WelTec
Te Whare Wānanga o te Awakairangi

西安科技大学高新学院
XI'AN KEDAGAOXIN UNIVERSITY

IT5507 Fundamentals of Data Science

Chapter 1
Database Systems

CENGAGE

Learning Objectives

- After completing this chapter, you will be able to:
  - Define the difference between data and information
  - Describe what a database is, various types, and why they are valuable assets for decision making
  - Explain the importance of database design
  - See how modern databases evolved from file systems
  - Understand flaws in file system data management
  - Outline the main components of the database system
  - Describe the main functions of a database management system (DBMS)

Why Databases?

- Characteristics of data in today's world
  - Ubiquitous (i.e., abundant, global, and everywhere)
  - Pervasive (i.e., unescapable, prevalent, and persistent)
- Databases make data persistent and shareable in a secure way
  - Specialized structures that allow computer-based systems to store, manage, and retrieve data very quickly

So why Database?

A database is like a super-smart filing system for storing, organizing, and managing information. It helps keep data in order, makes it easy to find, and ensures that different parts of a system can share and use the same information efficiently. Whether it's a list of students, financial transactions, or product details, a database makes handling large amounts of data way more organized and accessible.

Data versus Information

- Data consists of raw facts
  - Not yet processed to reveal meaning to the end user
  - Building blocks of information
- Information results from processing raw data to reveal meaning
  - Requires context, Bedrock of knowledge , Should be accurate, relevant, and timely

Data: Raw facts or figures without context or interpretation. It lacks meaning on its own. E.g. "1234567890"

Information: Processed data that has been organized, analyzed, and given context, making it meaningful and useful for decision-making.

E.g. "Customer ID: 1234567890, John Doe, purchased item XYZ on 2022-03-04."

Introducing the Database

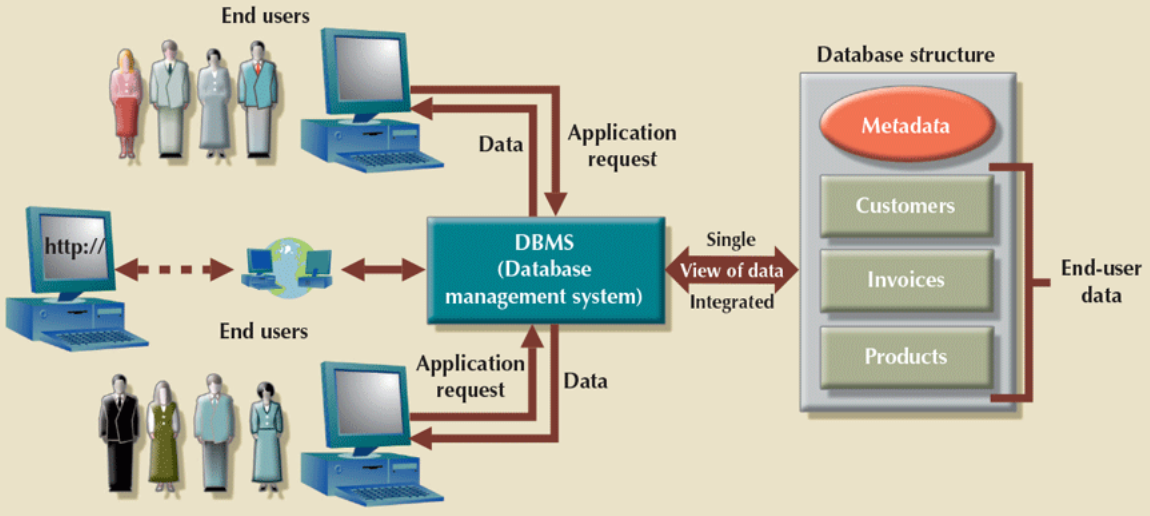
- Shared, integrated computer structure that stores data
  - End-user data: raw facts of interest to end user
  - Metadata: data about data, through which the end-user data is integrated and managed
    - Describes data characteristics and relationships
- Database management system (DBMS)
  - Collection of programs
  - Manages the database structure
  - Controls access to data stored in the database
- A structured collection of data organized for efficient storage, retrieval, and management, typically stored electronically.

Role and Advantages of the DBMS (1 of 2)

- Database management system (DBMS): intermediary between the user and the database
  - Enables data to be shared
  - Presents the end user with an integrated view of data
  - Provides more efficient and effective data management
  - Improves sharing, security, integration, access, decision-making, productivity, etc.
  - Software that manages and facilitates the organization, storage, retrieval, and manipulation of data within a database.
  - E.g. MS Access, Microsoft SQL Server, Oracle Database, MySQL, and PostgreSQL

Role and Advantages of the DBMS (2 of 2)

FIGURE 1.4 THE DBMS MANAGES THE INTERACTION BETWEEN THE END USER AND THE DATABASE



The Database Management System (DBMS) acts as an intermediary between the end user and the database. It facilitates the interaction by managing tasks such as data retrieval, storage, updating, and ensuring data integrity. The DBMS provides a systematic way for users to interact with the database without needing to handle the underlying complexities of data storage and manipulation directly.

Types of Databases (1 of 5)

- **Single-user database:** supports one user at a time
  - Desktop database: single-user database on a personal computer
  - E.g. MS Access
- **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
  - E.g. Oracle Database

A single-user database system allows only one user to access and manipulate data at a time, commonly utilized for individual or small-scale applications like personal projects. In contrast, a multiuser database system supports concurrent access by multiple users, making it suitable for larger enterprises where simultaneous data interaction is essential. Multiuser databases, exemplified by systems like Oracle Database in organizational settings, require more sophisticated management to handle concurrent access, ensuring data integrity and security.

Types of Databases (2 of 5)

- Classification by location

  - **Centralized database:** data located at a single site

  Data is stored in a single location, making it a hub for information management.

  - **Distributed database**: data distributed across different sites

  Data is spread across various locations, enabling decentralized access and reducing the risk of a single point of failure.

  - **Cloud database:** created and maintained using cloud data services that provide defined performance measures for the database

  Data is hosted and managed through cloud services, offering scalability, accessibility, and defined performance metrics without the need for physical infrastructure.

Types of Databases (3 of 5)

- Classification by data type
  - **General-purpose database:** contains a wide variety of data used in multiple disciplines

Holds diverse data used across various disciplines, catering to broad informational needs. Microsoft SQL Server or Oracle Database, capable of storing a wide range of data types for various purposes like business, research, and more.

  - **Discipline-specific database:** contains data focused on specific subject areas

Focuses on data relevant to specific subject areas, providing specialized information for targeted applications. PubMed for biomedical literature or IEEE Xplore for engineering and technology-focused documents, tailoring data to specific fields.

  - **Operational database:** designed to support a company's day-to-day operations

Geared towards supporting daily business operations, ensuring real-time data access and transactional processes essential for day-to-day functioning. An example is a Customer Relationship Management (CRM) system like Salesforce, designed to handle and process operational data related to customer interactions, sales, and support.

Types of Databases (4 of 5)

- 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
  - Online analytical processing (OLAP): tools for 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 Databases (5 of 5)

- Databases can be classified to reflect the degree to which the data is structured
  - Unstructured data exists in its original (raw) state
  - Structured data 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, It provides a flexible and extensible framework for encoding information in a format that is both machine-readable and easy for humans to understand.

```
<book>
 <title>Harry Potter</title>
 <author>J.K. Rowling</author>
 <genre>Fantasy</genre>
</book>
```

In this basic XML example, a book is represented with elements for its title, author, and genre. XML's simplicity and readability make it suitable for organizing and exchanging structured data.

Why Database Design Is Important

- Focuses on design of database structure that will be used to store and manage end-user data
  - Well-designed database: facilitates data management and generates accurate and valuable information
  - Poorly designed database: causes difficult-to-trace errors that may lead to poor decision making
  - Hence, database design is crucial as it structures the organization and relationships of data, ensuring efficient storage, retrieval, and management, ultimately leading to accurate information and optimal system performance.

Evolution of File System Data Processing (1 of 3)

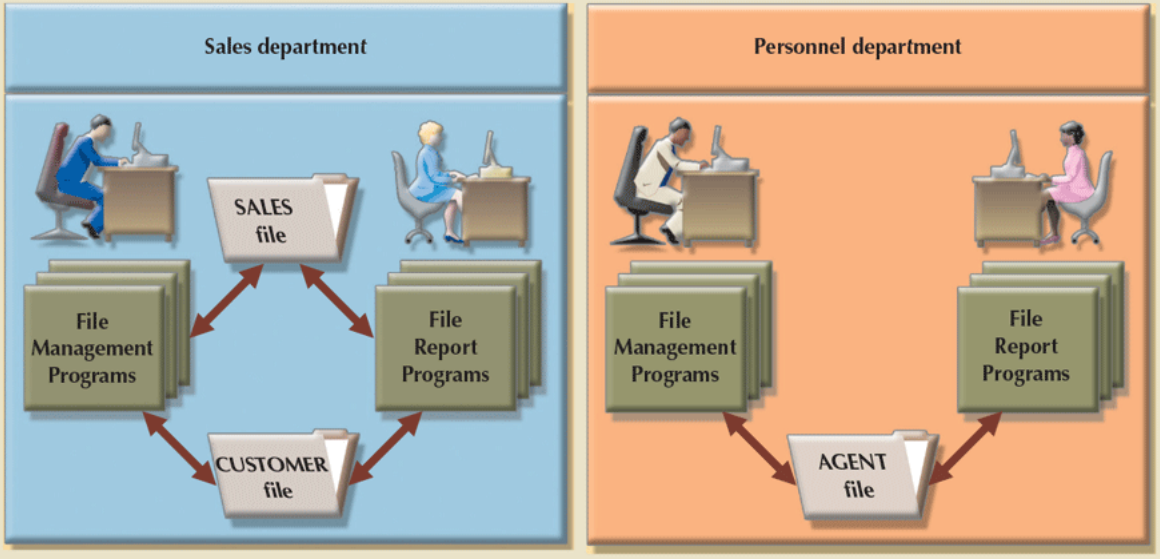
- Manual file systems
  - Accomplished through a system of file folders and filing cabinets
  - Utilize physical file folders and cabinets for data storage and retrieval, a labor-intensive process.
- Computerized file systems
  - Data processing (DP) specialist created a computer-based system to track data and produce required reports
  - Transition to computer-based systems managed by Data Processing (DP) specialists for efficient data tracking and report generation.
- File system redux: modern end-user productivity tools
  - Includes spreadsheet programs such as Microsoft Excel
  - Modernized with end-user productivity tools like spreadsheet programs enhancing data organization and analysis capabilities.

Evolution of File System Data Processing (2 of 3)

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 has little meaning unless it has 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.

Evolution of File System Data Processing (3 of 3)

FIGURE 1.9 A SIMPLE FILE SYSTEM



Simple File System: A basic method of organizing and storing data on a computer, typically consisting of files and folders. In a simple file system, data is stored in a hierarchical structure with directories (folders) containing files. Access and management are performed through basic file commands, and it lacks the advanced features and relational capabilities found in modern database systems.

Problems with File System Data Processing

- Problems with file systems challenge the types of information that can be created from data as well as information accuracy
  - Lengthy development times

Creating and modifying file systems can be time-consuming.

  - Difficulty of getting quick answers

Retrieving specific information may be cumbersome and time-intensive.

  - Complex system administration

Managing file structures and access can be administratively challenging.

  - Lack of security and limited data sharing

File systems often lack robust security measures, and sharing data across systems can be limited.

  - Extensive programming

Implementing changes or retrieving data may require significant programming efforts.

Structural and Data Dependence (1 of 2)

- **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

  Access to a file is tied to its specific structure; any changes require modifying all related programs. For example, if a file containing employee data is restructured, programs accessing that data need adjustment.

- **Structural independence**
  - File structure is changed without affecting the application's ability to access the data

  The file structure can be altered without impacting the application's ability to access data. An example is changing the internal file organization without requiring modifications to applications interfacing with that data. This enhances flexibility and reduces the need for extensive program adjustments.

Structural and Data Dependence (2 of 2)

- Data dependence
  - Data access changes when data storage characteristics change, If alterations are made to the way employee information is stored, it may impact how programs retrieve and manipulate that data.
- Data independence
  - Data storage characteristics are changed without affecting the program's ability to access the data. For instance, adjusting the database's internal structure should not impede an application's capability to retrieve employee details.
- The difference between logical and physical formats exemplifies the practical impact of data dependence. A change in the physical storage shouldn't disrupt the logical representation, maintaining consistency for applications utilizing the data.

Dat Redundancy (1 of 2)

Data Redundancy:

Storing identical data in multiple locations without necessity, leading to inefficiencies and increased risk of inconsistencies. For example, duplicating customer contact information in separate databases for sales and marketing purposes.

Islands of Information:

Scattered data locations create isolated sets of information, complicating data management. An illustration is having customer details stored independently in various departments' databases, hindering unified access and updates.

Increased Probability of Different Versions:

Data redundancy raises the likelihood of having divergent versions of the same data. For instance, if product prices are duplicated in separate systems, updating one may be overlooked in another, causing discrepancies in pricing information.

Data Redundancy (2 of 2)

- **Uncontrolled Data Redundancy Consequences:**

1. **Poor Data Security:** Increased redundancy heightens the risk of unauthorized access and compromises data security.
2. **Data Inconsistency:** Duplicated data leads to discrepancies, making it challenging to maintain consistency across different sources.
3. **Data-Entry Errors:** Repetitive data entry increases the likelihood of introducing errors during input or updates.
4. **Data Integrity Problems:** The unregulated proliferation of redundant data undermines the overall integrity of the database, impacting the reliability of stored information.

Data Anomalies

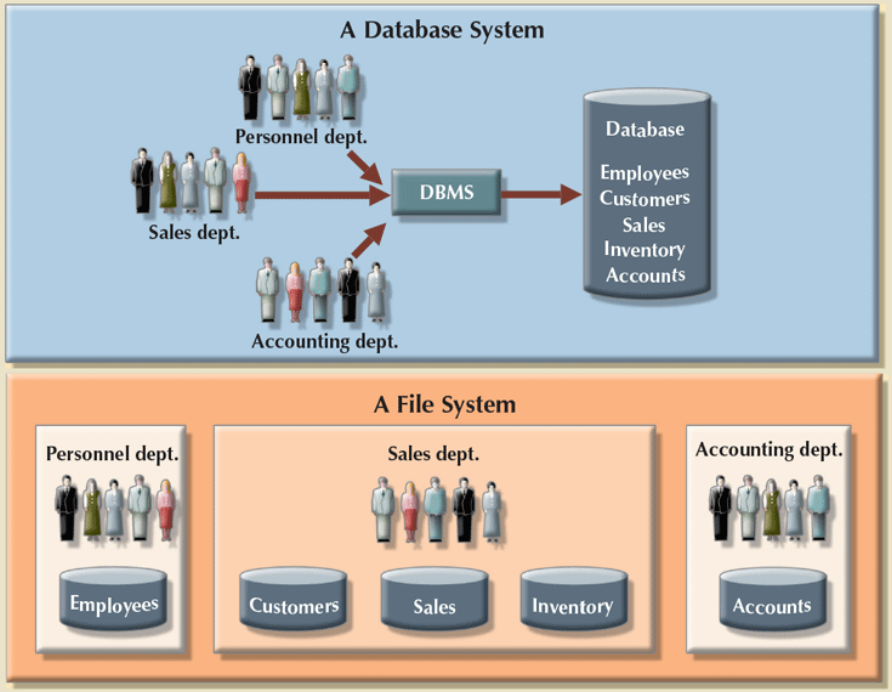
- **Data Anomalies:** Issues that arise when changes to redundant data are not uniformly applied, leading to inconsistencies.
- **Update Anomalies:** Occurs when modifying data in one place but not in all instances, resulting in disparities. For example, updating an employee's title in one record but not in another, causing inconsistency.
- **Insertion Anomalies:** Problems arising during the addition of new data when mandatory attributes are missing, impacting the completeness of information. For instance, adding a new customer record without specifying the associated sales representative.
- **Deletion Anomalies:** Problems occurring when deleting data unintentionally removes essential information, affecting data integrity. For example, deleting a product from a database inadvertently removes associated sales records, leading to incomplete data.

Database Systems (1 of 2)

- **Logical Data Repository:** Data that is logically related is stored together in a single repository, ensuring cohesive organization and facilitating efficient data management.
- **Physically Distributed Storage:** Despite logical cohesion, data can be physically distributed across multiple storage facilities, potentially improving accessibility and redundancy.
- **DBMS Elimination of Issues:** Database Management Systems (DBMS) address and resolve problems such as data inconsistency, anomalies, data dependence, and structural dependence inherent in traditional file systems.
- **Current Generation DBMS:** Modern DBMS software not only stores data structures but also manages relationships between structures and access paths. It defines, stores, and oversees all components critical for efficient data storage and retrieval.

Database Systems (2 of 2)

FIGURE 1.10 CONTRASTING DATABASE AND FILE SYSTEMS



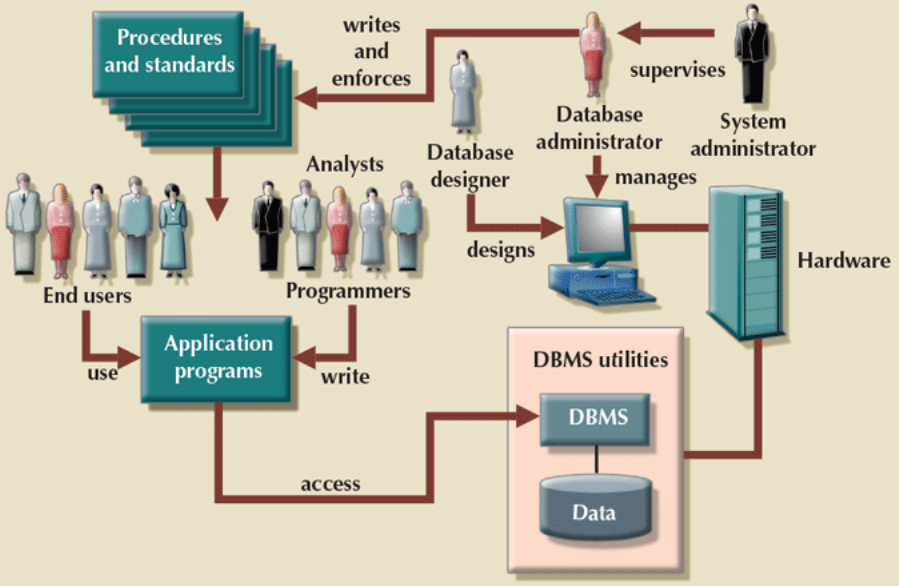
File systems are traditionally flat and lack relational structures, causing redundancy and difficulties in data management, while databases, facilitated by relational database management systems (RDBMS), provide structured, efficient storage, ensuring data integrity and enabling complex queries.

The Database System Environment (1 of 2)

- Database system: organization of components that define and regulate the collection, storage, management, and use of data within a database environment
- **Hardware:** The physical devices and equipment, like servers and storage, essential for the functioning of the database system.
- **Software:** The set of programs and applications enabling data processing, management, and user interactions within the database environment.
- **People:** The individuals involved in the design, maintenance, and usage of the database system, including administrators, developers, and end-users.
- **Procedures:** Established guidelines and protocols governing how data is managed, accessed, and modified within the database system.
- **Data:** The core information stored in the database, representing the valuable content managed and manipulated by the system.

The Database System Environment (2 of 2)

FIGURE 1.11 THE DATABASE SYSTEM ENVIRONMENT



Database System Environment: A dynamic ecosystem encompassing hardware, software, people, procedures, and data, intricately organized to facilitate efficient data collection, storage, management, and utilization.

DBMS Functions (1 of 3)

- **Data Dictionary Management:** Involves the maintenance of a data dictionary that stores comprehensive definitions of data elements and their relationships within the database system.
- **Data Storage Management:** Optimizes performance through performance tuning techniques, ensuring efficient storage and retrieval of data.
- **Data Transformation and Presentation:** Involves formatting data to align with logical expectations, enhancing clarity and usability.
- **Security Management:** Enforces user security and data privacy measures, safeguarding against unauthorized access and ensuring the integrity of sensitive information.

DBMS Functions (2 of 3)

- **Multiuser Access Control:** Incorporates sophisticated algorithms to enable multiple users to access the database simultaneously, ensuring concurrent access without compromising data integrity.
- **Backup and Recovery Management:** Facilitates database recovery after a failure, preserving data integrity by regularly creating backups and implementing recovery procedures.
- **Data Integrity Management:** Minimizes redundancy and maximizes consistency, ensuring the accuracy and reliability of data through systematic checks and controls.

DBMS Functions (3 of 3)

- **Database Access Languages and APIs:** Includes query languages and application programming interfaces (APIs) facilitating user interactions with the database system.
- **Query Language:** Allows users to specify desired actions without detailing how to perform them, enhancing ease of use and abstraction.
- **Structured Query Language (SQL):** A widely adopted query language and data access standard supported by most Database Management System (DBMS) vendors for efficient and standardized communication.
- **Database Communication Interfaces:** Accepts end-user requests through various network environments, providing flexibility and compatibility with different systems and platforms.

Managing the Database System: A Shift in Focus

1. **Increased Costs:** Implementation and maintenance of database systems can lead to higher initial and ongoing expenses.
2. **Management Complexity:** The complexity of managing databases, including design, administration, and security measures, can be challenging.
3. **Maintaining Currency:** Keeping the database current with evolving business requirements and technological advancements requires ongoing efforts.
4. **Vendor Dependence:** Relying on specific database vendors may limit flexibility and increase dependence on their technologies.
5. **Frequent Upgrade/Replacement Cycles:** Database systems may necessitate regular updates or replacements to adapt to changing needs, leading to potential disruptions and costs.

Preparing for Your Database Professional Career

TABLE 1.3	DATABASE CAREER OPPORTUNITIES	
JOB TITLE	DESCRIPTION	SAMPLE SKILLS REQUIRED
Database Developer	Create and maintain database-based applications	Programming, database fundamentals, SQL
Database Designer	Design and maintain databases	Systems design, database design, SQL
Database Administrator	Manage and maintain DBMS and databases	Database fundamentals, SQL, vendor courses
Database Analyst	Develop databases for decision support reporting	QL, query optimization, data warehouses
Database Architect	Design and implementation of database environments (conceptual, logical, and physical)	DBMS fundamentals, data modeling, SQL, hardware knowledge, etc.
Database Consultant	Help companies leverage database technologies to improve business processes and achieve specific goals	Database fundamentals, data modeling, database design, SQL, DBMS, hardware, vendor-specific technologies, etc.
Database Security Officer	Implement security policies for data administration	DBMS fundamentals, database administration, SQL, data security technologies, etc.
Cloud Computing Data Architect	Design and implement the infrastructure for next-generation cloud database systems	Internet technologies, cloud storage technologies, data security, performance tuning, large databases, etc.
Data Scientist	Analyze large amounts of varied data to generate insights, relationships, and predictable behaviors	Data analysis, statistics, advanced mathematics, SQL, programming, data mining, machine learning, data visualization

Summary

- Data consists of raw facts and is usually stored in a database
  - Database design defines the database structure
    - Can be classified according to the number of users, location, as well as data usage and structure
  - Databases evolved from manual and computerized file systems
    - There are some limitations of file system data management
    - DBMSs were developed to address the file system's inherent weaknesses