Lecture 1

DATABASE MANAGEMENT SYSTEM

Introduction (1/5)

- What is a database?

  Database : Collection of inter-related data

- What is data?

  Data : known (raw) facts that can be recorded and that have implicit meaning

- Examples of data:

  Names, Address, Telephone numbers, etc

Introduction (2/5)

- Databases play important role in all field – medicine, education, e-commerce, business, engineering, law and library science
- Database represents some aspects of the real world – mini-world
- **Changes to the mini-world are reflected in the database**
- **Database is designed, built and populated with data for a specific purpose**

Introduction (3/5)

- Database Management System (DBMS)
  - **Collection of programs that enables users to create and maintain a database**
  - It is a general purpose software system that facilitates the process of – **defining, constructing, manipulating and sharing databases** among various users and applications

Introduction (4/5)

- **Defining** : specify data types, structures, and constraints of data to be stored
- Database information (descriptive) is stored in the database in the form of a catalog called **meta-data**
- **Constructing** : process of storing the data on some storage medium
- **Manipulating** : functions like query, update and generate reports
- **Sharing** : allow multiple users and programs to access the database

Introduction (5/5)

- **Application program** : access the database by sending queries or requests for data to DBMS
- **Query** : cause some data to be retrieved
- **Transaction** : cause some data to be read from and some data to be written into the database

Characteristics of a Database approach (1/7)

- **Traditional file processing**
  - Each user defines and implements the files needed for a specific software application
  - Eg. Grade reporting officer – keep a file on students and their grades
  - Accounting officer – keep track of students' fees and their payments
  - Both users are interested in data about students, they each maintain separate files

Characteristics of a Database approach (2/7)

- **Redundancy** (duplicates) which leads to waste of storage space and efforts
- In database approach, a **single repository** of data is maintained that is defined once and then accessed by many users
- **Four characteristics**
  - Self-describing nature of database system
  - Insulation between programs and data, data abstraction
  - Support of multiple views of data
  - Sharing of data and multi user transaction processing

Characteristics of a Database approach (3/7)

- **Self describing nature of database system**
  - Database system not only contains database but also a **complete definition** of the database structure and constraints
  - Stored in **DBMS catalog** – information like structure of each file, type, storage format of each item and constraints on the data
  - Catalog is called **meta – data**

Characteristics of a Database approach (4/7)

- **Insulation between programs and data, data abstraction**
  - In traditional file processing, the structure of data files is embedded in the application programs
  - Any change to structure of a file needs to change programs that access that file
  - In database, structure is stored in DBMS catalog separately from access programs
  - This is **Program-Data independence**

Characteristics of a Database approach (5/7)

- **Data abstraction** : does not include many details of how the data is stored or how operations are implemented
- DBMS provides users with a **conceptual representation of the data**
- A **data model** is a type of data abstraction
- The data model hides storage and implementation details that are not of interest to most database users

Characteristics of a Database approach (6/7)

- **Support multiple views of the data**
  - Database has many users that may require different perspective (view) of the database
  - **View** : subset of the database, contains virtual data that is derived from the database files but is not explicitly stored

Characteristics of a Database approach (7/7)

- **Sharing of data and multiuser transaction processing**
  - Multiple users can access the database at the same time
  - Essential that multiple applications be **integrated** and **maintained** in a single database
  - DBMS include the **concurrency control** software to ensure that several users trying to update the same data do so in a controlled manner, such that the result of the update is correct

Database Administrator

- Chief administrator that oversee and manage the database resources
- Responsible for authorizing access to the database, coordinating, monitoring its use, and acquiring software and hardware resources as needed
- Accountable for problems like breach of security or poor system response time

End users (1/5)

- They are people whose jobs require access to the database for querying, updating and generating reports

- Categories
  - Casual end users
  - Naïve (parametric) end users
  - Sophisticated end users
  - Standalone users

End users (2/5)

- **Casual end users**
  - Access the database occasionally
  - Need different information each time
  - Use sophisticated database query language to specify their requests
  - Example : Managers

End users (3/5)

- **Naïve (Parametric) end users**
  - Constantly querying and update the database
  - Use standard types of queries and updates called canned transactions
  - Example : Bank tellers, Clerks

End users (4/5)

- **Sophisticated end users**
  - Thoroughly familiarize themselves with the facilities of the DBMS in order to implement their applications to meet their complex requirements
  - Example : Engineers, scientists, business analysts

End users (5/5)

- **Standalone end users**
  - Maintain personal database
  - Use ready-made program packages that provide easy-to-use menu-based or graphic-based interfaces
  - Example : user of a tax package

Database designers (1/2)

- Identify the data to be stored in database
- Choose appropriate structures to represent and store this data
- These tasks are mostly undertaken before database is actually implemented and populated with data
- Responsible to communicate with all prospective database users, in order to understand their requirements, and come up with a design that meets the requirements

Database designers (2/2)

- Interact with each potential group of users and develop a view of the database that meets the data and processing requirements of this group
- These views are analyzed and integrated with the views of other user groups
- Final database design must be capable of supporting requirements of all user groups

Advantages of using DBMS

1. Controlling redundancy
2. Restricting unauthorized access
3. Providing persistent storage for program objects and data structures
4. Permitting inference and actions using rules
5. Providing multiple user interfaces
6. Representing complex relationships among data
7. Enforcing integrity constraints
8. Providing backup and recovery

Controlling Redundancy (1/1)

- For consistency, we should have a database design that stores each logical data item in only one place in database
- Controlled redundancy may be useful for improving performance of queries

Restricting unauthorized access (1/1)

- Multiple users share a database hence need authorization to access all information in database
- Some users may be permitted only to retrieve data, whereas others are allowed both to retrieve and update
- Types of operations – retrieval and update must also be controlled
- Users or user groups are give accounts protected by passwords which they can use to gain access to database

Provide persistent storage for Program objects & Data structures (1/1)

- Database can be used to provide persistent storage for program objects and data structures
- Traditional database systems often suffered from mismatch problem since data structures provided by DBMS were incompatible with the programming language's data structures

Permitting inference and actions using rules (1/1)

- Some database systems provide capabilities for defining deduction rules for inferencing new information from stored database facts

- Such systems are called deductive database systems

Provide multiple user interfaces (1/1)

- DBMS should provide variety of user interfaces as many types of users with varying levels of technical knowledge use a database

- Includes
  - Query languages for casual users
  - Programming language interfaces for application programmers
  - Forms and command codes for parametric users
  - Menu-driven interfaces for stand-alone users

Represent complex relationships among data (1/1)

- A database may include numerous varieties of data that are interrelated in many ways
- DBMS must have capability to represent a variety of complex relationships among data as well as to retrieve and update related data easily and efficiently

Enforce integrity constraints (1/1)

- Simplest type of integrity constraint involves specifying a data type for each data item
- More complex type of constraint that occurs frequently involves specifying that a record in one file must be related to records in other files
- Another constraint specifies uniqueness on data item values
- It is the database designer's responsibility to identify integrity constraints during database design

Provide backup and recovery (1/1)

- DBMS must provide facilities for recovering from hardware or software failures
- The backup and recovery subsystem of DBMS is responsible for recovery
- The recovery subsystem should ensure that the program is resumed from the point at which it was interrupted so that its full effect is recorded in database

Comparison between File System & DBMS (1/2)

	File Based System	Database Approach
Reduction of redundancies	Each user maintains its own files → lots of duplicates, Waste of storage space, inconsistent data, update same data more than once, lots of effort	Data is stored in a single place → save space, time, no inconsistent data
Shared data	Not possible as each user maintains its own files	Possible
Data independence	Application program is more dependent on data. Descriptions of data and logic for accessing data are maintained in the system, built into each application program	Separates data descriptions from data. Details of data are not exposed

Comparison between File System & DBMS (2/2)

	File Based System	Database Approach
Data integrity	Has inconsistent data → redundancy	Provides constraints and checks on entered data
Multiple user interface	Only single user interface	Different interfaces for different types of users
Efficient data access	Problem when volume of data is huge. No cross referencing	Cross referencing of data is possible
Data security	Minimal	Authentication
Data backup & recovery	Not possible	When a program fails, it restores back the previous state of the database
Concurrent transactions	Not supported	Supported

Database System

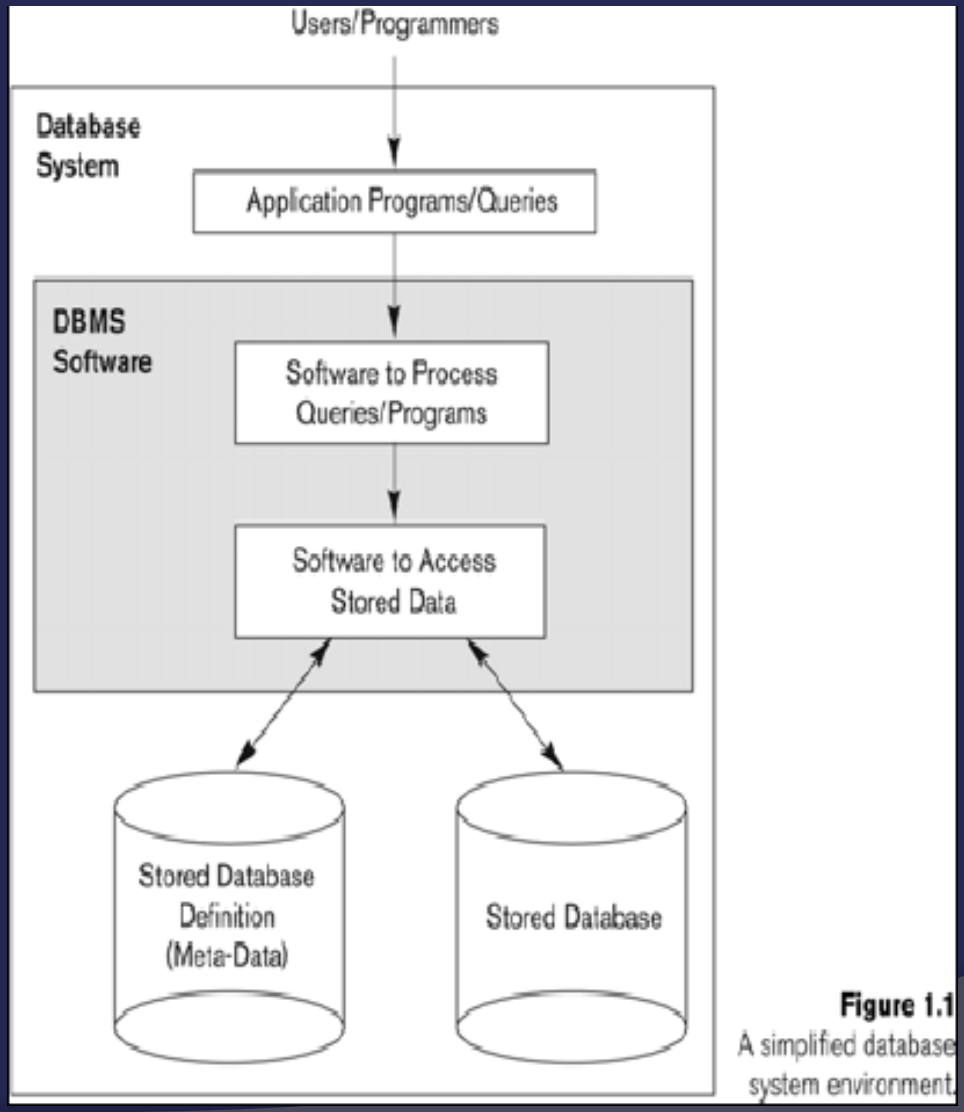

Figure 1.1
A simplified database system environment.