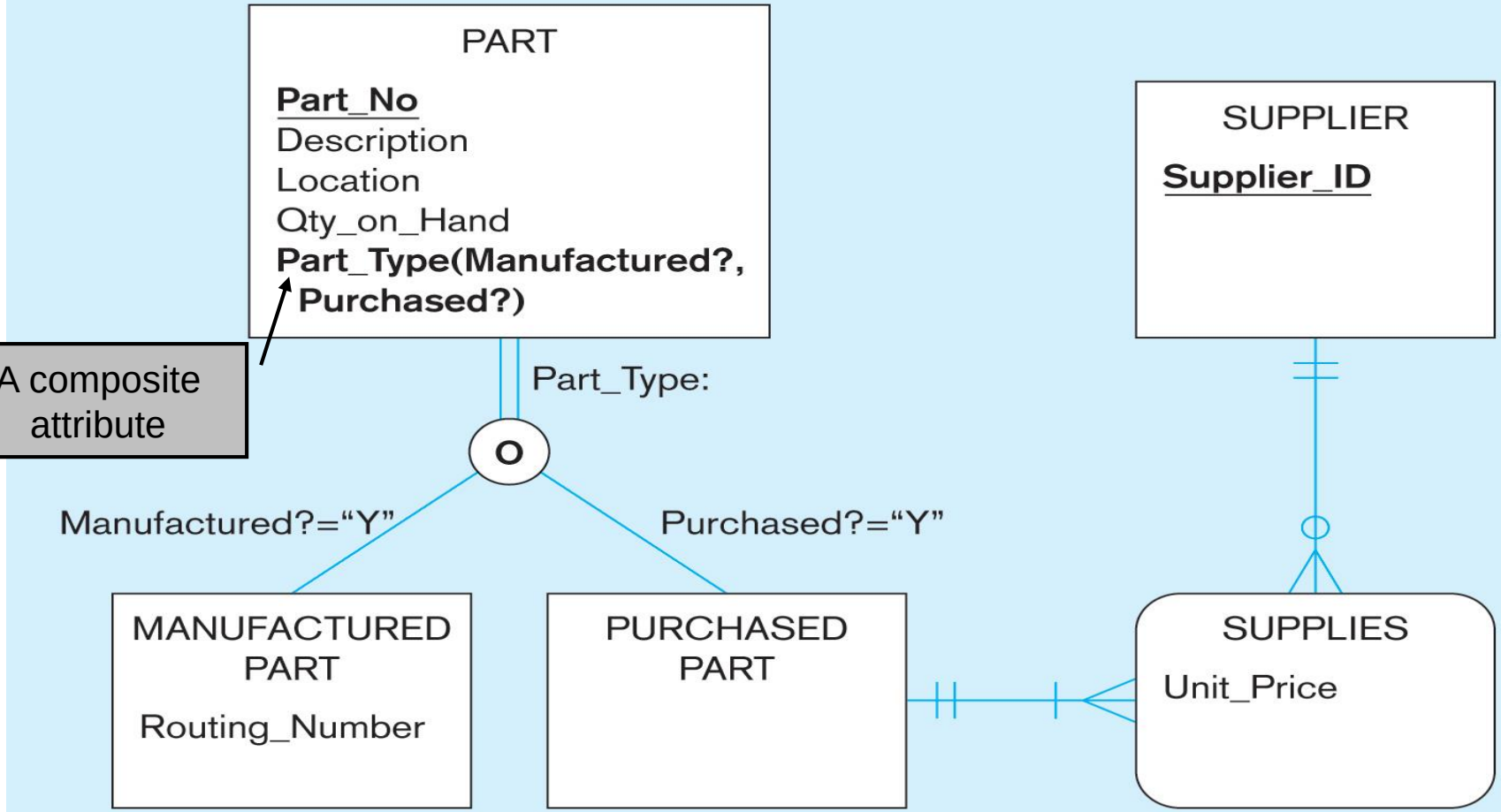
Subtype Discriminator

- An attribute of the supertype whose values determine the target subtype(s):
  - Disjoint – a *simple* attribute with alternative values to indicate the possible subtypes.
  - Overlapping – a *composite* attribute whose subparts pertain to different subtypes. Each subpart contains a boolean value to indicate whether or not the instance belongs to the associated subtype.

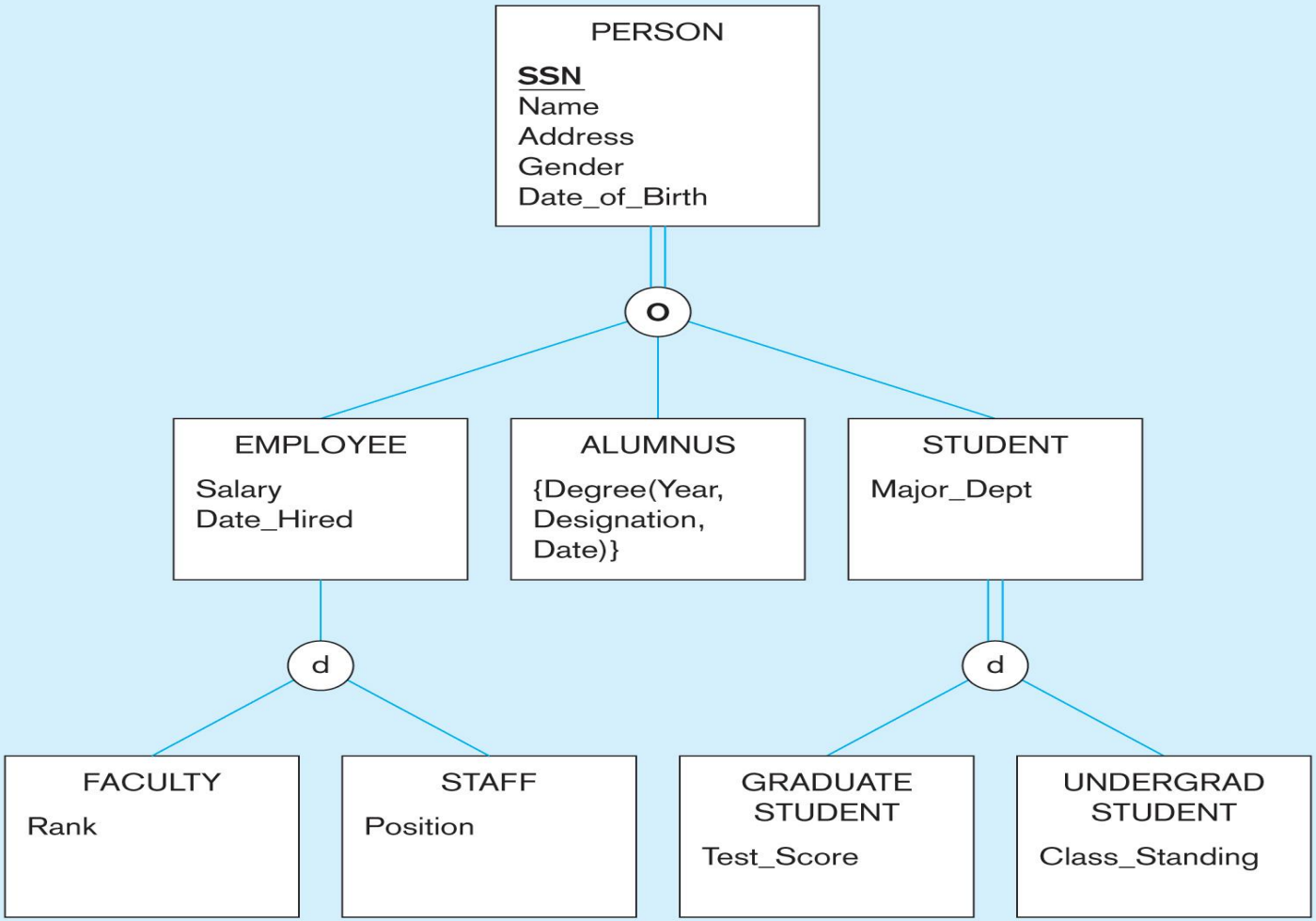
Defining Subtype Discriminators Disioint



Defining Subtype Discriminators Overlapping



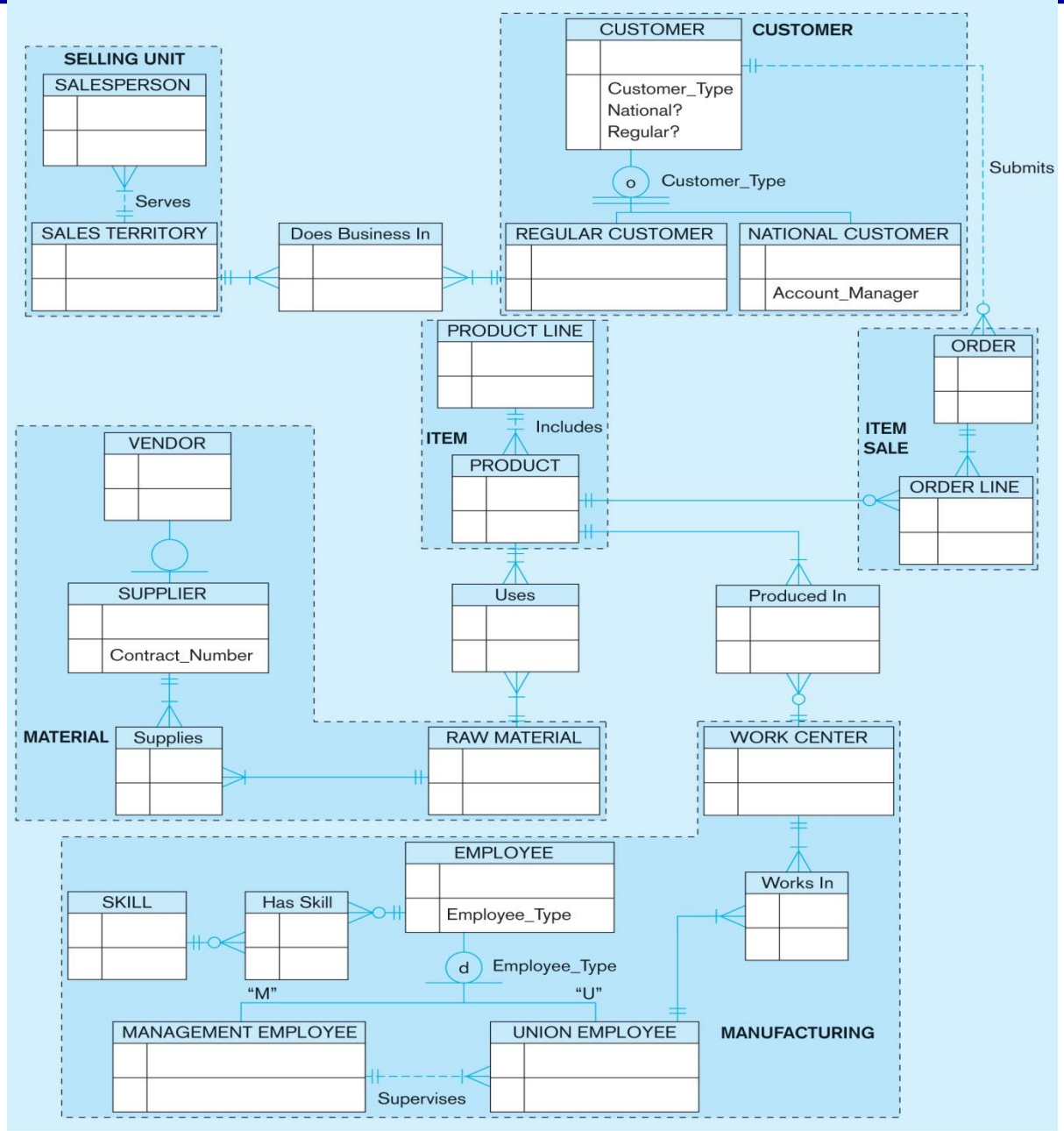
Supertype/Subtype Hierarchies



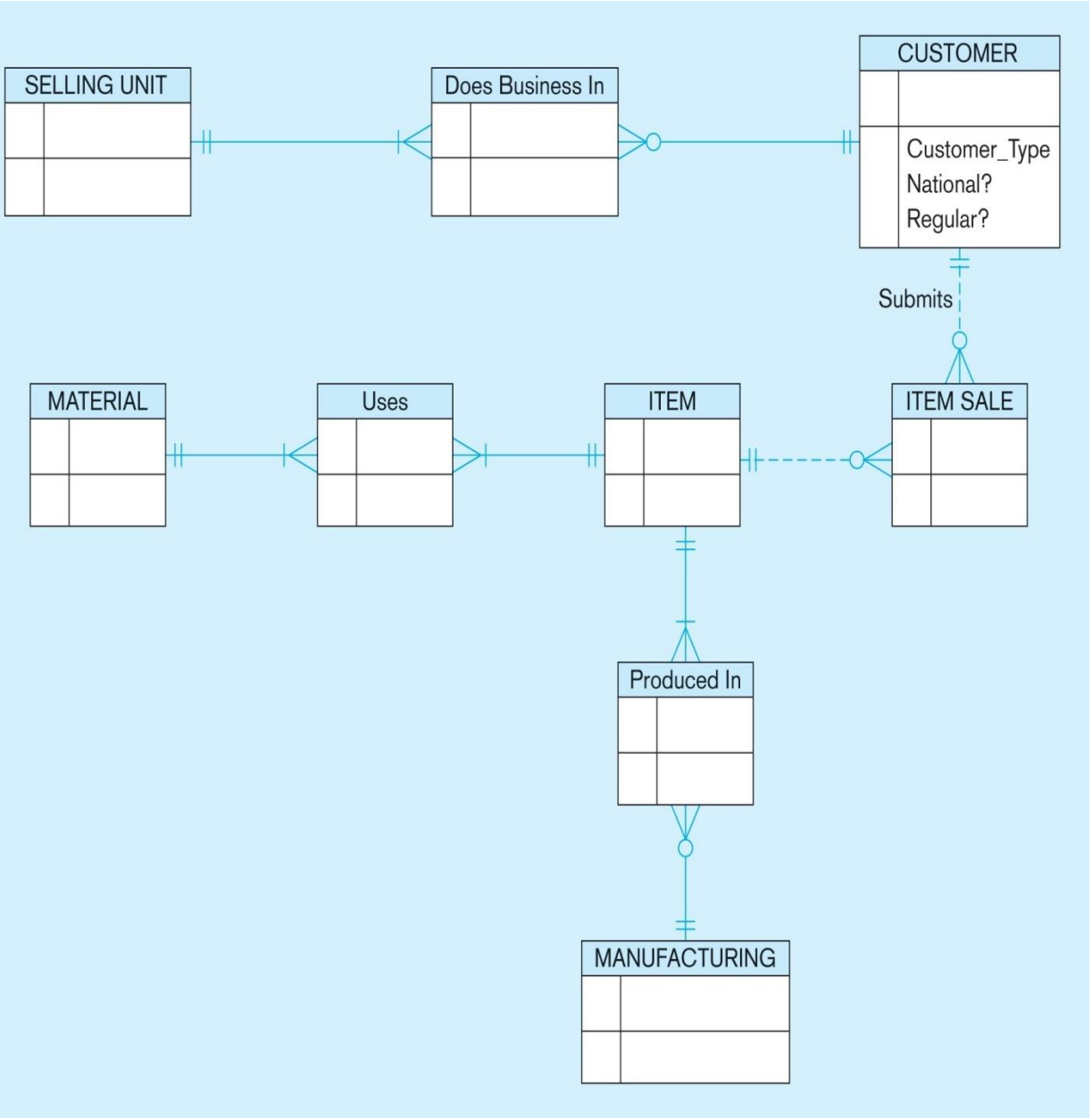
Entity Clusters

- EER diagrams are difficult to read when there are too many entities and relationships.
- Solution: group entities and relationships into entity clusters.
- Entity cluster: set of one or more entity types and associated relationships grouped into a single abstract entity type.

Entity Clusters



Entity Clusters



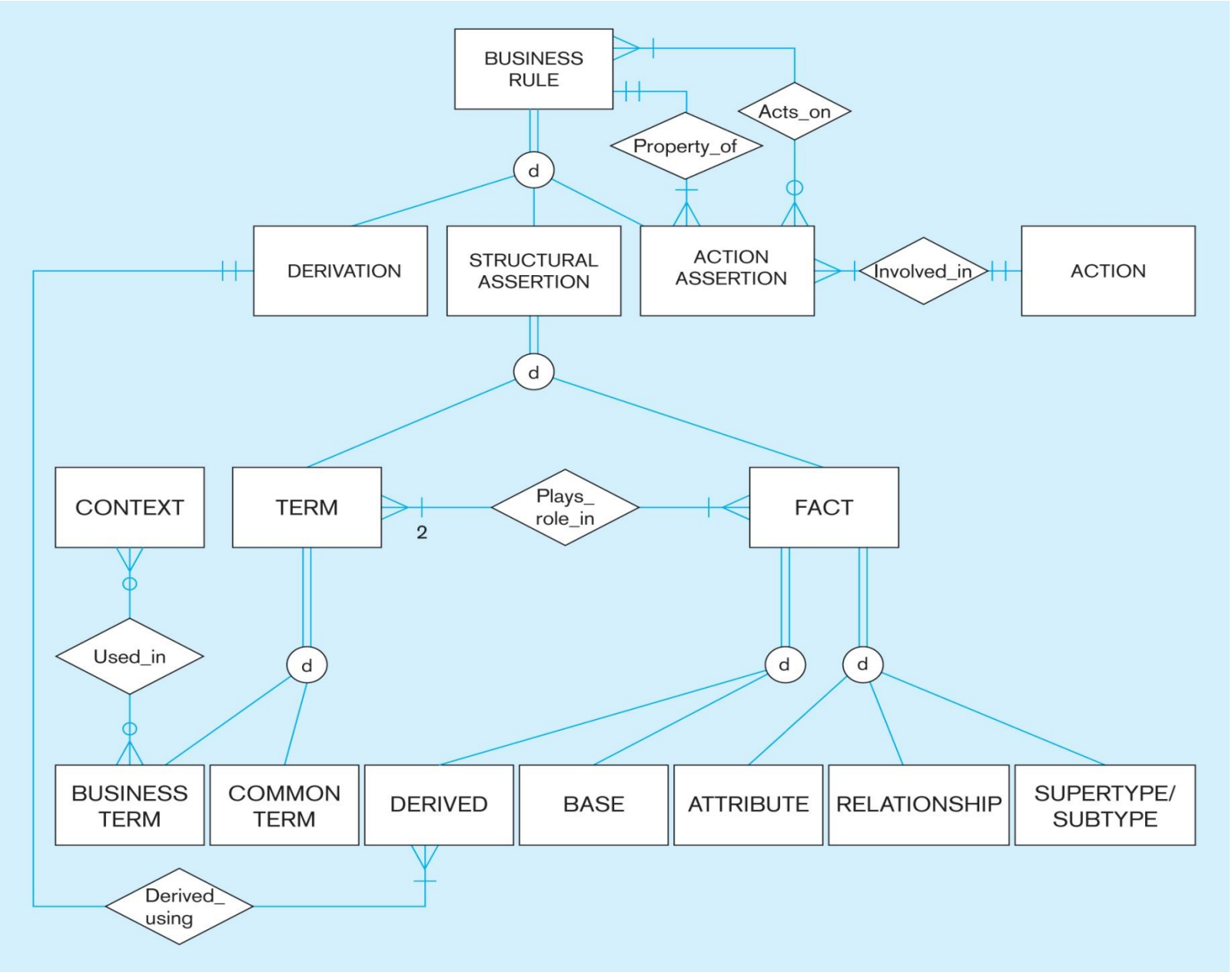
More On Business Rules

- ER and EER diagrams can be used to express many different types of business rules, such as participation constraints in supertype/subtype relationships as well as many others we have seen.
- However, there are many other types of business rules that cannot be expressed by these models.
- We need some mechanism for incorporating these rules into our conceptual model.

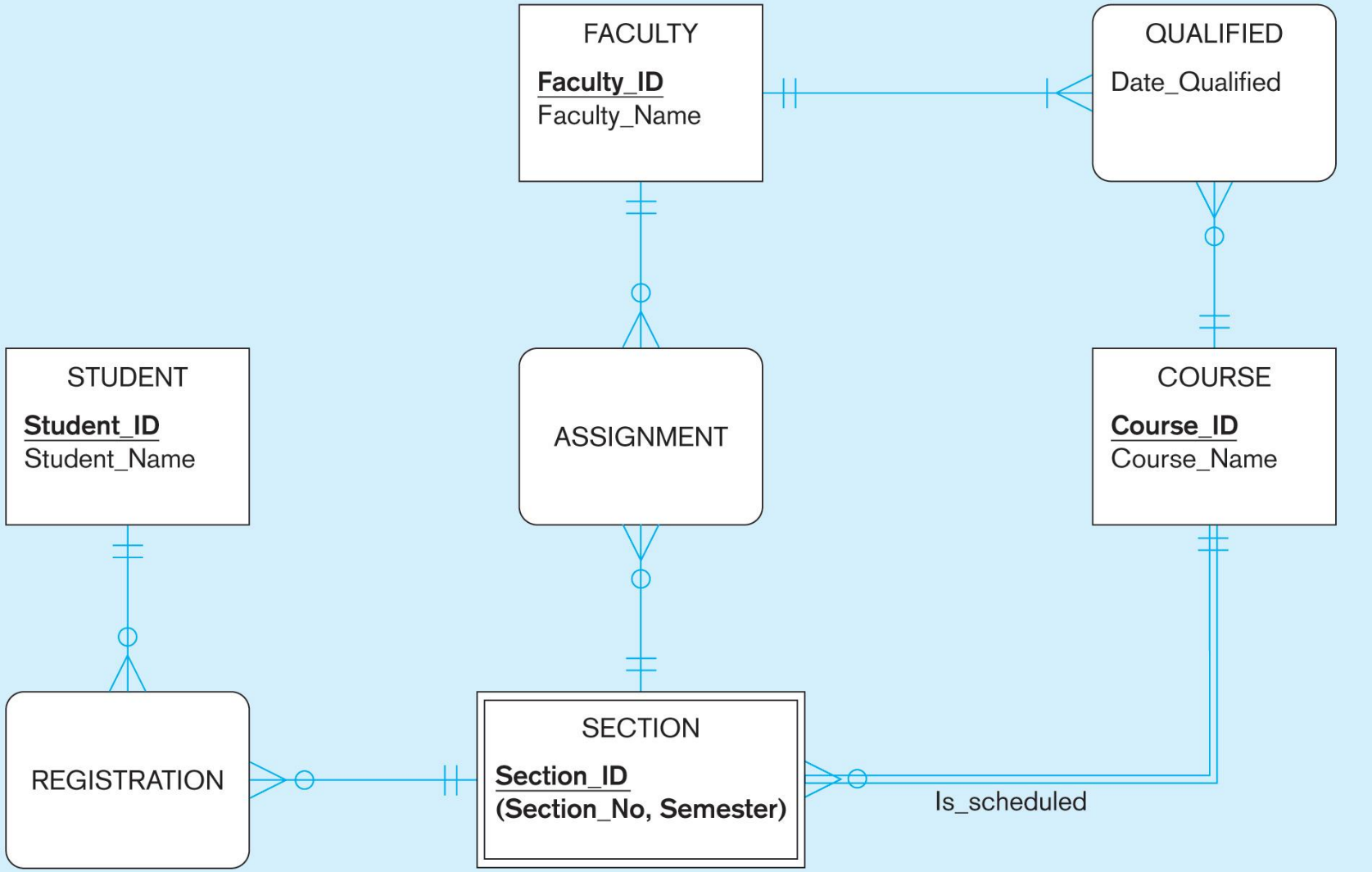
Types Of Business Rules

- There are three main types of business rules: derivation, structural assertion, and action assertion.

1. A derivation is a statement derived from other knowledge in the business.
   - Commonly result from mathematical or logic inference.
2. A structural assertion is a statement that expresses some aspect of the static structure of the organization. Either a term or a fact.
3. An action assertion is a statement of a constraint or a control on the actions of the organization.
   - A property of some business rule and states under what conditions a particular action can be performed on which business rules.


BUSINESS RULE
Acts_on
Property_of
d
DERIVATION
STRUCTURAL ASSERTION
ACTION ASSERTION
Involved_in
ACTION
d
CONTEXT
TERM
2
Plays_ role_in
FACT
Used_in
d
d
d
BUSINESS TERM
COMMON TERM
DERIVED
BASE
ATTRIBUTE
RELATIONSHIP
SUPERTYPE/ SUBTYPE
Derived_ using

ER Diagram Example For Business Rule Modeling



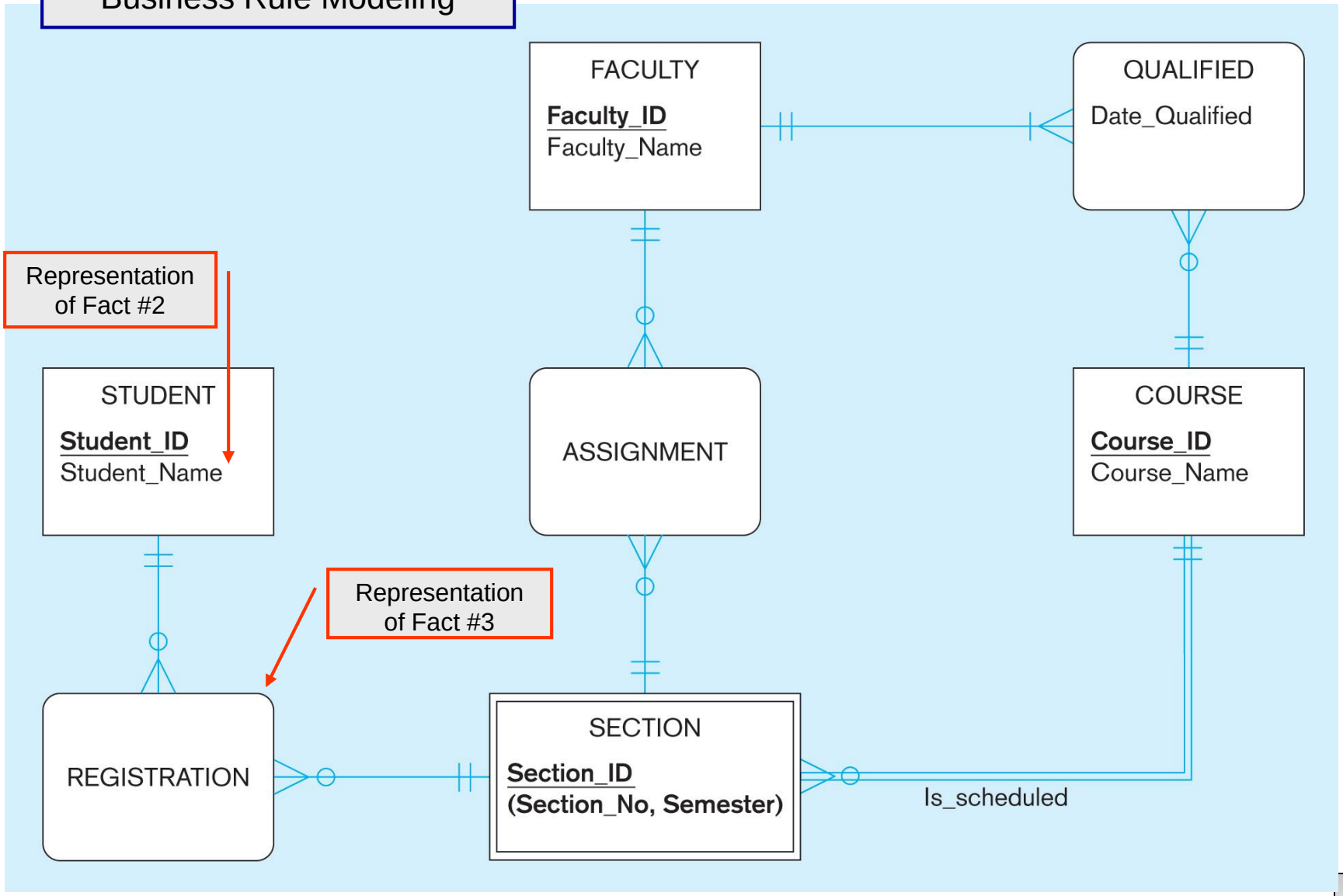
Stating a Structural Assertion

- A structural assertion states that something of importance to the organization either exists or exists in relationship with other things of interest to the organization.
- Can be as simple as the definition of a term or a fact. (A fact is a statement of a relationship between terms.)

Examples of Facts

1. A course is a module of instruction in a particular subject area.
   - This is a *definition of the term* course, associates two terms: module of instruction and subject area.
2. Student name is an attribute of student.
3. A student may register for many sections, and a section may be registered for by many students. Both student and section are business terms that require definitions.

ER Diagram Example For Business Rule Modeling



Stating a Structural Assertion: Derived Facts

- The facts we just defined are called base facts.
- Base facts are fundamental facts that cannot be derived from other terms or facts.
- A derived fact is a fact that is derived from business rules using some algorithm or inference.
  - A derived attribute (recall this from our earlier look at the ER model) is just one type of derived fact.
  - Examples:
    - A student enrolled in CGS 2545 during Fall 2011 will have Mark Llewellyn as an instructor. This fact can be derived from the fact that Mark Llewellyn is the instructor for CGS 2545 during Fall 2011.
    - Your GPA at UCF. How is this determined? It is derived using an algorithm which involves a number of different facts about you.

Stating an Action Assertion

- Action assertions deal with the dynamic structure of an organization. (Structural assertions deal with the static structure of an organization.)
- Action assertions impose "must" ("must not") and "should" ("should not") constraints on handling data.
- An action assertion is the property of some business rule (called the anchor object); for a data handling action (e.g. create, update, delete, or read), it states how other business rules (called the corresponding objects) act on the anchor object.

Stating an Action Assertion

Examples

1. A student (anchor object) must have a GPA (corresponding object) of 2.0 or greater to graduate (action). In this case, the anchor object is a structural assertion, but it is also possible for the anchor object to be another action assertion.

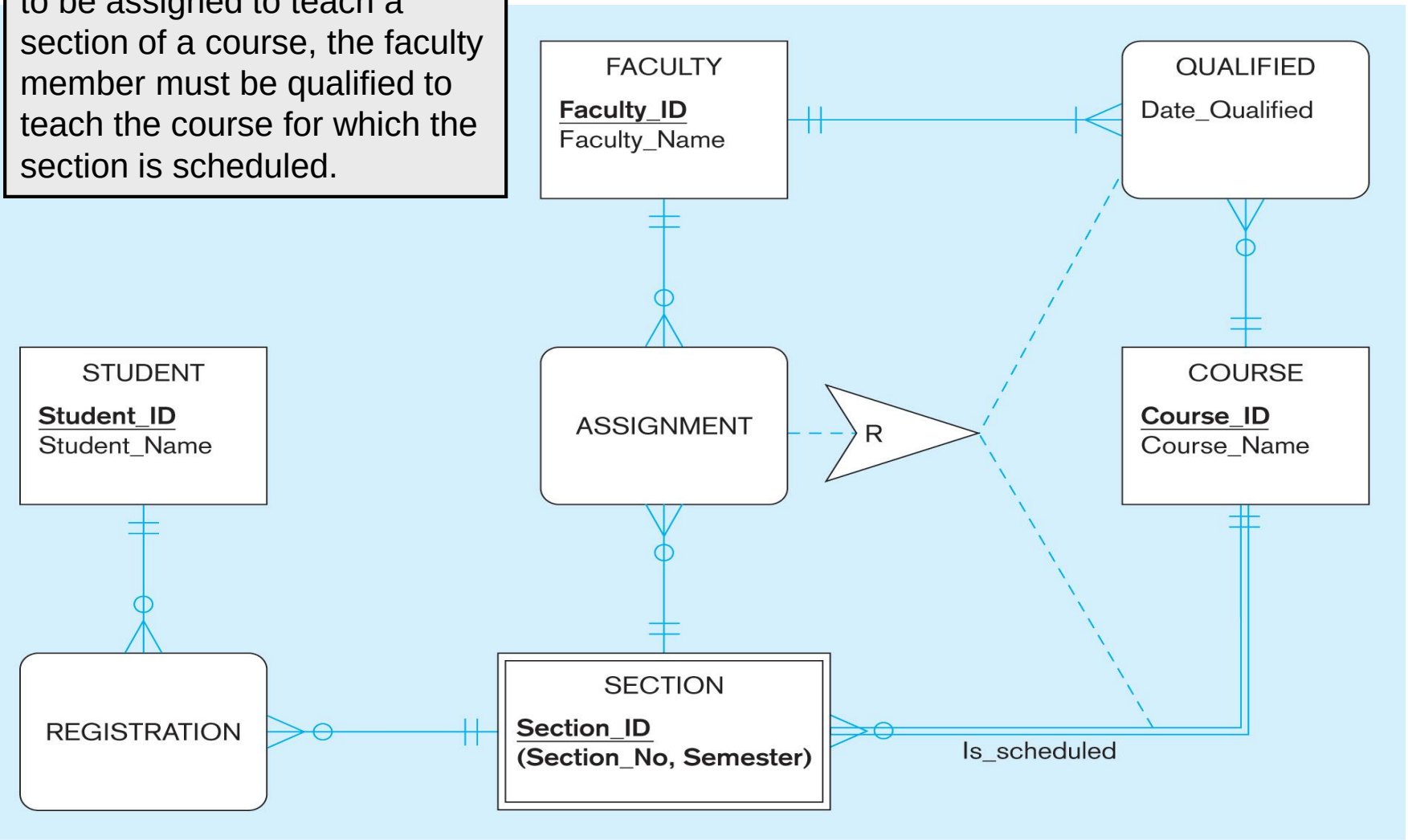
2. A student cannot register for (the anchor object is the Registration associative entity) a section of a course for which there is no qualified faculty (the corresponding object is the Qualified associative entity).

3. A course (anchor object) must have a course name (corresponding object). In this case, the action is updating the course name property of a course.

Representing and Enforcing Business Rules

- Most organizations have hundreds or thousands of business rules similar to the ones we've just seen.
- Action assertions have traditionally been implemented in procedural logic buried deep within individual application programs in a form which is virtually unrecognizable, unmanageable, and inconsistent.
  - A heavy burden is placed on the programmer to know all of the constraints and to properly implement them. Mistakes, omissions, and misunderstandings will potentially leave the database in an inconsistent state.
- The more modern approach is to declare action assertions at a conceptual level without specifying how they are to be implemented.

An ERD showing the business rule that for a faculty member to be assigned to teach a section of a course, the faculty member must be qualified to teach the course for which the section is scheduled.



An ERD showing the business rule that for a faculty member to be assigned to teach a section of a course, the faculty member must not be assigned to teach a total or more than three course sections.