Database Management

Database Management Approaches

Objectives

- Describe distributed database management systems (DDBMSs)
- Discuss client/server systems
- Examine the ways databases are accessed on the Web
- Discuss XML and related document specification standards

Objectives (continued)

- Define data warehouses and explain their structure and access
- Discuss the general concepts of object-oriented DBMSs

Distributed Databases

- Computers at various sites
- Connected with **communications network** or **network**
- **Distributed database**: single logical database physically divided among networked computers
- **Distributed database management system (DDBMS)**: supports and manipulates distributed databases

Distributed Databases (continued)

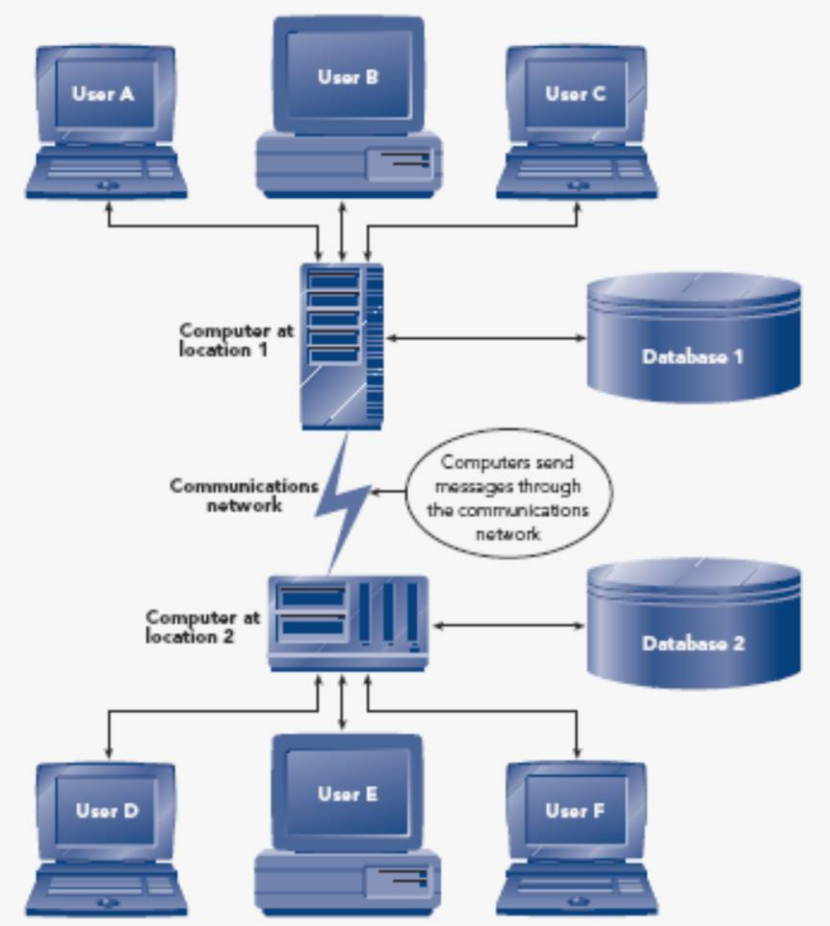

FIGURE 9-1: Communications network

Distributed Databases (continued)

- Computers in a network communicate through **messages**
- **Access delay** required for every message
  - Fixed amount of time
- Communication time = access delay + (data volume / transmission rate)

Characteristics of Distributed DBMSs

- **Homogeneous DDBMS**: same local DBMS at each site
- **Heterogeneous DDBMS**: at least two sites at which local DBMSs are different
- Shared characteristics of DDBMSs
  - Location transparency
  - Replication transparency
  - Fragmentation transparency

Location Transparency

- **Remote site**: site other than one where user is
- **Local site**: site where user is
- **Location transparency**: users do not need to be aware of location of data in a distributed database

Replication Transparency

- Data replication creates update problems that can lead to data inconsistencies
- **Replication transparency**: users unaware of steps taken by DDBMS to update various copies of data

Fragmentation Transparency

- **Data fragmentation**: DDBMS can divide and manage a logical object among various locations under its control
  - Data placed at the location where it is most often accessed
- **Fragmentation transparency**: users unaware of fragmentation

Fragmentation Transparency (continued)

Part

PartNum	Description	OnHand	Class	Warehouse	Price
AT94	Iron	50	HW	3	$24.95
BV06	Home Gym	45	SG	2	$794.95
CD52	Microwave Oven	32	AP	1	$165.00
DL71	Cordless Drill	21	HW	3	$129.95
DR93	Gas Range	8	AP	2	$495.00
DW11	Washer	12	AP	3	$399.99
FD21	Stand Mixer	22	HW	3	$159.95
KL62	Dryer	12	AP	1	$349.95
KT03	Dishwasher	8	AP	3	$595.00
KV29	Treadmill	9	SG	2	$1,390.00

FIGURE 9-2: Premiere Products Part table data

Fragmentation Transparency (continued)

Fragment Part1

PartNum	Description	OnHand	Class	Warehouse	Price
CD52	Microwave Oven	32	AP	1	$165.00
KL62	Dryer	12	AP	1	$349.95

Fragment Part2

PartNum	Description	OnHand	Class	Warehouse	Price
BV06	Home Gym	45	SG	2	$794.95
DR93	Gas Range	8	AP	2	$495.00
KV29	Treadmill	9	SG	2	$1,390.00

Fragment Part3

PartNum	Description	OnHand	Class	Warehouse	Price
AT94	Iron	50	HW	3	$24.95
DL71	Cordless Drill	21	HW	3	$129.95
DW11	Washer	12	AP	3	$399.99
FD21	Stand Mixer	22	HW	3	$159.95
KT03	Dishwasher	8	AP	3	$595.00

FIGURE 9-3: Fragmentation of Part table data by warehouse

Advantages of Distributed Databases

- Local control of data
- Increased database capability
- System availability
- Improved performance

Disadvantages of Distributed Databases

- Update of replicated data
  - Primary copy
- More complex query processing
- More complex treatment of concurrent update
  - **Local deadlock**: occurs at a single site in a distributed database
  - **Global deadlock**: involves more than one site
- More complex recovery measures
  - **Two-phase commit**: one site acts as **coordinator**

Disadvantages of Distributed Databases (continued)

- More difficult management of data dictionary
- More complex database design
- More complicated security and backup requirements

Rules for Distributed Databases

- Local autonomy
- No reliance on a central site
- Continuous operation
- Location transparency
- Fragmentation transparency
- Replication transparency

Rules for Distributed Databases (continued)

- Distributed query processing
- Distributed transaction management
- Hardware independence
- Operating system independence
- Network independence
- DBMS independence

Client/Server Systems

- File server architecture
  - **File server**: stores user files on the network
- **Client/server** architecture
  - **Server**: computer providing data to clients
    - **Back-end processor** or **back-end machine**
  - **Clients**: computers connected to a network and used by users to access data
    - **Front-end processor** or **front-end machine**

Client/Server Systems (continued)

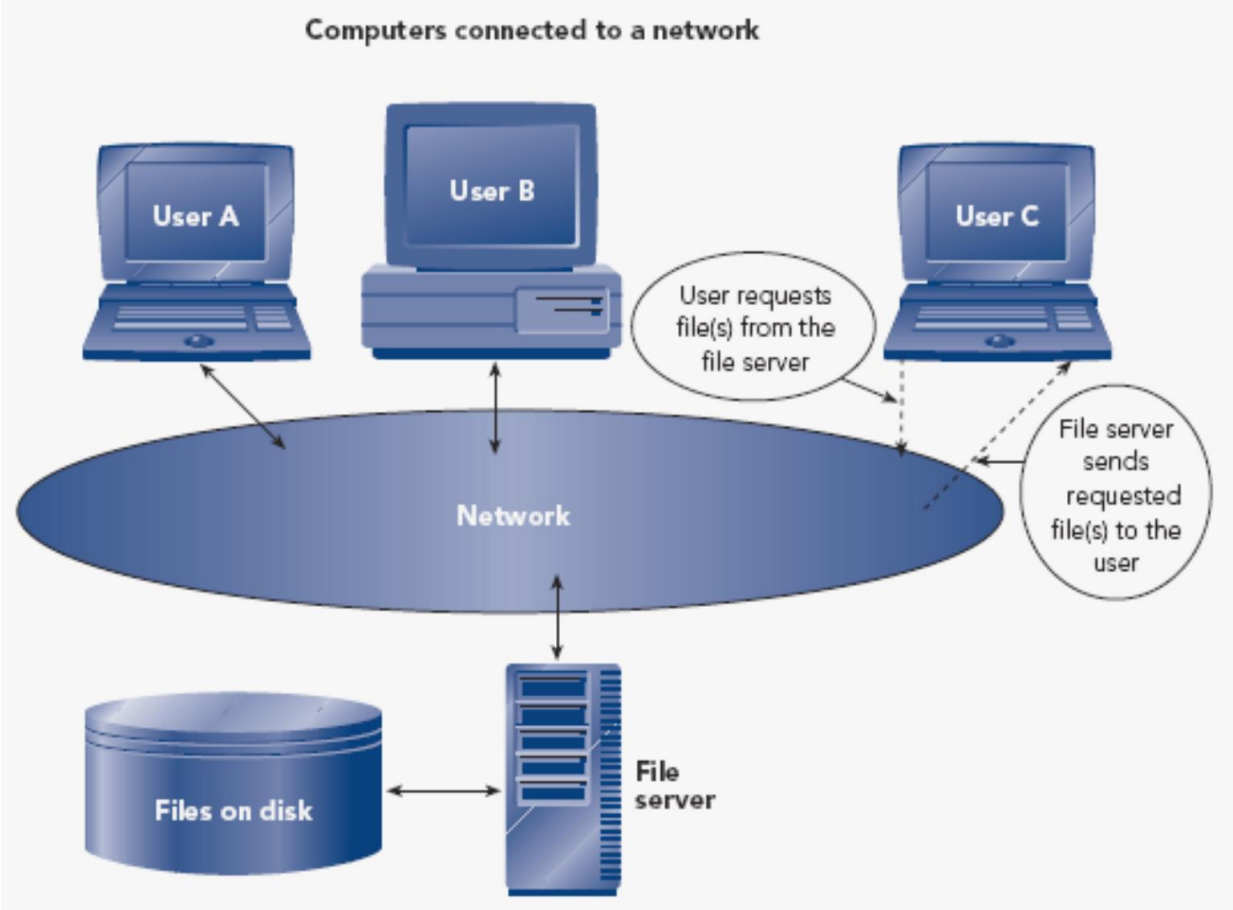

FIGURE 9-4: File server architecture

Client/Server Systems (continued)

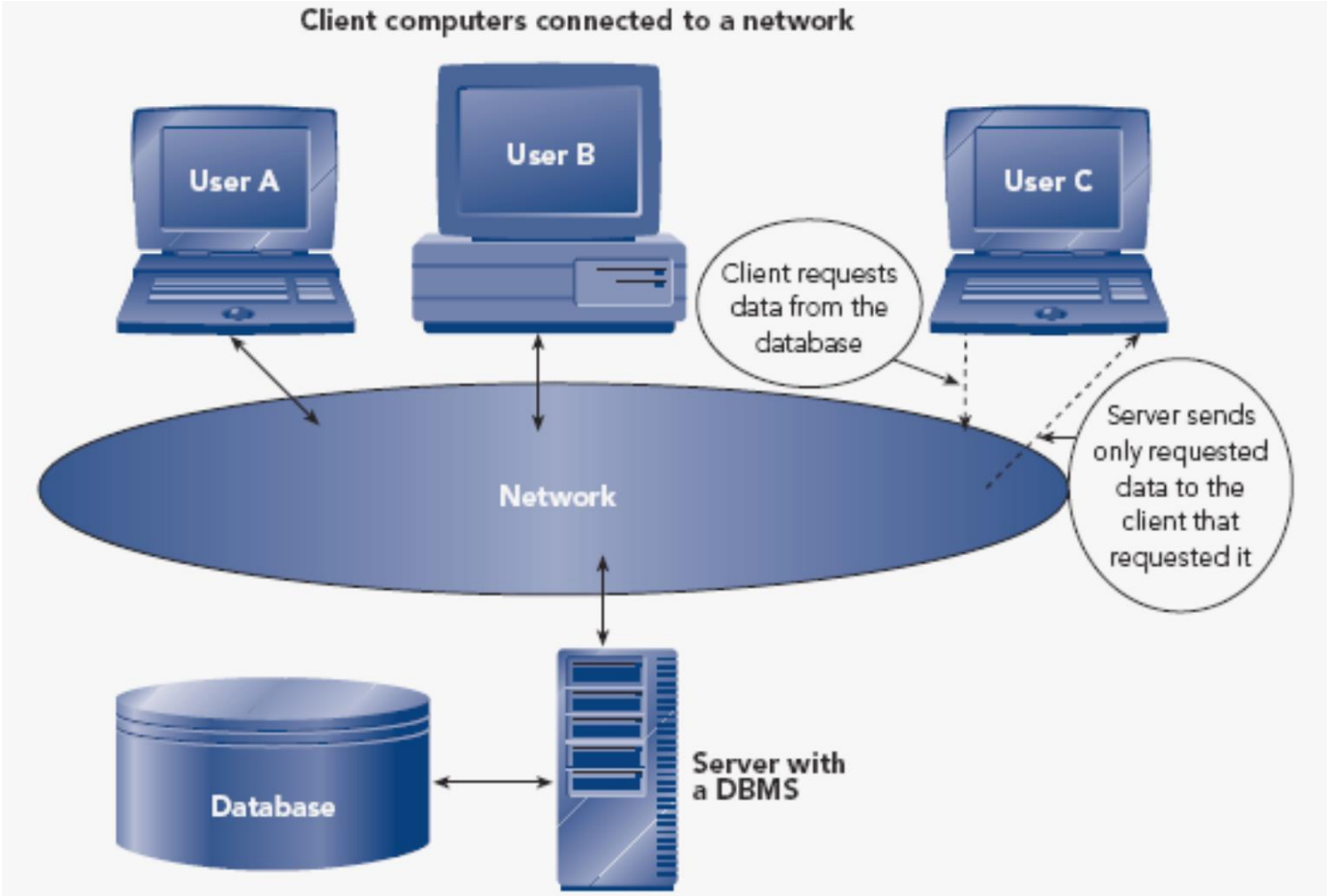

FIGURE 9-5: Two-tier client/server architecture

Client/Server Systems (continued)

- **Two-tier architecture**
  - Server performs database functions
  - Clients perform presentation functions
    - Fat client
    - Thin client
- **Three-tier architecture**
  - Clients perform presentation functions
  - **Database server** performs database functions
  - **Application servers** perform business functions and interface between clients and database server

Client/Server Systems (continued)

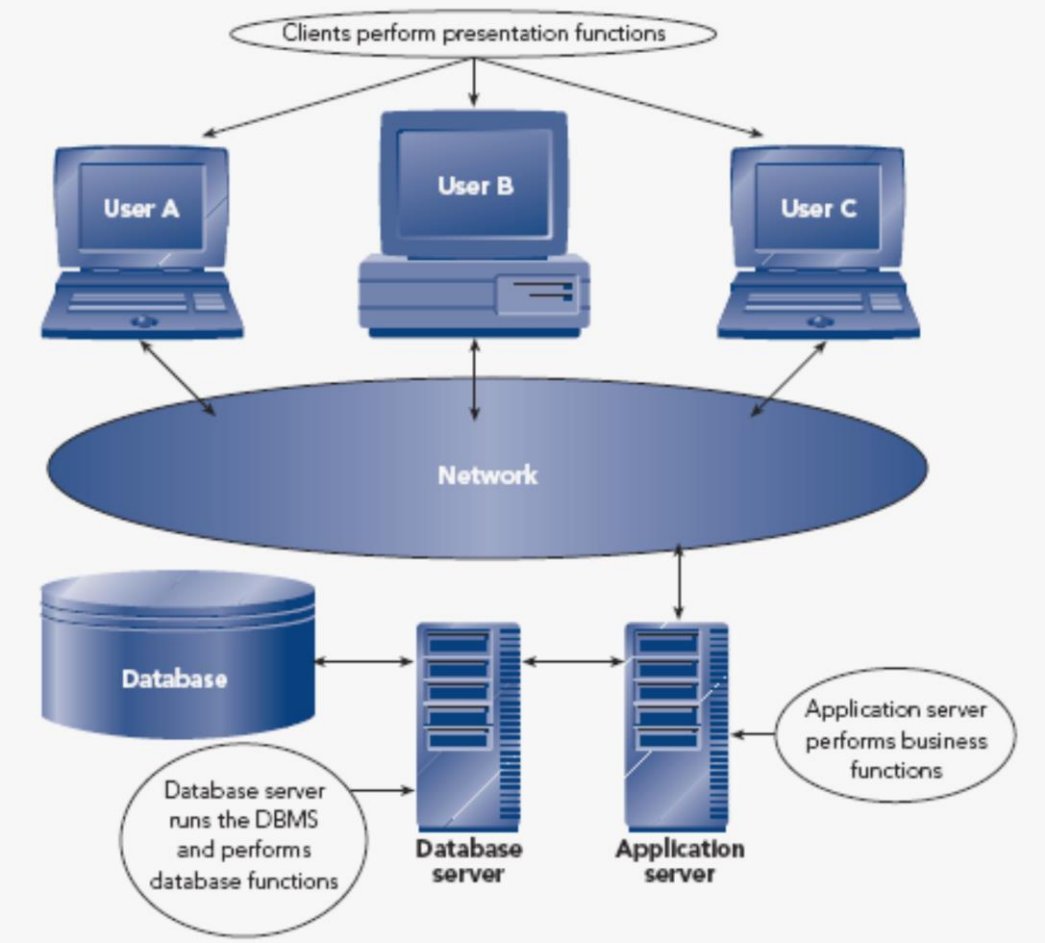

FIGURE 9-6: Three-tier client/server architecture

Advantages of Client/Server Systems

- Lower network traffic
- Improved processing distribution
- Thinner clients
- Greater processing transparency
- Increased network, hardware, and software transparency
- Improved security
- Decreased costs
- Increased scalability

Web Access to Databases

- **Internet** and **World Wide Web** (or the **Web**)
- **Web page**: digital document on the Web
- **Web server**: stores Web pages
- **Web client**: computer requesting a Web page
- Each Web page has a **Uniform Resource Locator (URL)**
- **Hypertext Transfer Protocol (HTTP)**: data communication method used to exchange data on the Internet

Web Access to Databases (continued)

- **Web browser**: computer program that retrieves a Web page from a Web client
- **Transmission Control Protocol/Internet Protocol (TCP/IP)**: standard protocol for communication on the Internet
- Web pages usually created using **Hypertext Markup Language (HTML)**

Web Access to Databases (continued)

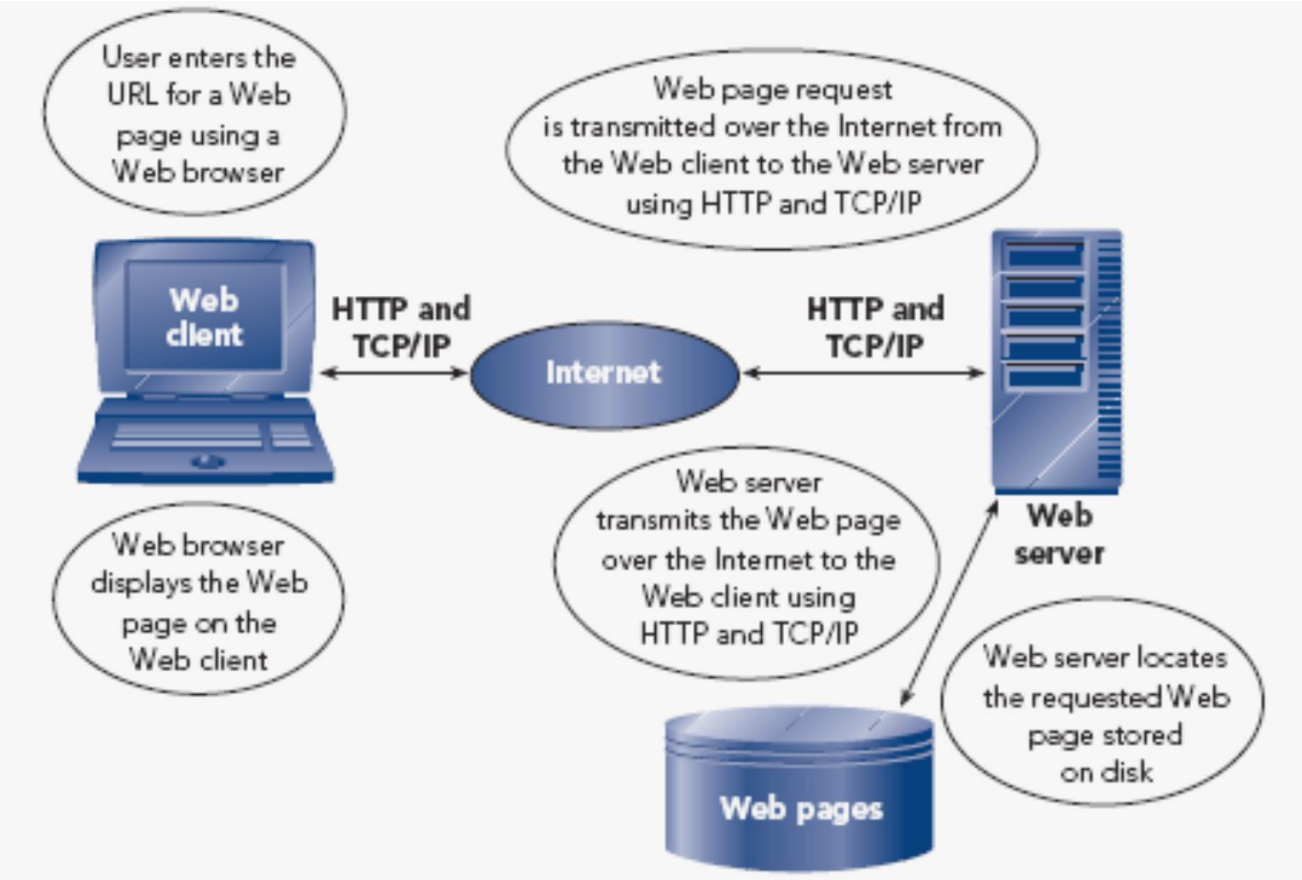

FIGURE 9-7: Retrieving a Web page on the Internet

Web Access to Databases (continued)

- Static vs. dynamic Web pages
  - **Static Web pages**: same content for all Web clients
  - **Dynamic Web pages**: content changes in response to inputs and choices from Web clients
- **Server-side extensions** or **server-side scripts**
- **Client-side extensions** or **client-side scripts**
- Three-tier Web-based architecture
  - Web clients
  - Web server
  - Database server

Web Access to Databases (continued)

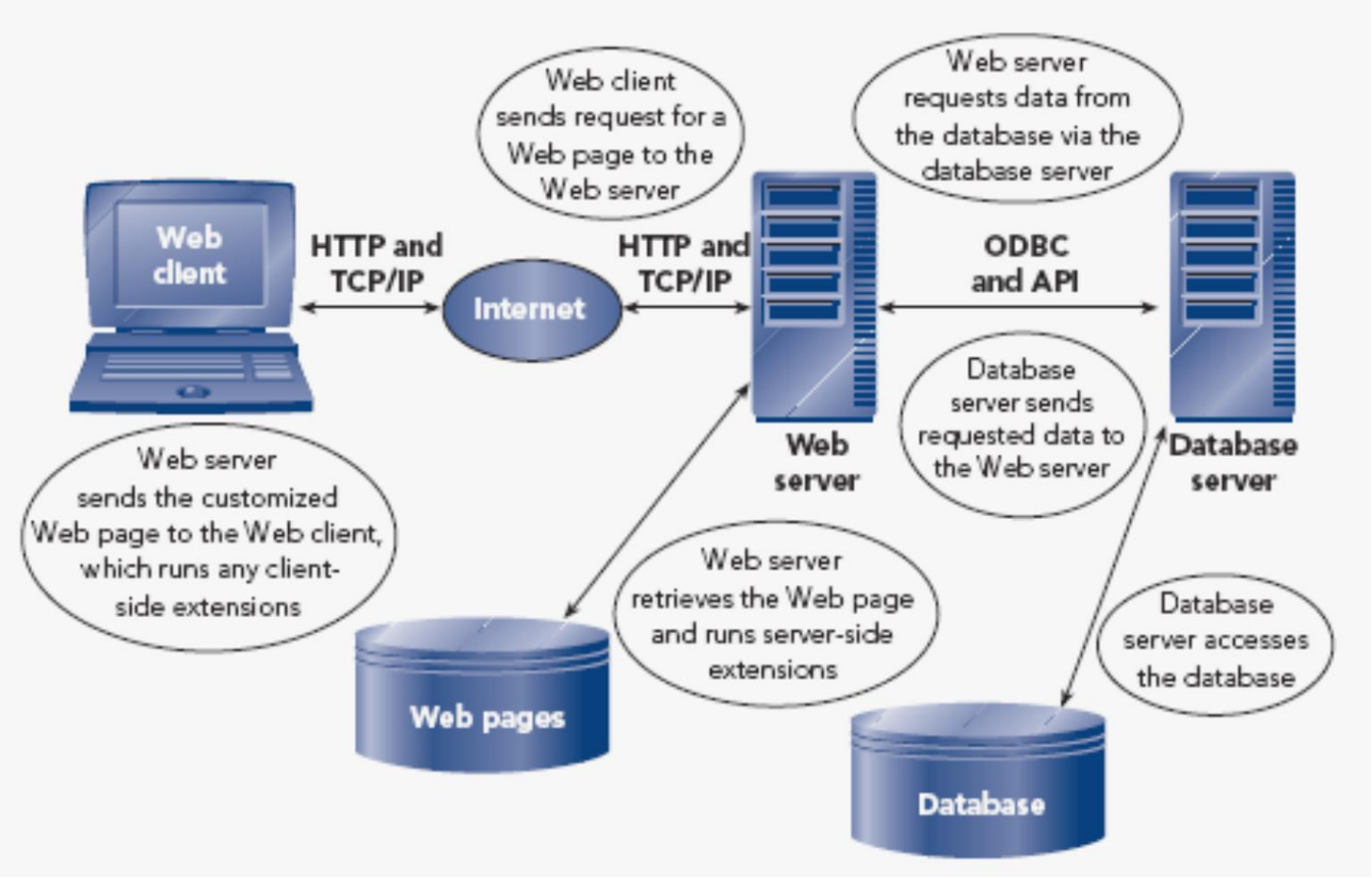

FIGURE 9-8: Three-tier Web-based architecture

XML

- HTML
  - Describes content and appearance of Web pages
  - Does not describe structure and meaning of data
- **Extensible Markup Language (XML)**
  - Tags can define meaning and structure of data
  - An XML document should begin with an **XML declaration**

XML (continued)

- **Extensible Hypertext Markup Language (XHTML)**
  - Markup language based on XML
  - Stricter version of HTML
- Defining structure, characteristics, and relationships of data
  - **Document Type Definition (DTD)**
  - **XML schema**
- Presentation of data
  - **Stylesheet**

XML (continued)

```
<xsd:element name="Rate" minOccurs="0" jetType="double"
       sqlSType="float" type="xsd:double">
<xsd:annotation>
<xsd:appinfo>
<fieldProperty name="ColumnWidth" type="3" value="840"/>
<fieldProperty name="ColumnOrder" type="3" value="0"/>
<fieldProperty name="ColumnHidden" type="1" value="0"/>
<fieldProperty name="DecimalPlaces" type="2" value="255"/>
<fieldProperty name="Required" type="1" value="0"/>
<fieldProperty name="DisplayControl" type="3" value="109"/>
<fieldProperty name="TextAlign" type="2" value="0"/>
<fieldProperty name="AggregateType" type="4" value="-1"/>
<fieldProperty name="GUID" type="9"
       value="CgLbv43o5ECFLODxDEetHA==">
</xsd:appinfo>
</xsd:annotation>
</xsd:element>
```

FIGURE 9-10: XML schema for the Rate element from the Rep table

XML (continued)

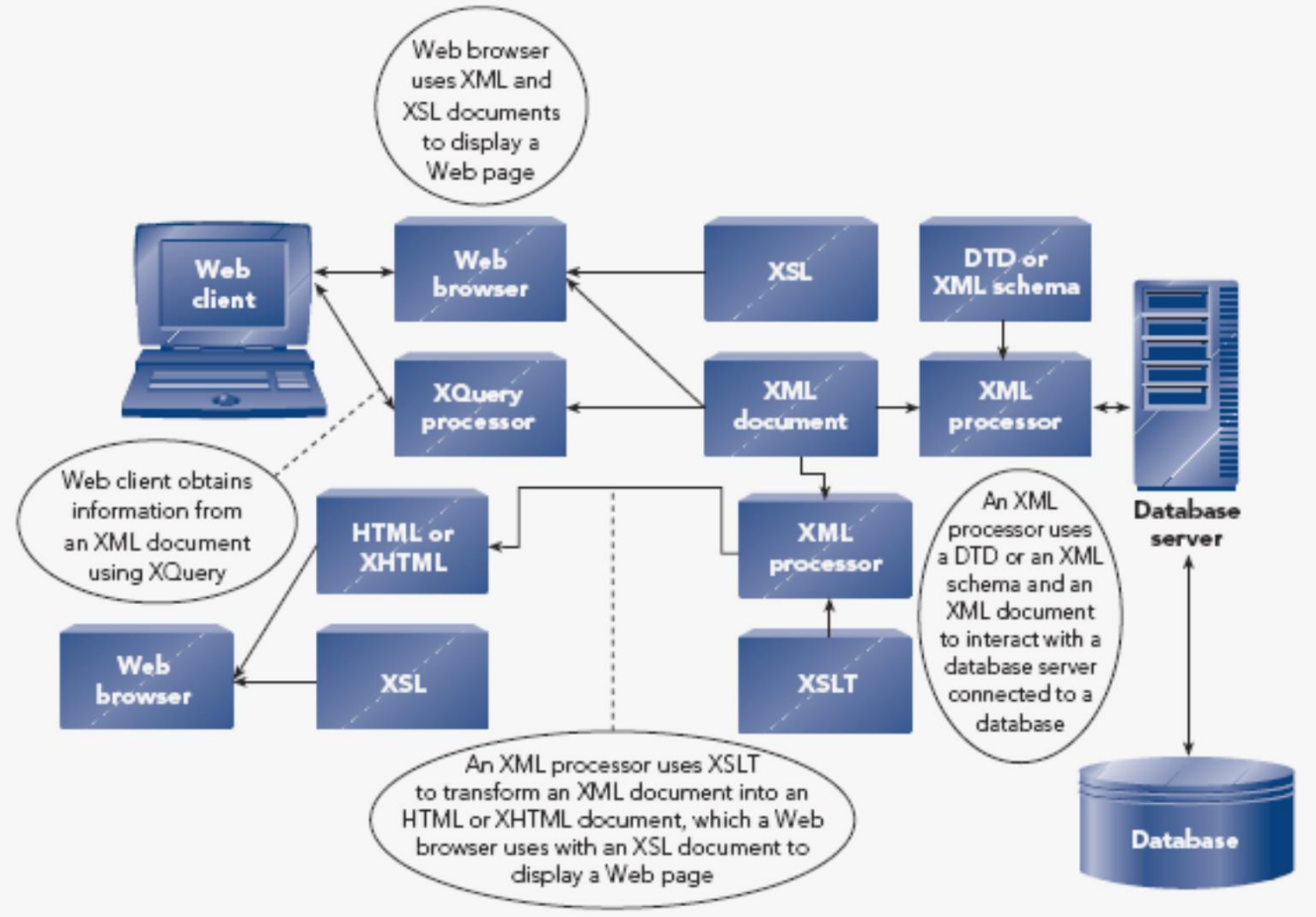

FIGURE 9-11: Interaction among XML and related languages

Data Warehouses

- **Online transaction processing (OLTP)** systems
  - Users use transactions when interacting with an RDBMS
- **Data warehouse**
  - Subject-oriented, integrated, time-variant, nonvolatile collection of data in support of management's decision-making process
  - Used for analysis of existing data
  - Resolves performance issues suffered by operational RDBMSs and OLTPs

Data Warehouses (continued)

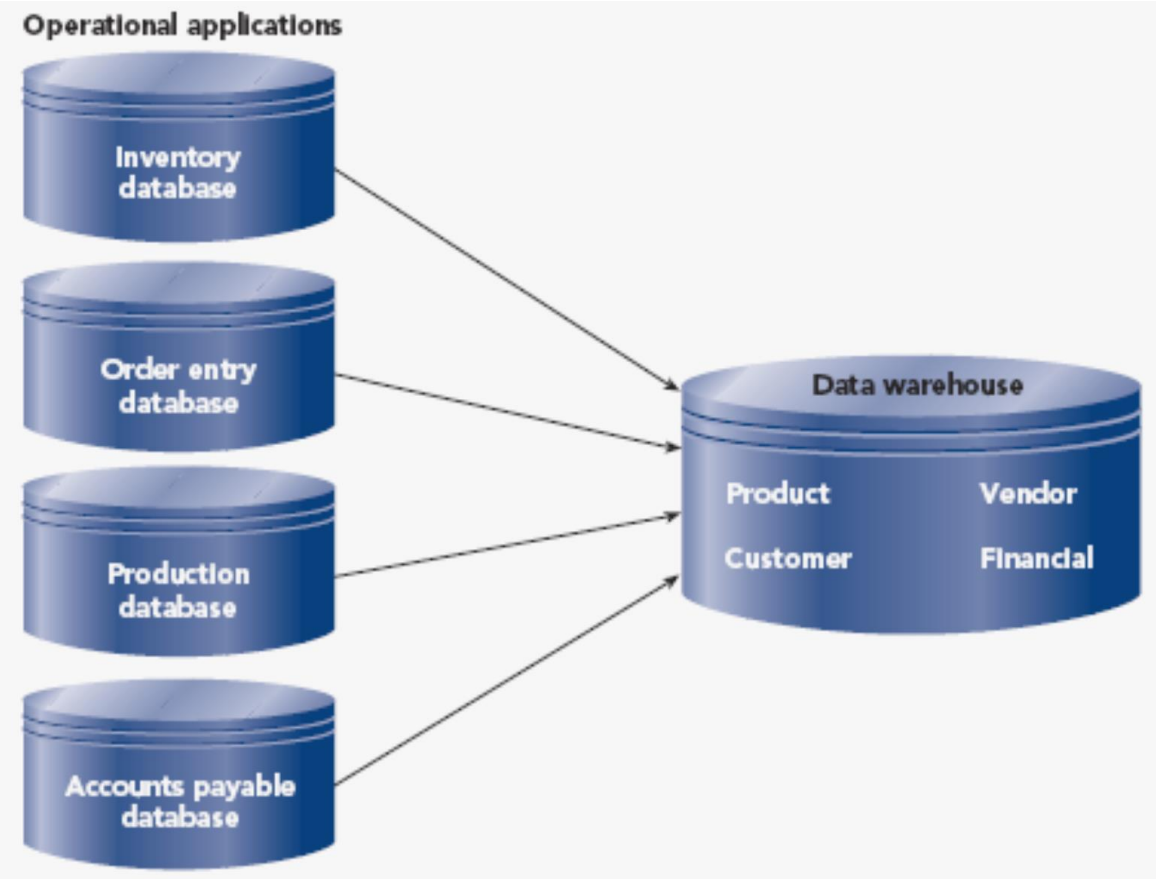

FIGURE 9-12: Data warehouse architecture

Data Warehouse Structure and Access

- **Star schema**
  - **Fact table**
  - **Dimension table**
- **Online analytical processing (OLAP)** software: for access to a data warehouse
- **Data cube**: a shape for visualizing a data warehouse as a multidimensional database
- **Data mining**: uncovering new knowledge, patterns, trends, and rules from data in a data warehouse

Data Warehouse Structure and Access (continued)

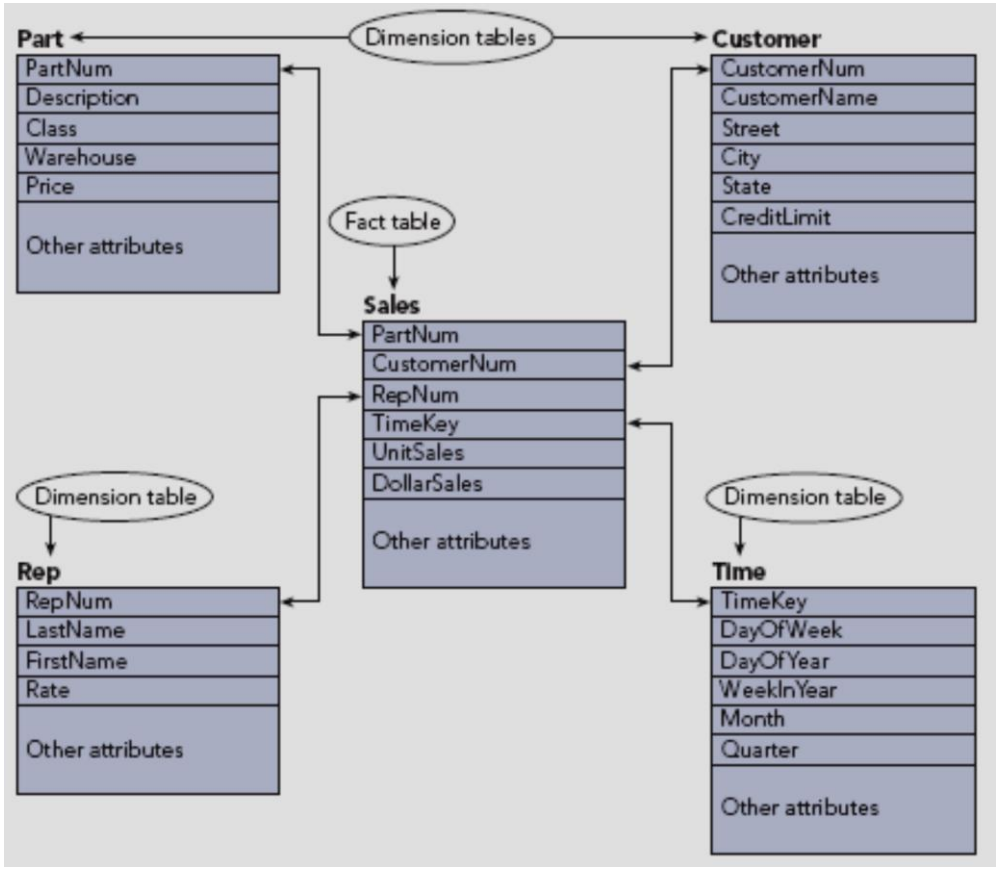

FIGURE 9-13: A star schema with four dimension tables and a central fact table

Data Warehouse Structure and Access (continued)

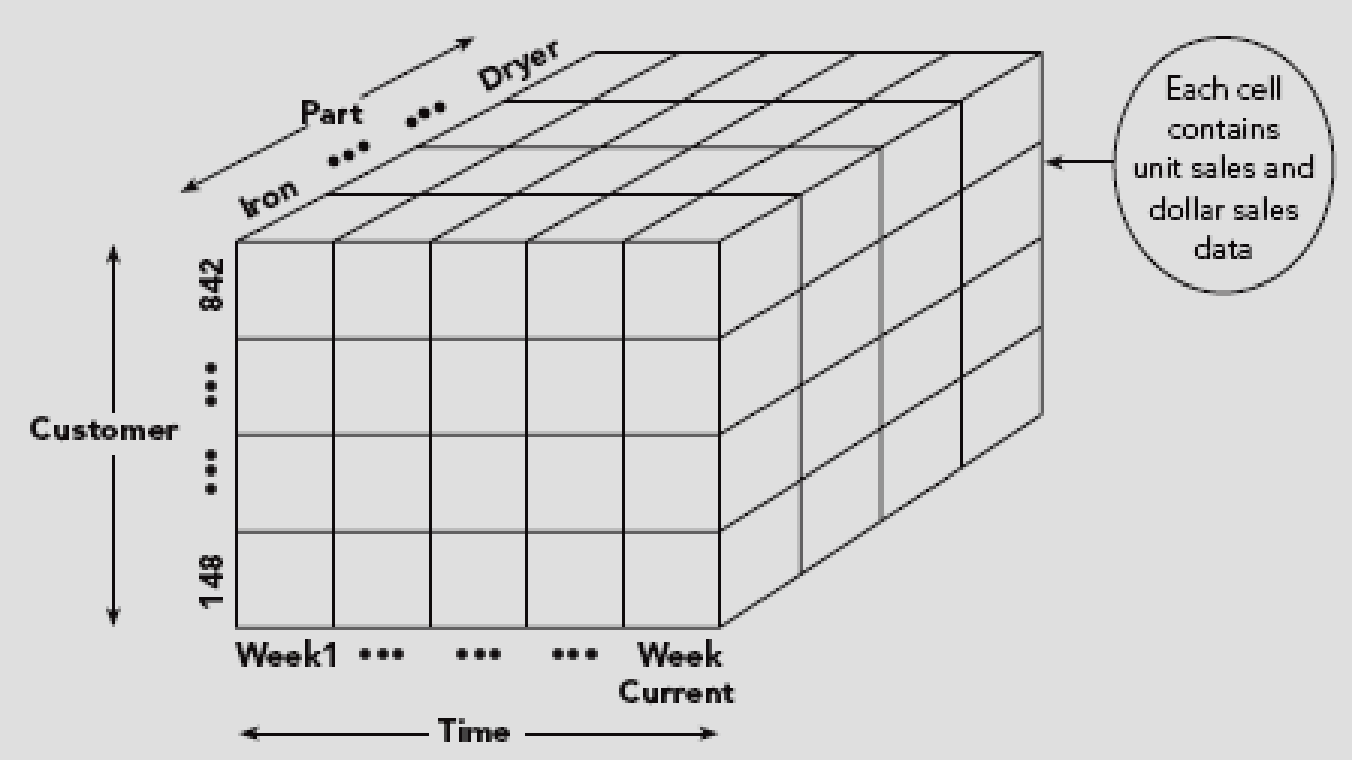

FIGURE 9-14: A data cube representation of the Part, Customer, and Time dimensions

Rules for OLAP Systems

- Multidimensional conceptual view
- Transparency
- Accessibility
- Consistent reporting performance
- Client/server architecture
- Generic dimensionality

Rules for OLAP Systems (continued)

- Dynamic sparse matrix handling
- Multiuser support
- Unrestricted, cross-dimensional operations
- Intuitive data manipulation
- Flexible reporting
- Unlimited dimensions and aggregation levels

Object-Oriented DBMSs

- Complex objects: graphics, drawings, photographs, video, sound, voice mail, spreadsheets, etc.
- RDBMSs store complex objects using special data types
  - **Binary large objects (BLOBs)**
- Object-oriented DBMSs used with applications whose focus is on complex objects

What Is an Object-Oriented DBMS?

- **Object**: set of related attributes along with associated actions
- **Object-oriented database management system (OODBMS)**: database management system in which data and associated actions are **encapsulated** into objects

Objects and Classes

- Represent each entity as an *object* rather than a relation
- List attributes vertically below object names
  - Follow each attribute by name of **domain**
- Objects can contain other objects
- An object can contain a portion of another object

Methods and Messages

- **Methods**: actions defined for a class
- Defined during data definition process
- Executed when user sends a message to the object

Methods and Messages (continued)

```
Add Order (WOrders)
        Add row to Orders table
                OrderNum      = WOrderNum
                OrderDate     = WOrderDate
                CustomerNum = WCustomerNum
        For each order line record in WOrders DO
                Add row to OrderLine table
                        OrderNum      = WOrderNum
                        PartNum       = WPartNum
                        NumOrdered    = WNumOrdered
                        QuotedPrice   = WQuotedPrice
                Update Part table (WHERE PartNum = WPartNum)
                        Allocated     = Allocated + WNumOrdered

Delete Order (WOrderNum)
        Delete row from Orders table (WHERE OrderNum = WOrderNum)
        For each OrderLine record (WHERE OrderNum = WOrderNum) DO
                Delete row from OrderLine table
                Update Part table (WHERE Part.PartNum = OrderLine.PartNum)
                        Allocated     = Allocated – NumOrdered
```

FIGURE 9-22: Two methods for the Premiere Products object-oriented database

Inheritance

- **Subclass**
  - Every occurrence of subclass is considered an occurrence of the class
  - Subclass *inherits* structure and methods of the class

Unified Modeling Language (UML)

- Used to model all aspects of software development for object-oriented systems
  - Includes a way to represent database designs
- **Class diagram**: most relevant diagram type for database design
  - Rectangles represent classes
  - Lines joining classes represent relationships; called **associations**
  - **Visibility symbol** indicates whether other classes can view or update value in attribute

Unified Modeling Language (UML) (continued)

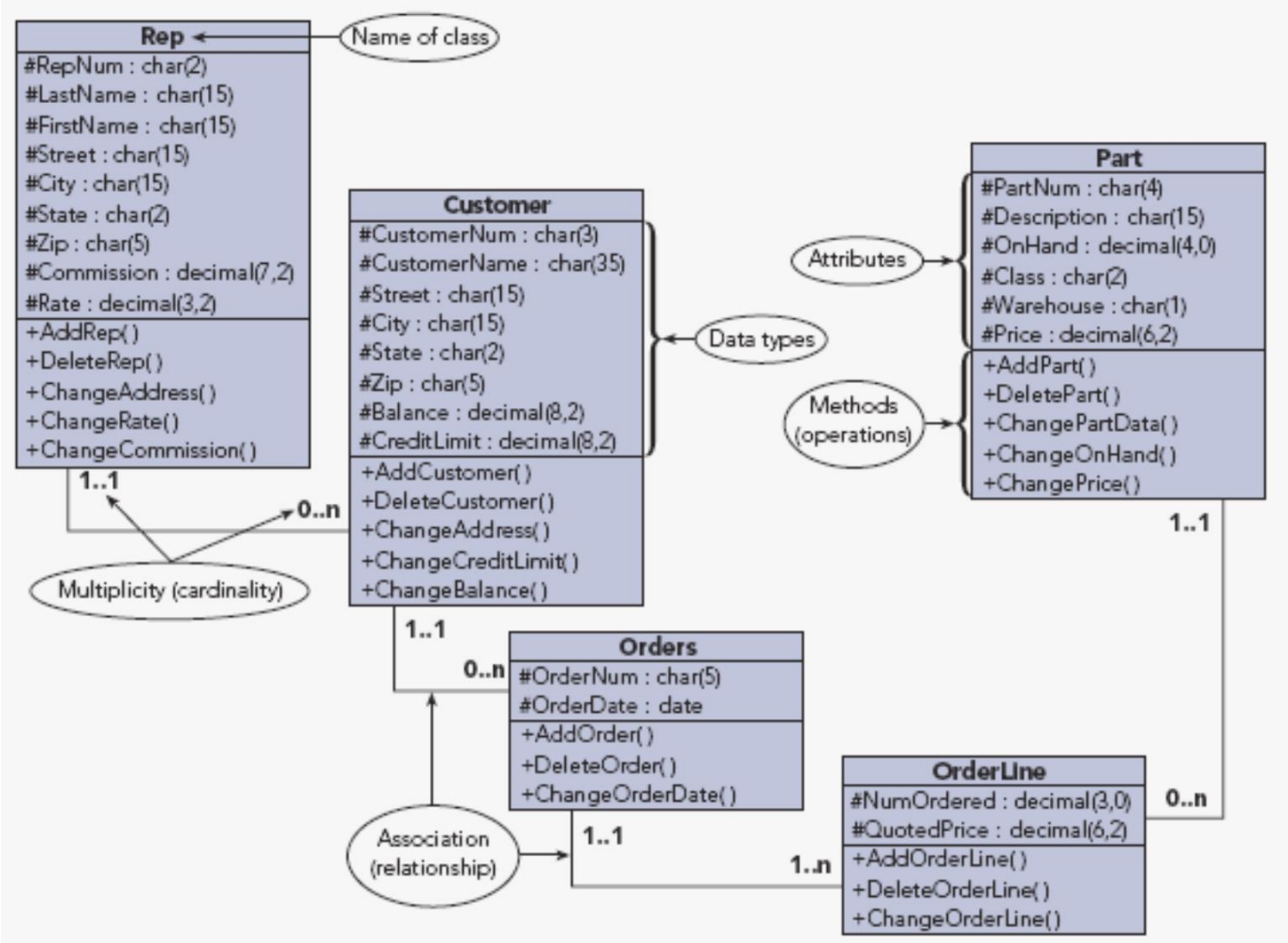

FIGURE 9-24: Class diagram for the Premiere Products database

Unified Modeling Language (UML) (continued)

- **Multiplicity**: number of objects that can be related to an individual object
- Constraints
- **Superclass**
- **Generalization**: relationship between a superclass and a subclass

Unified Modeling Language (UML) (continued)

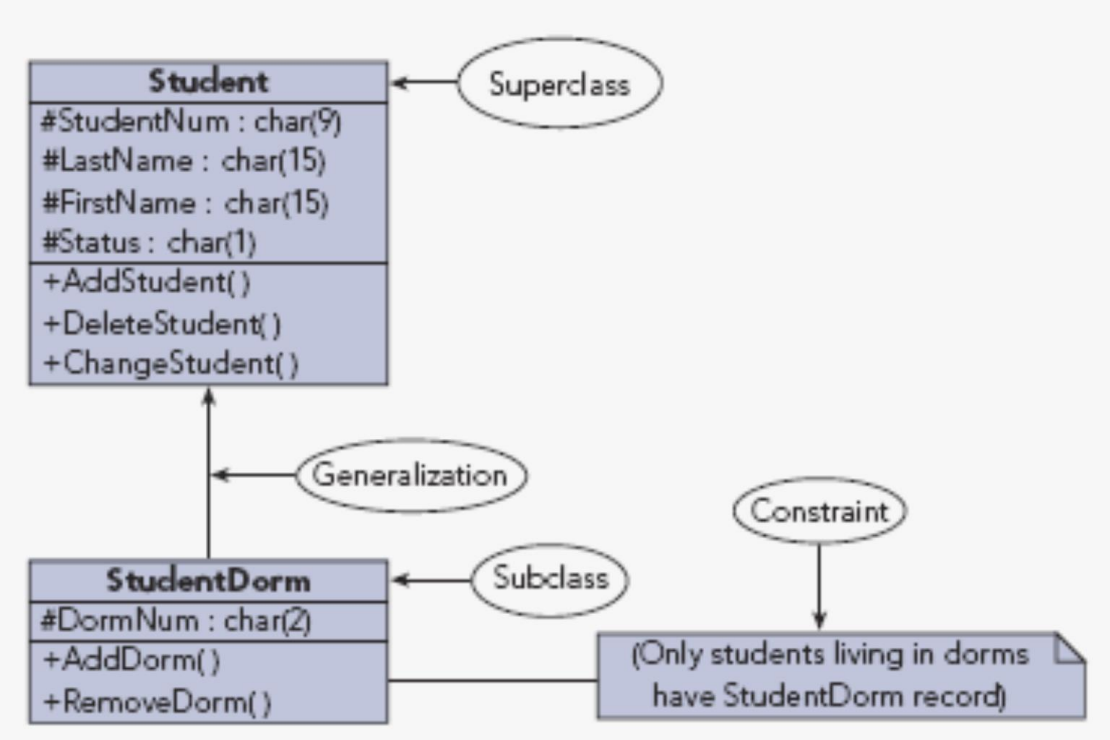

FIGURE 9-26: Class diagram with a generalization and a constraint

Rules for OODBMSs

- Complex objects
- Object identity
- Encapsulation
- Information hiding
- Types of classes
- Inheritance
- Late binding

Rules for OODBMSs (continued)

- Computational completeness
- Extensibility
- Persistence
- Performance
- Concurrent update support
- Recovery support
- Query facility

Summary

- Distributed database: single logical database physically divided among computers at several sites on a network
- Location transparency, replication transparency, and fragmentation transparency are important characteristics of DDBMSs
- Two-tier client/server architecture: DBMS runs on file server and server sends only the requested data to the clients

Summary (continued)

- Three-tier client/server architecture: clients perform presentation functions, database servers perform database functions, and application servers perform business functions
- Web servers interact with Web clients using HTTP and TCP/IP to display HTML Web pages
- Dynamic Web pages, not static Web pages, are used in e-commerce
- XML was developed because of need for data exchange between organizations and inability of HTML to specify structure and meaning of data

Summary (continued)

- XHTML: markup language based on XML; stricter version of HTML
- Data warehouse: subject-oriented, integrated, time-variant, nonvolatile collection of data in support of management's decision-making process
- Users perceive data in a data warehouse as a multidimensional database in data cube shape
- Data mining: uncovering new knowledge, patterns, trends, and rules from data stored in a data warehouse

Summary (continued)

- Object-oriented DBMSs deal with data as objects
  - Object: set of related attributes and actions associated with the attributes
  - OODBMS: database management system in which data and actions that operate on the data are encapsulated into objects
- UML: an approach to model all aspects of software development for object-oriented systems