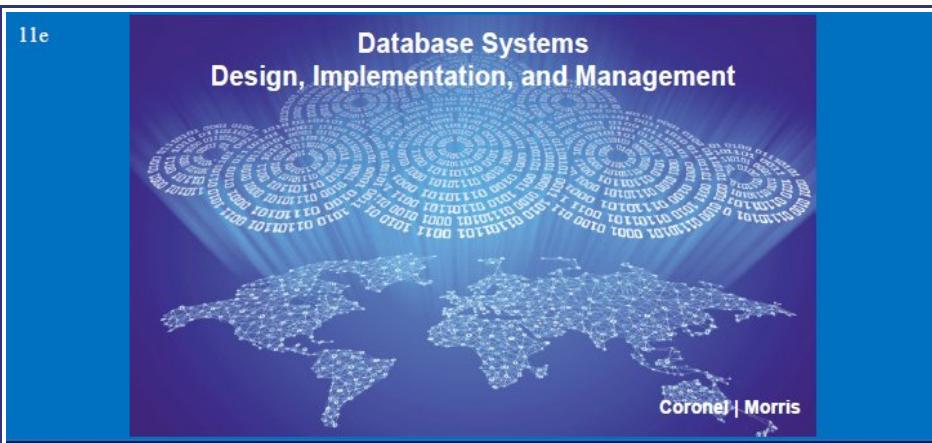


1/30 7:18:39 \*\*\*

← →

# Introduction

# Database systems



## Chapter 1

### Database Systems

# What you will learn:

## Learning Objectives

- In this chapter, you will learn:
  - The difference between data and information
  - What a database is, the various types of databases, and why they are valuable assets for decision making
  - The importance of database design
  - How modern databases evolved from file systems

# What else you will learn:

## Learning Objectives

- In this chapter, you will learn:
  - About flaws in file system data management
  - The main components of the database system
  - The main functions of a database management system (DBMS)

# Data != information!

## Data vs. Information

### Data

- Raw facts
  - Raw data - Not yet been processed to reveal the meaning
- Building blocks of information
- **Data management**
  - Generation, storage, and retrieval of data

### Information

- Produced by processing data
- Reveals the meaning of data
- Enables **knowledge** creation
- Should be accurate, relevant, and timely to enable good decision making

# DB, DBMS

## Database

- Shared, integrated computer structure that stores a collection of:
  - End-user data - Raw facts of interest to end user
  - **Metadata:** Data about data, which the end-user data are integrated and managed
    - Describe data characteristics and relationships
- **Database management system (DBMS)**
  - Collection of programs
  - Manages the database structure
  - Controls access to data stored in the database

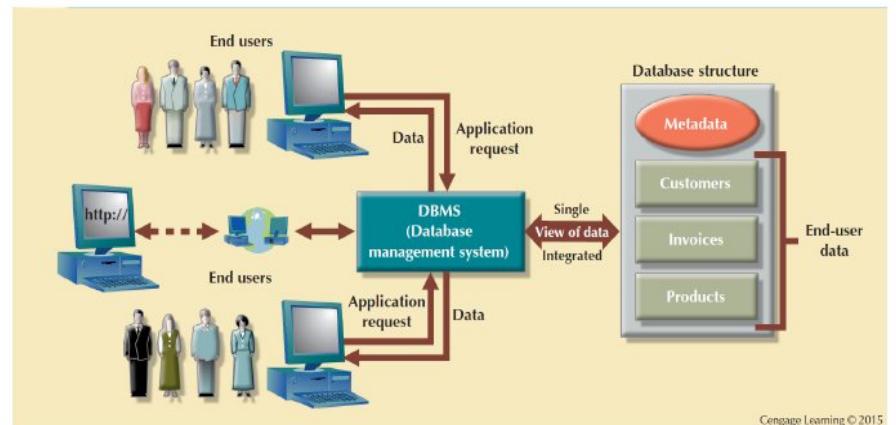
# Why DBMS?

## Role of the DBMS

- Intermediary between the user and the database
- Enables data to be shared
- Presents the end user with an integrated view of the data
- Receives and translates application requests into operations required to fulfill the requests
- Hides database's internal complexity from the application programs and users

# DBMS is a go-between

Figure 1.2 - The DBMS Manages the Interaction between the End User and the Database



# DBMS: advantages

## Advantages of the DBMS

- Better data integration and less data inconsistency
  - **Data inconsistency:** Different versions of the same data appear in different places
- Increased end-user productivity
- Improved:
  - Data sharing
  - Data security
  - Data access
  - Decision making
    - **Data quality:** Promoting accuracy, validity, and timeliness of data

# Types of DBs: based on user count

## Types of Databases

- **Single-user database:** Supports one user at a time
  - **Desktop database:** Runs on PC
- **Multiuser database:** Supports multiple users at the same time
  - **Workgroup databases:** Supports a small number of users or a specific department
  - **Enterprise database:** Supports many users across many departments

# Types of DBs: based on location

## Types of Databases

- **Centralized database:** Data is located at a single site
- **Distributed database:** Data is distributed across different sites
- **Cloud database:** Created and maintained using cloud data services that provide defined performance measures for the database

# Types of DBs: based on content

## Types of Databases

- **General-purpose databases:** Contains a wide variety of data used in multiple disciplines
- **Discipline-specific databases:** Contains data focused on specific subject areas

# Types of DBs: based on data currency

## Types of Databases

- **Operational database:** Designed to support a company's day-to-day operations
- **Analytical database:** Stores historical data and business metrics used exclusively for tactical or strategic decision making
- **Data warehouse:** Stores data in a format optimized for decision support

# Types of DBs [cont'd]

## Types of Databases

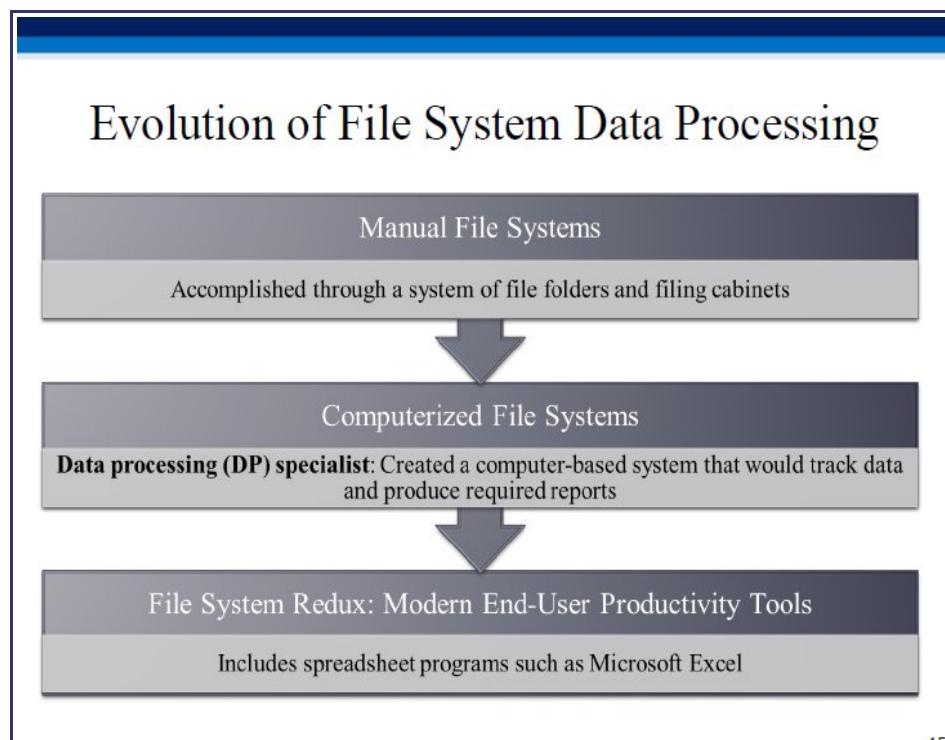
- **Online analytical processing (OLAP)**
  - Enable retrieving, processing, and modeling data from the data warehouse
- **Business intelligence:** Captures and processes business data to generate information that support decision making

# Types of DBs: based on the structure of contained data

## Types of Databases

- **Unstructured data:** It exists in their original state
- **Structured data:** It results from formatting
  - Structure is applied based on type of processing to be performed
- **Semistructured data:** Processed to some extent
- **Extensible Markup Language (XML)**
  - Represents data elements in textual format

# Early DBs: file systems



# File systems

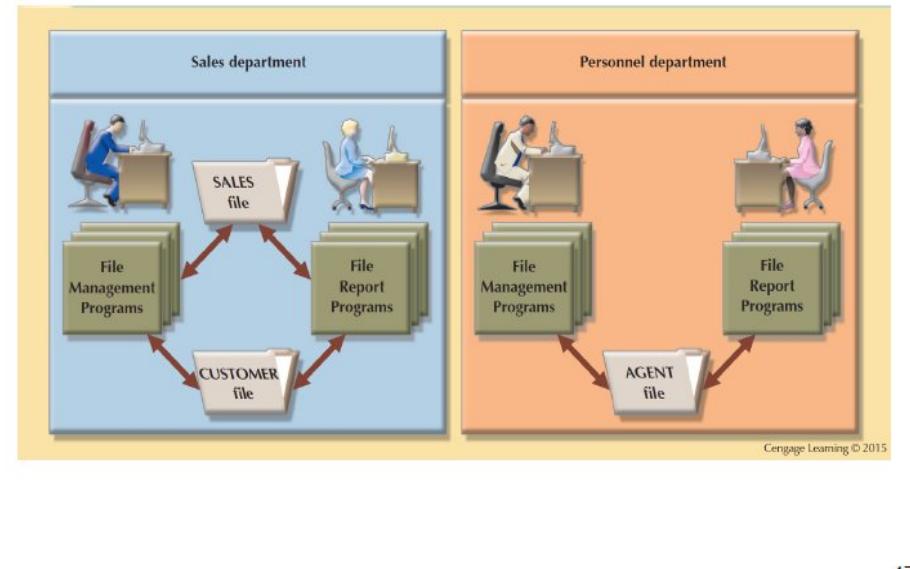
Table 1.2 - Basic File Terminology

TERM	DEFINITION
Data	Raw facts, such as a telephone number, a birth date, a customer name, and a year-to-date (YTD) sales value. Data have little meaning unless they have been organized in some logical manner.
Field	A character or group of characters (alphabetic or numeric) that has a specific meaning. A field is used to define and store data.
Record	A logically connected set of one or more fields that describes a person, place, or thing. For example, the fields that constitute a record for a customer might consist of the customer's name, address, phone number, date of birth, credit limit, and unpaid balance.
File	A collection of related records. For example, a file might contain data about the students currently enrolled at Gigantic University.

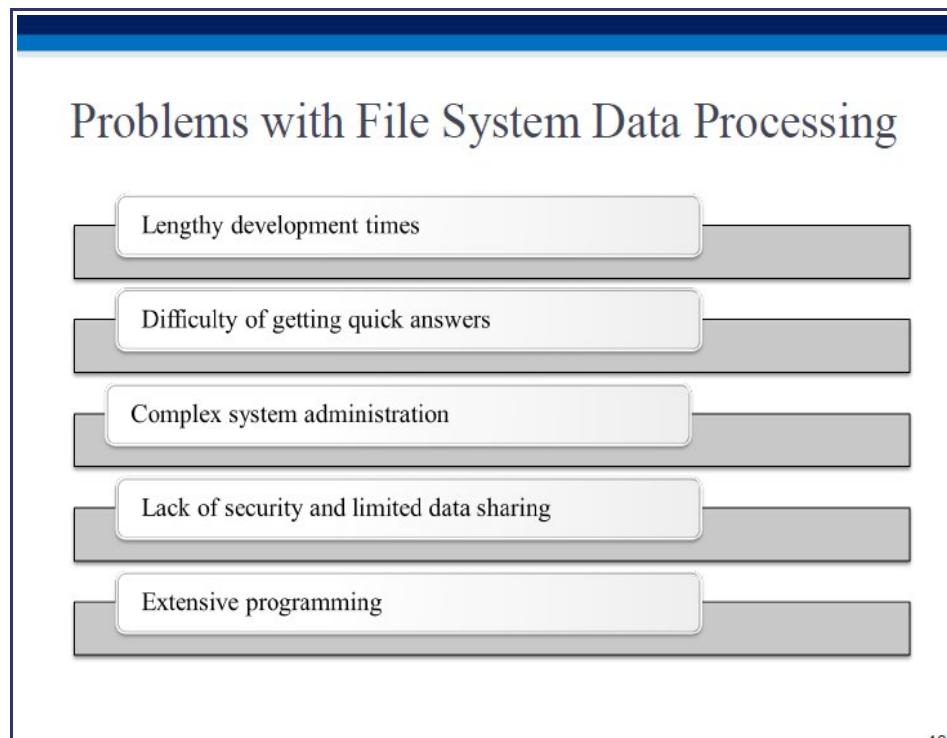
Cengage Learning © 2015

# File system

Figure 1.6 - A Simple File System



# File systems: problems



# 'Structural' dependence (not a good thing!)

## Structural and Data Dependence

- **Structural dependence:** Access to a file is dependent on its own structure
  - All file system programs are modified to conform to a new file structure
- **Structural independence:** File structure is changed without affecting the application's ability to access the data

# Structural dependence [cont'd]

## Structural and Data Dependence

- Data dependence
  - Data access changes when data storage characteristics change
- Data independence
  - Data storage characteristics is changed without affecting the program's ability to access the data
- Practical significance of data dependence is difference between logical and physical format

# Redundancy of data (again, not a good thing!)

## Data Redundancy

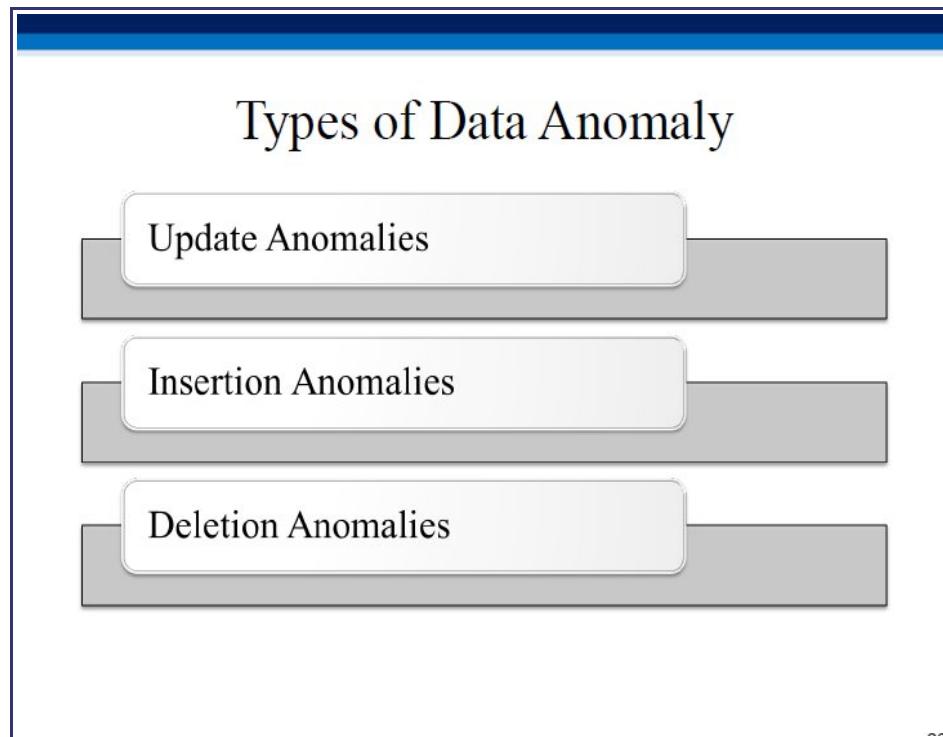
- Unnecessarily storing same data at different places
- **Islands of information:** Scattered data locations
  - Increases the probability of having different versions of the same data

# Why is redundancy not a good thing?

## Data Redundancy Implications

- Poor data security
- Data inconsistency
- Increased likelihood of data-entry errors when complex entries are made in different files
- **Data anomaly:** Develops when not all of the required changes in the redundant data are made successfully

# The three types of data anomalies



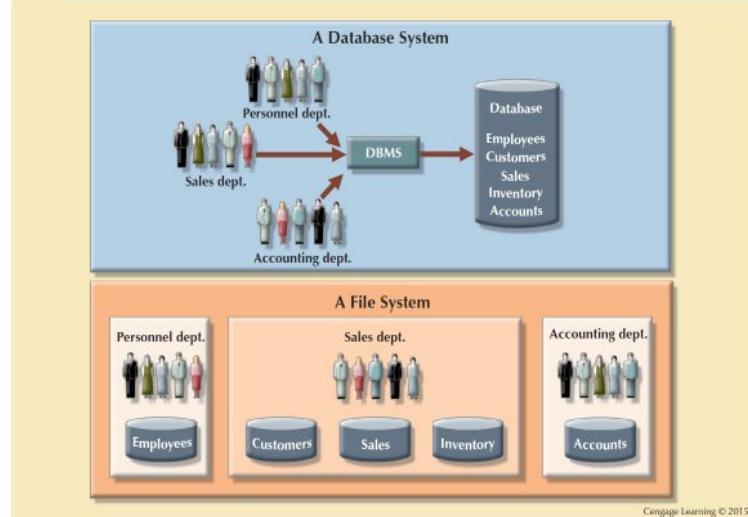
# DB systems

## Database Systems

- Logically related data stored in a single logical data repository
  - Physically distributed among multiple storage facilities
  - DBMS eliminates most of file system's problems
- Current generation DBMS software:
  - Stores data structures, relationships between structures, and access paths
  - Defines, stores, and manages all access paths and components

# DB vs file system

Figure 1.8 - Contrasting Database and File Systems



# DBMS

## DBMS Functions

### Data dictionary management

- **Data dictionary:** Stores definitions of the data elements and their relationships

### Data storage management

- **Performance tuning:** Ensures efficient performance of the database in terms of storage and access speed

### Data transformation and presentation

- Transforms entered data to conform to required data structures

### Security management

- Enforces user security and data privacy

..

# DBMS [cont'd]

## DBMS Functions

### Multiuser access control

- Sophisticated algorithms ensure that multiple users can access the database concurrently without compromising its integrity

### Backup and recovery management

- Enables recovery of the database after a failure

### Data integrity management

- Minimizes redundancy and maximizes consistency

# DBMS [cont'd]

## DBMS Functions

Database access languages and application programming interfaces

- **Query language:** Lets the user specify what must be done without having to specify how
- **Structured Query Language (SQL):** De facto query language and data access standard supported by the majority of DBMS vendors

Database communication interfaces

- Accept end-user requests via multiple, different network environments

# How DBs could be "bad"

