Modern Systems Analysis and Design

Sixth Edition

Jeffrey A. Hoffer
Joey F. George
Joseph S. Valacich

Chapter 8
Structuring System Data Requirements

Learning Objectives

- ✓ Concisely define each of the following key data modeling terms: entity type, attribute, multivalued attribute, relationship, degree, cardinality, business rule, associative entity, trigger, supertype, subtype.
- ✓ Draw an entity-relationship (E-R) diagram to represent common business situations.
- ✓ Explain the role of conceptual data modeling in the overall analysis and design of an information system.

Learning Objectives (Cont.)

- ✓ Distinguish between unary, binary, and ternary relationships and give an example of each.
- ✓ Define four basic types of business rules in a conceptual data model.
- ✓ Relate data modeling to process and logic modeling as different views of describing an information system.

Conceptual Data Modeling

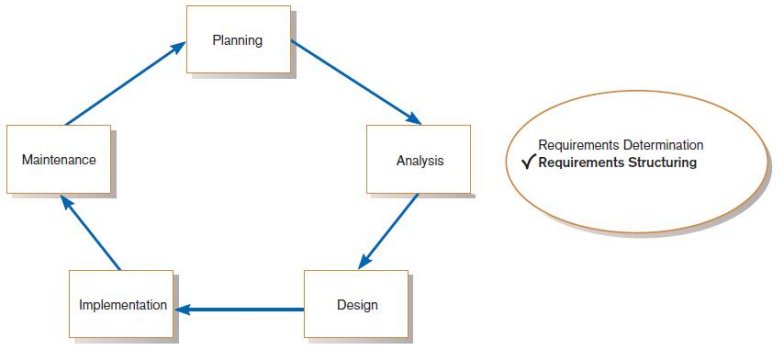


FIGURE 8-1
Systems development life cycle with analysis phase highlighted

"Conceptual" Data Modeling

- **Conceptual data modeling**: a detailed model that captures the overall structure of data in an organization
  - Independent of any database management system (DBMS) or other implementation considerations

Data Modeling Process

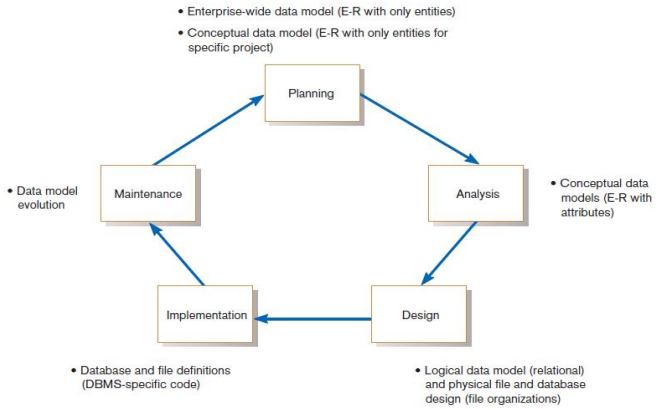


FIGURE 8-2
Relationship between data modeling and the systems development life cycle

Deliverables and Outcome

- Primary deliverable is an entity-relationship (E-R) diagram or class diagram.
- A set of entries about data objects to be stored in repository or project dictionary.
  - Each data store in a process model must relate to business objects represented in the data model.

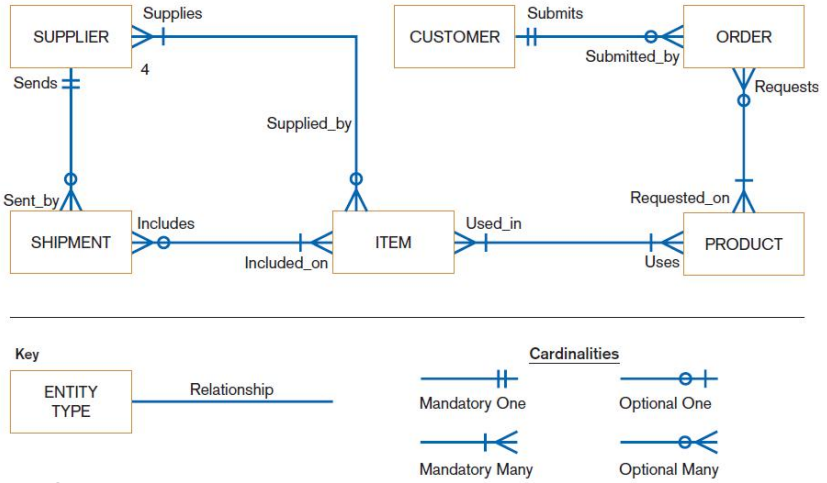


FIGURE 8-3
Sample conceptual data model

Gathering Information for Conceptual Data Modeling

- Two perspectives on data modeling:
  - *Top-down approach* for a data model is derived from an intimate understanding of the business.
  - *Bottom-up approach* for a data model is derived by reviewing specifications and business documents.

Gathering Information

TABLE 8-1 Requirements Determination Questions for Data Modeling

1. *What are the subjects/objects of the business?* What types of people, places, things, materials, events, etc., are used or interact in this business, about which data must be maintained? How many instances of each object might exist?—**data entities and their descriptions**
2. *What unique characteristic (or characteristics) distinguishes each object from other objects of the same type?* Might this distinguishing feature change over time or is it permanent? Might this characteristic of an object be missing even though we know the object exists?—**primary key**
3. *What characteristics describe each object?* On what basis are objects referenced, selected, qualified, sorted, and categorized? What must we know about each object in order to run the business?—**attributes and secondary keys**
4. *How do you use these data?* That is, are you the source of the data for the organization, do you refer to the data, do you modify it, and do you destroy it? Who is not permitted to use these data? Who is responsible for establishing legitimate values for these data?—**security controls and understanding who really knows the meaning of data**
5. *Over what period of time are you interested in these data?* Do you need historical trends, current "snapshot" values, and/or estimates or projections? If a characteristic of an object changes over time, must you know the obsolete values?—**cardinality and time dimensions of data**
6. *Are all instances of each object the same?* That is, are there special kinds of each object that are described or handled differently by the organization? Are some objects summaries or combinations of more detailed objects?—**supertypes, subtypes, and aggregations**
7. *What events occur that imply associations among various objects?* What natural activities or transactions of the business involve handling data about several objects of the same or a different type?—**relationships, and their cardinality and degree**
8. *Is each activity or event always handled the same way or are there special circumstances?* Can an event occur with only some of the associated objects, or must all objects be involved? Can the associations between objects change over time (for example, employees change departments)? Are values for data characteristics limited in any way?—**integrity rules, minimum and maximum cardinality, time dimensions of data**

E-R Model Fundamental

- **Entity-Relationship data model** (**E-R model**): a detailed, logical representation of the entities, associations and data elements for an organization or business area
- **Entity-relationship diagram** (**E-R diagram**): a graphical representation of an E-R model

Entity, Type, and Instance

- **Entity**: a person, place, object, event or concept in the user environment about which data is to be maintained
- **Entity type**: collection of entities that share common properties or characteristics
- **Entity instance**: single occurrence of an entity type

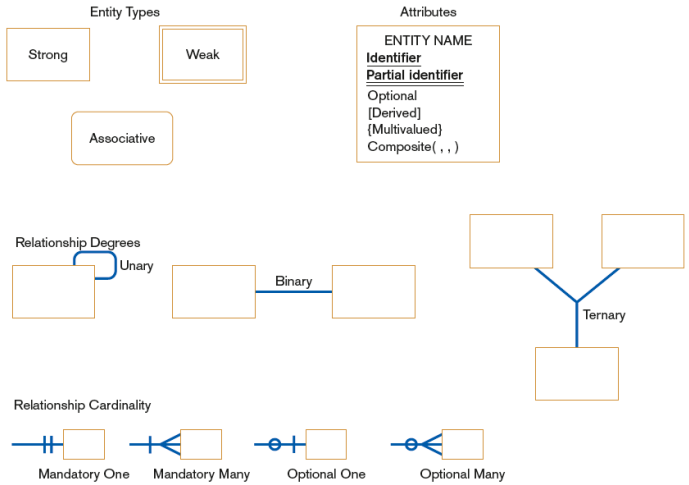


FIGURE 8-5 Basic E-R notation

Naming and Defining Entity Types

- An entity type definition should:
  - Include a statement of *what the unique characteristic(s) is (are) for each instance*.
  - Make clear *what entity instances are included and not included* in the entity type.
  - Often include a description of *when an instance of the entity type is created or deleted*.

Attributes

- **Attribute**: a named property or characteristic of an entity that is of interest to the organization
  - Naming an attribute: i.e. Vehicle_ID
  - Place its name inside the rectangle for the associated entity in the E-R diagram.

Naming and Defining Attributes

- An attribute name is a *noun* and should be *unique*.
- To make an attribute name unique and for clarity, *each attribute name should follow a standard format*.
- *Similar attributes of different entity types should use similar but distinguishing names*.

Candidate Keys and Identifiers.

- **Candidate key**: an attribute (or combination of attributes) that uniquely identifies each instance of an entity type
- **Identifier**: a candidate key that has been selected as the unique, identifying characteristic for an entity type

Candidate Keys and Identifiers (Cont.)

- Selection rules for an identifier
  - Choose a candidate key that will not change its value.
  - Choose a candidate key that will never be null.
  - Avoid using intelligent keys.
  - Consider substituting single value surrogate keys for large composite keys.

Other Attribute Types

- **Multivalued attribute**: an attribute that may take on more than one value for each entity instance
- **Repeating group**: a set of two or more multivalued attributes that are logically related

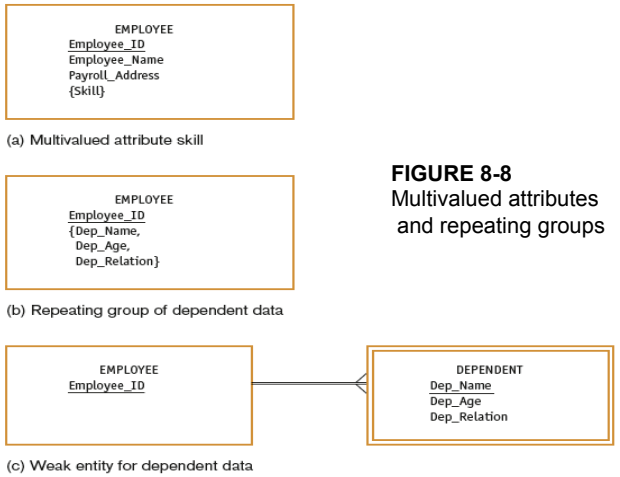


FIGURE 8-8
Multivalued attributes and repeating groups

Other Attribute Types

- **Required attribute**: an attribute that must have a value for every entity instance
- **Optional attribute**: an attribute that may not have a value for every entity instance
- **Composite attribute**: an attribute that has meaningful component parts
- **Derived attribute**: an attribute whose value can be computed from related attribute values

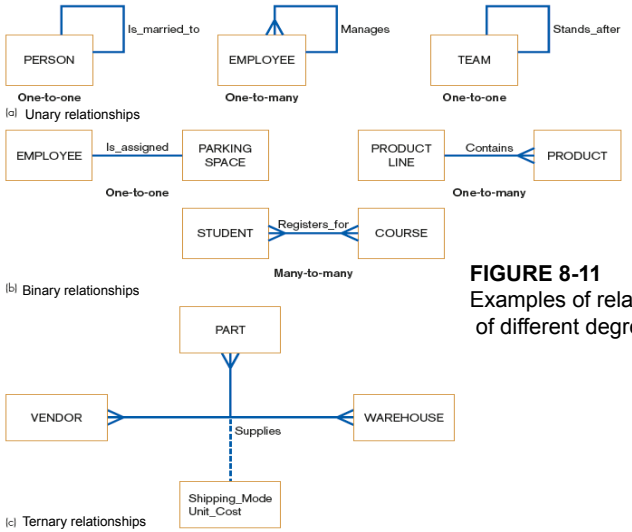


FIGURE 8-11
Examples of relationships of different degrees

Relationships

- **Relationship**: an association between the instances of one or more entity types that is of interest to the organization
- **Degree**: the number of entity types that participate in a relationship

Conceptual Data Modeling and the E-R Model

- **Unary relationship**: a relationship between the instances of one entity type
  - Also called a *recursive relationship*
- **Binary relationship**: a relationship between instances of two entity types
  - Most common type of relationship encountered in data modeling
- **Ternary relationship**: a simultaneous relationship among instances of three entity types

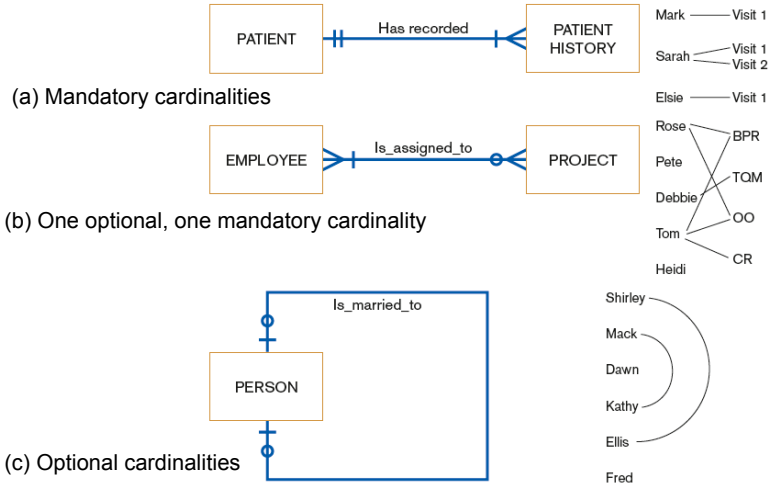
Cardinalities in Relationships

- **Cardinality**: the number of instances of entity B that can (or must) be associated with each instance of entity A
- Minimum Cardinality
  - The minimum number of instances of entity B that may be associated with each instance of entity A
- Maximum Cardinality
  - The maximum number of instances of entity B that may be associated with each instance of entity A

Cardinalities in Relationships (Cont.)

- Mandatory vs. Optional Cardinalities
  - Specifies whether an instance must exist or can be absent in the relationship.

FIGURE 8-14 Examples of cardinality constraints



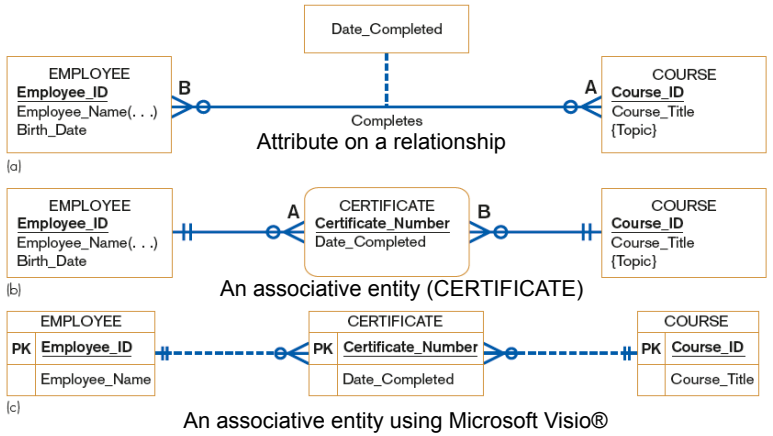
Naming and Defining Relationships

- A relationship name is a verb phrase; avoid vague names.
- A relationship definition:
  - Explains what action is to be taken and possibly why it is important.
  - Gives examples to clarify the action.

Associative Entities

- **Associative Entity**: 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
  - Sometimes called a gerund
- The data modeler chooses to model the relationship as an entity type.

FIGURE 8-15 An associative entity



Summary of Conceptual Data Modeling with E-R Diagrams

- The purpose of E-R diagramming is to capture the richest possible understanding of the meaning of the data necessary for an information system or organization.

Representing Supertypes and Subtypes

- **Subtype**: a subgrouping of the entities in an entity type
  - Is meaningful to the organization
  - Shares common attributes or relationships distinct from other subgroupings
- **Supertype**: a generic entity type that has a relationship with one or more subtypes

Representing Supertypes and Subtypes (Cont.)

- Business Rules for Supertype/subtype Relationships:
  - **Total specialization** specifies that each entity instance of the supertype must be a member of some subtype in the relationship.
  - **Partial specialization** specifies that an entity instance of the supertype does not have to belong to any subtype, and may or may not be an instance of one of the subtypes.

Representing Supertypes and Subtypes (Cont.)

  - **Disjoint rule** specifies that if an entity instance of the supertype is a member of one subtype, it cannot simultaneously be a member of any other subtype.
  - **Overlap rule** specifies that an entity instance can simultaneously be a member of two (or more) subtypes.

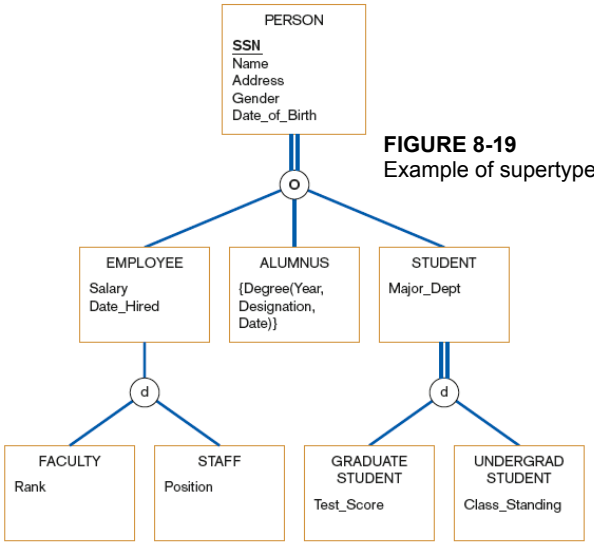


FIGURE 8-19
Example of supertype/subtype hierarchy

Business Rules

- **Business rules**: specifications that preserve the integrity of the logical data model
  - Captured during requirements determination
  - Stored in CASE repository as they are documented

Business Rules (Cont.)

- Four basic types of business rules are:
  - *Entity integrity*: unique, non-null identifiers
  - *Referential integrity constraints*: rules governing relationships between entity types
  - *Domains*: constraints on valid values for attributes
  - *Triggering operations*: other business rules that protect the validity of attribute values

Triggering Operations

- Includes the following components:
  - *User rule*: statement of the business rule to be enforced by the trigger
  - *Event*: data manipulation operation that initiates the operation
  - *Entity Name*: name of entity being accessed or modified
  - *Condition*: condition that causes the operation to be triggered
  - *Action*: action taken when the operation is triggered

Summary

- In this chapter you learned how to:
  - ✓ Concisely define each of the following key data modeling terms: entity type, attribute, multivalued attribute, relationship, degree, cardinality, business rule, associative entity, trigger, supertype, subtype.
  - ✓ Draw an entity-relationship (E-R) diagram to represent common business situations.
  - ✓ Explain the role of conceptual data modeling in the overall analysis and design of an information system.

Summary (Cont.)

- In this chapter you learned how to:
  - ✓ Distinguish between unary, binary, and ternary relationships and give an example of each.
  - ✓ Define four basic types of business rules in a conceptual data model.
  - ✓ Relate data modeling to process and logic modeling as different views of describing an information system.

Modern Systems Analysis and Design

Sixth Edition

Jeffrey A. Hoffer
Joey F. George
Joseph S. Valacich

Chapter 8 Appendix
Object-Oriented Analysis and Design: Object Modeling – Class Diagrams

Learning Objectives

- ✓ Concisely define each of the following key data modeling terms: object, state, behavior, object class, class diagram, operation, encapsulation, association role, abstract class, polymorphism, aggregation, and composition.
- ✓ Draw a class diagram to represent common business situations.
- ✓ Explain the unique capabilities of class diagrams compared with E-R diagrams for modeling data.

Representing Objects and Classes

- **Object**: an entity with a well-defined role in an application domain, and has state, behavior, and identity characteristics
- **State**: encompasses an object' s properties (attributes and relationships) and the values of those properties

Representing Objects and Classes (Cont.)

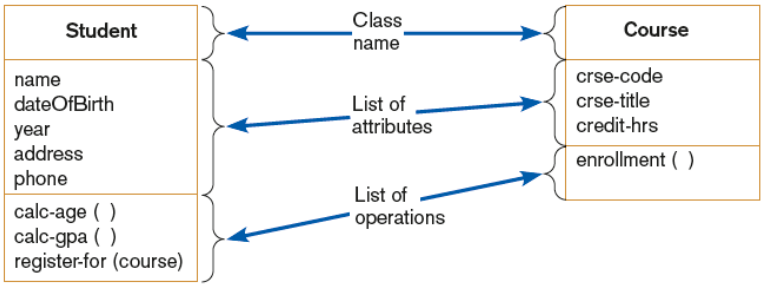
- **Behavior**: represents how an object acts and reacts
- **Identity**: uniqueness, no two objects are the same
- **Object class** (**class**): a logical grouping of objects that have the same (or similar) attributes, relationships, and behaviors (methods)

Representing Objects and Classes (Cont.)

- **Class diagram**: a diagram that shows the static structure of object classes, their internal structure, and the relationships in which they participate
- UML classes are analogous to E-R entities.

Representing Objects and Classes (Cont.)

Figure 8-26 UML class diagram showing two classes



Representing Objects and Classes (Cont.)

- **Operation**: a function or a service that is provided by all the instances of a class to invoke behavior in an object by passing a message
- **Encapsulation**: the technique of hiding the internal implementation details of an object from its external view

Representing Associations

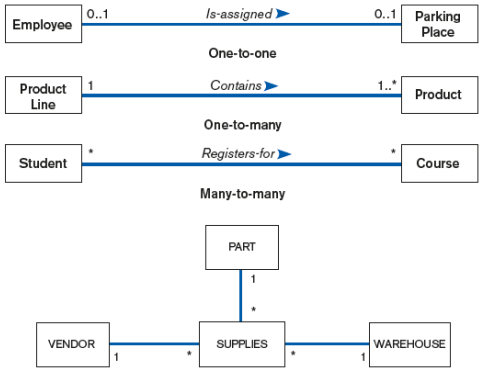
- **Association**: a relationship among instances of object classes
- **Association role**: the name given to the end of an association where it connects to a class

Representing Associations

- **Multiplicity**: indicates how many objects participate in a given relationship:
  - 0..10 means minimum of 0 and maximum of 10
  - 1, 2 means can be either 1 or 2
  - * means any number
- UML associations are analogous to E-R relationships and UML multiplicities are analogous to E-R cardinalities.

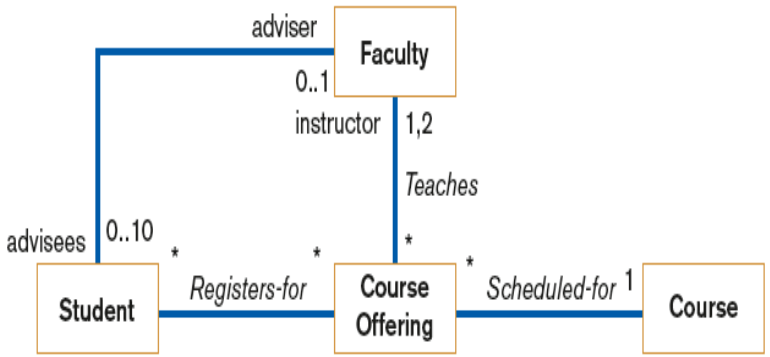
Representing Associations (Cont.)

Figure 8-27 Examples of association relationships of different degrees



Representing Associations (Cont.)

Figure 8-28 Examples of binary associations

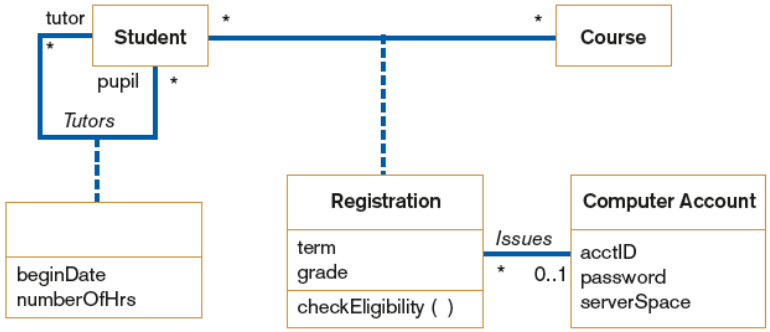


Representing Associative Classes

- **Associative class**: an association that has attributes or operations of its own or that participates in relationships with other classes
- UML association classes are analogous to E-R associative entities.
- Generalization and inheritance implemented via superclass/subclasses in UML, supertypes/ subtypes in E-R.

Representing Associative Classes (Cont.)

Figure 8-29 Class diagram showing associative classes



Representing Stereotypes for Attributes

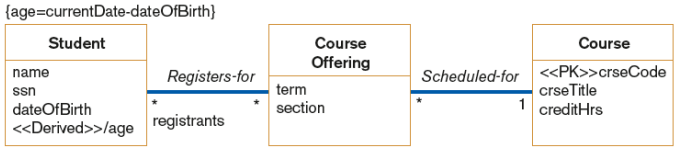


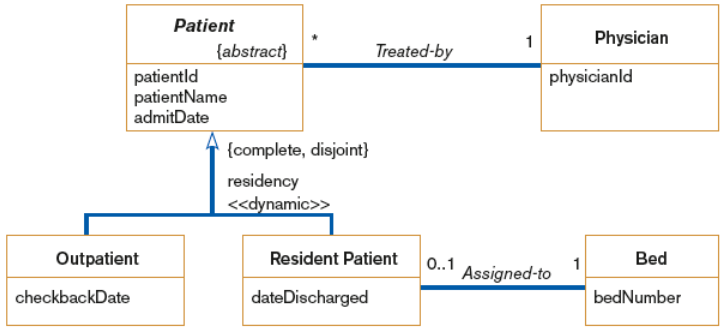
Figure 8-31 Stereotypes

Representing Generalization

- **Abstract class**: a class that has no direct instances but whose descendants may have direct instances
- **Concrete class**: a class that can have direct instances

Representing Generalization (Cont.)

Figure 8-32 Example of generalizations, inheritance, and constraints



Representing Generalization (Cont.)

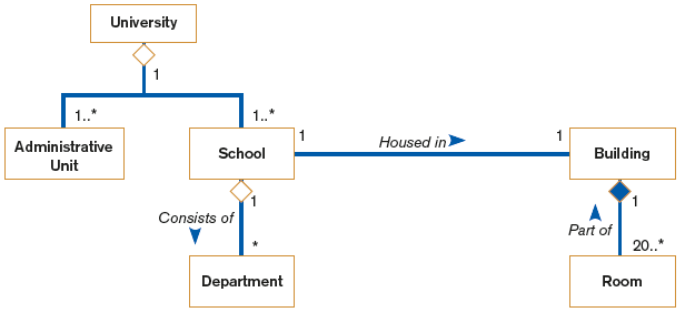
- **Abstract operation**: defines the form or protocol of the operation, but not its implementation
- **Method**: the implementation of an operation
- **Polymorphism**: the same operation may apply to two or more classes in different ways

Representing Aggregation

- **Aggregation**: a part-of relationship between a component object and an aggregate object
  - Represented with open diamonds
- **Composition**: a part object that belongs to only one whole object and that lives and dies with the whole
  - Represented with filled diamonds

Aggregation and Composition (Cont.)

Figure 8-34 Aggregation and composition



An Example of Conceptual Data Modeling at Hoosier Burger

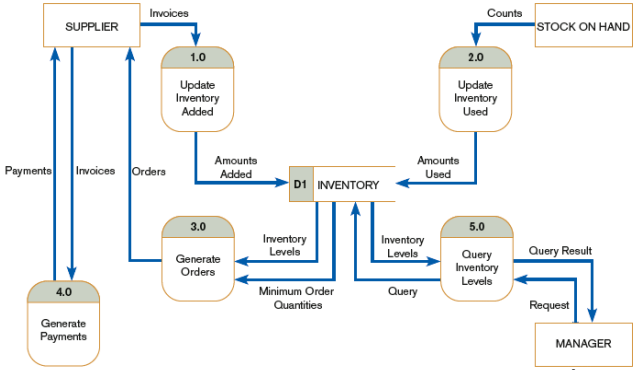


Figure 8-35 Level-0 data flow diagram for Hoosier Burger's new logical inventory control system

An Example of Conceptual Data Modeling at Hoosier Burger (Cont.)

Conditions/ Courses of Action	Rules						
	1	2	3	4	5	6	7
Type of item	P	P	P	P	P	P	N
Time of week	D	W	D	W	D	W	–
Season of year	A	A	S	S	H	H	–
Standing daily order	X		X		X		
Standing weekend order		X		X		X	
Minimum order quantity							X
Holiday reduction					X	X	
Summer reduction			X	X			

Figure 8-36 Reduced decision table for Hoosier Burger's inventory reordering

An Example of Conceptual Data Modeling at Hoosier Burger (Cont.)

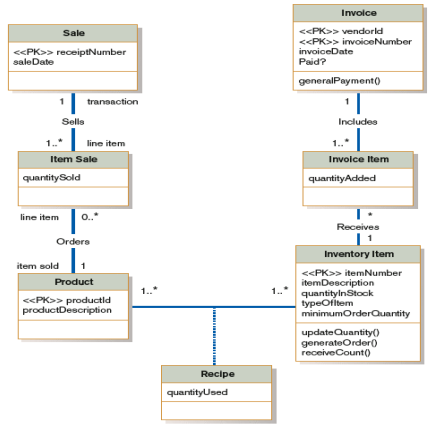


Figure 8-38 Final class diagram for Hoosier Burger's inventory control system

Summary

- In this appendix you learned how to:
- ✓ Concisely define each of the following key data modeling terms: object, state, behavior, object class, class diagram, operation, encapsulation, association role, abstract class, polymorphism, aggregation, and composition.
- ✓ Draw a class diagram to represent common business situations.
- ✓ Explain the unique capabilities of class diagrams compared with E-R diagrams for modeling data.