

16<sup>th</sup> Edition

# Understanding Computers

Today and Tomorrow

Comprehensive

## Chapter 12

# Databases and Database Management Systems

Deborah Morley  
Charles S. Parker

© 2017 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part, except for use as permitted in a license distributed with a certain product or service or otherwise on a password-protected website for classroom use.



# What Is a Database?

- A **database** is a collection of related data stored in a manner that enables information to be retrieved as needed
- A **database management system (DBMS)** is the software used to create, maintain, and access databases
  - Data Dictionary describes data
  - Data Definition Language (DDL) defines data structure
  - Data Manipulation Language (DML) manipulates data

**Textbook** => Database Engine is the part of the program that actually stores and retrieves data



# Advantages and Disadvantages of the DBMS Approach

- Advantages
  - Low level of redundancy
    - data is normalized into separate tables
      - Faster response time
      - Lower storage requirements
      - Easier to secure
      - Increased data accuracy
- Disadvantages
  - Increased vulnerability
  - Security and backup procedures extremely important

# DBMS Advantages – No Redundancy

## File Management Systems

- File management systems cannot retrieve data from more than one table at a time, requiring redundancy

PRODUCT NUMBER	PRODUCT NAME	SUPPLIER	PRICE	CURRENT STOCK	ON ORDER?
A202	Skis	Ellis Ski Co.	90.00	15	Yes
A211	Boots	Ajax Bros.	60.00	90	Yes
A220	Poles	Bent Corp.	25.00	30	Yes
A240	Bindings	Acme Corp.	35.00	25	Yes
A351	Wax	Candle Industries	3.00	8	No


**PRODUCT TABLE**

ORDER NUMBER	PRODUCT NUMBER	SHIPMENT DATE	PRODUCT NAME	SUPPLIER	PRICE	QUANTITY
1001	A202	1/8	Skis	Ellis Ski Co.	90.00	30
1002	A240	1/8	Bindings	Acme Corp.	35.00	15
1003	A211	1/9	Boots	Ajax Bros.	60.00	50
1004	A202	1/9	Skis	Ellis Ski Co.	90.00	40
1005	A220	1/10	Poles	Bent Corp.	25.00	35
1006	A211	1/12	Boots	Ajax Bros.	60.00	60

**INVENTORY ON ORDER TABLE**

**REDUNDANT FIELDS**  
Instead of just having one key field duplicated like in the databases created using a DBMS (green shaded columns), flat files created using file management systems require many more fields to be duplicated (green and blue shaded columns). Notice that the blue shaded columns shown here appear only in the Product table in Figure 12-1; when using a DBMS, these fields are not included in the Inventory and Inventory on Order tables.

Example of file management system tables.



# Individuals Involved with a Database Management System

- Database designers design the database
- Database developers create the database and get it ready for data entry
- Database programmers write the programs needed to access the database or tie the database to other programs
- Database administrators are responsible for managing the databases within an organization
- Users are individuals who enter data, update data, and retrieve information from the database



# Database Models

- Relational databases are the most common today
  - => Relational Database Management System (RDBMS)
  - => or DBMS
- Hierarchical databases - Rare
  - Organizes data in a tree structure
  - Typically a one-to-many relationship between data entities
- Network databases - Rare
  - Allow both one-to-many and many-to-many relationships between data elements
- Object-Oriented databases
  - Next slide



# The Object-Oriented Database Model

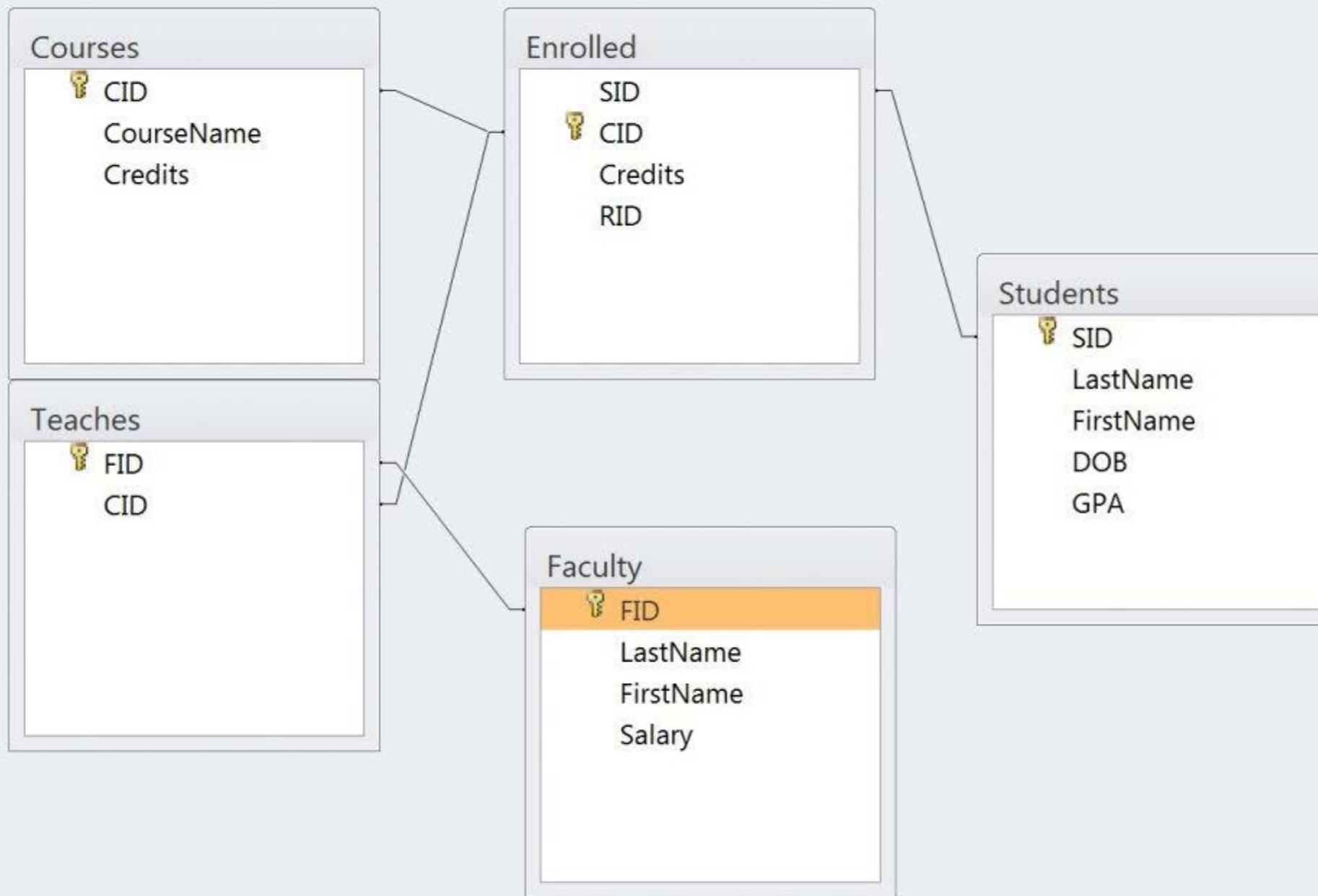
- **Object-oriented database management system (OODBMS)** is a database system in which multiple types of data are stored as objects along with their related code
  - Can contain virtually any type of data (video clip, text with music, etc.) along with the methods to be used with that data (encapsulation)
  - Objects can be retrieved using queries (object query language or OQL)
  - Objects can be reused in other applications to create new applications quickly
  - Note: RDBMS evolved to support this OO functionality



# Relational Database Model (RDBMS)

- With a **relational database management system (RDBMS)**, data is organized in tables related by common fields (Primary Key – Foreign Key)
- Designing a relational database
  - Identify the purpose of the database and the activities it will be used for
  - Determine the necessary fields and tables
  - Assign each field to a table
  - Reorganize as needed to minimize **redundancy** (normalization)
    - ZNF to 3NF
  - Finalize the structure (primary keys, field properties, etc.)







# Entities and Entity Relationships

- An **entity** is a person, object, or event of importance to the organization
  - Entities that the organization wants to store data about typically becomes a database table
  - Single entity presented as a row in the table
    - RDBMS is set theory so order irrelevant
- An **attribute** is a characteristic of an entity
  - Become fields represented as columns
    - RDBMS is set theory so order irrelevant
- A relationship is an association between two or more entities (i.e. across tables)



# DBMS Tables (Data Structure)

## Tables (normalized data)

- Collection of related **records/entities**
  - **Interrelated** with other tables in the database through **primary/foreign keys**
  - Data separated into separate tables in lossless decomposition (keys facilitate re-aggregation).
- **Attributes/Fields (columns)**
    - Single category of data (mathematical domain) to be stored in a database (name, telephone number, etc.)
  - **Entities/Records (rows)**
    - Collection of related fields in a database (all the fields for one customer, for example)

# Primary Key

- A **primary key** is a field that uniquely identifies the records in a table
  - Used in a relational database to relate that table to other tables

The yellow fields will always contain unique data and so are good to use as primary keys.		The blue fields will not always contain unique data and so are not good to use as primary keys.	
Product Number	Product Name	Supplier	Price
Student ID Number	Student Name	Address	Phone
Customer Number	Address	Phone	Balance

**FIGURE 12-2**

**Primary key fields.**

A primary key field must contain unique data so it can be used to identify each record in the table.

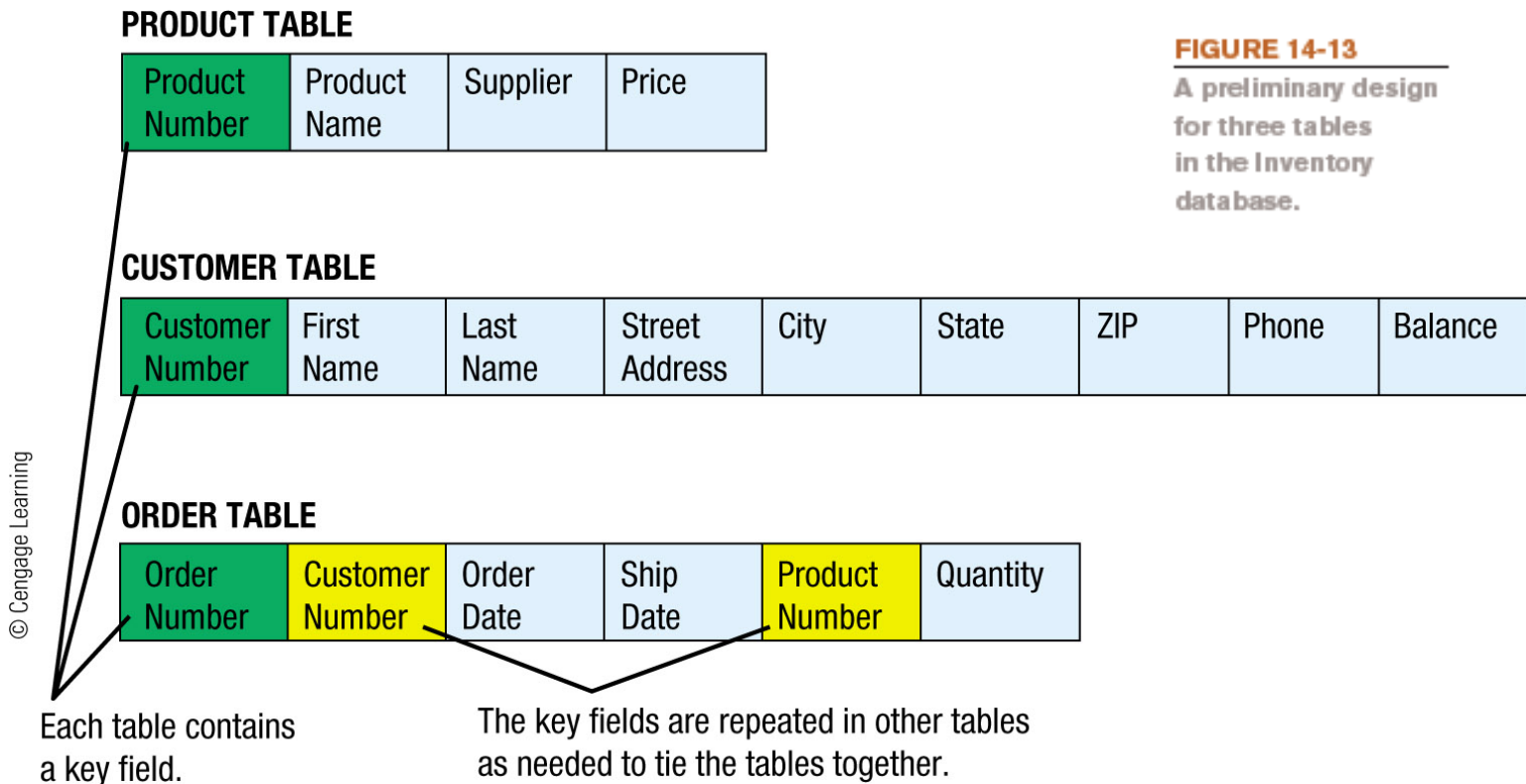


# Relating Tables

- Once all tables have been created, they can be related to one another using their primary-foreign key pairs
  - Primary key in one table used to reconnect/extract related data from other tables as needed
  - The same primary key field must in both tables to be related (called foreign key in 2<sup>nd</sup> table)
  - Once tables are related, data from both tables can be displayed when viewing one of those tables, a report, or other object associated with one of the tables

# Normalization, Primary & Foreign Keys

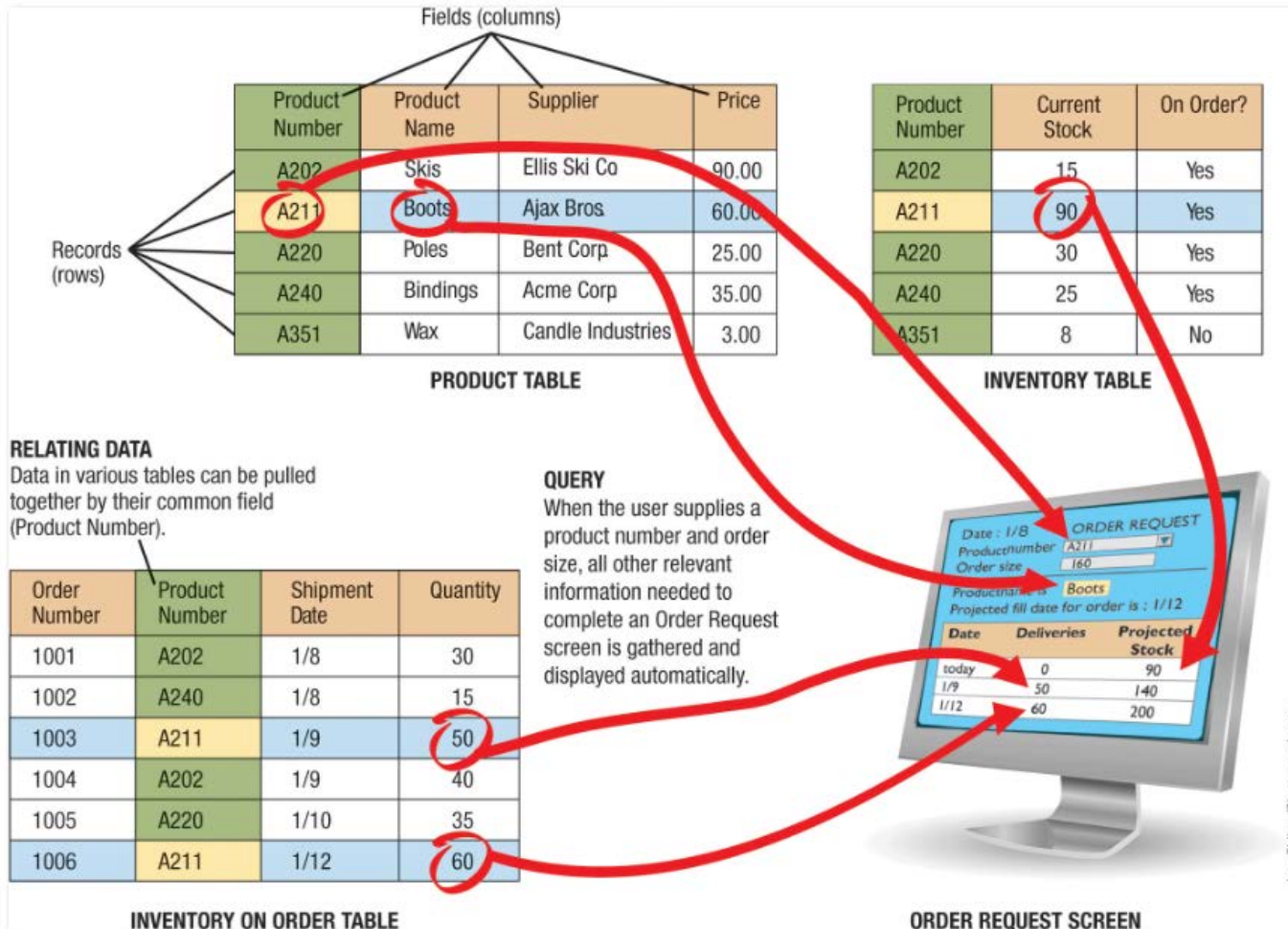
Primary keys become Foreign keys in related tables



**FIGURE 14-13**

A preliminary design for three tables in the Inventory database.

# Related Entities



**FIGURE 12-1**  
Using a relational database in an inventory system.

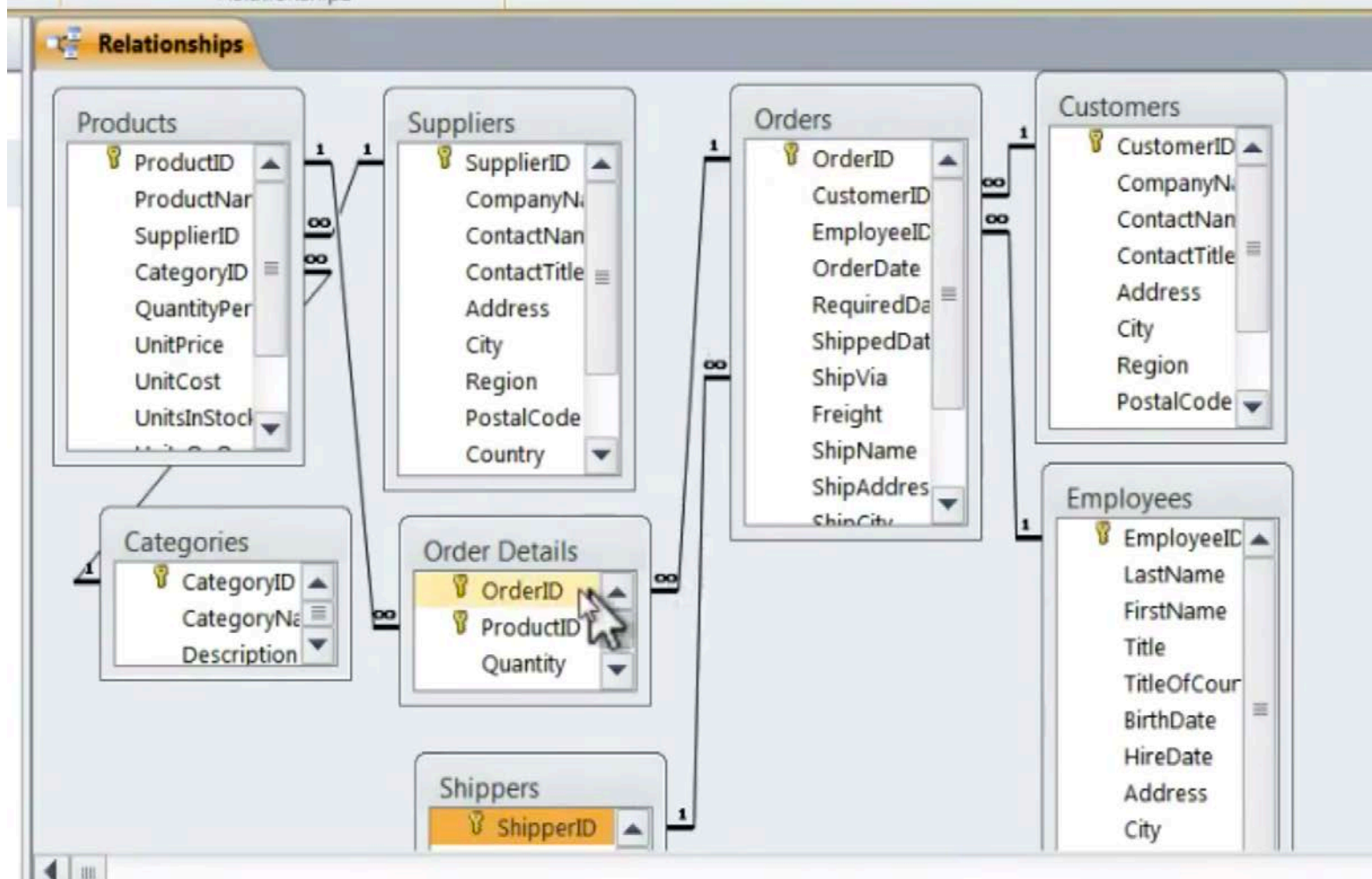


# Entity Relationships - Cardinality

- One to one (1:1) entity relationships
  - One entity is related to only one other entity of a particular type
  - Not a common type of relationship
    - Ex: employee and company car
- One to many (O:M) entity relationship
  - Most common type of relationship
  - One entity can be related to more than one other entity
    - Ex: Department has many employees
- Many to many (M:M) entity relationships
  - One entity can be related to more than one other entity, that are also related to multiple entities
    - Ex: Many employees working on many projects



# Cardinality





# Data Definition

- **Data definition** is the process of describing the properties of data to be included in a database table
  - During data definition, each field is assigned:
    - Name (must be unique within the table)
    - Data type (such as Text, Number, Currency, Yes/No, etc.)
    - Description (optional description of the field)
    - Properties (field size, format of the field, allowable range, if field is required, initial value, etc.)
  - Finished specifications for a table become the table structure
  - Recorded in Data Definition Language (DDL)

# Data Definition Example

**Indicates this field is the primary key.**

**Fields, data types, and descriptions.**

**Field size for Product Number.**

**Indicates the pattern Product Number data must follow (one letter followed by three numbers).**

**A validation rule can be entered here.**

**Product Number field is required and cannot be left blank.**

**TABLE STRUCTURE**  
The table structure specifies the fields and their characteristics.

**Properties of current field (Product Number).**

**TABLE DATA**  
The data is entered into the table in the appropriate fields.

**A new record can be added here; it would become the 6th record in this table.**

Field Name	Data Type	Description (Optional)
Product Number	Short Text	Company assigned product number
Current Stock	Number	Quantity available for sale
On Order?	Yes/No	If the product is on order

Product Number	Current Stock	On Order?	Click to Add
A202	15	<input checked="" type="checkbox"/>	
A211	90	<input checked="" type="checkbox"/>	
A220	30	<input checked="" type="checkbox"/>	
A240	25	<input checked="" type="checkbox"/>	
A351	8	<input type="checkbox"/>	
*	0	<input type="checkbox"/>	



# Data Dictionary

- The **data dictionary** contains all data definitions in a database
  - Table structures
  - Security information (passwords, etc.)
  - Relationships between the tables in the database
  - Basic information about each table (e.g., # of records)
    - Contains **metadata**, or information about the database tables
    - Does not contain any of the data in the tables
  - Ensures that data being entered into the database does not violate any of its assigned properties



# Tables, Queries, Forms, Reports

- External schema/view
- Tables contain the data
- Queries retrieve but also can insert and update
- Forms provide safe mechanism to add/update data
- Reports present retrieved information



## 3 Views/Schemas to DBMS

3 layers of abstraction supported by the DBMS (Known as schemas or views – not in textbook)

1. Physical schema – Physical storage & access (DB has a single physical schema)
2. Conceptual schema – Designed by DBA => normalized data (DB has a single conceptual schema)
3. External schema – Users perception & data access (many conceptual schemas for different users)



# Queries (External Schema)

- Retrieving Information from a Relational Database:
- A **query** is a request to see information from a database that matches specific criteria (data driven)
  - In Access, can create a query object that specifies what fields and records should be displayed
  - Or can write a query using **structured query language (SQL)**
  - Each time a query is run, the data currently meeting the specified conditions is displayed
  - Must be designed to extract information as efficiently as possible
  - Poorly written queries can impact the overall performance of the system
- Queries can also update and insert data in DBMS



# Example SQL

## Generic Format

**SELECT:** Fields (columns) to be shown

**FROM:** Table

*JOINED Tables/Criteria (generic pseudocode)*

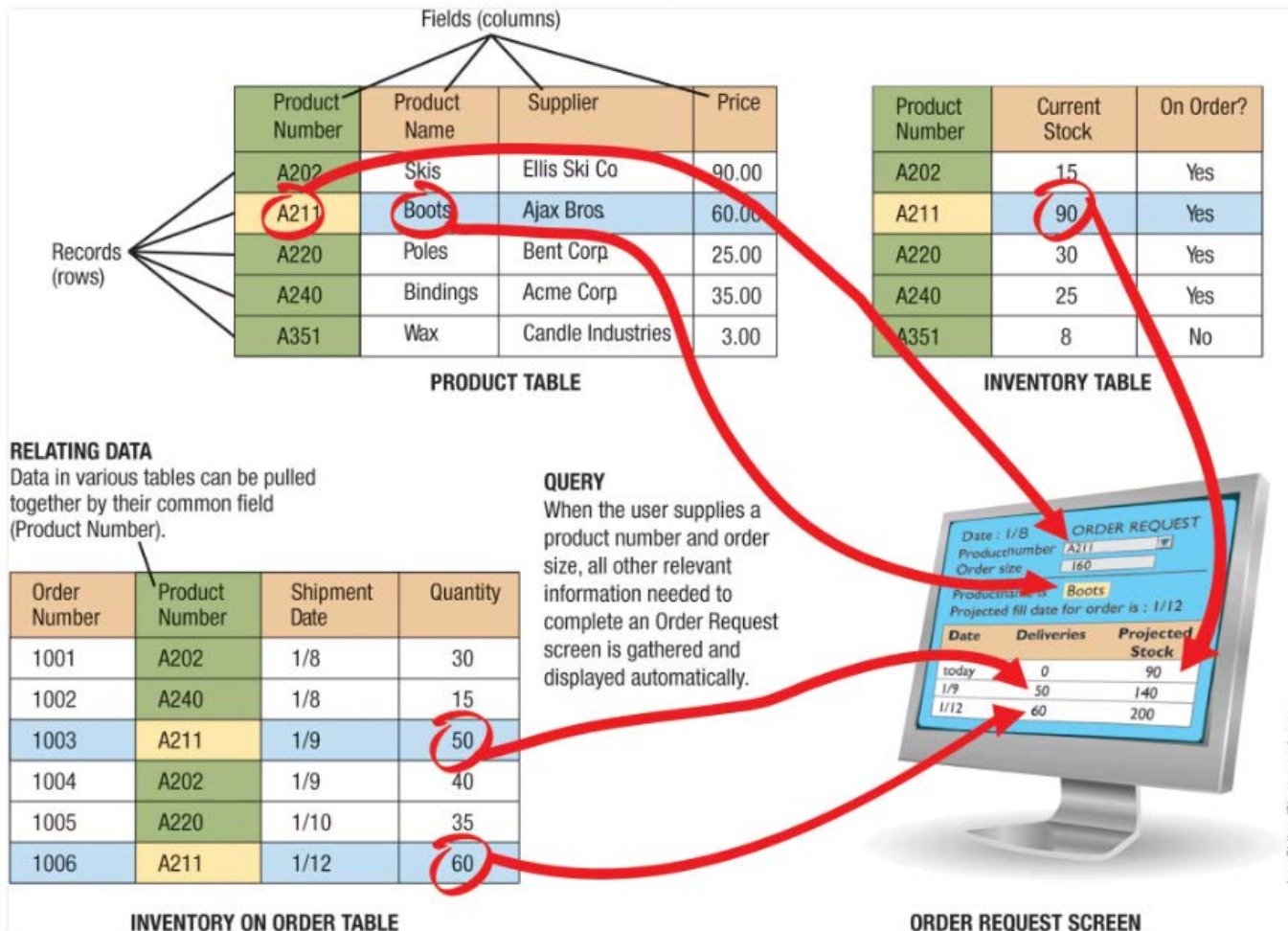
**WHERE:** Specify some criteria/condition

**Ex:** Joining Data from Products and Inventory tables

```
SELECT Products.ProductName, Products.Supplier, Products.Price, Inventory.CurrentStock  
FROM Products  
INNER JOIN Inventory  
On Products.ProductNumber = Inventory.ProductNumber
```



# Related Entities



**FIGURE 12-1**  
Using a relational database in an inventory system.



# Forms

- Forms used to enter and edit data
- Used when data needs to be migrated into the tables
- New data can be added using the table's Datasheet view or via a **form** but => in truth, never allow direct entry
  - Datasheet view displays more than one record at a time
  - Forms
    - Must be created for a particular table
    - Must be opened to be used to view or edit data
    - Display one record at a time
  - In either case, the same data is being viewed or manipulated



# Reports

- **Reports** are a formatted way of looking at a database table or the results of a query
  - Can pull data from more than one table (if related)
  - Many programs have wizards or other tools to make it easy to create a report
  - In Access, reports are often created using the Report Wizard and then modified as needed using the report's Design view
  - Reports in Microsoft Access are saved as objects in the database file
  - When a report is opened, the current data is displayed in the specified format



# Maintaining a Relational Database

- Common maintenance tasks:
  - Upgrading database software and installing patches as they become available
  - Repairing or restoring data that has become corrupt
  - Modifying the table structures to add new fields or change properties of fields as needed
  - Adding new indexes to speed up queries
  - Deleting obsolete data
  - Evaluating and improving security



# DBMS Data Organization/Access

- Data organization arranges data for efficient retrieval
  - (based on data not computer addresses)
- **Indexed organization**
  - Uses an **index** to keep track of where data is stored in a database
- **Direct organization**
  - Uses hashing algorithms to specify the exact storage location
  - Algorithms should be designed to limit collisions
- Some systems use a combination of both indexed and direct organization

# Example of Direct Organization

## HASHING PROCEDURE

Prime number

$$\begin{array}{r} 4 \\ 23 \overline{) 102} \\ \underline{92} \\ 10 \end{array}$$

Remainder:

10

1. The primary key value (in this case the Customer Number) is divided by a prime number.
2. The remainder indicates the location to be used for that record (in this case, 10).

**FIGURE 12-8**

Direct organization is frequently used for faster real-time processing.



# DBMS Advantages: Integrity & Validation

- **Data integrity** refers to the accuracy of data
  - Quality of data entered determines the quality of generated information
- **Data validation** refers to the process of ensuring that data entered into the database is valid
  - Ensures entered data matches the specified data type, format, and allowable value for each field
  - Can include record validation rules (checking the value of a field with the value of another field to ensure validity)
  - If data is invalid, an error message is usually displayed
  - Can be enforced on a per transaction basis so that the entire transaction will fail if one part is invalid
- Database locking prevents two individuals from changing the same data at the same time

# Data Validation Example

## WRONG DATA TYPE

Only data matching a field's assigned data type may be entered into that field.

The screenshot shows the Microsoft Access interface with the 'Inventory' table open. The 'Current Stock' field for product 'A500' contains the value 'A'. A yellow warning icon is visible next to the field. A tooltip message reads: "The value you entered does not match the Number data type in this column." Below the tooltip are options: "Enter new value.", "Convert the data in this column to the Text data type.", and "Help with data types and formats." A red circle labeled 'A' highlights the value 'A' in the 'Current Stock' column.

Product Number	Current Stock
A202	15
A211	90
A220	30
A240	25
A351	8
A500	A

The screenshot shows a dialog box titled "Microsoft Access" with a yellow warning icon. The message reads: "One or more values are prohibited by the validation rule '>=0' set for 'Inventory.Current Stock'. Enter a value that the expression for this field can accept." There are "OK" and "Help" buttons. In the background, the 'Inventory' table is visible with the 'Current Stock' field for product 'A500' containing the value '-1'. A red circle labeled '-1' highlights this value. A red circle labeled 'A' is also visible on the 'Current Stock' field in the table view, indicating the source of the error.

**VALIDATION RULE VIOLATION**  
Only data conforming to a field's assigned validation rule may be entered into that field.

**FIGURE 12-5**

## Data validation.

Using appropriate data properties can prevent invalid data from being entered into a database table.





# DBMS Advantages => Security

- **Data security** protects data against destruction and misuse (both intentional and accidental)
  - Protects against unauthorized access to and unauthorized use of a database and data loss
    - Firewalls, access controls, access privileges, etc.
  - Database activity monitoring programs can be used to detect possible intrusions and risks
  - Database encryption should be used
  - Strict backup and disaster-recovery procedures can protect against data loss due to database failure, accidental deletions, disasters, etc.



# DBMS Advantages: Privacy

- **Data privacy** addresses protecting the privacy of personal data stored in databases
  - Many states require businesses to notify customers when their personal data has been compromised
  - Data breaches can be costly (Recall ROI of Security)
    - One estimate is \$200 per breached record
  - To protect the privacy of data, businesses should:
    - Make sure all data they are collecting and storing is necessary
    - Make sure they use adequate security measures



# DBMS Types/Classifications

- **\*Note types/classifications can overlap or belong to 2 different classifications**



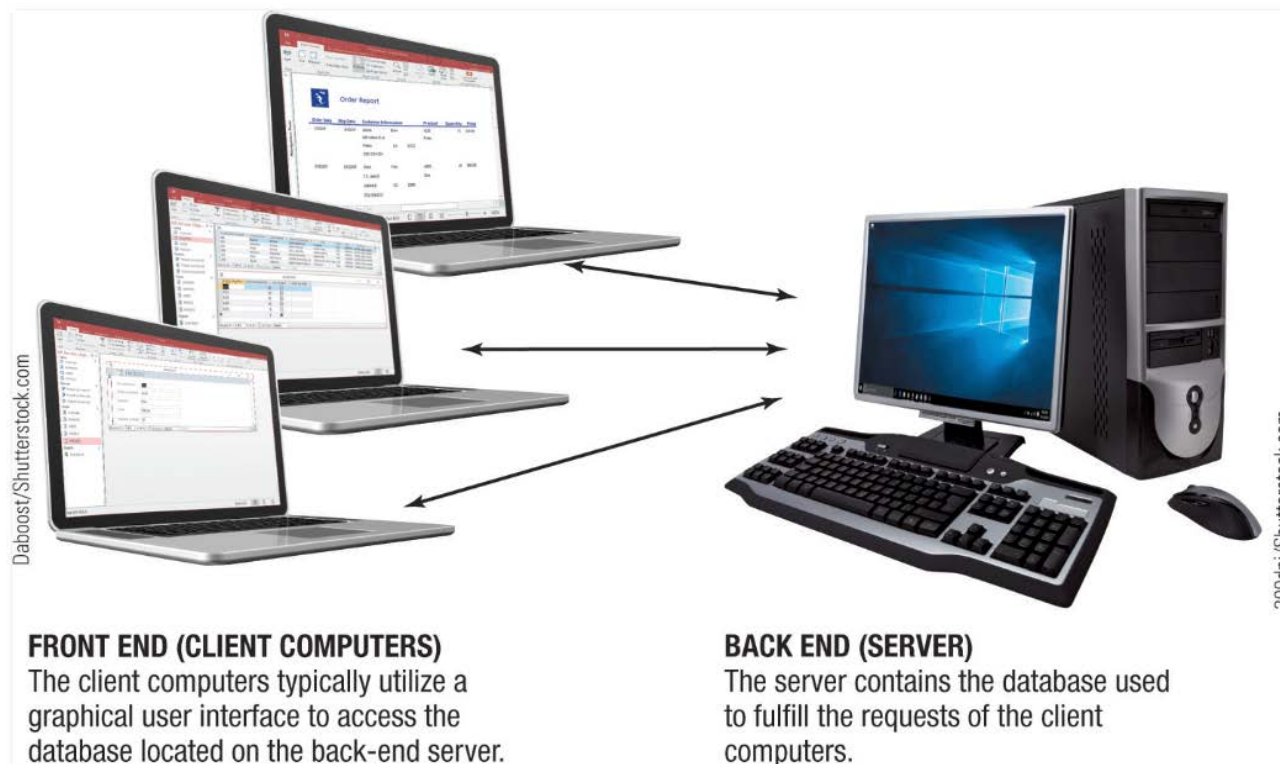
# DBMS Types –Single/Multi-User

**Note types/classifications can overlap or belong to 2 different classifications**

- **Single-user database system**
  - Located on a single computer
  - Designed to be accessed by one user
  - Widely used for personal applications and very small businesses
- **Multiuser database system**
  - Designed to be accessed by multiple users (most business databases today)

# DBMS Types: Client-Server

- A **client-server database system** has both clients (front end) and at least one database server (back end)



**FIGURE 12-9**  
Client-server  
database systems.

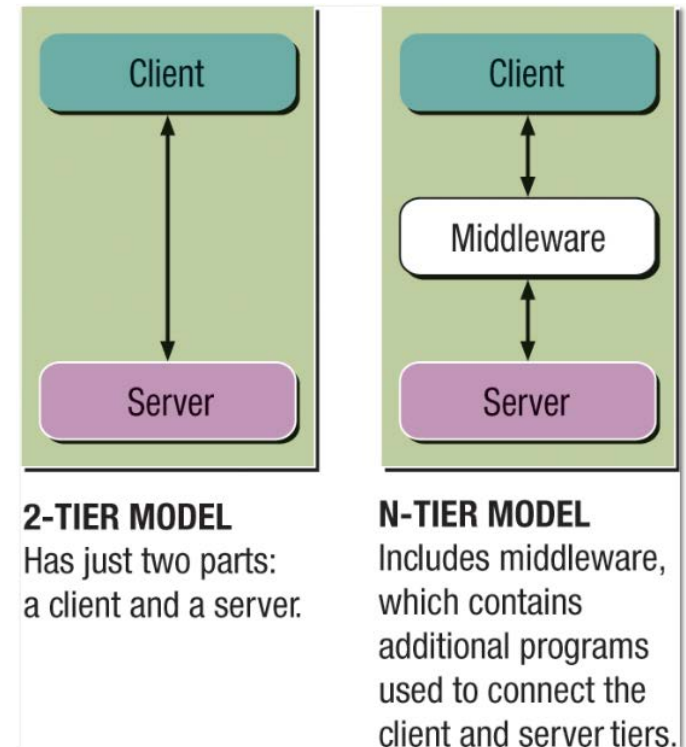


# Cloud Databases


- **Cloud databases** are accessible to users via the Web
  - Used in conjunction with businesses Web sites to display product information, facilitate online ordering, etc.
  - Used to facilitate searching for information
  - Store user-generated content (Flickr, YouTube, Facebook, etc.)
  - Allow Web pages to be dynamic Web pages

# DBMS Types: N-Tier DBMS

- N-tier database systems have at least one middle component between the client and the server
  - Additional tiers typically contain middleware to connect to a database
  - Allows program code to be separate from the database
  - Code can be divided into any number of logical components
  - Tiers can be used with different platforms



**FIGURE 12-10**  
A 2-tier vs. an n-tier database model.



# DBMS Types: Centralized vs. Distributed

- **Centralized database system**
  - Database is located on a single computer, such as a server or mainframe
- **Distributed database system**
  - Data is physically divided among several computers connected by a network, but appears as a single database to users
  - Allows data to be stored at the site where it is needed most frequently or that makes data retrieval most efficient
  - DNS, Cryptocurrency





# DBMS Types: Disk-Based vs. In-Memory

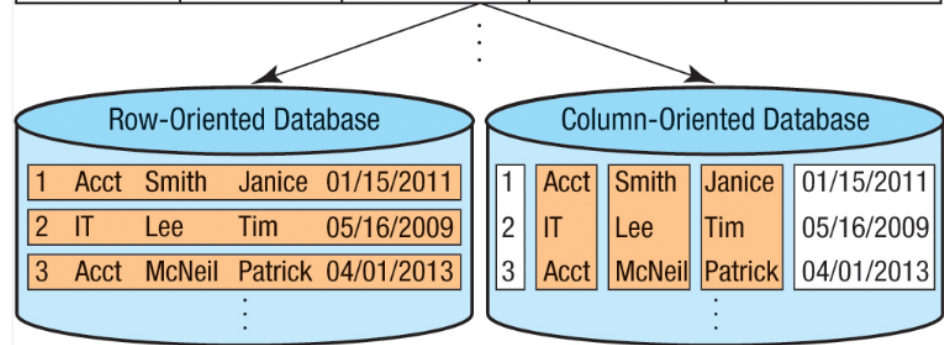
- Disk-based systems
  - Data is stored on hard drives
- **In-memory databases (IMDBs)** (main memory databases (MMDBs))
  - All data is stored in main memory
  - Use is growing as memory costs fall
  - Dramatically faster than disk-based databases
  - Good backup procedures are essential because RAM is volatile
  - Used both in high-end systems where performance is crucial and in small-footprint, embedded applications

# DBMS Types: Column Databases

## Column Databases

- Store data in columns instead of rows
- Improve performance by minimizing the time needed to read the disk
- Used with data warehouses and other big data applications to increase database performance

Emp_no	Dept	Emp_last	Emp_first	Hire_date
1	Acct	Smith	Janice	01/15/2011
2	IT	Lee	Tim	05/16/2009
3	Acct	McNeil	Patrick	04/01/2013
4	Sales	Wilson	Sammy	08/11/2015
5	IT	Morales	Jack	11/16/2014



Database needs to read all columns (because data is read by rows) to access the requested data.

Database needs to read only three columns to access the requested data.

Example of retrieving the names of all IT employees using a row vs. a column database.

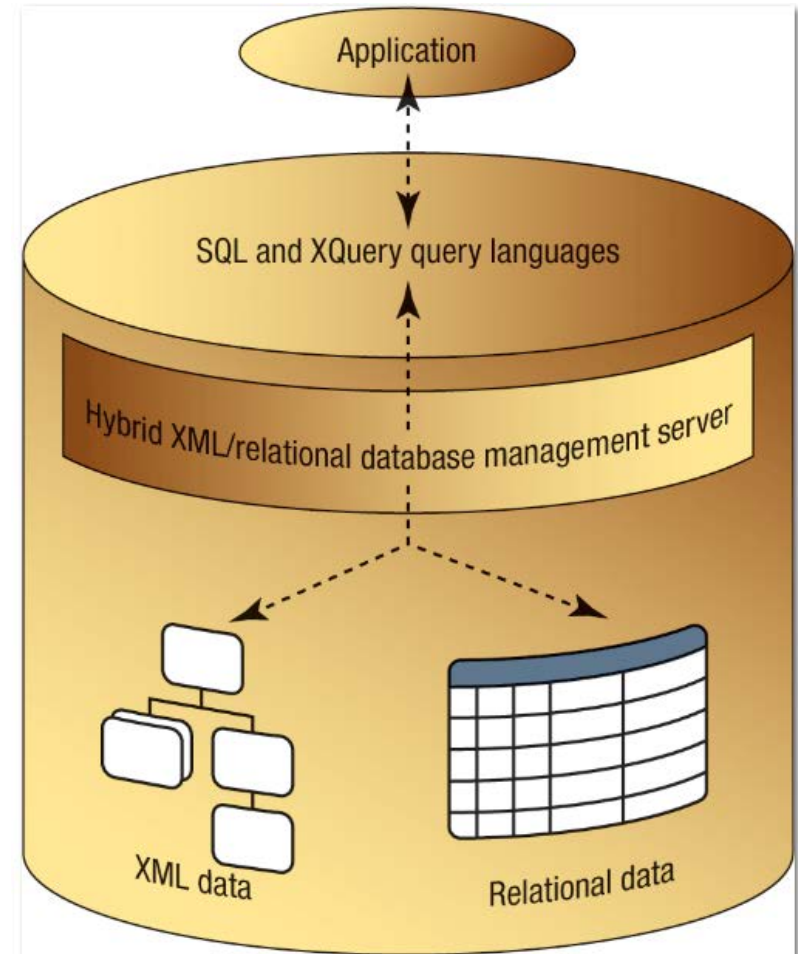


# Multidimensional Databases (MDDDB)

- **Multidimensional databases (MDDDB)** are designed to be used with data warehousing
  - Designed to store a collection of summarized data for quick and easy data analysis
  - Data is collected from a variety of activities and then summarized and restructured to be viewed from multiple perspectives (dimensions)
  - One of the most common types of software used is Online Analytical Processing (OLAP)

# Hybrid Database Models

- Hybrid databases are a combination of two or more database types or models
- **Hybrid XML/relational databases** can store and retrieve both XML data and relational data
  - DB2



**FIGURE 12-21**  
Hybrid XML/  
relational databases.



# NoSQL Databases

- NoSQL databases include database technologies that were developed to overcome some of the limitations of relational databases
  - Object-oriented databases
  - Multidimensional databases
- Are organized differently than relational databases
  - Usually, they do not use tables and SQL
- Increasingly used for big data, IoT applications, and cloud databases



# Summary

- What Is a Database?
- Data Concepts and Characteristics
- Database Classifications
- Database Models
- Cloud Databases