Learning Objectives (1)

• Explain what a database is, including common database terminology, and list some of the advantages and disadvantages of using databases.

• Discuss some basic concepts and characteristics of data, such as data hierarchy, entity relationships, and data definition.

• Describe the importance of data integrity, security, and privacy and how they affect database design.

• Identify some basic database classifications and discuss their differences.
Learning Objectives (2)

• List the most common database models and discuss how they are used today.
• Understand how a relational database is designed, created, used, and maintained.
• Describe some ways databases are used on the Web.
Overview

• This chapter covers:
  – What a database is, the individuals who use them, and the software used to create them
  – Important database concepts and vocabulary
  – Database classifications and models
  – How relational databases and created and used
  – How databases are used on the Web
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
  – Database engine is the part of the program that actually stores and retrieves data
  – Most DBMSs also come bundled with a set of tools to perform a variety of necessary tasks
Components of a Database

• A database typically consists of:
  
  – **Fields (columns)**
    • Single category of data to be stored in a database (name, telephone number, etc.)
  
  – **Records (rows)**
    • Collection of related fields in a database (all the fields for one customer, for example)
  
  – **Tables**
    • Collection of related *records*
    • Are often interrelated with other tables in the database
Example of a Simple Relational Database

![Diagram of a relational database]

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

**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.
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
The Evolution of Databases

- Have evolved due to our increased reliance on information systems, the need to store and retrieve a variety of complex data, etc.

<table>
<thead>
<tr>
<th>MODEL</th>
<th>FLAT FILES</th>
<th>HIERARCHICAL</th>
<th>NETWORK</th>
<th>RELATIONAL</th>
<th>OBJECT-ORIENTED</th>
<th>MULTI-DIMENSIONAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>YEAR BEGAN</td>
<td>1940s</td>
<td>1960s</td>
<td>1960s</td>
<td>1970s</td>
<td>1980s</td>
<td>1990s</td>
</tr>
<tr>
<td>DATA ORGANIZATION</td>
<td>Flat files</td>
<td>Trees</td>
<td>Trees</td>
<td>Tables and relations</td>
<td>Objects</td>
<td>Data cubes, tables and relations, or a combination</td>
</tr>
<tr>
<td>DATA ACCESS</td>
<td>Low-level access</td>
<td>Low-level access with a standard navigational language</td>
<td>Low-level access with a standard navigational language</td>
<td>High-level, nonprocedural languages</td>
<td>High-level, nonprocedural, object-oriented languages</td>
<td>OLAP (Online Analytical Processing) tools or programming languages</td>
</tr>
<tr>
<td>SKILL LEVEL REQUIRED TO ACCESS DATA</td>
<td>Programmer</td>
<td>Programmer</td>
<td>Programmer</td>
<td>User</td>
<td>User</td>
<td>User</td>
</tr>
<tr>
<td>ENTITY RELATIONSHIPS SUPPORTED</td>
<td>One-to-one</td>
<td>One-to-one, one-to-many</td>
<td>One-to-one, one-to-many, many-to-many</td>
<td>One-to-one, one-to-many, many-to-many</td>
<td>One-to-one, one-to-many, many-to-many</td>
<td>One-to-one, one-to-many, many-to-many</td>
</tr>
<tr>
<td>DATA AND PROGRAM INDEPENDENCE</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

FIGURE 12-3
The evolution of databases. Databases have evolved over the years, becoming more flexible, more capable, and easier to use.
Advantages and Disadvantages of the DBMS Approach

• Advantages
  – Low level of redundancy
    • Faster response time
    • Lower storage requirements
    • Easier to secure
    • Increased data accuracy

• Disadvantages
  – Increased vulnerability
  – Security and backup procedures extremely important
Inside the Industry

File Management Systems

- Because file management systems cannot retrieve data from more than one table at a time, there is a much higher level of redundancy.

Example of file management system tables.
Data Concepts and Characteristics

- Important to understand the concepts and characteristics of a database in order to successfully design, create, and use a database

- Data hierarchy
  - Fields/columns: Hold single pieces of data
  - Records/rows: Groups of related fields
  - Tables: Collection of related records
  - Database: Contains a group of related tables
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
- **An attribute** is a characteristic of an entity
  - Typically become fields in the entity’s database table
- A relationship is an association between two or more entities
  - There are three basic entity relationships
Entity Relationships

• 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

• One to many (O:M) entity relationship
  – Most common type of relationship
  – One entity can be related to more than one other entity

• Many to many (M:M) entity relationships
  – One entity can be related to more than one other entity, and those entities can be related to multiple entities of the same type as the original entity
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
Example of Data Definition

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

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

**FIGURE 12-4**
Data definition. Each field in a database has a defined data type and properties that can be assigned to that field.

- 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.

A new record can be added here; it would become the 6th record in this table.
The 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)
    • Does not contain any of the data in the tables
    • Does contain **metadata**, or information about the database tables
  – Ensures that data being entered into the database does not violate any of its assigned properties
Data Integrity

- **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
Example of Data Validation

**WRONG DATA TYPE**
Only data matching a field’s assigned data type may be entered into that field.

**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.
Data 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.
Example of Database Security Tools

**FIGURE 12-6**

Database security tools. This program secures databases and displays alerts for vulnerabilities and attacks.
Data 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
    - 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
Data Organization

• Data organization arranges data for efficient retrieval

• **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 Indexed Organization

FIGURE 12-7
Indexed organization is often used for real-time transaction processing.
Example of Direct Organization

**FIGURE 12-8**
Direct organization is frequently used for faster real-time processing.

**HASHING PROCEDURE**

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).
How It Works

Column Databases

- Store data by 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

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

Example of retrieving the names of all IT employees using a row vs. a column database.
Quick Quiz (1)

1. A column in a database in which customer names are stored would be referred to as a __________.
   a. field
   b. record
   c. table

2. True or False: One advantage to the DBMS approach is the low level of redundancy.

3. The field in a table that uniquely identifies each record in that table and relates that table to other tables is called the __________.

Answers:
1) a; 2) True; 3) primary key
Database Classifications: Single-User and Multiuser

• **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)
A client-server database system has both clients (front end) and at least one database server (back end).
N-Tier Database Systems

- 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
Centralized vs. Distributed Database Systems

• **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
  – Cloud databases
Example of Centralized vs. Distributed Databases

CENTRALIZED DATABASE
The databases are stored on a single server.

DISTRIBUTED DATABASE
The data is divided among multiple databases stored on more than one server, though it acts as a single database to the users.

FIGURE 12-11
Centralized vs. distributed databases.
Disk-Based vs. In-Memory Database Systems

• **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
Quick Quiz (2)

1. Which type of database system is beginning to be used in high-end systems where performance is crucial?
   a. in-memory databases
   b. disk-based databases
   c. single-user databases

2. True or False: With the n-tier database model, there is at least one middle piece of software between the client and the server.

3. With a(n) __________ database system, the data in a database is stored on more than one computer.

**Answers:**

1) a; 2) True; 3) distributed
Database Models

• Hierarchical databases
  – Organizes data in a tree structure
  – Typically a one-to-many relationship between data entities

• Network databases
  – Allow both one-to-many and many-to-many relationships between data elements

• Most databases today are neither hierarchical or network models

• Relational databases are the most common today
The Relational Database Model

• With a **relational database management system (RDBMS)**, data is organized in tables related by common fields

• 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.)
Example of a Relational Database Preliminary Design

**FIGURE 12-14**
A preliminary design for three tables in the Inventory database.

<table>
<thead>
<tr>
<th>PRODUCT TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Number</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CUSTOMER TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Number</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ORDER TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order Number</td>
</tr>
</tbody>
</table>

Each table contains a primary key field.

The primary key fields are repeated in other tables as needed to tie the tables together.
Creating a Relational Database: File and Tables

• Create the database file
• Creating the tables
  – Use the table structures developed during the database design process
  – In Access, can use the table’s:
    • Design view – the table structure is created first and then data is input into the table
    • Datasheet view – the table structure is created as data is entered into the table
      – Structure often needs to then be modified
Example of Creating a Database Table

- Use the View button to select the desired view.
- You specify the name of the database file when the database file is created.

**TABLE DESIGN VIEW**
The table structure is created before data is entered.

- You create all fields and set the primary key.

**TABLE DATASHEET VIEW**
The table structure is created as table data is entered.

- Entering data creates fields, each with the appropriate data type and a generic field name, which you can later rename; an ID field primary key is created by default.

**FIGURE 12-15**
Tables can be created using Design view or Datasheet view.
Entering and Editing Data

• Existing data needs to be migrated into the tables
• New data can be added using the table’s Datasheet view or via a form
  – 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
Example of Creating and Using a Form

FIGURE 12-16
Forms. Forms can be used to view, edit, and add table data.
Relating Tables

• Once all tables have been created, they can be related to one another using their primary keys
  – Primary key in one table can be used to extract data from other tables as needed
  – The same primary key field must in both tables to be related
  – Once tables are related, data from both tables can be display when viewing one of those tables, a report, or other object associated with one of the tables
Example of Relating Tables

1. Click to open Relationships.

2. Drag a primary key field to a related table and then click the Create button to create the relationship between those two tables.

3. Once the tables are related, data from one table (Order table, in this example) can be displayed within a related table (Customer table, in this example).

FIGURE 12-17
Relating tables.
Retrieving Information from a Relational Database: Queries

- A **query** is a request to see information from a database that matches specific criteria
  - 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
Example of Querying a Database

FIGURE 12-18
Querying a database. This example pulls information from the Product table in the Inventory database.
• **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
Example of Creating and Viewing a Report

FIGURE 12-19

Reports. Display table information with a more formal, businesslike appearance.

© 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.
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
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
  - Objects can be retrieved using queries (object query language or OQL)
  - Objects can be reused in other applications to create new applications quickly
Example of an Object-Oriented Database

FIGURE 12-20
The Sloan Digital Sky Survey object-oriented database.

A variety of search tools are available to retrieve images from the database.
Trend

**Law Enforcement Databases**

- New database are now emerging that hold non-traditional data like photos and biometric data
- Next Generation Identification (NGI) components:
  - AFIT to store and match fingerprints
  - IPS to store and match photos
  - NPPS to store and match palm prints
  - RISC provides mobile access for law enforcement officers
  - Rap Back sends status notifications of criminal history or activities for individuals holding positions of trust

Databases are widely used in law enforcement.

© 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.
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.
Multidimensional Databases (MDDB)

• **Multidimensional databases (MDDB)** 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)
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
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
Cloud Databases

• Use is growing rapidly
• Typically built using a cloud provider (Windows Azure, Amazon SimpleDB, or Google Cloud SQL)
• Requires less in-house hardware and maintenance
• Individuals can create via Microsoft Access web apps
  – Stored on a Sharepoint site

With Microsoft Access, you can create custom cloud databases.
Examples of Web Pages that Use Cloud Databases

**REFERENCE SITE**
This site stores address information for individuals and businesses in the United States. After the user enters a name or category and a location (in this case, pizza in Pismo Beach, California), the matching information is retrieved from the database.

**PERSONALIZED SITE**
This site retrieves information from its database to create personalized pages for signed-in viewers, such as recently viewed products and recommendations based on viewed and purchased products.

**FIGURE 12-22**
Many Web pages are used in conjunction with a cloud database.
How Cloud Databases Work

- Visitor makes request by filling out a Web page form or selecting an option from a Web page menu
- Web server converts the request into a database query, passes it onto the database server, and then sends the results back to the visitor
  - Middleware connects the Web server and the database
    - Commonly written as scripts
      - CGI scripts are written in programming languages (C, Perl, Java, etc.)
      - Active Server Pages (ASPs) are written in JavaScript or VBScript
    - PHP scripts use PHP tags inserted into the Web page HTML
Example of a Cloud Database in Action

1. The user fills out the search box and either presses Enter or clicks the Search button, sending the "rocker" data to the Web server.

2. The Web server converts the data entered (rocker) into a database query and sends it to the database server via middleware.

3. The database server performs the query on the database and sends the results back to the Web server via middleware.

4. The middleware program converts the query results to HTML, and then the Web server sends the results in the form of a Web page that is displayed on the user’s screen.

FIGURE 12-23
A cloud database in action.
Quick Quiz (3)

1. What is the most widely used type of database today?
   a. network
   b. relational
   c. object-oriented

2. True or False: Cloud databases are often used in conjunction with dynamic Web pages.

3. A(n) ____________ is used to extract specific information from a database by specifying particular conditions about the data to be retrieved.

Answers:
1) b; 2) True; 3) query
Summary

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