










Lecture 1

Databases and Database Users

Agenda

-  Introduction
-  Basic Concepts
-  An Example
-  Characteristics of the Database Approach
-  Actors on the Scene
-  Advantages of Using the DBMS Approach
-  When Not to Use a DBMS



Introduction

Introduction

➤ What is a database?

- A **database** is a collection of related data.

➤ What is data?

- Known facts that can be recorded such as the names, telephone numbers, and addresses of people.

Introduction

- Databases and database systems are an essential component of life in modern society.
- We encounter several activities every day that involve some interaction with a database.
- For example:
 - if we go to the bank to deposit or withdraw funds
 - if we make a hotel or airline reservation
 - if we access a computerized library catalog to search for a book



Basic Concepts

Basic Concepts

➤ A database has the following properties:

- A database represents some aspect of the real world, sometimes called the miniworld. Changes to the miniworld are reflected in the database.
- A database is a logically coherent collection of data with some meaning. A random assortment of data cannot correctly be referred to as a database.
- A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some applications in which these users are interested.

Basic Concepts

- A database has some source from which *data* is derived, some degree of *interaction* with events in the real world, and an *audience* that is actively interested in its contents.
- In order for a database to be accurate and reliable at all times, it must be a true reflection of the miniworld that it represents; therefore, changes must be reflected in the database as soon as possible.
- For example:
 - A bank customer withdraws money → Update his account in the database
 - An employee changes his address → Update his address in the database

Basic Concepts - DBMS

➤ Database management system (DBMS)

- DBMS is a computerized system that enables users to create and maintain a database.
- DBMS is a software system that facilitates the processes of *defining*, *constructing*, *manipulating*, and *sharing* databases among various users and applications.

Basic Concepts - DBMS

- **Defining** a database involves specifying the data types, structures, and constraints of the data to be stored in the database.
- **Constructing** the database is the process of storing the data on some storage medium.
- **Manipulating** a database includes retrieving data, modifying the data to reflect changes in the miniworld, and generating reports from the data.
- **Sharing** a database allows multiple users and programs to access the database simultaneously.

Basic Concepts - DBMS

- An **application program** accesses the database by sending queries or requests for data to the DBMS. This causes some data to be read or written into the database.
- Other important function provided by the DBMS include *protecting* the database. **Protection** includes *system protection* against hardware or software crashes and *security protection* against unauthorized or malicious access.
- We will call the **database** and **DBMS** software together a **database system**.

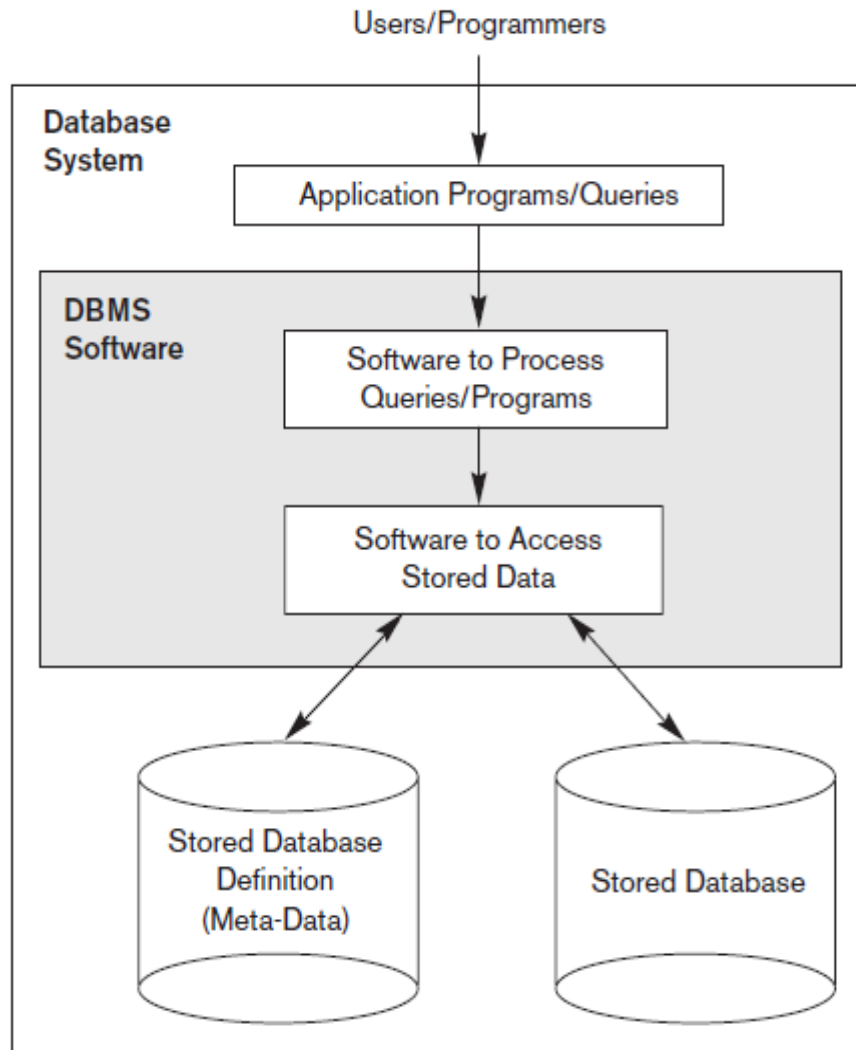


Figure 1.1
A simplified database
system environment.



An Example

An Example - UNIVERSITY

- UNIVERSITY here is our miniworld.
- For a certain UNIVERSITY we have the following **Entities**:
 - **Students** enrolled at this UNIVERSITY
 - **Instructors** that work at this UNIVERSITY
 - **Courses** taught at this UNIVERSITY
- **Entities** are real-world objects.

Columns names

An Example - UNIVERSITY

STUDENT

Student_ID	Student_First_Name	Student_Last_Name
1123	Smith	John
1124	Brown	Sam

INSTRUCTOR

Instructor_ID	Instructor_First_Name	Instructor_Last_Name
11	Anderson	Tom
12	Stone	Barlet

COURSE

Course_ID	Course_Name
C260	Database
C270	Data structures

An Example - UNIVERSITY

Records

STUDENT

Student_ID	Student_First_Name	Student_Last_Name
1123	Smith	John
1124	Brown	Sam

INSTRUCTOR

Instructor_ID	Instructor_First_Name	Instructor_Last_Name
11	Anderson	Tom
12	Stone	Barlet

COURSE

Course_ID	Course_Name
C260	Database
C270	Data structures

An Example - UNIVERSITY

- What is missing?
 - Each **Student** is enrolled in which **Courses**
 - Each **Course** is given by which **Instructor**
- Two additional **Tables (called Relations)** are added to represent the **Relationships** above.

An Example - UNIVERSITY

Students_Enrollments

Student_ID	Course_ID
1123	C260
1123	C270
1124	C260

Courses_Instructors

Course_ID	Instructor_ID
C260	11
C270	12

An Example - UNIVERSITY

- Database **manipulation** involves retrieving and modifying.
- Examples of retrieving are as follows:
 - Get the names of students enrolled in Database course
 - Get the name of the instructor that teaches Data structures
- Examples of modifying are as follows:
 - **Update** the Student_ID of student Smith to 1122
 - **Insert** a new student
 - **Delete** the enrollment of student Smith from Data structures course



Characteristics of the Database Approach

Characteristics of the Database Approach

- A number of characteristics distinguish the database approach from the much older approach of writing customized programs to access data stored in files.
- In **traditional file processing**, each user defines and implements the files needed for a specific software application as part of programming the application.

Characteristics of the Database Approach

- For example, one user, may keep files on students and their grades.
- A second user, may keep track of students' fees and their payments.
- Although both users are interested in data about students, each user maintains separate files because each requires some data not available from the other user's files.

Characteristics of the Database Approach

- This redundancy in defining and storing data results in wasted storage space and in redundant efforts to maintain common up-to-date data.
- In the database approach, a single repository maintains data that is defined only once and then accessed by various users through queries, transactions, and application programs.

Characteristics of the Database Approach

- The main characteristics of the database approach versus the file-processing approach are the following:
 - Self-describing nature of a database system
 - Insulation between programs and data, and data abstraction
 - Support of multiple views of the data
 - Sharing of data and multiuser transaction processing

Characteristics of the Database Approach

➤ **Self-Describing Nature of a Database System:**

- A fundamental characteristic of the database approach is that the database system contains not only the database itself but also a complete definition or description of the database structure and constraints.
- This definition is stored in the DBMS catalog, which contains information such as the type and storage format of each data item, and various constraints on the data.
- The information stored in the catalog is called **meta-data**.

Characteristics of the Database Approach

➤ Self-Describing Nature of a Database System:

- An example of a database catalog:

Relations

Relation_name	No_Of_Columns
STUDENT	2
COURSE	2
INSTRUCTOR	2
.....

Columns

Column_Name	Data_type	Belongs_to_relation
Student_ID	Character (30)	STUDENT
Student_First_Name	Character (40)	STUDENT
.....
Instructor_ID	Character (30)	INSTRUCTOR
.....

Characteristics of the Database Approach

➤ Insulation between Programs and Data, and Data Abstraction:

- In traditional file processing, the structure of data files is embedded in the application programs, so any changes to the structure of a file may require *changing all programs* that access that file.
- By contrast, DBMS access programs do not require such changes in most cases. The structure of data files is stored in the DBMS catalog separately from the access programs. We call this property **program-data independence**.

Characteristics of the Database Approach

➤ Insulation between Programs and Data, and Data Abstraction:

- For example, if we want to add another piece of data to each STUDENT record, say the Birth_date, **traditional file access program** may no longer work and should be changed.
- By contrast, in a **DBMS** environment, we only need to change the *description* of STUDENT records in the catalog.

Characteristics of the Database Approach

➤ **Support of Multiple Views of the Data:**

- A database typically has many types of users, each of whom may require a different perspective or **view** of the database. A view may be a subset of the database.
- For example, one user may be interested only in accessing and printing the transcript of each student; A second user, may be only interested in checking that students have taken all the prerequisites of each course for which the student registers.

Characteristics of the Database Approach

➤ **Sharing of Data and Multiuser Transaction Processing:**

- A multiuser DBMS, as its name implies, must allow multiple users to access the database at the same time.
- The DBMS must include concurrency control software to ensure that several users trying to update the same data do so in a controlled manner so that the result of the updates is correct.
- For example, when several reservation agents try to assign a seat on an airline flight, the DBMS should ensure that each seat can be accessed by only one agent at a time for assignment to a passenger.



Actors on the Scene

Actors on the Scene

- In large organizations, many people are involved in the *design, use, and maintenance* of a large database with hundreds or thousands of users.
- We identify the people whose jobs involve the day-to-day use of a large database; we call them the *actors on the scene*.

Actors on the Scene

➤ Database Administrators:

- In a database environment, the primary resource is the database itself, and the secondary resource is the DBMS and related software.
- Administering these resources is the responsibility of the database administrator (DBA).
- The DBA is responsible for authorizing access to the database, coordinating and monitoring its use, and acquiring software and hardware resources as needed.

Actors on the Scene

➤ Database Designers:

- They are responsible for identifying the data to be stored in the database and for choosing appropriate structures to represent and store this data.
- It is the responsibility of database designers to communicate with all database users in order to understand their requirements and to create a design that meets these requirements.

Actors on the Scene

➤ End Users:

- End users are the people whose jobs require access to the database for querying, updating, and generating reports; the database primarily exists for their use.
- For example,
 - ☐ **Bank customers** and **tellers** check account balances and post withdrawals and deposits.
 - ☐ **Reservation agents** or **customers** for airlines, hotels, and car rental companies check availability for a given request and make reservations.
 - ☐ **Social media users** post and read items on social media Web sites.

Actors on the Scene

- **System Analysts and Application Programmers (Software Engineers):**
 - **System analysts** determine the requirements of end users and develop specifications for transactions that meet these requirements.
 - **Application programmers** implement these specifications as programs; then they test, debug, document, and maintain these transactions.
 - Such analysts and programmers—commonly referred to as **software developers or software engineers**—should be familiar with the full range of capabilities provided by the DBMS to accomplish their tasks.



Advantages of Using the DBMS Approach

Advantages of Using the DBMS Approach

- **Controlling Redundancy**
- **Restricting Unauthorized Access**
- **Providing Backup and Recovery**
- **Providing Multiple User Interfaces**
- **Representing Complex Relationships among Data**
- **Enforcing Integrity Constraints**



When Not to Use a DBMS

When Not to Use a DBMS

- In spite of the advantages of using a DBMS, there are a few situations in which a **DBMS** may involve unnecessary overhead costs that would not be incurred in **traditional file processing**.
- The overhead costs of using a DBMS are due to the following:
 - High initial investment in hardware, software, and training.
 - Overhead for providing security, concurrency control, recovery, and integrity functions.

When Not to Use a DBMS

- Therefore, it may be more desirable not to use a DBMS under the following circumstances:
- Simple, well-defined applications that are not expected to change at all.
 - Embedded systems with limited storage capacity, where a general-purpose DBMS would not fit.
 - No multiple-user access to data



Thank You