Chapter 2

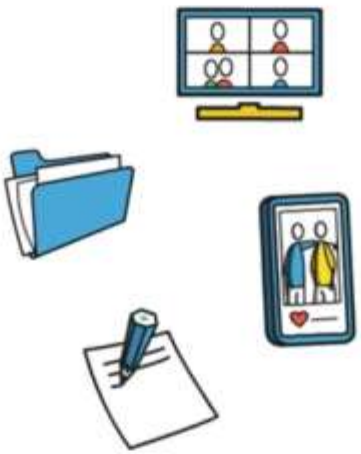
Database Concepts

In this chapter, you will learn:

- The difference between data and information
- What a database is, about different types of databases, and why they are valuable assets for decision making
- Why database design is important
- How modern databases evolved from files and file systems

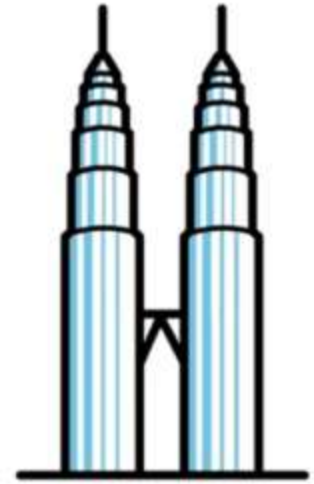
In this chapter, you will learn (continued):

- About flaws in file system data management
- How a database system differs from a file system, and how a DBMS functions within the database system
- About data storage and retrieval strategies



What is Database?

Different types of Database

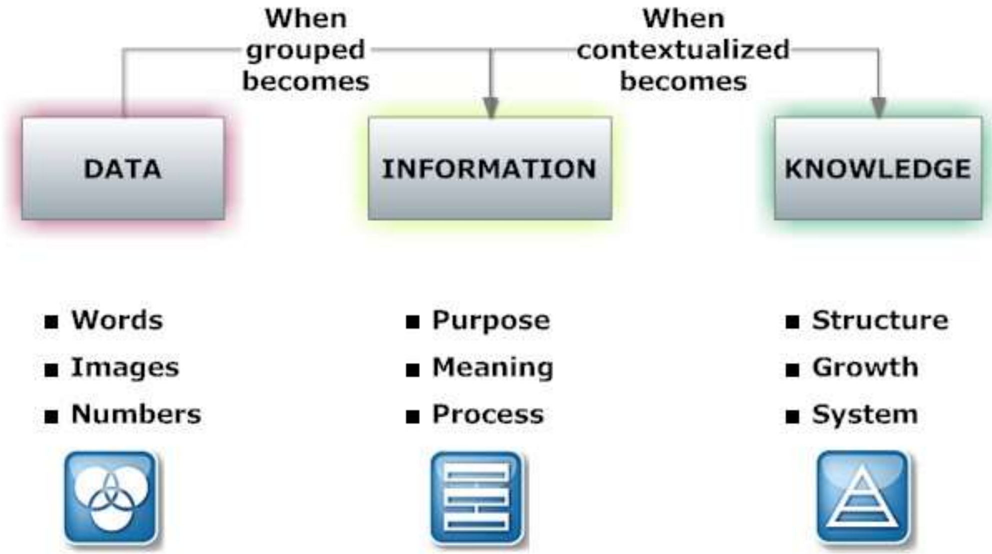


Data vs. Information

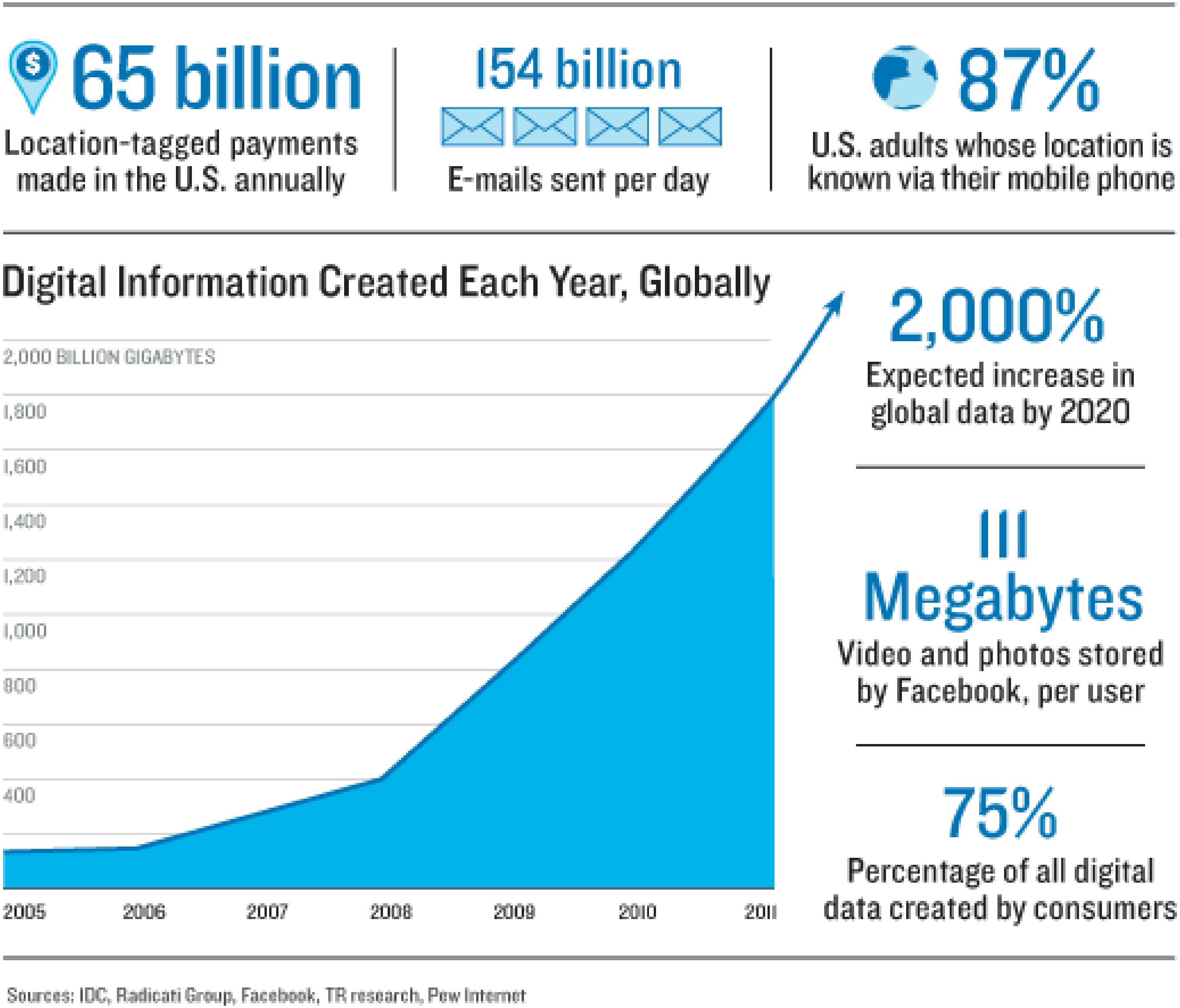
Data	Information
● Raw facts ○ Raw data - Not yet been processed to reveal the meaning ● Building blocks of information ● Data management ○ Generation, storage, and retrieval of data	● Produced by processing data ● Reveals the meaning of data ● Enables knowledge creation ● Should be accurate, relevant and timely to enable good decision making

➔ Accurate, relevant, and timely information is key to good decision making

➔ Good decision making is key to survival in global environment



Meaningful Information - example



Transforming raw data into information - example

a) Data entry screen

b) Raw data

c) Information in summary format

Rank	COUNT	%INFS	TOT/COL	%COL. TOT.	%COL. FAC.
Adjunct	5	20.00%	23	21.74%	3.27%
Assistant Professor	2	8.00%	28	7.14%	1.31%
Associate Professor	9	36.00%	37	24.32%	5.88%
Instructor	2	8.00%	18	11.11%	1.31%
Professor	7	28.00%	47	14.89%	4.58%

d) Information in graphical format

Data Management Trends

- **Paper File-based**

→

- **Computer File-based System**

→

- **Database**

↓

- **Data Warehouse And Data mining**

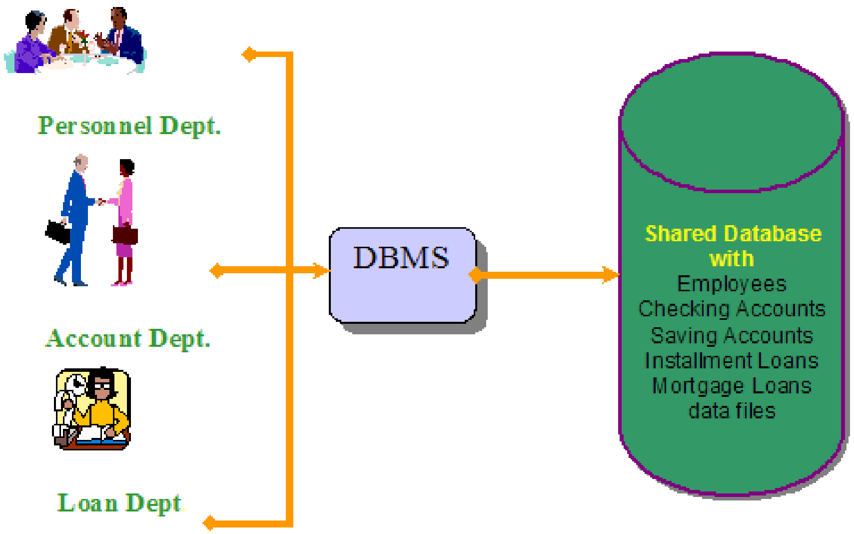
Read:

http://cnx.org/content/m28156/latest/

Sighted 09/05/2014

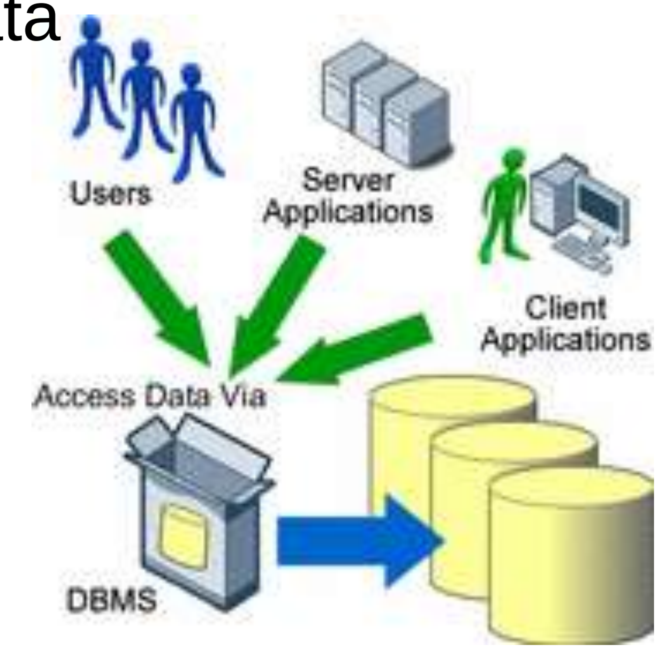
Database Approach

- Database—shared, integrated computer structure that houses:
  - End user data: Raw facts of interest to end user
  - Metadata: Data about data, which end-user data are integrated and managed



Database Approach

- DBMS (database management system):
  - Collection of programs that manages database structure and controls access to data

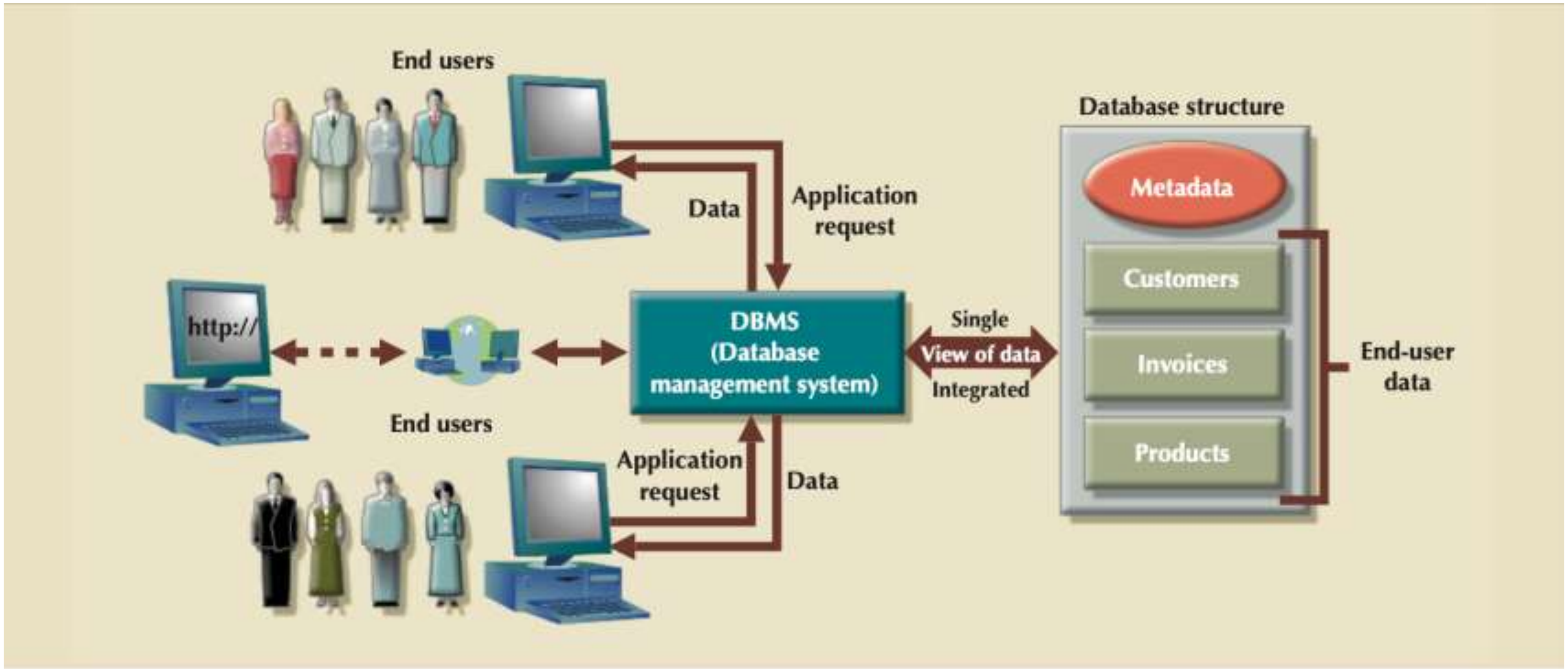


  - Possible to share data among multiple applications or users
  - Makes data management more efficient and effective

DBMS Makes Data Management More Efficient and Effective

- End users have better access to more and better-managed data
  - Promotes integrated view of organization's operations
    - Better understanding and minimize errors VS individualistic view (the story of the 4 blind man describing an elephant)
  - Probability of data inconsistency is greatly reduced
  - Possible to produce quick answers to ad hoc queries

The DBMS Manages the Interaction Between the End User and the Database



Types of Databases

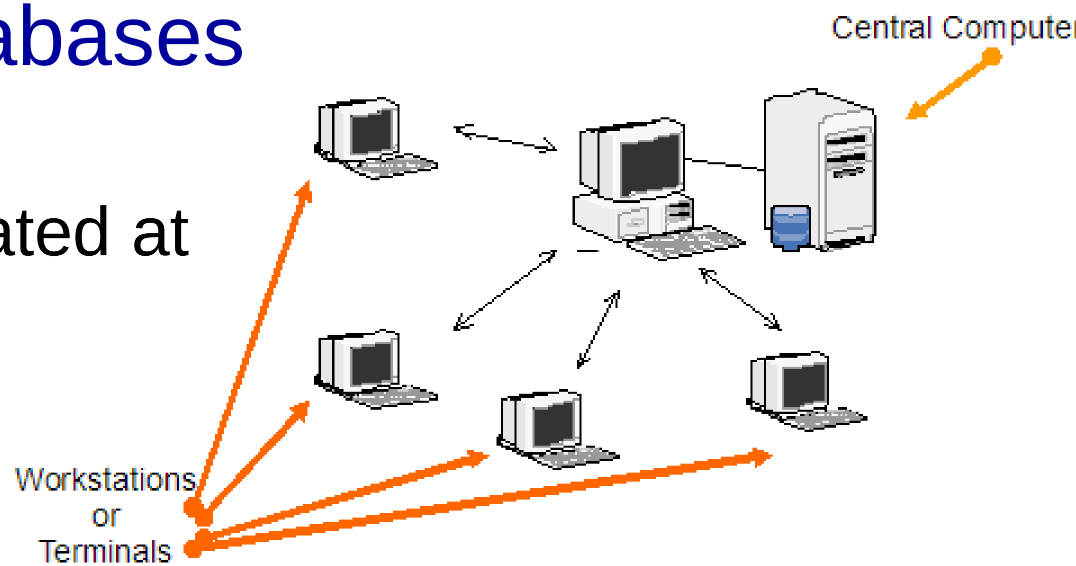
- Single-user:
  - Supports only one user at a time
- Desktop:
  - Single-user database running on a personal computer
- Multi-user:
  - Supports multiple concurrent users at the same time

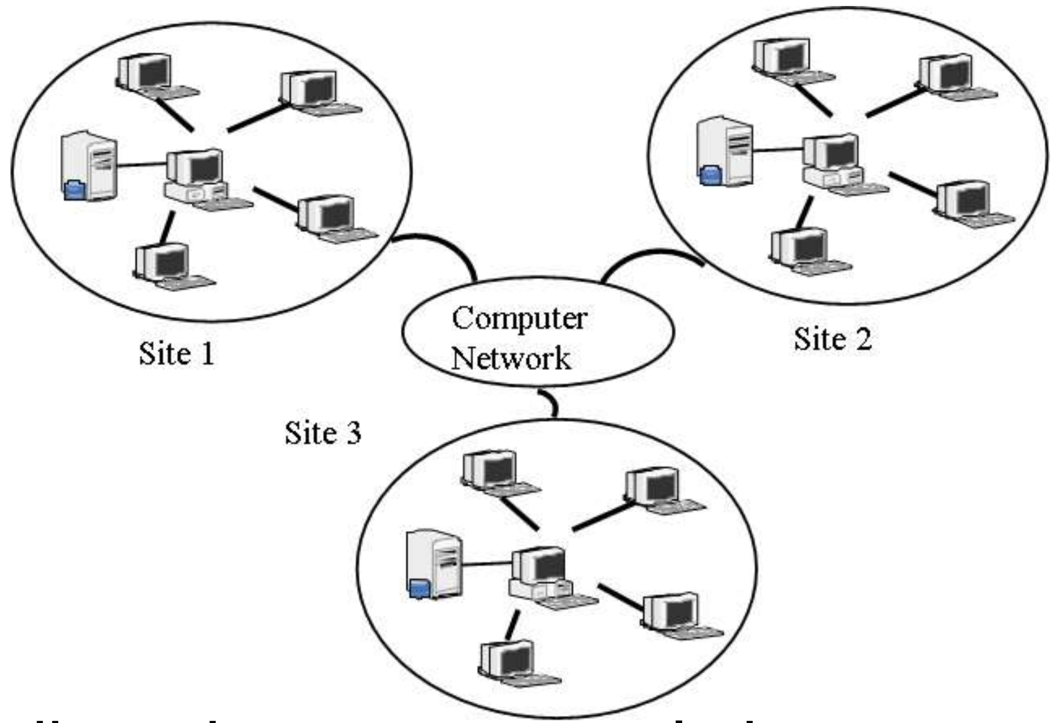
Types of Databases (continued)

- Workgroup:
  - Multi-user database that supports a small group of users or a single department
- Enterprise:
  - Multi-user database that supports a large group of users or an entire organization

Location of Databases

- Centralized:
  - Supports data located at a single site





- Distributed:
  - Supports data distributed across several site

Uses of Databases

- Transactional (or production):
  - Supports a company’s day-to-day operations
- Data warehouse:
  - Stores data used to generate information required to make tactical or strategic decisions
    - Such decisions typically require “data massaging”
  - Often used to store historical data

Why Database Design is Important

- Defines the database’s expected use
- Different approach needed for different types of databases
- Avoid redundant data (unnecessarily duplicated)
- A poorly designed database may lead to poor decision making—and poor decision making can lead to the failure of an organization.

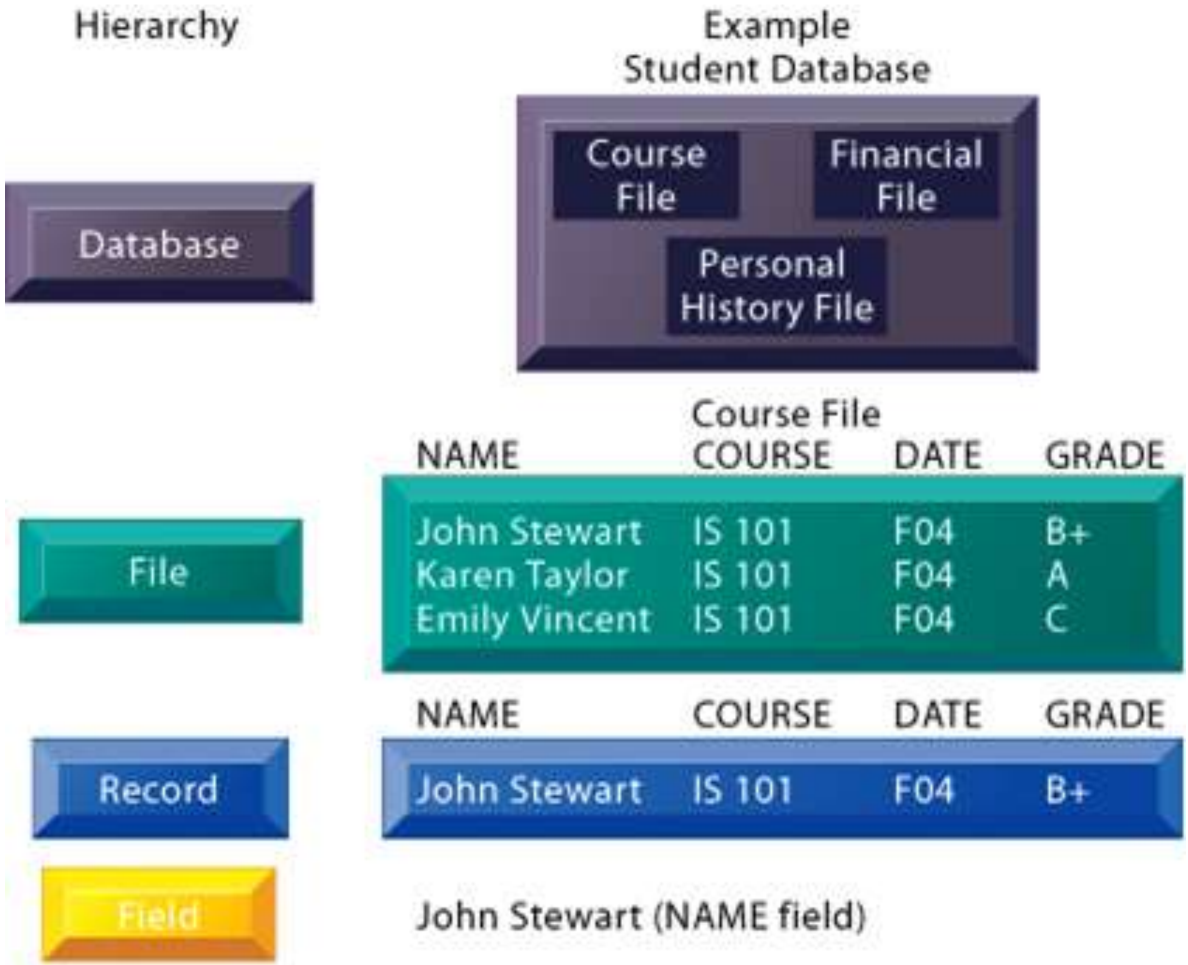
Manual File Systems

- Traditionally composed of collection of file folders kept in file cabinet
- Organization within folders was based on data’s expected use (ideally logically related)
- System was adequate for small amounts of data with few reporting requirements
- Finding and using data in growing collections of file folders became time-consuming and cumbersome

Conversion from Manual File System to Computer File System

- Could be technically complex, requiring hiring of data processing (DP) specialists
- DP specialists created file structures, wrote software, and designed application programs
- Resulted in numerous “home-grown” systems being created
- Initially, computer files were similar in design to manual files

Components of a File



Example of Early Database Design

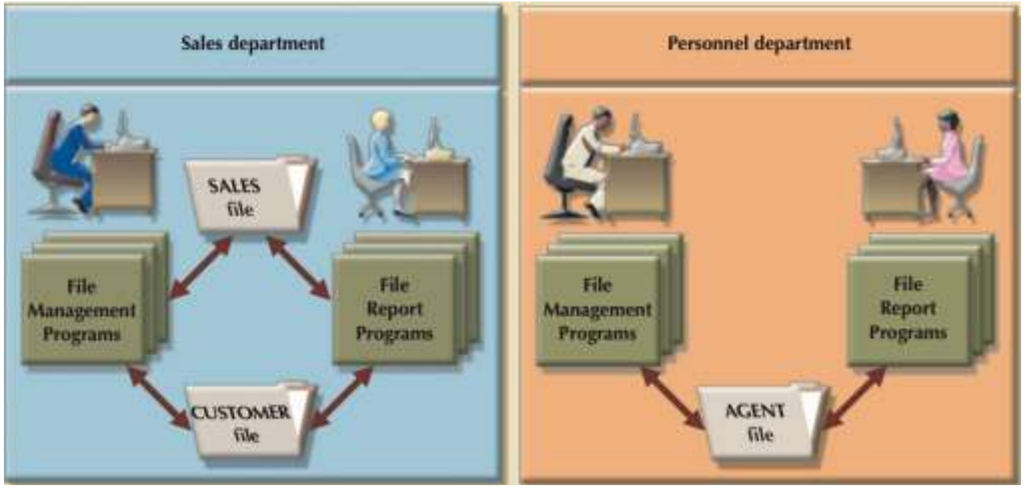
- DP specialist wrote programs for reports:
  - Monthly summaries of types and amounts of insurance sold by agents
  - Monthly reports about which customers should be contacted for renewal
  - Reports that analyzed ratios of insurance types sold by agent
  - Customer contact letters summarizing coverage
- Additional reports were written as required

Example of Early Database Design (continued)

- Other departments requested databases be written for them
  - SALES database created for sales department
  - AGENT database created for personnel department

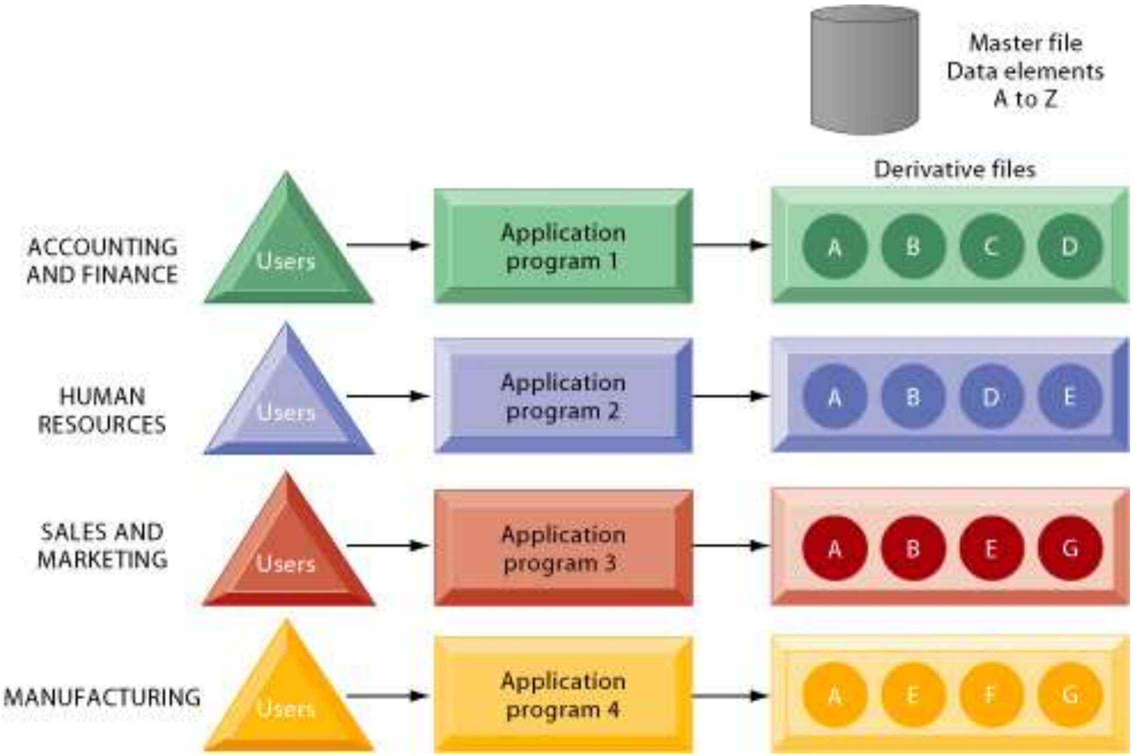
Evolution of Simple File System

- As number of databases increased, small file system evolved
- Each file used its own application programs
- Each file was owned by individual or department who commissioned its creation



File-based System

- Each of the files in the systems used its own application programs to store, retrieve and modify data
- Each file was owned by the department that created it.



Problems with File System Data Management

- Every task requires extensive programming in a third-generation language (3GL)
  - Programmer must specify task and *how* it must be done
- Modern databases use fourth-generation language (4GL)
  - Allows user to specify what must be done *without specifying how* it is to be done

Programming in 3GL

- Time-consuming, high-level activity
- Programmer must be familiar with physical file structure
- As system becomes complex, access paths become difficult to manage and tend to produce malfunctions
- Complex coding establishes precise location of files and system components and data characteristics

Programming in 3GL (continued)

- Ad hoc queries are impossible
- Writing programs to design new reports is time consuming
- As number of files increases, system administration becomes difficult
- Making changes in existing file structure is difficult
- File structure changes require modifications in all programs that use data in that file
- Modifications are likely to produce errors, requiring additional time to "debug" the program
- Security features hard to program and therefore often omitted

Structural and Data Dependence

- Structural dependence
  – Access to a file depends on its own structure
- Data dependence
  – Changes in database structure affect program’s ability to access data
  – Logical data format
    - How a human being views the data
  – Physical data format
    - How the computer “sees” the data

Data Redundancy

- Data redundancy results in data inconsistency
  - Different and conflicting versions of the same data appear in different places
- Errors more likely to occur when complex entries are made in several different files and recur frequently in one or more files
- Data anomalies develop when required changes in redundant data are not made successfully

Data Anomalies

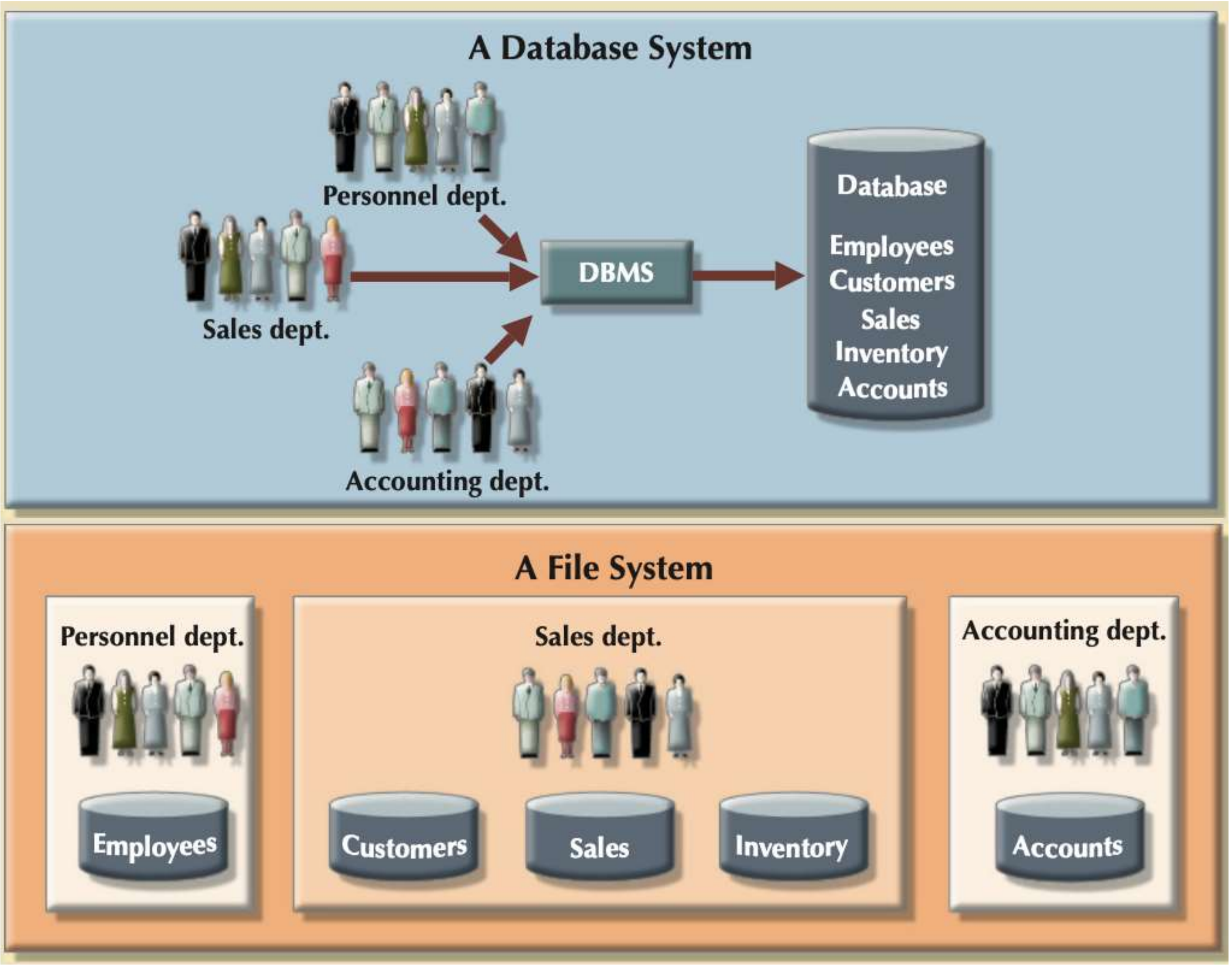
Unable to perform certain data maintenance due to errors in database design

- Modification anomalies
  - Occur when changes must be made to existing records
- Insertion anomalies
  - Occur when entering new records
- Deletion anomalies
  - Occur when deleting records

Database vs. File System

- Problems inherent in file systems make using a database system desirable
- File system
  – Many separate and unrelated files
- Database
  – Logically related data stored in a single logical data repository

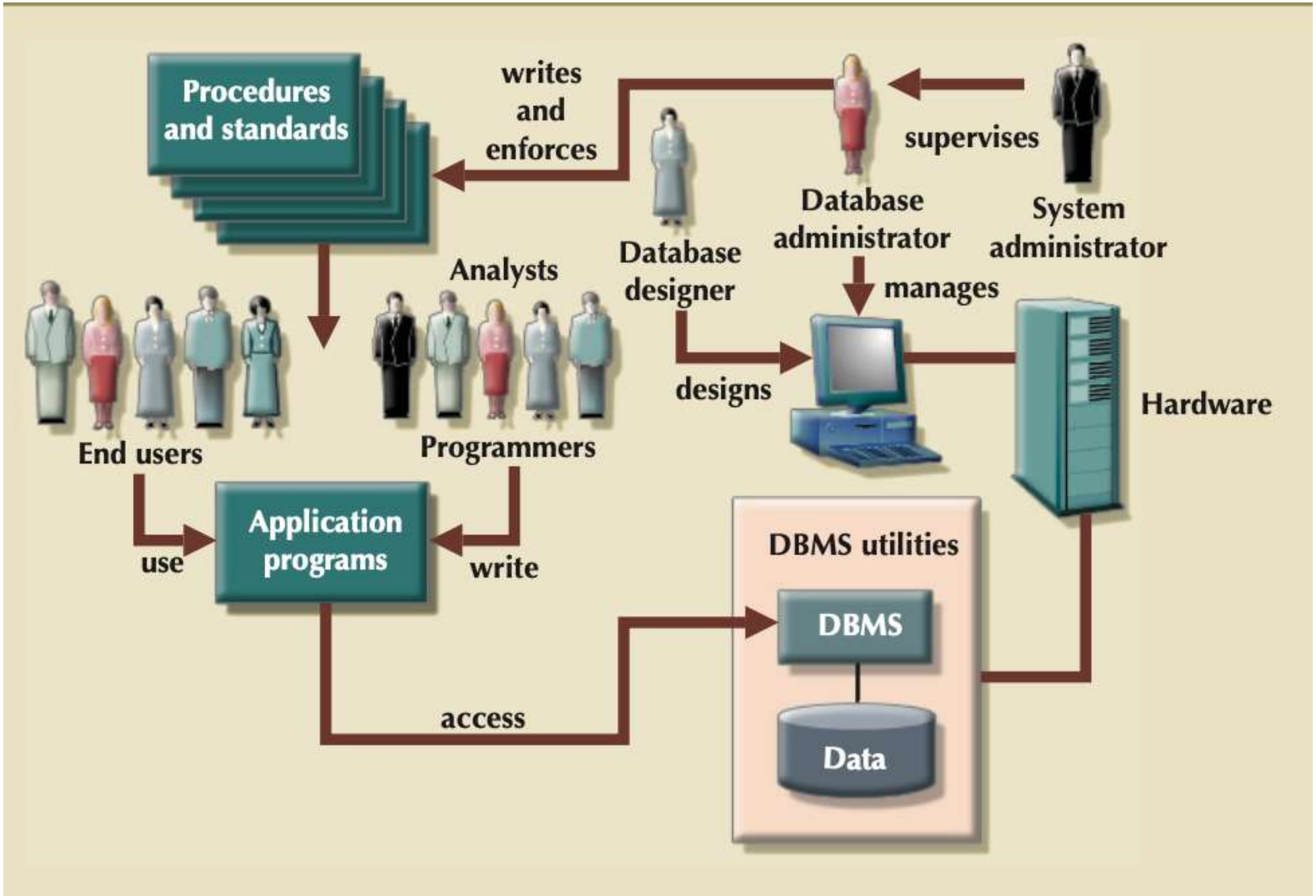
Contrasting Database and File Systems



The Database System Environment

- Database system is composed of 5 main parts:
  1. Hardware
  2. Software
     - Operating system software
     - DBMS software
     - Application programs and utility software
  3. People
  4. Procedures
  5. Data

The Database System Environment (continued)



DBMS Functions

- Performs functions that guarantee integrity and consistency of data
  - Data dictionary management
    - defines data elements and their relationships
  - Data storage management
    - stores data and related data entry forms, report definitions, etc.
  - Data transformation and presentation
    - translates logical requests into commands to physically locate and retrieve the requested data

DBMS Functions (continued)

– Security management
  - enforces user security and data privacy within database
– Multi-user access control
  - creates structures that allow multiple users to access the data
– Backup and recovery management
  - provides backup and data recovery procedures

DBMS Functions (continued)

- Data integrity management
  - promotes and enforces integrity rules to eliminate data integrity problems
- Database access languages and application programming interfaces
  - provides data access through a query language
- Database communication interfaces
  - allows database to accept end-user requests within a computer network environment

Summary

- Information is derived from data, which is stored in a database
- To implement and manage a database, use a DBMS
- Database design defines its structure
- Good design is important

Summary (continued)

- Databases were preceded by file systems
- Because file systems lack a DBMS, file management becomes difficult as a file system grows
- DBMS were developed to address file systems' inherent weaknesses

Homework – Reading Assignment

Physical data storage concepts

- What are the media and devices?
- How does it actually work?
- What are the most common operations on files?
- Typical files storage methods and its uses
- How to make access to data faster?
- How to make storage of data more permanent?
- Indexing files to facilitate faster records access

Reference: Fundamentals of Database Systems, 6th Ed, Elmasri, Navathe - chapters 17,18

Physical data storage concepts

- Disk Storage Devices
- Files of Records
- Operations on Files
- Unordered Files
- Ordered Files
- Hashed Files
  - Dynamic and Extendible Hashing Techniques
- RAID Technology
- Indexed File

Reference: Fundamentals of Database Systems, 6th Ed, Elmasri, Navathe - chapters 17,18

Operation on Files

- Typical file operations include:
  - **OPEN**: Readies the file for access, and associates a pointer that will refer to a *current* file record at each point in time.
  - **FIND**: Searches for the first file record that satisfies a certain condition, and makes it the current file record.
  - **FINDNEXT**: Searches for the next file record (from the current record) that satisfies a certain condition, and makes it the current file record.
  - **READ**: Reads the current file record into a program variable.
  - **INSERT**: Inserts a new record into the file & makes it the current file record.
  - **DELETE**: Removes the current file record from the file, usually by marking the record to indicate that it is no longer valid.
  - **MODIFY**: Changes the values of some fields of the current file record.
  - **CLOSE**: Terminates access to the file.
  - **REORGANIZE**: Reorganizes the file records.
    - For example, the records marked deleted are physically removed from the file or a new organization of the file records is created.
  - **READ_ORDERED**: Read the file blocks in order of a specific field of the file.