Object Oriented Databases

Database Systems:
Design, Implementation, and Management,

Seema Sirpal
Delhi University Computer Centre

You will learn:

- What basic concepts govern OO systems
- How OO features are related to the more traditional relational and ER models
- What the basic features of an OO database management system (OODBMS) are
- What effect OO concepts are likely to have on data modeling and design

2

You will learn: (continued)

- About the advantages and disadvantages of OODBMSs
- How OO concepts have influenced the relational model
- How Oracle*9i* implements object extensions

3

Object Orientation and Its Benefits

- A set of design and development principles based on conceptually autonomous computer structures known as objects
- Each object represents a real-world entity with the ability to act upon itself and interact with other objects
- Modularity is therefore almost inevitable

4

Object Orientation Contributions

TABLE 11.1 OBJECT ORIENTATION CONTRIBUTIONS

COMPUTER-RELATED AREA	OO CONTRIBUTIONS
Programming languages	Reduces the number of lines of code Decreases development time Enhances code reusability Makes code maintenance easier Enhances programmer productivity
Graphical user interfaces (GUIs)	Enhances ability to create easy-to-use interfaces Improves system end-user friendliness Makes it easier to define standards
Databases	Supports abstract data types Supports complex objects Supports multimedia data types
Design	Captures more of the data model's semantics Represents the real world more accurately Supports complex data manipulations in specialized applications that target graphics, imaging, mapping, financial modeling, telecommunications, geospatial applications, medical applications, and so on
Operating systems	Enhances system portability by creating layers of abstractions to handle hardware-specific issues Facilitates system extensibility through the use of inheritance and other object oriented constructs

Key Benefits of Object Technology

- Easy maintenance of applications
- Saves money on maintenance
- Faster product development
- Gives company a competitive advantage
- Higher quality of code at a lower cost
- Leads to more satisfied customers
- Reusability of code
- Allows multiple programmers to share a code repository, which saves time and money

6

The Evolution of Object Oriented Concepts

- Object oriented programming
  - Developed as an alternative to traditional programming methods
  - Programmer creates or uses objects:
    - Self-contained, reusable modules that contain data as well as the procedures used to operate on such data

7

The Evolution of Object Oriented Concepts (continued)

- Object oriented programming languages were developed to:
  - Provide an easy-to-use software development environment
  - Provide a powerful software modeling tool for application development
  - Decrease development time by reducing the amount of code
  - Improve programmer productivity by making that code reusable

8

The Evolution of Object Oriented Concepts (continued)

- Object oriented environment has several important attributes:
  - Data set is no longer passive
  - Data and procedures are bound together, creating an object
  - Object has an innate ability to act on itself

9

Object Oriented Concepts

- Have their roots in programming languages
- No knowledge of programming is necessary to understand these concepts

10

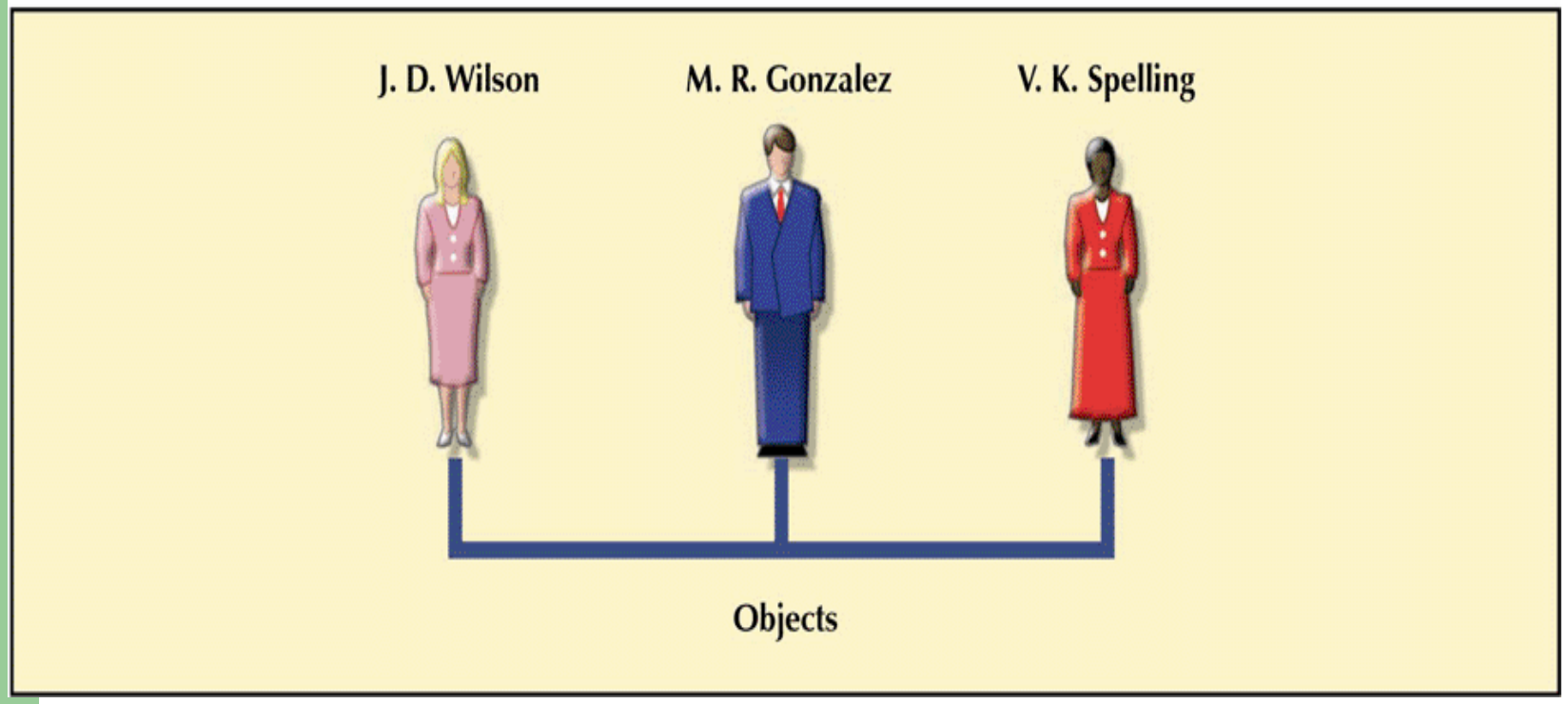
Objects: Components and Characteristics

- Object:
  - Abstract representation of a real-world entity
  - Has:
    - Unique identity
    - Embedded properties
    - Ability to interact with other objects and act upon itself
  - Defining characteristic is its unique identity

11

Real-World Student Objects

FIGURE 11.1 REAL-WORLD STUDENT OBJECTS



12

Object Identity

- Unique to that object
- Assigned by system at moment of object's creation
- Cannot be changed under any circumstances
- Can be deleted *only* if the object is deleted
- Can never be reused

13

Object Attributes

TABLE 11.2 OBJECT ATTRIBUTES

ATTRIBUTE NAME	ATTRIBUTE VALUE
SOCIAL_SECURITY_NUMBER	414-48-0944
FIRST_NAME	John
MIDDLE_INITIAL	D
LAST_NAME	Smith
DATE_OF_BIRTH	11/23/1966
MAJOR *	Accounting
SEMESTER_GPA	2.89
OVERALL_GPA	3.01

TABLE 11.2 OBJECT ATTRIBUTES (CONTINUED)

ATTRIBUTE NAME	ATTRIBUTE VALUE
COURSES_TAKEN *	ENG201;MATH243;HIST201;ACCT211;ECON210;ECON212;ACCT212; CIS220;ENG202;MATH301;HIST202;CIS310;ACCT343;ACCT345;ENG242; MKTG301;FIN331;ACCT355
ADVISOR*	Dr. W. R. Learned

* Represents an attribute that references one or more other objects

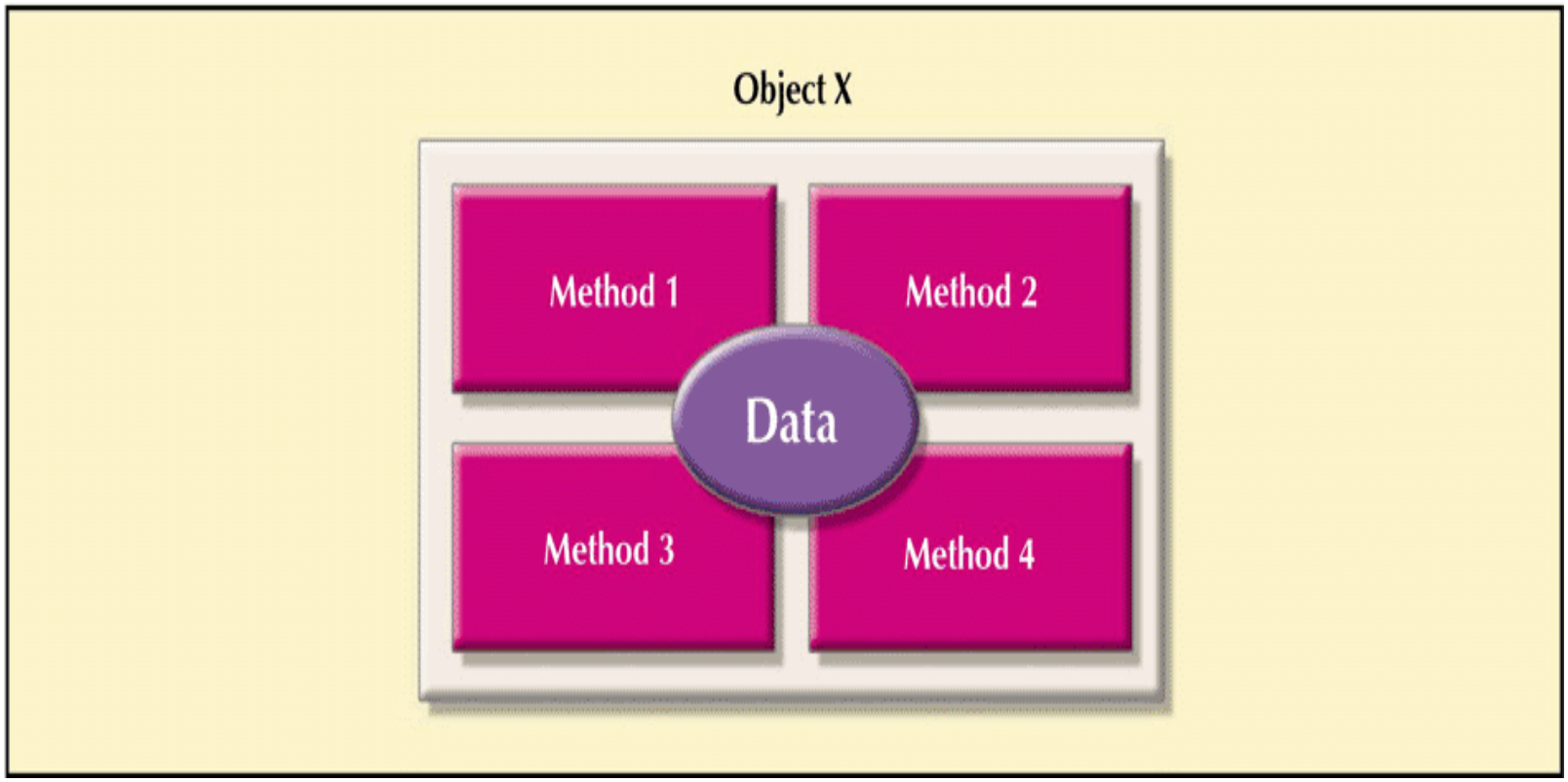
Methods

- Method:
  - Code that performs a specific operation on object's data
  - Protects data from direct and unauthorized access by other objects
  - Used to change the object's attribute values or to return the value of selected object attributes
  - Represent real-world actions

15

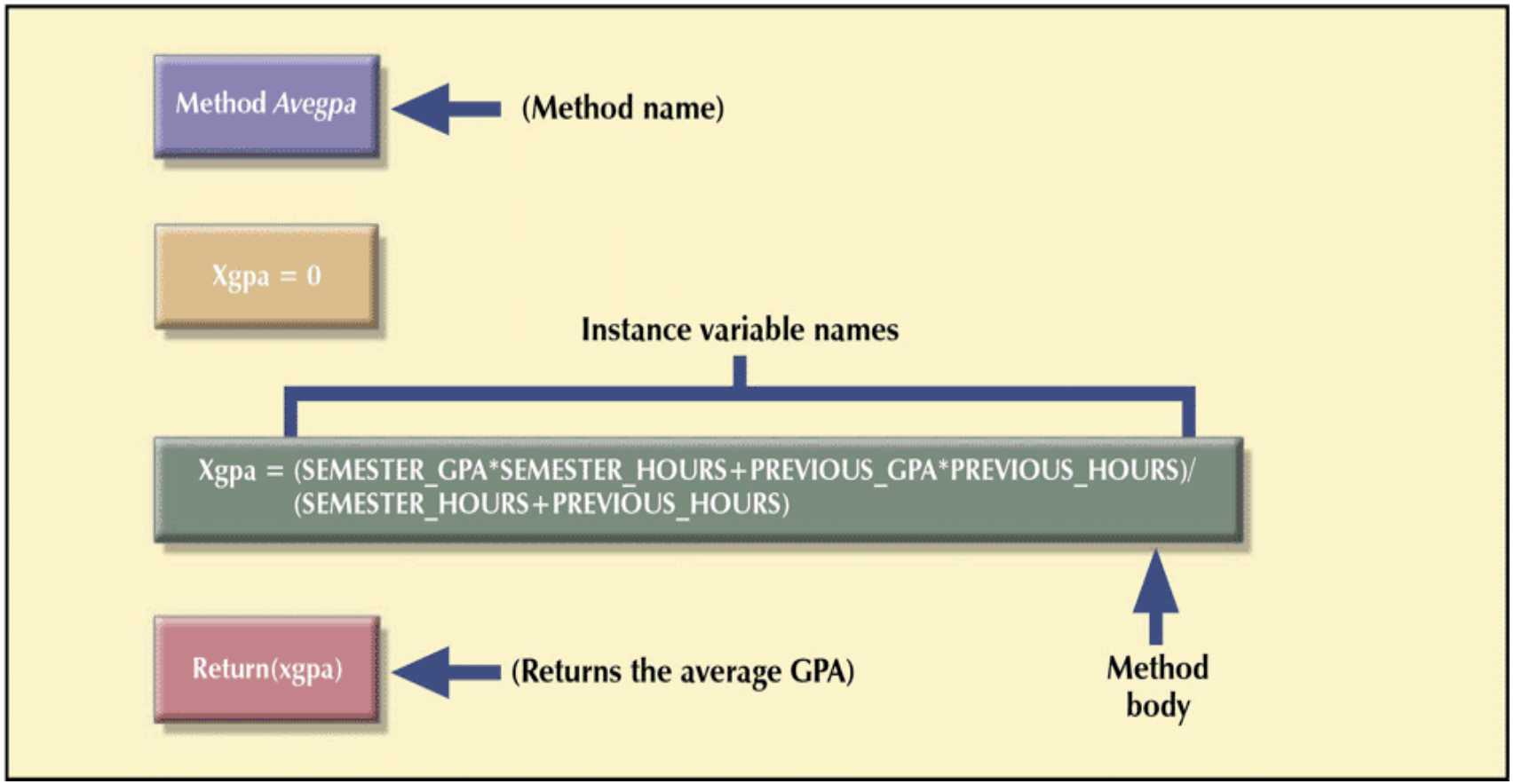
Depiction of an Object

FIGURE 11.2 DEPICTION OF AN OBJECT



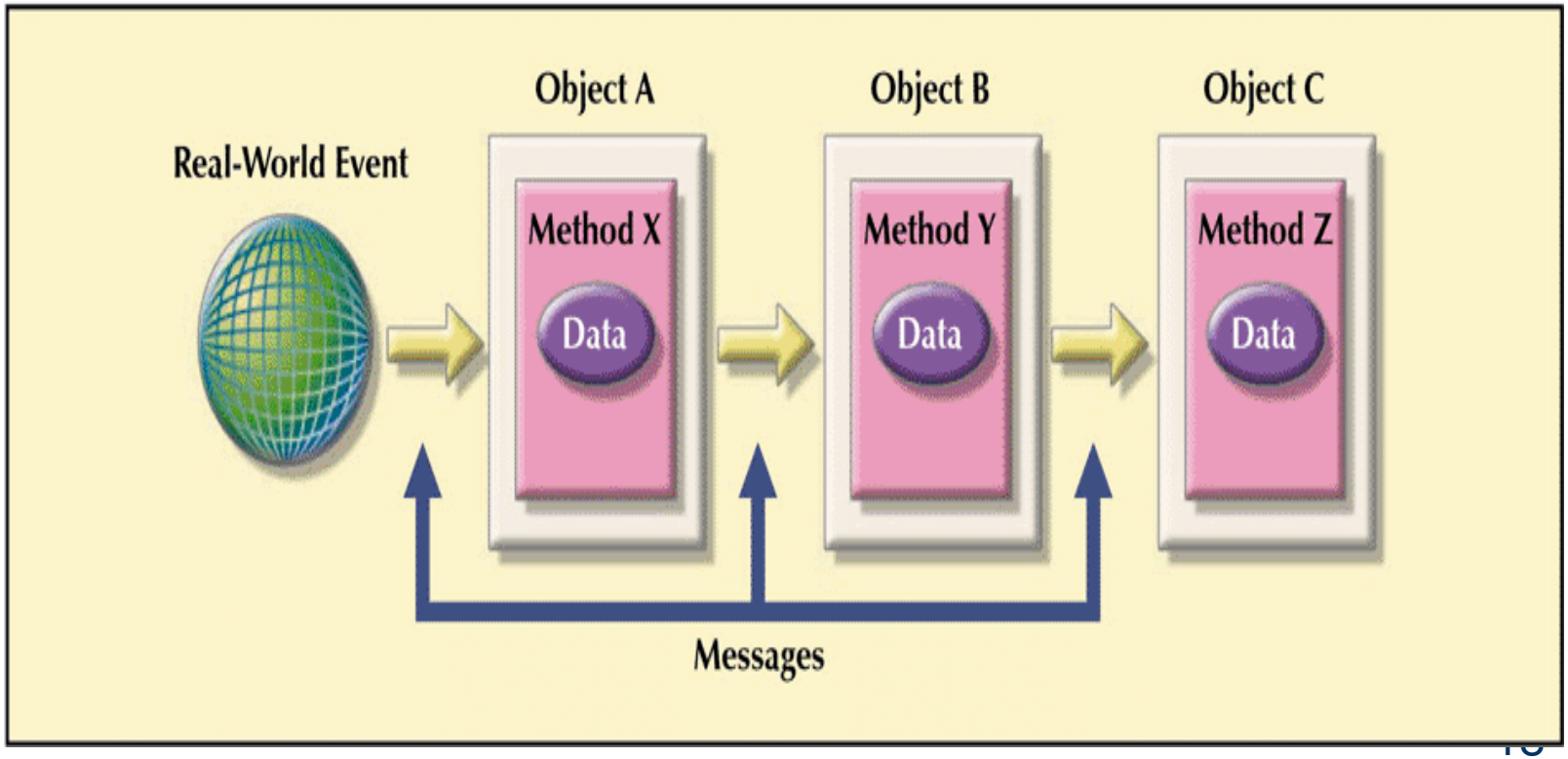
Method Components

FIGURE 11.3 METHOD COMPONENTS



Objects Send Messages to Each Other

FIGURE 11.4 OBJECTS SEND MESSAGES TO EACH OTHER



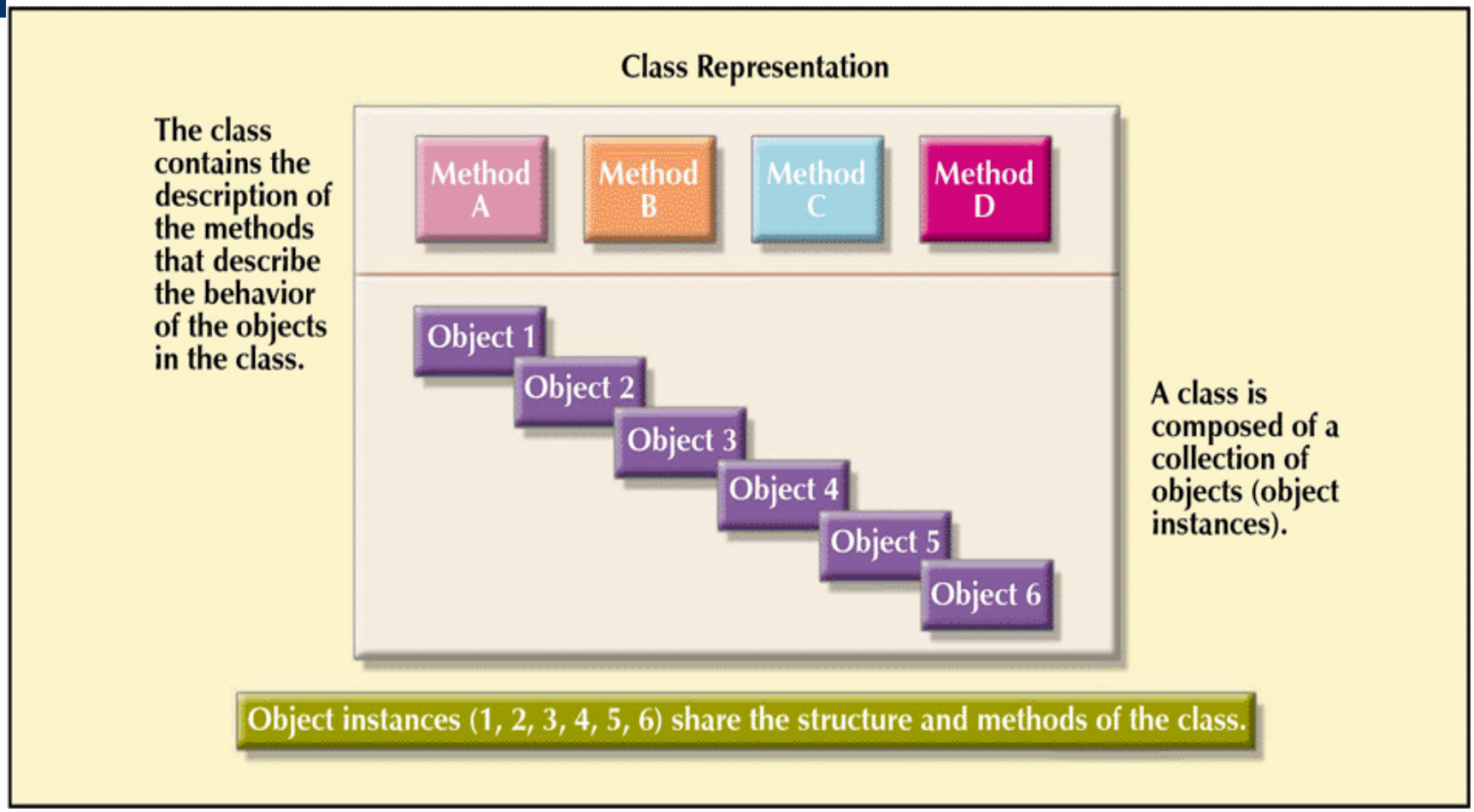
Classes

- Collection of similar objects with shared structure (attributes) and behavior (methods)
- Class instance or object instance
  - Each object in a class

19

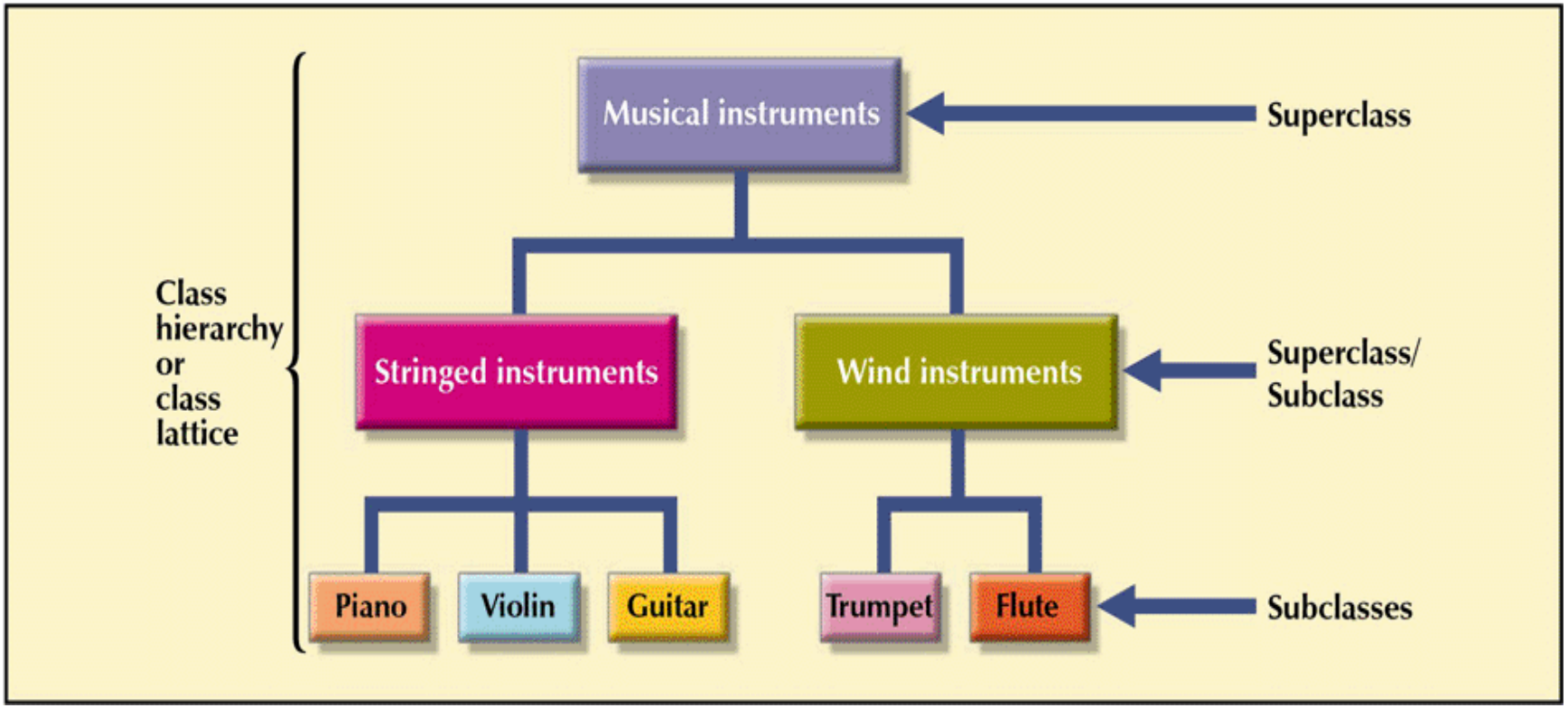
Class Illustration

FIGURE 11.5 CLASS ILLUSTRATION



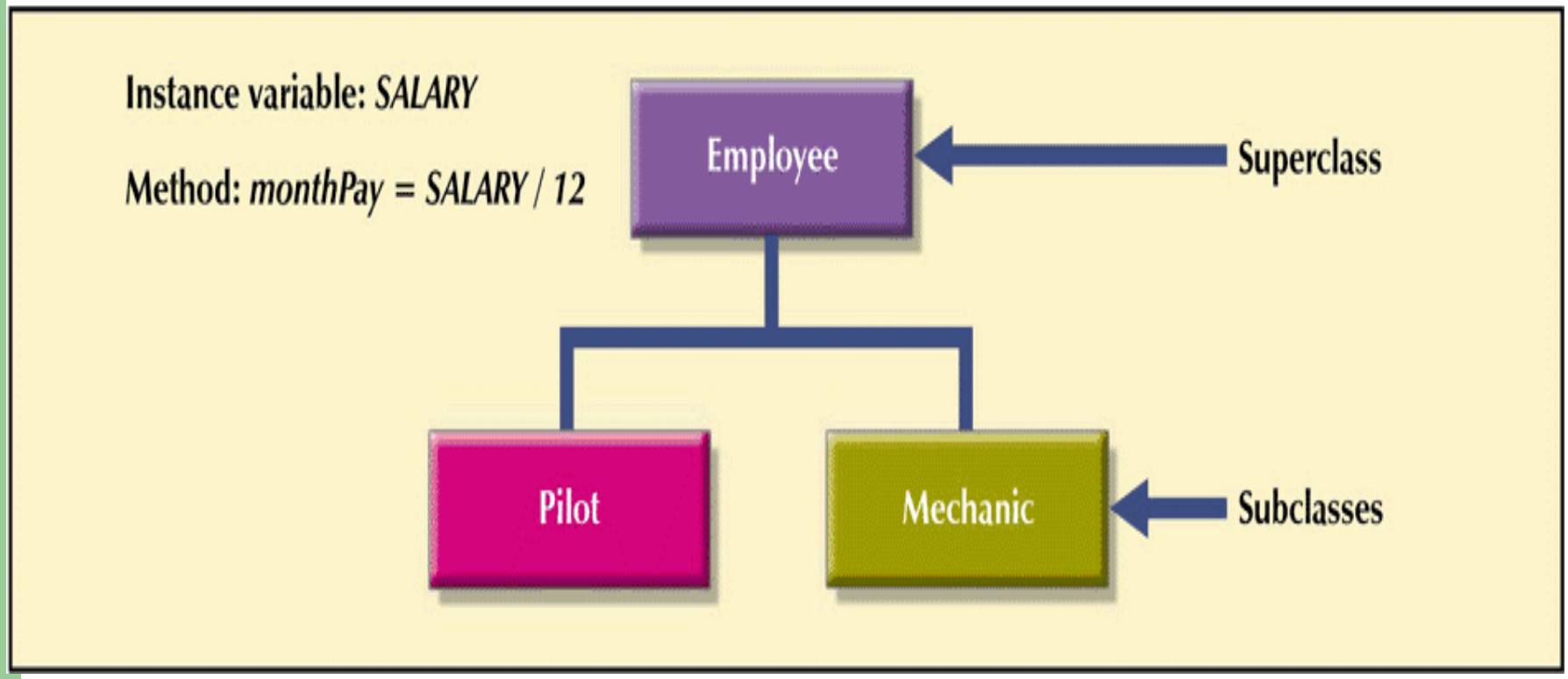
Musical Instruments Class Hierarchy

FIGURE 11.9 MUSICAL INSTRUMENTS CLASS HIERARCHY



Single Inheritance

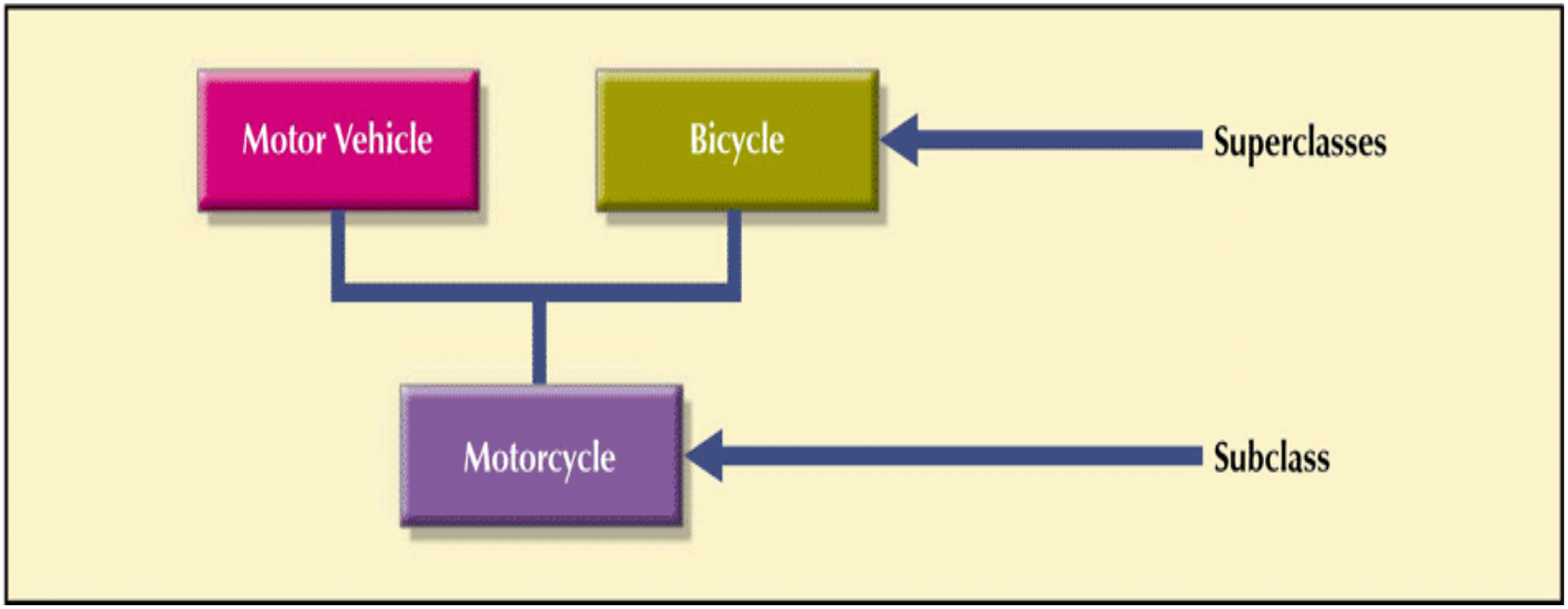
FIGURE 11.10 SINGLE INHERITANCE



22

Multiple Inheritance

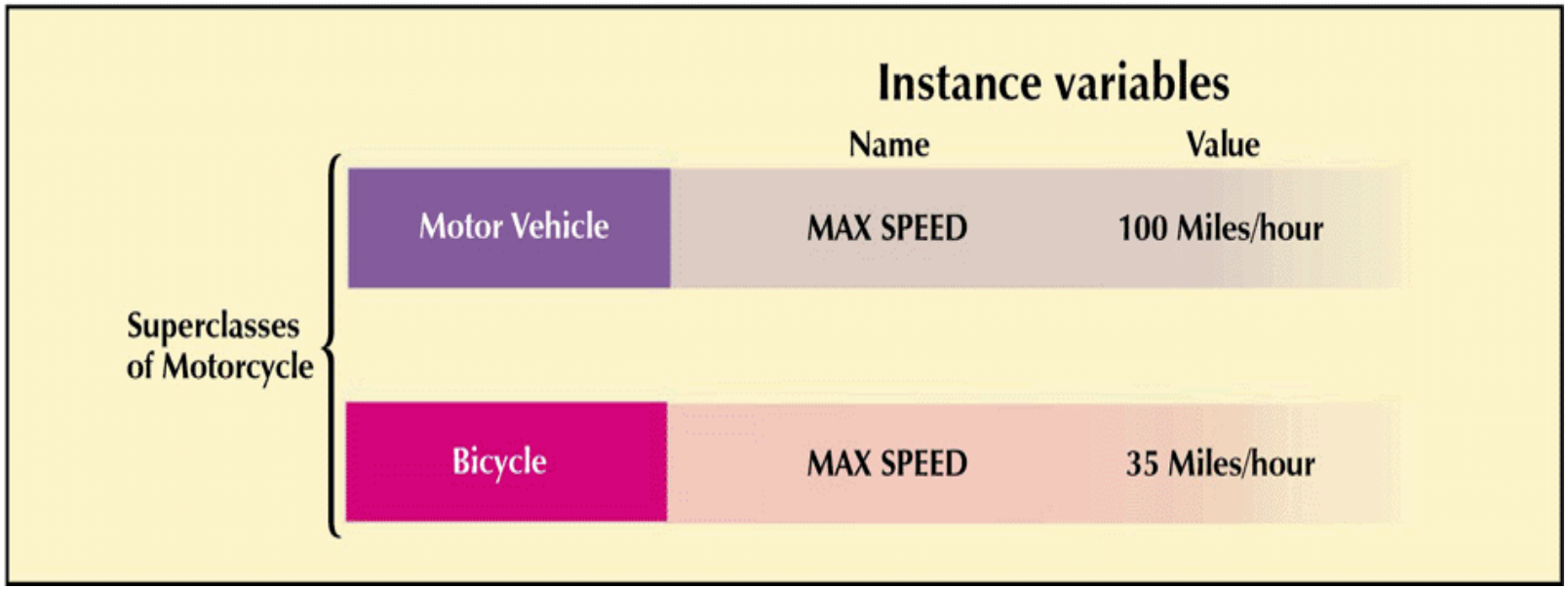
FIGURE 11.11 MULTIPLE INHERITANCE



23

Motor Vehicle and Bicycle Instance Variables

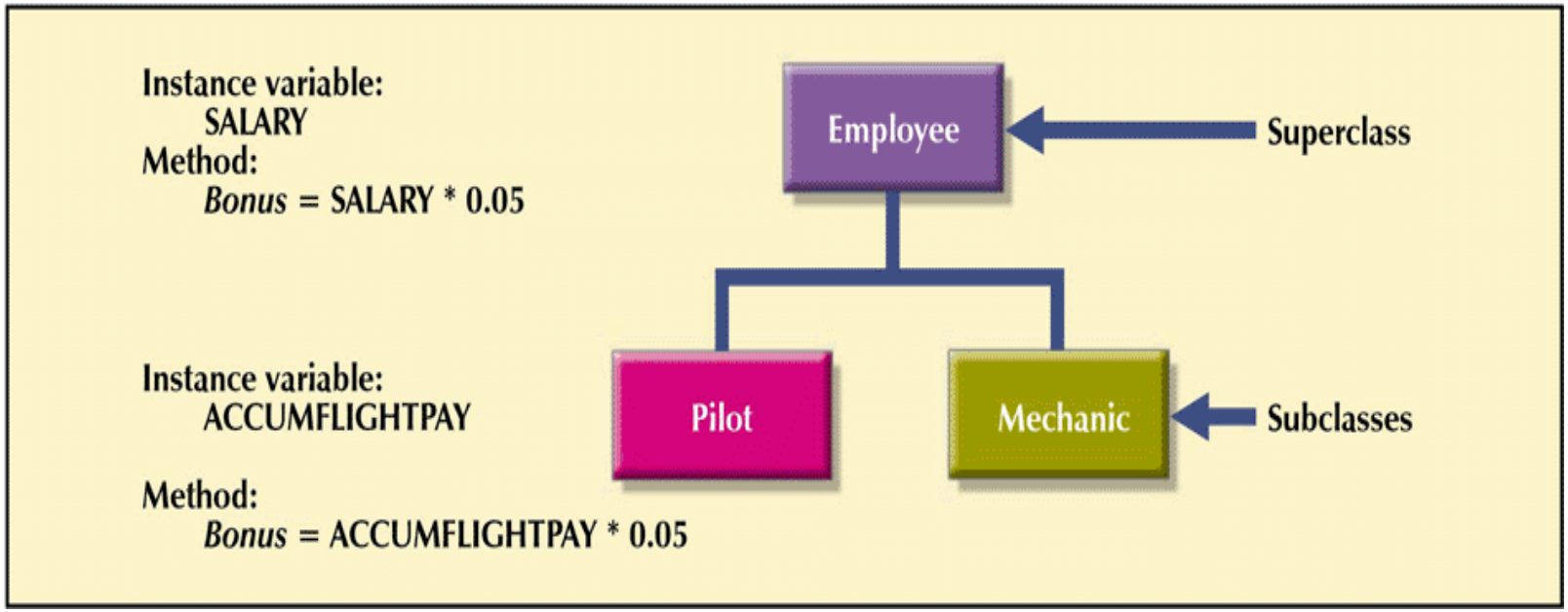
FIGURE 11.12 MOTOR VEHICLE AND BICYCLE INSTANCE VARIABLES



24

Employee Class Hierarchy Method Override

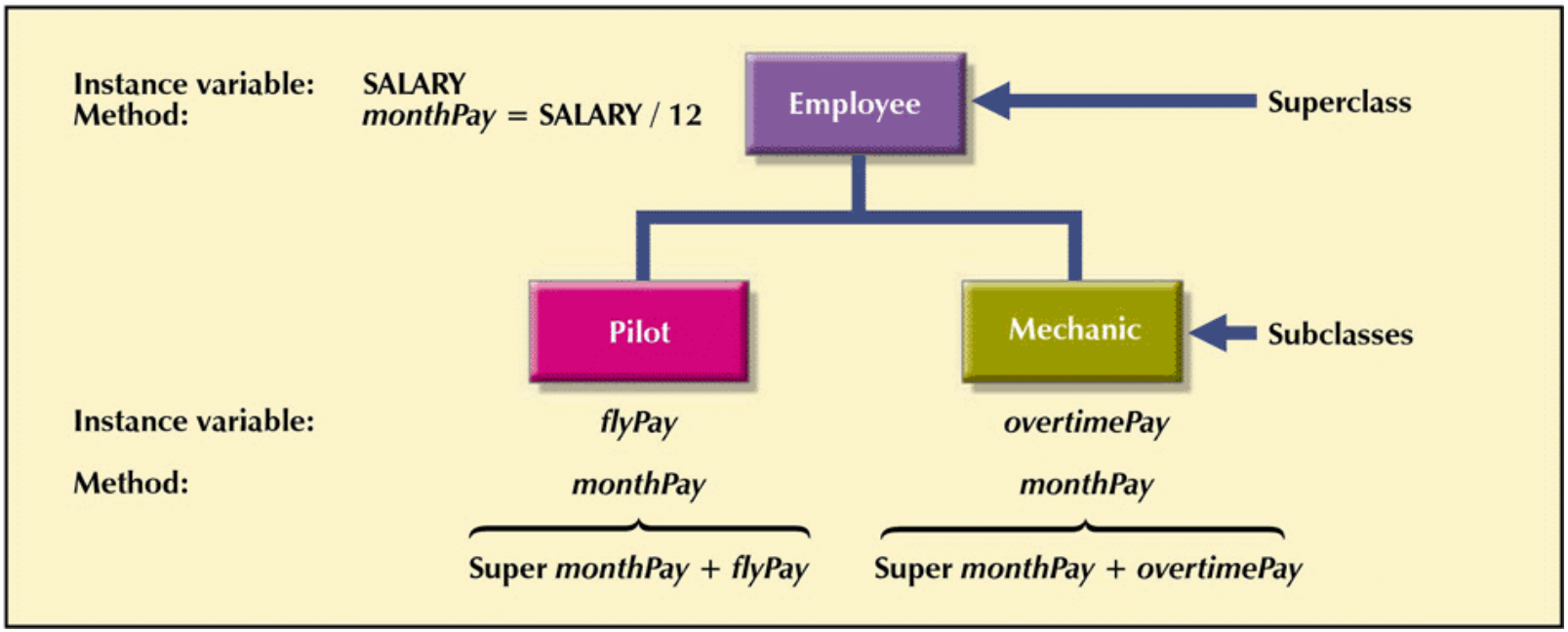
FIGURE 11.13 EMPLOYEE CLASS HIERARCHY METHOD OVERRIDE



25

Employee Class Hierarchy Polymorphism

FIGURE 11.14 EMPLOYEE CLASS HIERARCHY POLYMORPHISM



26

Characteristics of an Object Oriented Data Model

- Support the representation of complex objects
- Are extensible:
  - Capable of defining new data types as well as the operations to be performed on them
- Support encapsulation:
  - Data representation and method's implementation must be hidden from external entities

27

Characteristics of an Object Oriented Data Model (continued)

- Exhibit inheritance:
  - Object must be able to inherit properties (data and methods) of other objects
- Support the notion of object identity (OID)

28

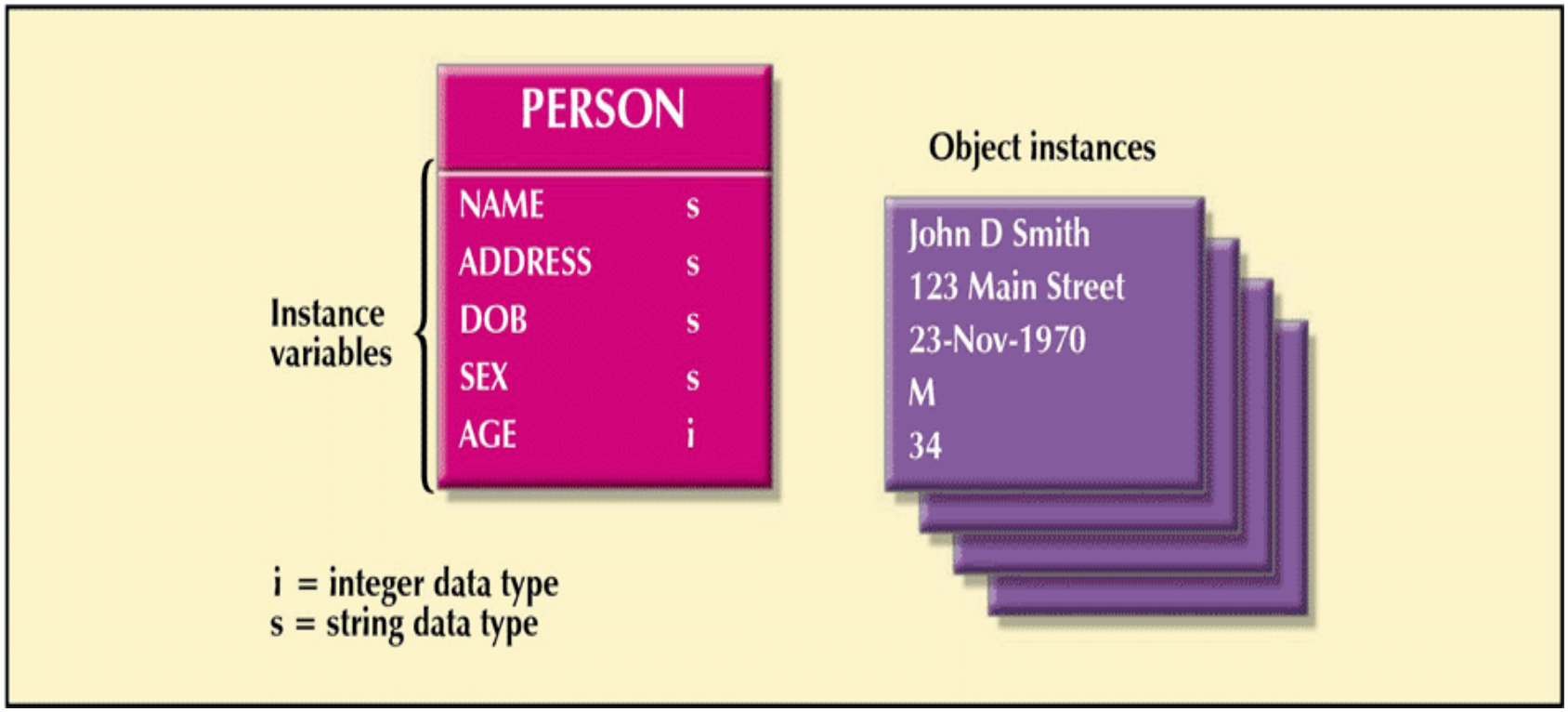
Comparing the OO and ER Model Components

TABLE 11.3 COMPARING THE OO AND ER MODEL COMPONENTS

OO DATA MODEL	ER MODEL
Type	Entity definition
Object	Entity
Class	Entity set
Instance variable	Attribute
N/A	Primary key
OID	N/A
Method	N/A
Class hierarchy	ER diagram

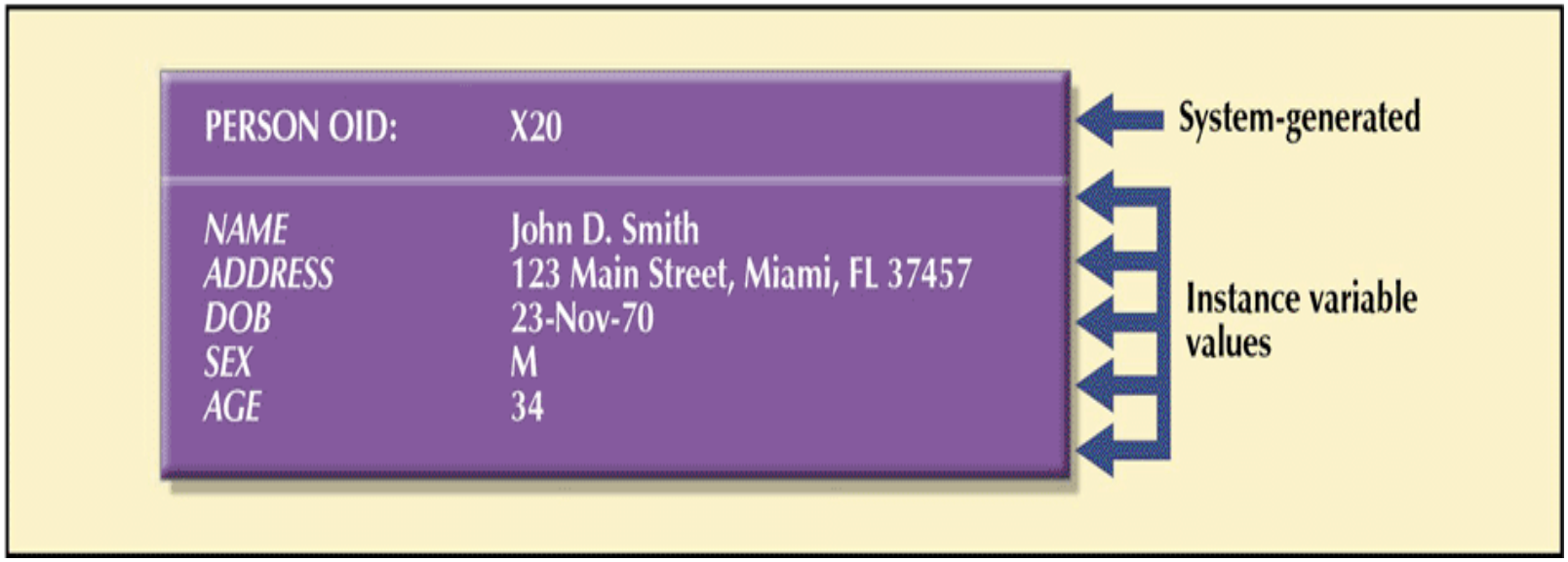
Shared Representation for All Objects of the Class Person

FIGURE 11.15 SHARED REPRESENTATION FOR ALL OBJECTS OF THE CLASS PERSON



State of a Person Object Instance

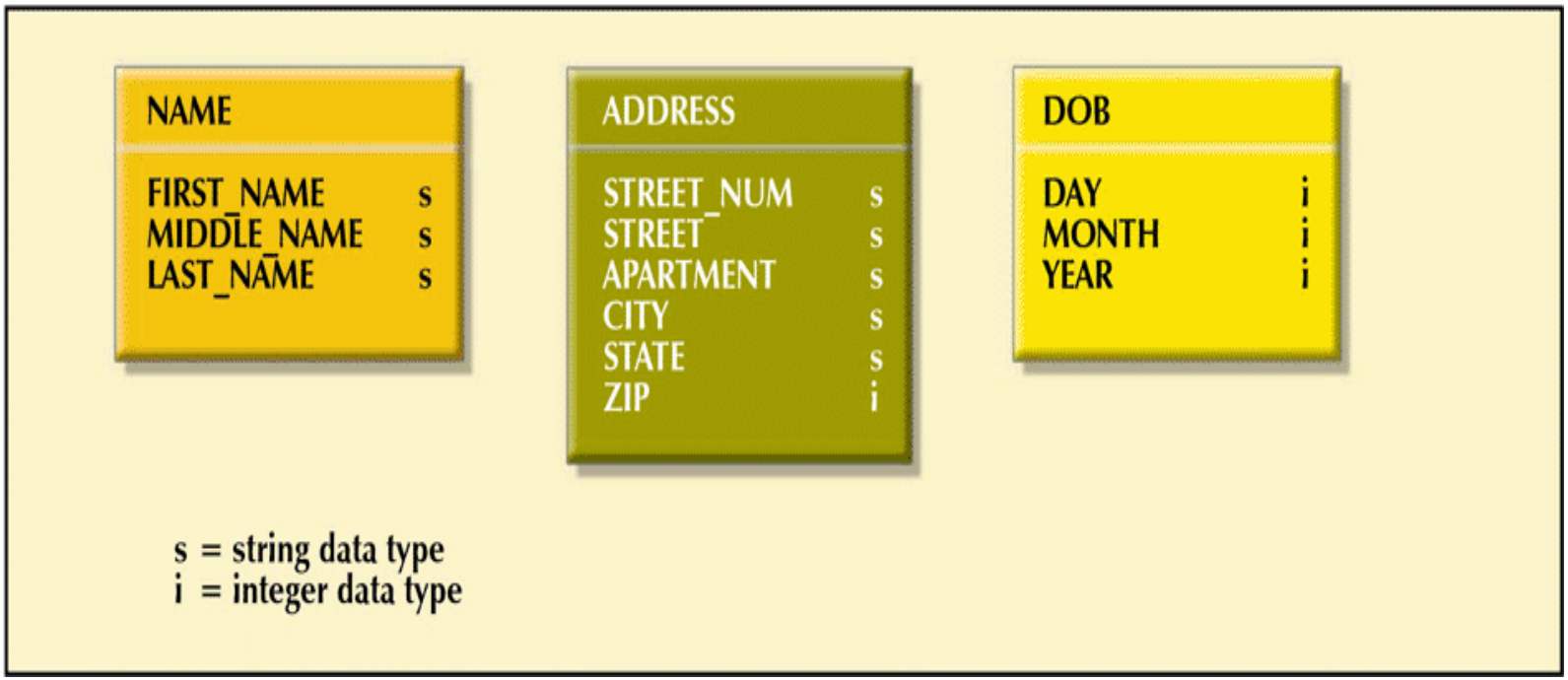
FIGURE 11.16 STATE OF A PERSON OBJECT INSTANCE



31

Defining Three Abstract Data Types

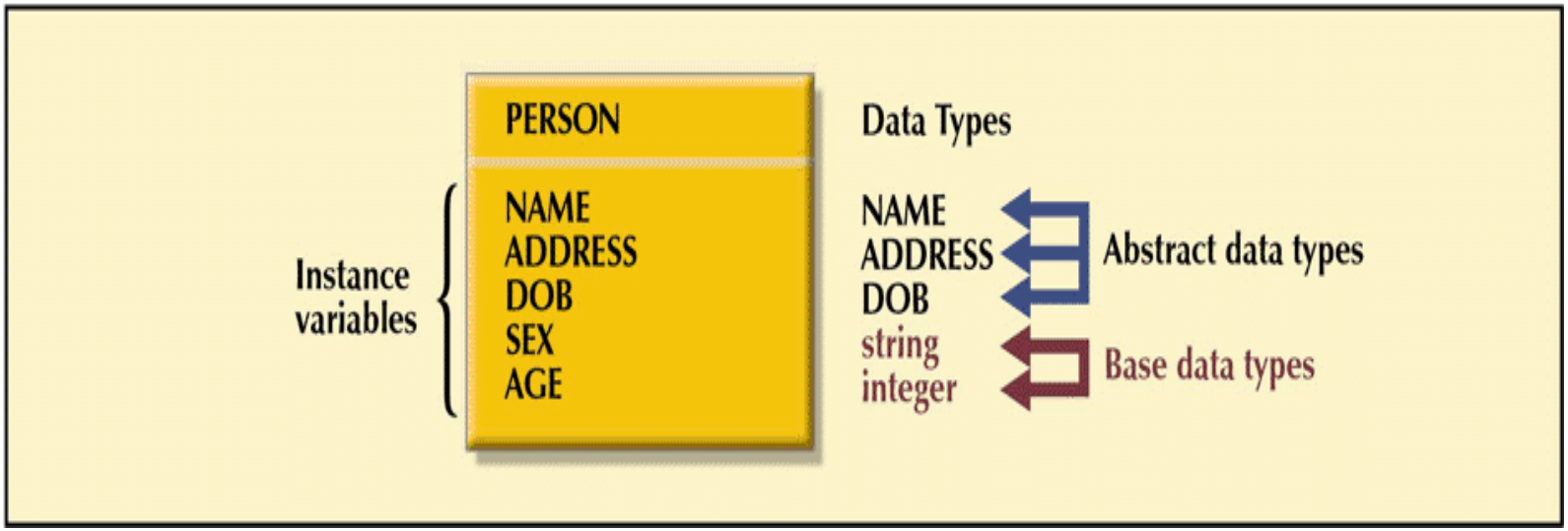
FIGURE 11.17 DEFINING THREE ABSTRACT DATA TYPES



32

Object Representation for Instances of the Class Person with ADTs

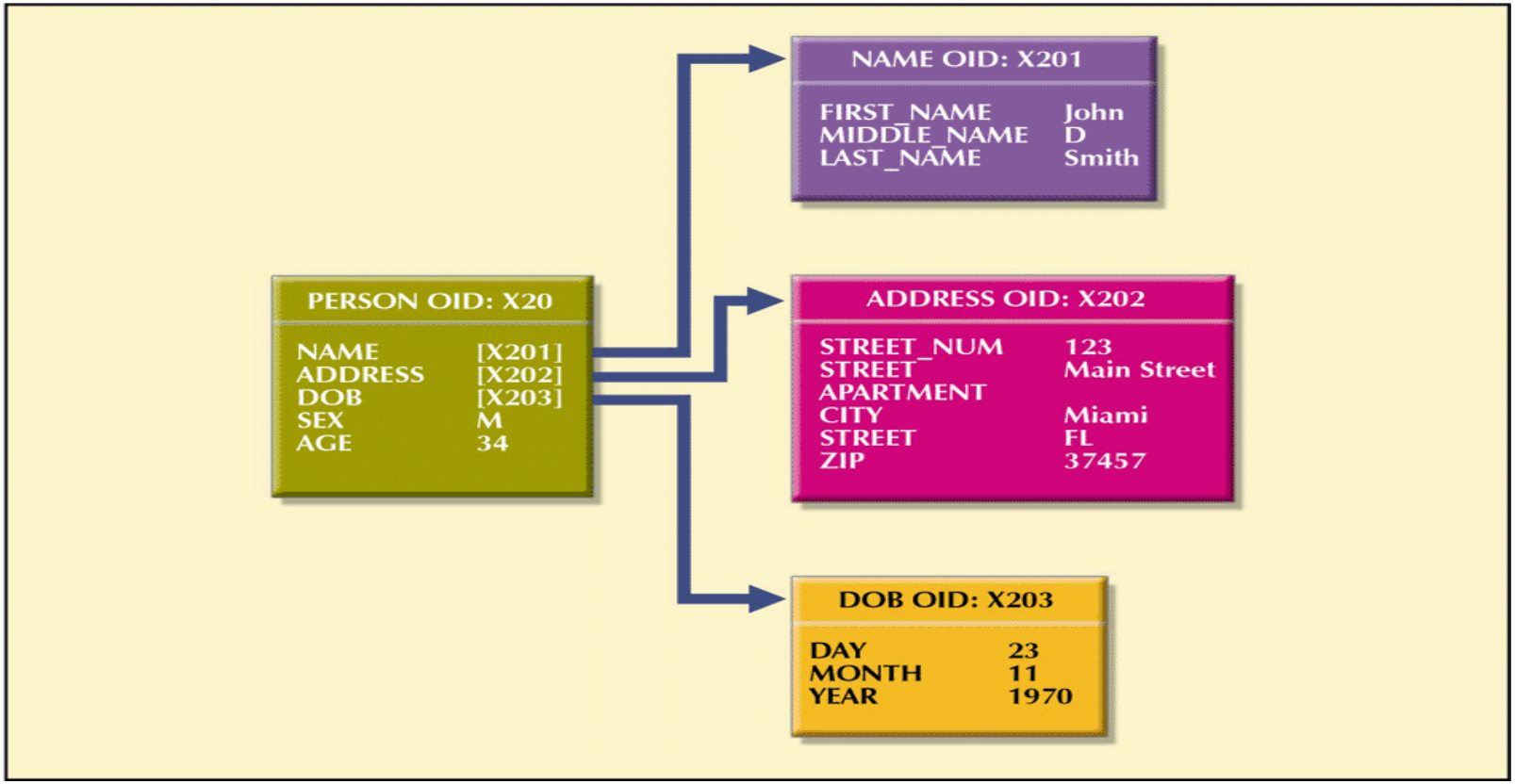
FIGURE 11.18 OBJECT REPRESENTATION FOR INSTANCES OF THE CLASS PERSON WITH ADTS



33

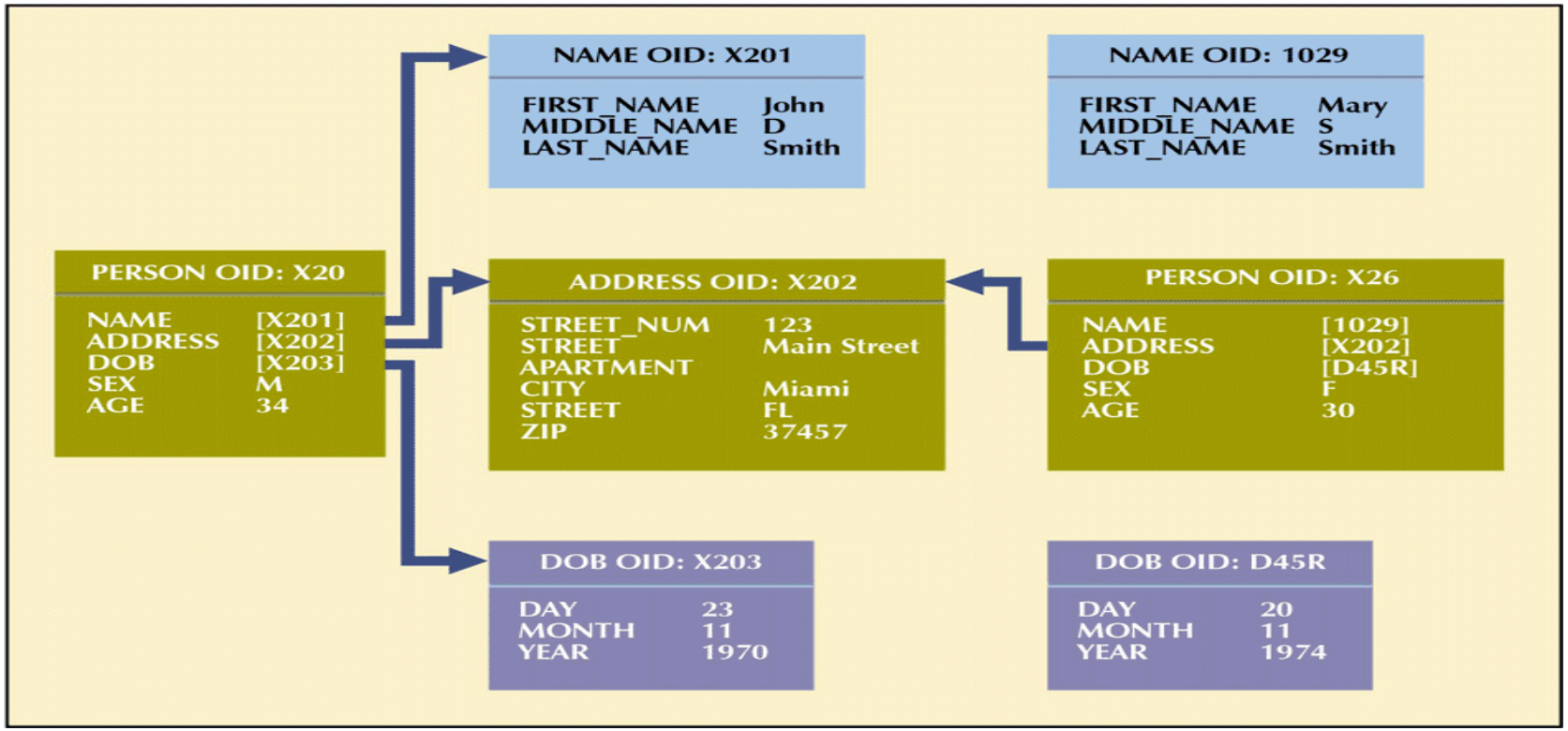
Object State for an Instance of the Class Person, Using ADTs

FIGURE 11.19 OBJECT STATE FOR AN INSTANCE OF THE CLASS PERSON, USING ADTS



Referential Object Sharing

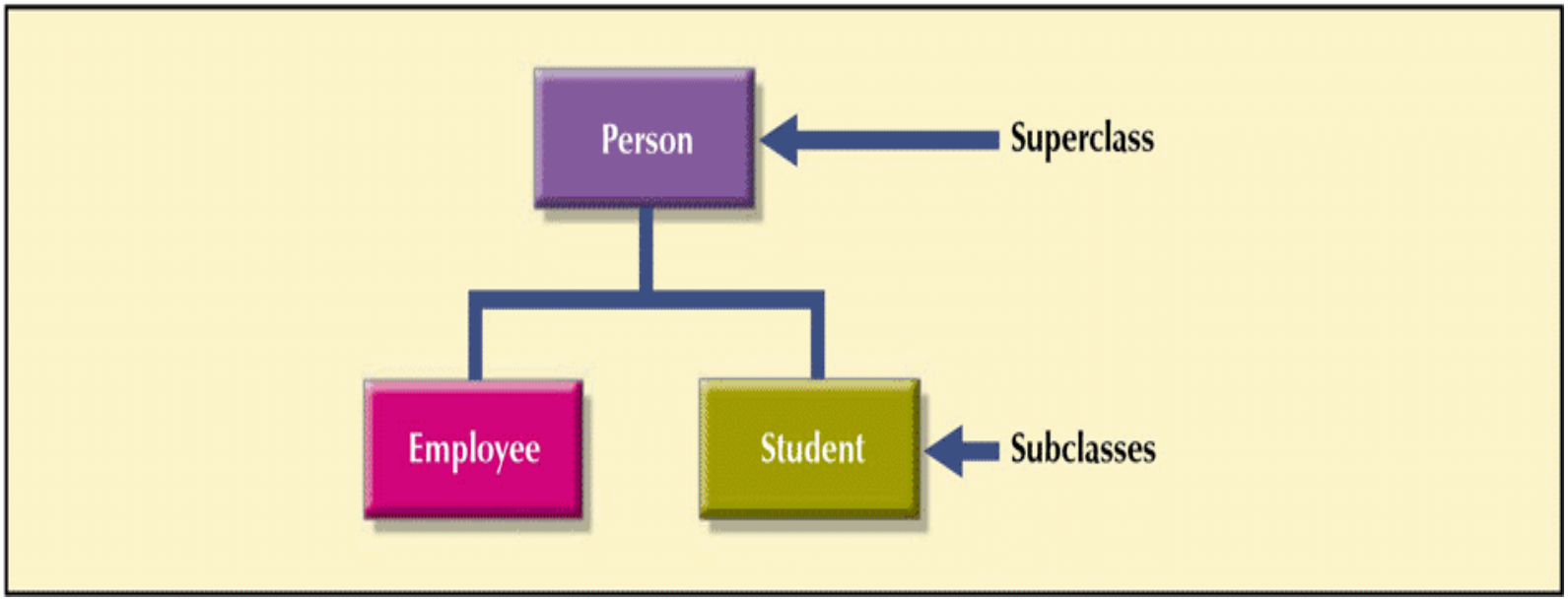
FIGURE 11.20 REFERENTIAL OBJECT SHARING



35

Class Hierarchy

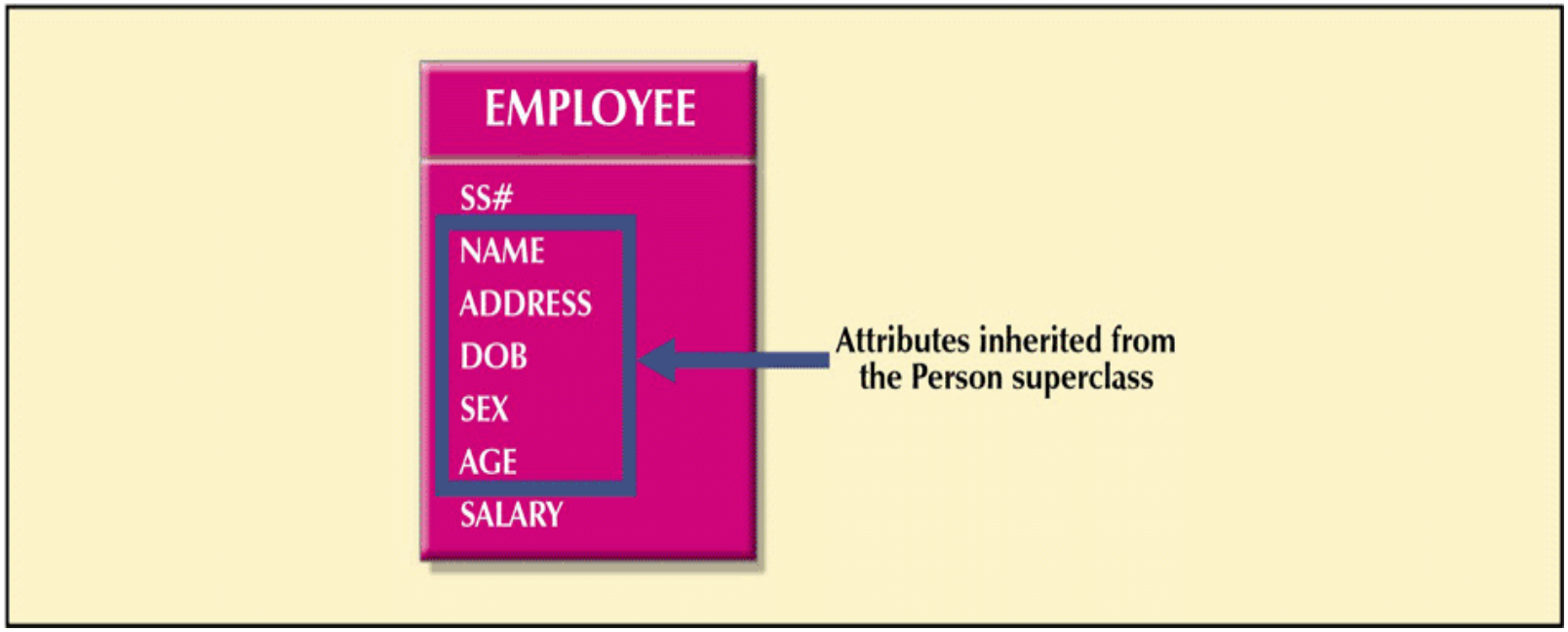
FIGURE 11.21 CLASS HIERARCHY



36

Employee Object Representation

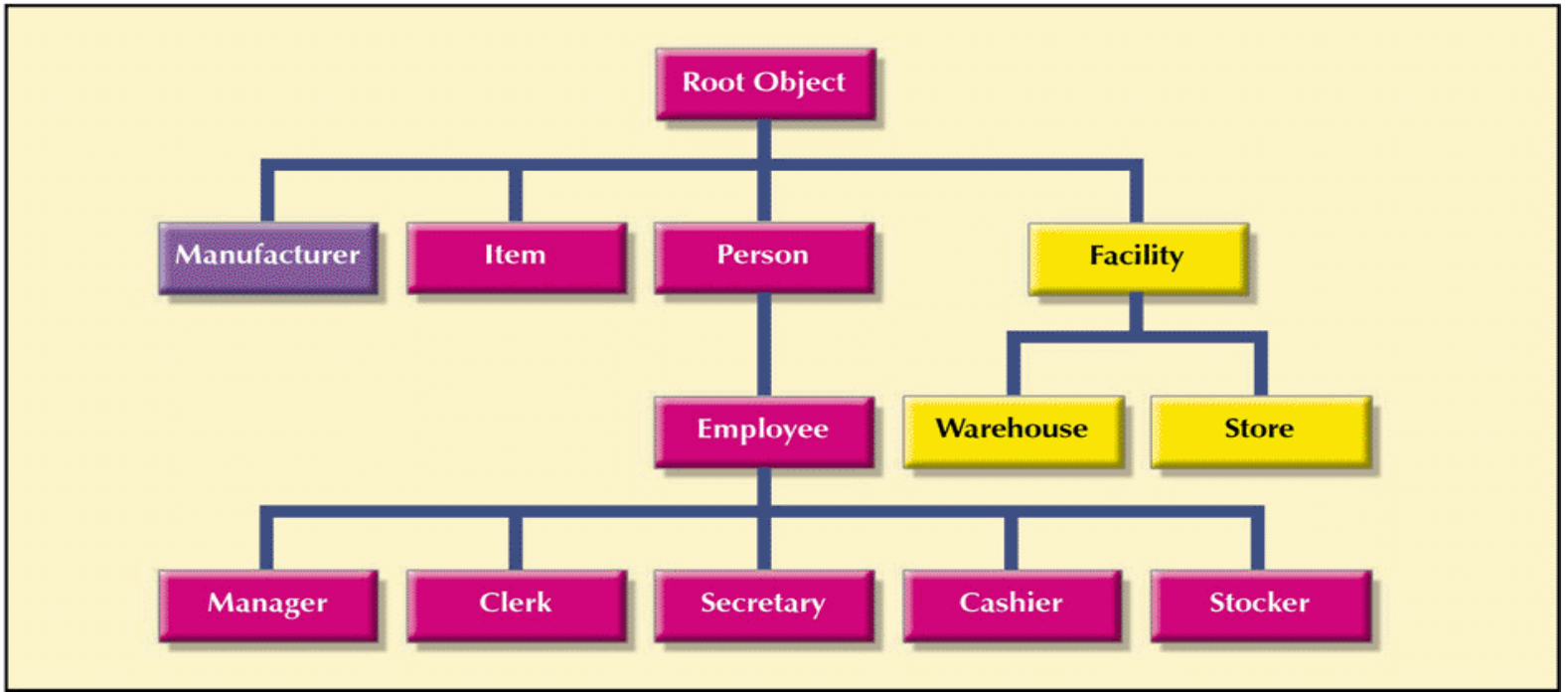
FIGURE 11.22 EMPLOYEE OBJECT REPRESENTATION



37

Class Hierarchy for the Retail Corporation

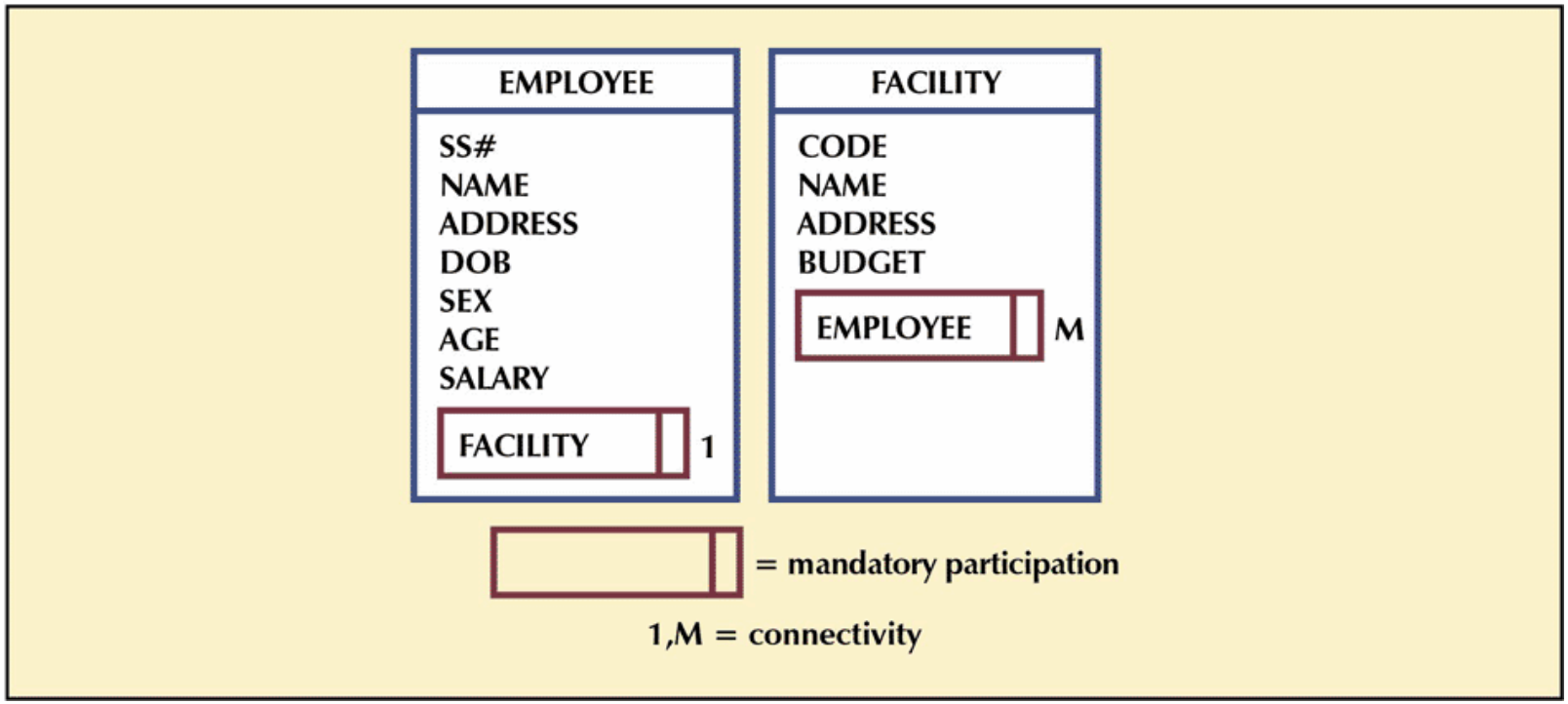
FIGURE 11.23 CLASS HIERARCHY FOR THE EDLP RETAIL CORPORATION



38

Representing a 1:M Relationship

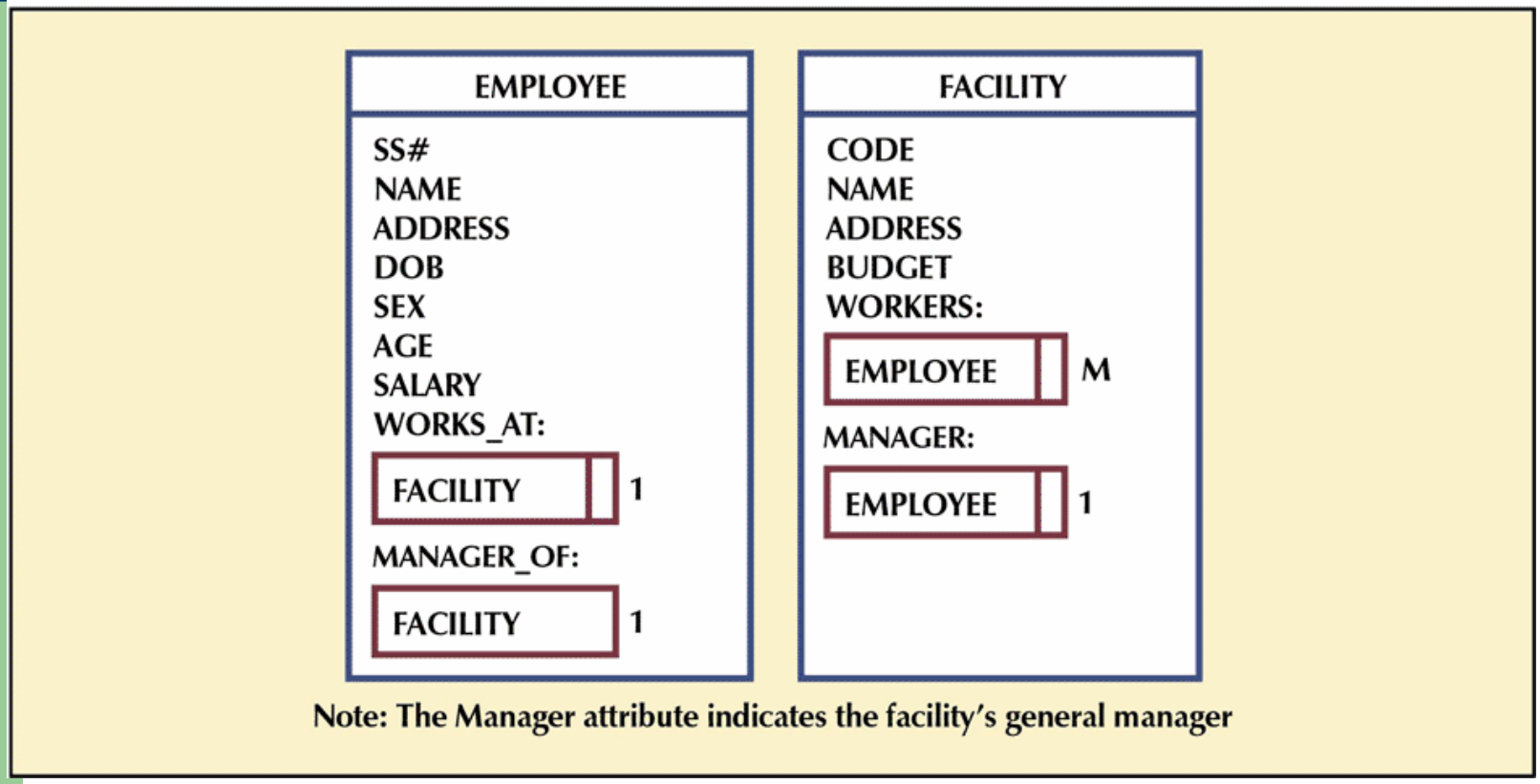
FIGURE 11.24 REPRESENTING A 1:M RELATIONSHIP



39

Representing 1:1 and 1:M Relationships

FIGURE 11.25 REPRESENTING 1:1 AND 1:M RELATIONSHIPS

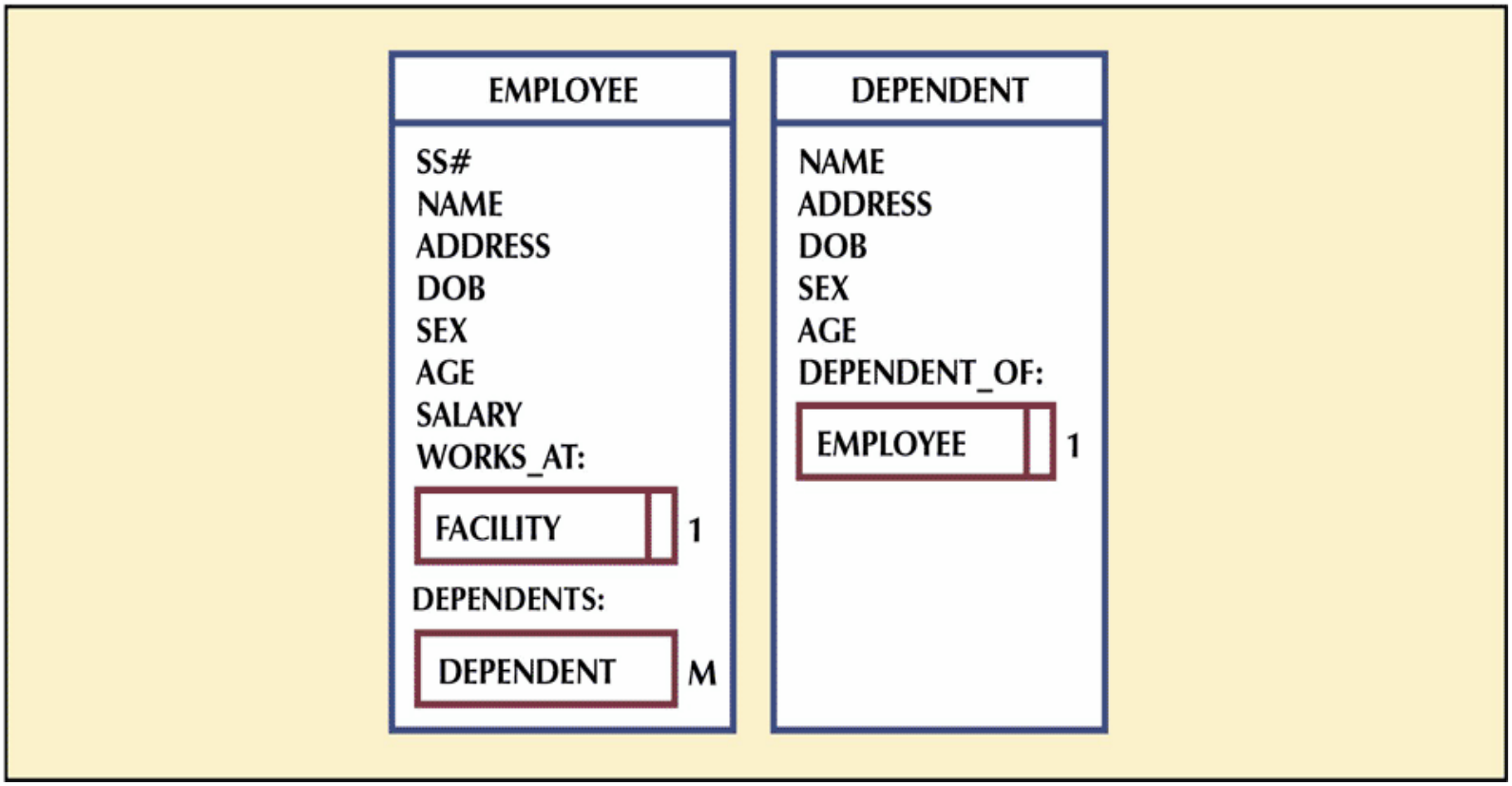


Note: The Manager attribute indicates the facility's general manager

40

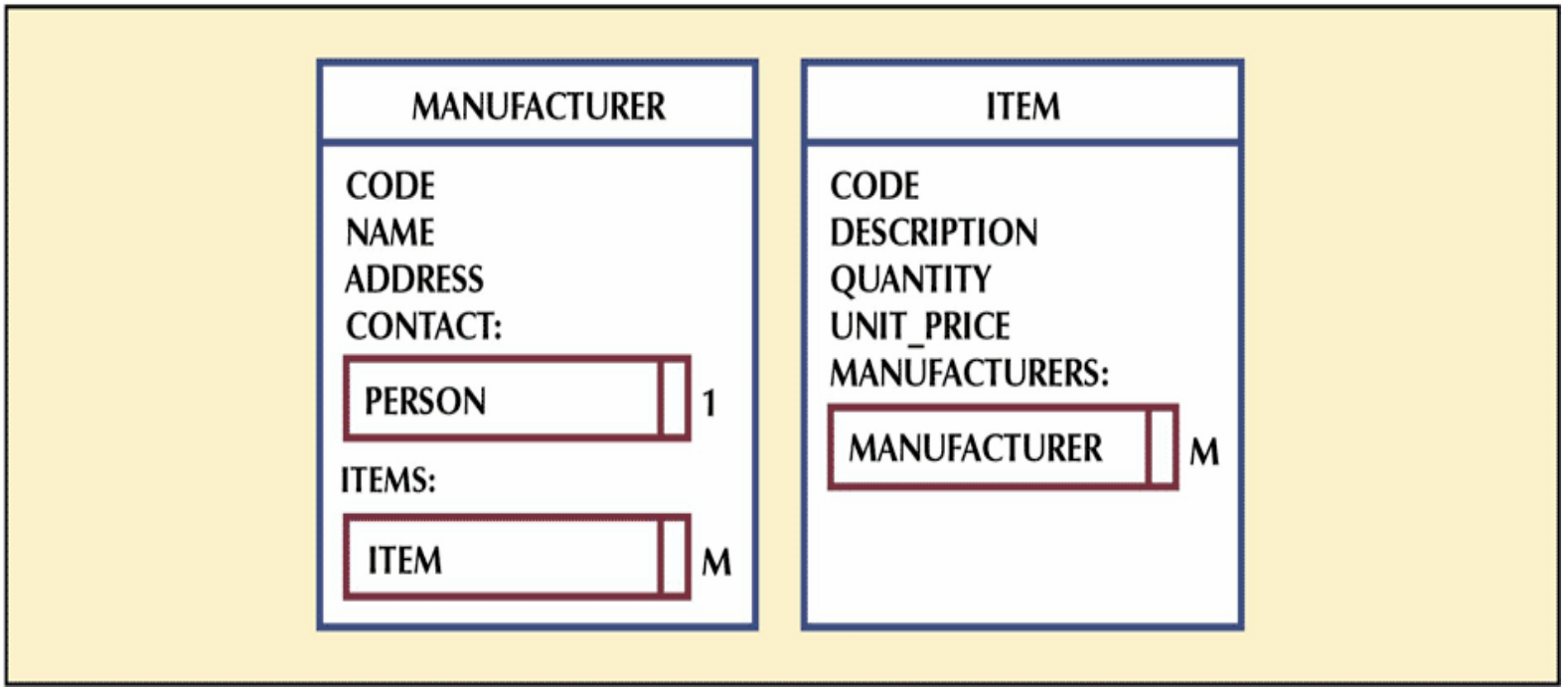
Employee-Dependent Relationship

FIGURE 11.26 EMPLOYEE-DEPENDENT RELATIONSHIP



Representing the M:N Relationship

FIGURE 11.27 REPRESENTING THE M:N RELATIONSHIP



42

INVENTORY Table with Predetermined (Base) Data Types

FIGURE 11.31 INVENTORY TABLE WITH PREDETERMINED (BASE) DATA TYPES

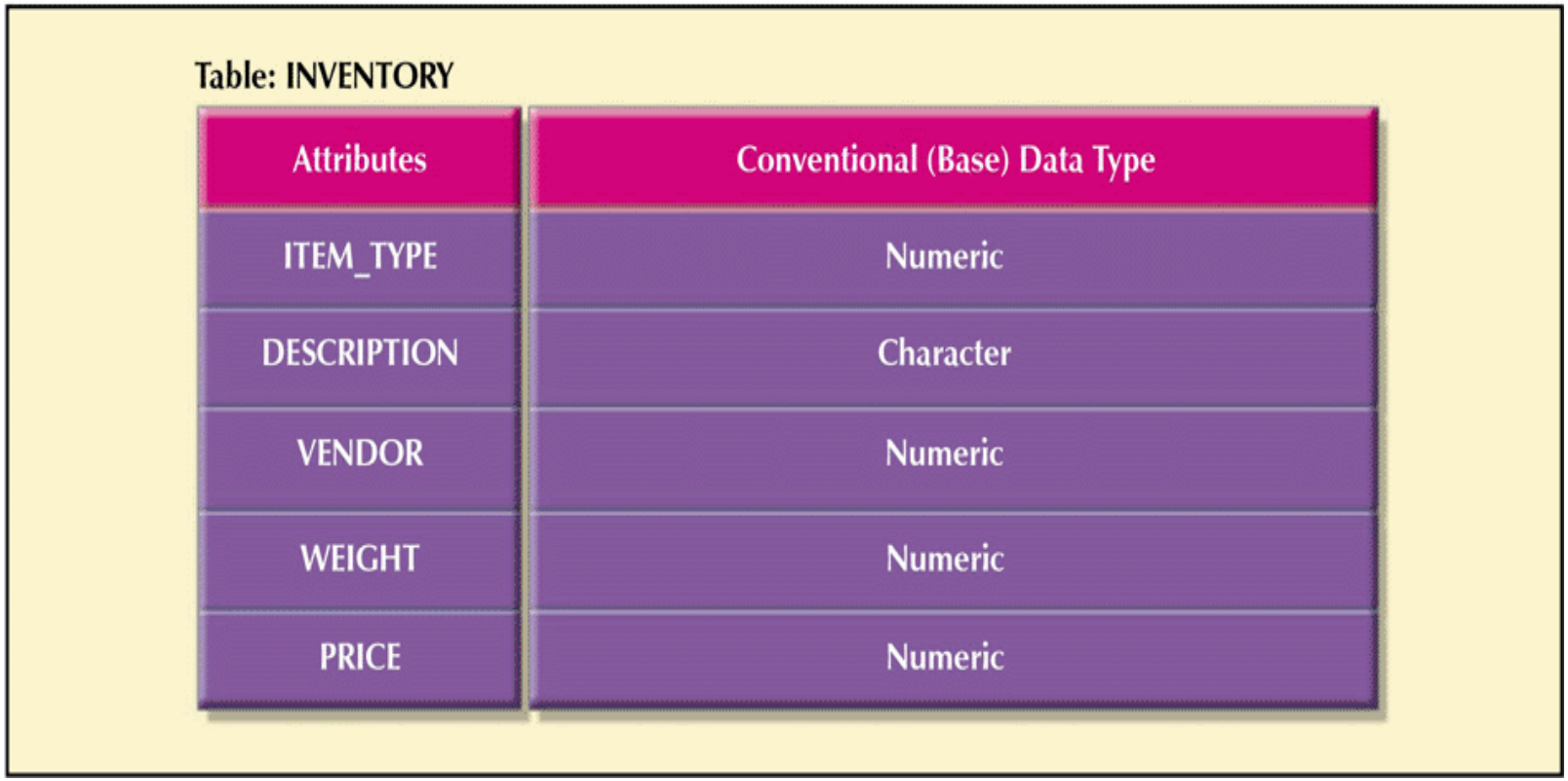
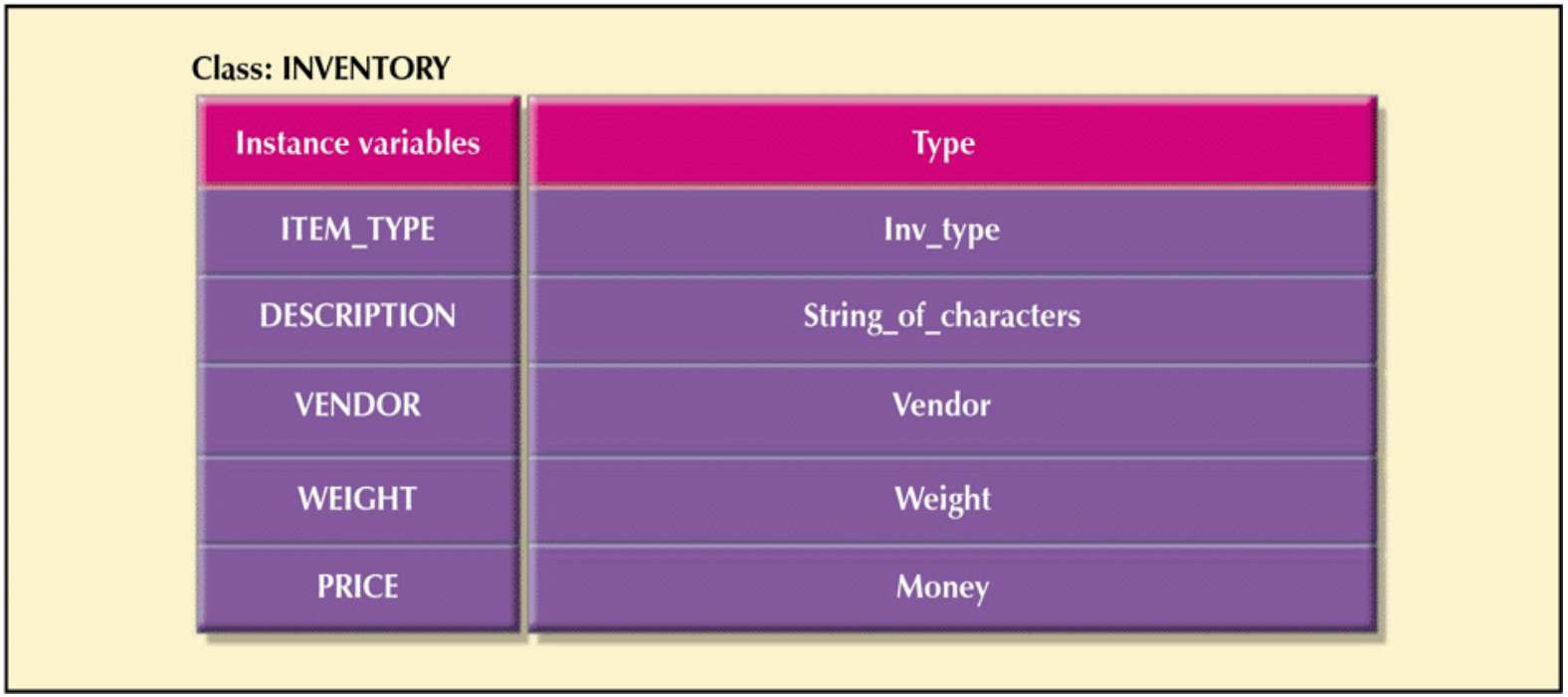


Table: INVENTORY

Attributes	Conventional (Base) Data Type
ITEM_TYPE	Numeric
DESCRIPTION	Character
VENDOR	Numeric
WEIGHT	Numeric
PRICE	Numeric

Inventory Class with Early Binding

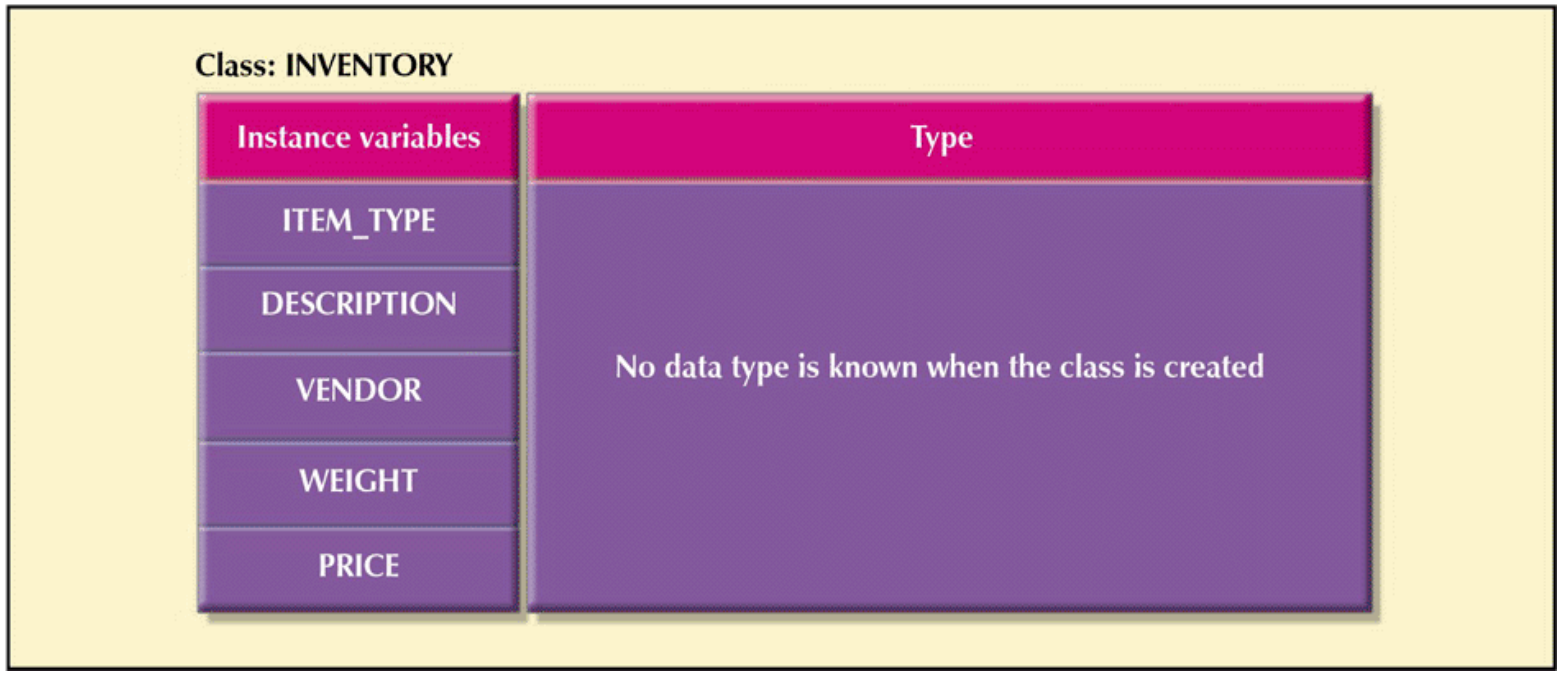
FIGURE 11.32 INVENTORY CLASS WITH EARLY BINDING



44

OODM Inventory Class with Late Binding

FIGURE 11.33 OODM INVENTORY CLASS WITH LATE BINDING



45

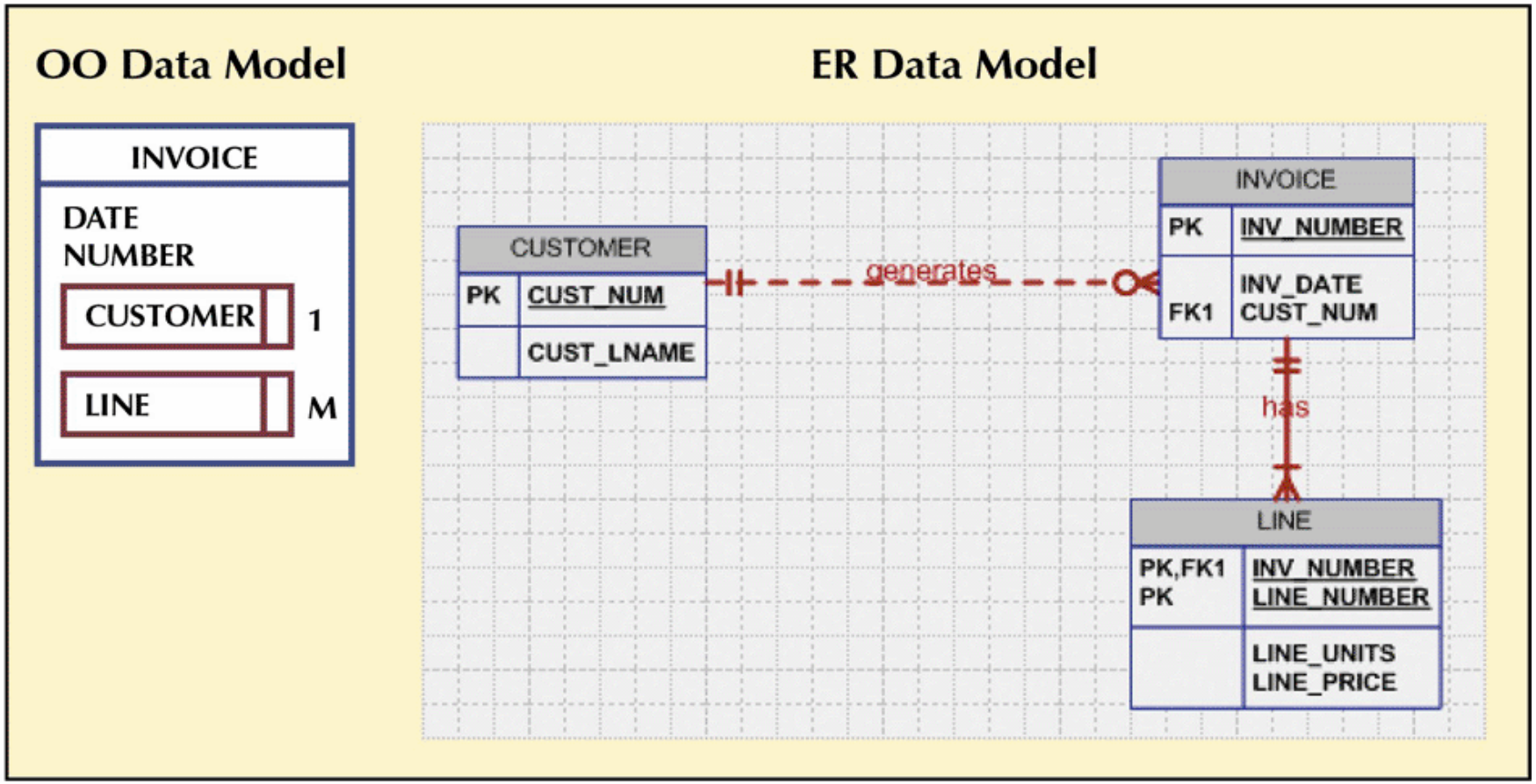
OODM and Previous Data Models: Similarities and Differences

- OODM object resembles entity and tuple in the ER and relational models but has additional characteristics
- Class
- Hierarchies
- Encapsulation
- Relationships
- OODM produces a schema in which relations form part of the structure of the database

46

An Invoice Representation

FIGURE 11.34 AN INVOICE REPRESENTATION



Object Oriented Database Management Systems

- Integrate benefits of typical database systems with the more powerful modeling and computational (programming) characteristics of the object oriented data model
- Used to develop complex systems

48

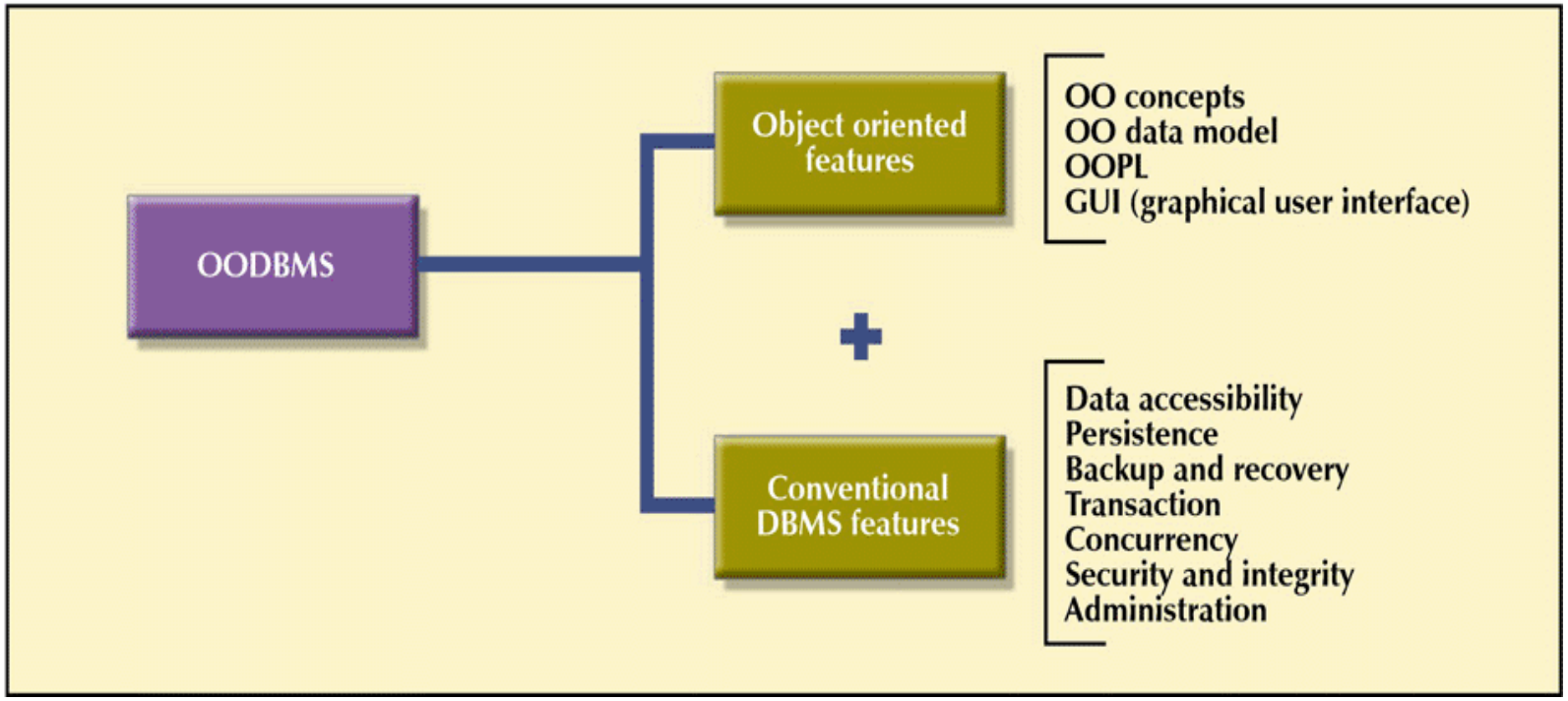
Object Oriented Database Management Systems

- Result of combining OO features, such as
  - class inheritance
  - encapsulation, and
  - polymorphism,
- With database features such as
  - data integrity, -- data manipulation,
  - security, -- system tuning and
  - persistence, -- recovery
  - transaction management,
  - concurrency control,
  - backup,

49

Object Oriented Database Management Systems

FIGURE 11.35 OBJECT ORIENTED DATABASE MANAGEMENT SYSTEMS



50

The Thirteen OODBMS Rules

TABLE 11.4 THE THIRTEEN OODBMS RULES

	RULES THAT MAKE IT AN OO SYSTEM
Rule 1	The system must support complex objects.
Rule 2	Object identity must be supported.
Rule 3	Objects must be encapsulated.
Rule 4	The system must support types or classes.
Rule 5	The system must support inheritance.
Rule 6	The system must avoid premature binding.
Rule 7	The system must be computationally complete.
Rule 8	The system must be extensible.
	RULES THAT MAKE IT A DBMS
Rule 9	The system must be able to remember data locations.
Rule 10	The system must be able to manage very large databases.
Rule 11	The system must accept concurrent users.
Rule 12	The system must be able to recover from hardware and software failures.
Rule 13	Data query must be simple.

How Object Orientation Affects Database Design

- Relational and ER models sometimes cannot adequately represent some objects
- Operations are not a part of the database model
- Object oriented design requires the database description to include objects and their data representation, constraints, and operations
- Few computerized OODB design tools exist
- Lack of standards affects object oriented database design

52

OODBMS: Advantages and Disadvantages

- OODBMS occupies a strong market
- Vehicle for technological innovation
- Has not been the beneficiary of market share growth

53

How OO Concepts Have Influenced the Relational Model

- RDBMS is beginning to reach its limits in a business data environment that is changing with the advent of mixed-media data storage and retrieval

54

How OO Concepts Have Influenced the Relational Model

- Extended relational model (ERM), or object/relational model (O/RM)
  - Extensibility of new user-defined (abstract) data types
  - Complex objects
  - Inheritance
  - Procedure calls (rules or triggers)
  - System-generated identifiers (OID surrogates)

55

The Next Generation of Database Management Systems

- Likely to incorporate features borrowed from object oriented database systems, artificial intelligence systems, expert systems, distributed databases, and the Internet
- Enable future DBMSs to handle more complex problems with both normalized and nonnormalized data
- OODBMS will probably continue to occupy a niche within the database market

56

Summary

- Object oriented concepts are derived from OO programming languages
- An object is a conceptual representation of a real-world entity
- Similar objects are grouped in classes
- Classes are organized into a class hierarchy
- Abstract data types are implemented through classes
- The OO data model allows the designer to create a more authentic representation of real-world objects

57

Summary (continued)

- OODBMSs combine OO features with database features
- OODBMSs yield several benefits over conventional DBMSs
  - more semantic information in the database
  - support for complex objects
  - better performance in complex applications
  - reusability of classes
  - extensibility of the data types supported by the database

58