Chapter 1 - Databases

Content

Introduction

- Databases and database systems are an essential component of life in modern society.
- Traditional database applications: Most of the information that is stored and accessed is either textual or numeric. Examples are:

Deposit or withdraw funds

Making a hotel or airline reservation

Access a computerized library catalog

Purchase something online

- Big data or No SQL Systems: Social media websites, such as Facebook, Twitter, instagram among many others
  - has required the creation of huge databases that store nontraditional data, such as posts, tweets, images, and video clips.
  - New types of database systems, often referred to as big data storage systems, or NOSQL systems, have been created to manage data for social media applications.

Introduction

- Database – **unified collection of related data**.
- **Data** means known **facts** that can be recorded and that have some meaning.
- For example, **mobile phones have their own simple database software** and typically store Names, Telephone numbers and Addresses of the people etc.
- A database has:

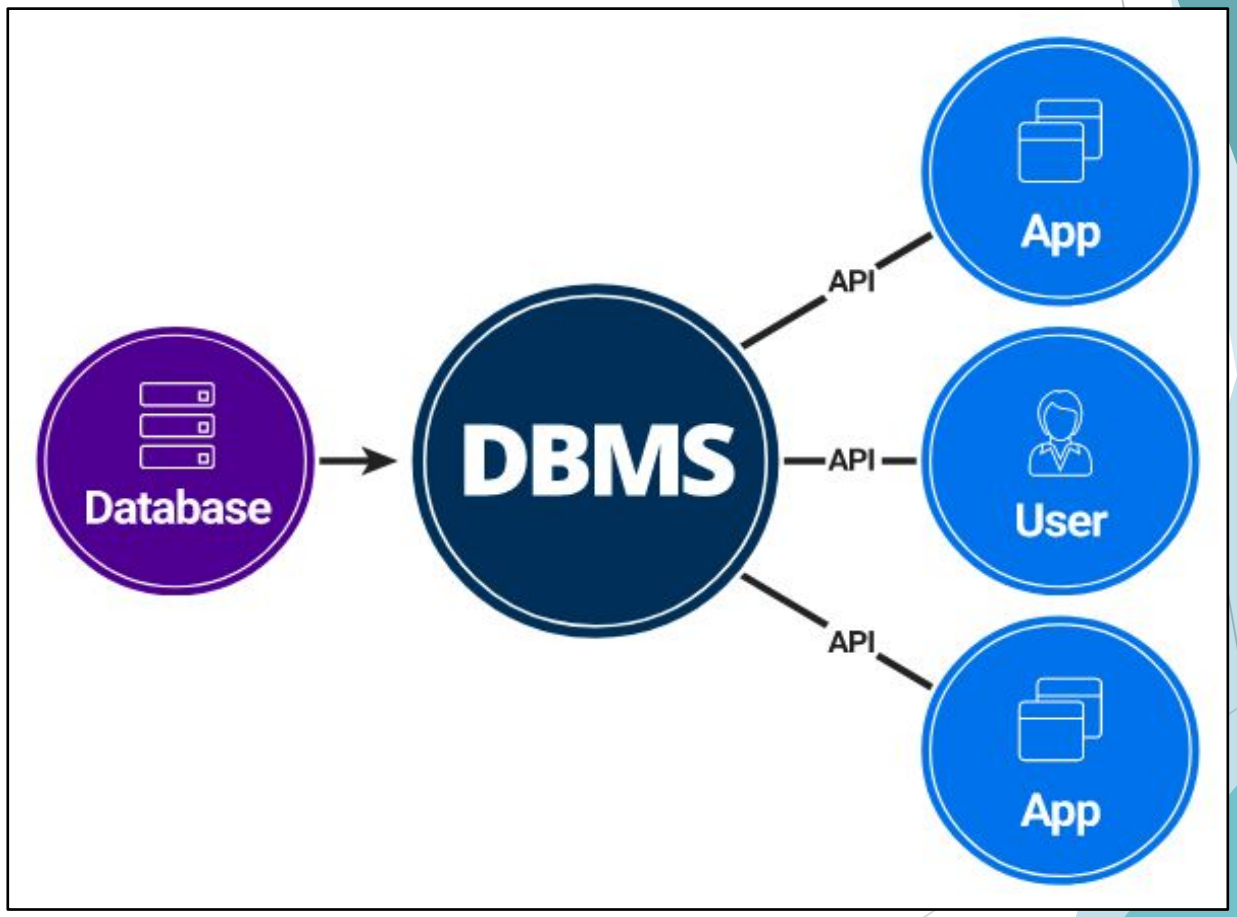
- **Changes are needed in DBMS because of any action performed by user** or (for example, a customer buys a camera) **or some events might occur** (for example, an employee's postal address is changed).
- To maintain accuracy and reliability, its much needed to **update changes in database as soon as possible**.
- A database can be of any **size and complexity**.
  - For example, the list of names and addresses may consist of only a few hundred records.
  - A database of even greater size and complexity would be maintained by a social media company such as Facebook, which has more than a billion users.
  - Example of a **large commercial database is Amazon.com**

Introduction

A database may be generated or maintained manually or computerized.

A computerized database is managed by group of application programs that are written to perform a task.

A database management system (DBMS) is a computerized system that enables users to create and maintain a database.



Introduction

- DBMS - General-purpose software system that facilitates the processes of:

1
Defining

Defining a database involves specifying:
- Data types,
- Structures,
- Constraints of the data.

Meta-data - The database definition stored by the DBMS in the form of a database catalog or dictionary.
Stored in: specialist document/metadata repository

2
Constructing

Process of **storing the data on some storage medium that is controlled by the DBMS.**

3
Manipulating

Includes functions such as querying the database to:
- retrieve specific data,
- updating the database
- and generating reports.

4
Sharing

Allows **multiple users and programs to access the database simultaneously.**

An application program accesses the database by sending queries or requests for data to the DBMS.
A **query** causes some data to be retrieved.
A **transaction** may cause some data to be read and some data to be written into the database.

Introduction

- DBMS provides data protection & maintenance over a longer period of time.
  - **Protection**
    - system protection against hardware or software malfunction (or crashes)
    - security protection against unauthorized or malicious access.
  - **Maintenance**
    - A typical large database may have a life cycle of many years, so the DBMS must be able to maintain the database system by allowing the system to evolve as requirements change over time.
- We can call database and database software together as a database system

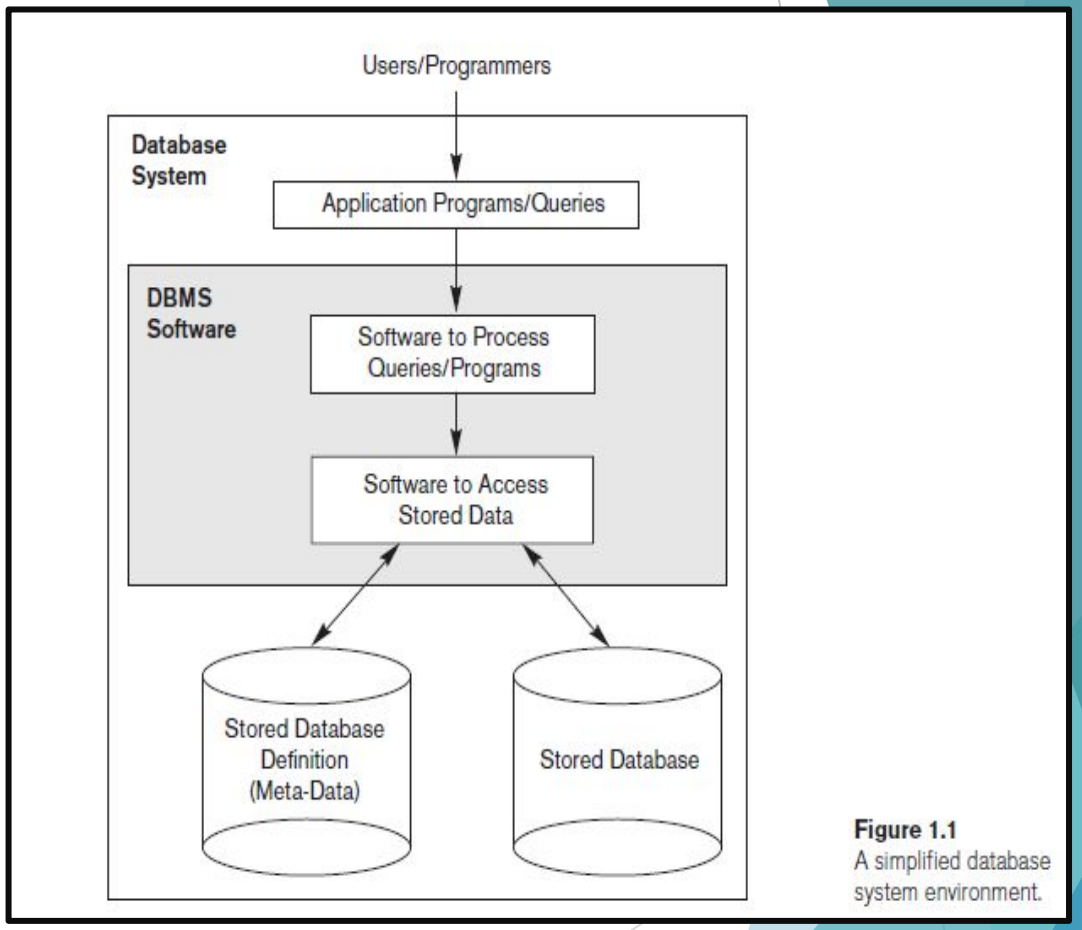


Figure 1.1
A simplified database system environment.

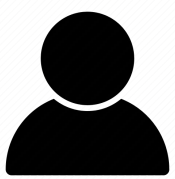
Data Storage

- How data is stored?
  - **file processing systems**
    - Stored using some spreadsheets or notepad files etc.
    - Problems:
      - Have data redundancy issue
      - Redundant efforts needed
      - Storage wastage
      - Inconsistent data formats
  - **Databases**
    - Store data in properly built and normalized data units
    - Centralized repository for entertaining multiple users at a time
    - Avoid data redundancy
- Which approach is better and why?

file system Approach

- Traditional file processing - each user defines and implements the files needed for a specific software application as part of application programming.

USER 1
Grade reporting office
Keep files of students and their grades



USER 2
Accounting office
Keep track of students' fees and their payments

- Similar information of student is stored in two separate files ~ data redundancy

Characteristics of Database Approach

- Database System - a **single repository** maintains data that is defined once and then **accessed by various users** repeatedly **through queries, transactions, and application programs.**
- 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 Database Approach

- **Self-Describing Nature of a Database System**
- A database system contains database as well as the **complete definition of the database structure and constraints**.
- This definition is stored in the DBMS catalog contains information such as:
  - the **structure** of each file,
  - the **data type and storage format** of each data item,
  - and various **constraints on the data**.
- This information is termed as **meta-data**.
- **NoSQL databases do not contain metadata** because data is stored in self describing manner as item name and value stored together.
- Whenever a request is made to access, say, the Name of a STUDENT record, the DBMS software refers to
  - the catalog to determine the structure of the STUDENT file
  - and the position and size of the Name data item within a STUDENT record.
- In file system, you have to give the location or information related to file and in db. it isn't required.

RELATIONS

Relation_name	No_of_columns
STUDENT	4
COURSE	4
SECTION	5
GRADE_REPORT	3
PREREQUISITE	2

COLUMNS

Column_name	Data_type	Belongs_to_relation
Name	Character (30)	STUDENT
Student_number	Character (4)	STUDENT
Class	Integer (1)	STUDENT
Major	Major_type	STUDENT
Course_name	Character (10)	COURSE
Course_number	XXXXNNNN	COURSE
....	....	.....
....	....	.....
....	....	.....
Prerequisite_number	XXXXNNNN	PREREQUISITE

Database catalog in a database

Characteristics of Database Approach

- Insulation between programs and data, and data abstraction

Traditional File Processing	DBMS
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.	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 known as program-data independence.

Characteristics of Database Approach

- Insulation between programs and data, and data abstraction
- For example, a file access program may be written in such a way that it can access only STUDENT records of the structure shown.
- If we want to add another piece of data to each STUDENT record, say the Birth_date, such a program will no longer work and must be changed.
- In a DBMS environment, we only need to change the description of STUDENT records in the catalog to show the new data item Birth_date; no programs are changed.

Data Item Name	Starting Position in Record	Length in Characters (bytes)
Name	1	30
Student_number	31	4
Class	35	1
Major	36	4

Figure 1.4
Internal storage format for a STUDENT record, based on the database catalog in Figure 1.3.

Characteristics of Database Approach

- Insulation between programs and data, and data abstraction
- In some types of database systems, such as object-oriented and object-relational systems users can **define operations** on data as part of the database definitions.
- Operation (also called a function or method) is specified in two parts.
  - The interface (or signature) - includes the operation name and the data types of its arguments (or parameters).
  - The implementation (or method) - is specified separately and can be changed without affecting the interface.
- User application programs can operate on the data by invoking these operations through their names and arguments called as **program-operation independence.**
- For example, an operation CALCULATE_GPA can be applied to a STUDENT object to calculate the grade point average.
- **Data abstraction** - The characteristic that allows program-data independence and program-operation.
- *A DBMS provides users with a conceptual representation of data that does not include many of the details of how the data is stored or how the operations are implemented.*

Characteristics of Database Approach

- Support of Multiple Views of the Data
- View - a subset of the database or it may contain virtual data that is derived from the database files but is not explicitly stored.
- A multiuser DBMS whose users have a variety of distinct applications must provide facilities for defining multiple views.
- For example:
  - One user of the database may be interested only in accessing and printing the transcript of each student; the view for this user is shown in Figure 1.5(a).
  - A second user, who is interested only in checking that students have taken all the prerequisites of each course for which the student registers, may require the view shown in Figure 1.5(b).

(a)

TRANSCRIPT

Student_name	Student_transcript				
	Course_number	Grade	Semester	Year	Section_id
Smith	CS1310	C	Fall	08	119
	MATH2410	B	Fall	08	112
Brown	MATH2410	A	Fall	07	85
	CS1310	A	Fall	07	92
	CS3320	B	Spring	08	102
	CS3380	A	Fall	08	135

(b)

COURSE_PREREQUISITES

Course_name	Course_number	Prerequisites
Database	CS3380	CS3320
		MATH2410
Data Structures	CS3320	CS1310

Figure 1.5
Two views derived from the database in Figure 1.2. (a) The TRANSCRIPT view. (b) The COURSE_PREREQUISITES view.

Characteristics of Database Approach

- Sharing of Data and Multiuser Transaction Processing

- DBMS **must allow multiple users to access the database at the same time.**
- DBMS should have a **concurrency control system** that ensures one transaction performance at a time. And avoid writing or reading conflicts.
- **Transaction** - A transaction is a unit of program execution that accesses and possibly updates various data items.
- ACID properties of transaction processing are:
  - **A**tomicity . Either all operations of the transaction are properly reflected in the database or none are.
  - **C**onsistency . Execution of a transaction in isolation preserves the consistency of the database.
  - **I**solation: effect of a transaction is independent of other transactions in a system
  - **D**urability: changes made by a transaction should not be lost by any failure

Files vs. Databases

File Systems	Databases
File System is a software that manages & organizes files on a storage medium in your PC	DBMS is a software for managing databases
Data redundancy	No data redundancy issue
No backup/recovery methods for data if its lost	Data backup & recovery system is available
No efficient query processing system as you get the whole file in return of a keyword search	Efficient query processing system
Inconsistent data may exist because of data redundancy issue	Data consistency ensured with normalization
Less security	More secure (views, stored procedures etc.)
Less expensive	DBMS creation and management needs a lot of efforts and time. So its costly.

Advantages Of Using DBMS

- Controlling Redundancy
- ***In file processing Systems, every user group maintains its own files for handling its data-processing applications.***
- For example: **UNIVERSITY database**
- Two groups of users: 1. Course registration personnel 2. Accounting office.
- In the traditional approach, each group independently keeps files on students.
  - 1.The accounting office keeps data on registration and related billing information
  - 2. The registration office keeps track of student courses and grades.

Advantages Of Using DBMS

- **Controlling Redundancy**
- This **redundancy** in storing the same data multiple times leads to several problems.

1. Need to perform a single logical update such as entering data on a new student multiple times, once for each file where student data is recorded. This leads to **duplication of effort**.

2. **Storage space is wasted** when the same data is stored repeatedly.

3. Files that represent the **same data may become inconsistent**. This may happen **because an update is applied to some of the files but not to others**.

Even if an update—such as adding a new student—is applied to all the appropriate files, the data concerning the student may still be inconsistent because **the updates are applied independently by each user group.**

- For example, one user group may enter a student's birth date erroneously as 'JAN-19-1988', whereas the other user groups may enter the correct value of 'JAN-29-1988'.

- This data redundancy can be controlled in databases using normalization.

Advantages Of Using DBMS

- **Restricting Unauthorized Access**
- When **multiple users share a large database,** most **users will not be authorized to access all information in the database.**
- **Users' access and data management should be controlled and security policies are enforced by DBA.**
  - For example, financial data such as salaries and bonuses is often considered confidential, and only authorized persons are allowed to access such data.
  - A DBMS should provide a security and authorization subsystem
    - the DBA uses to create accounts and specify account restrictions.
    - For example, only the DBA's staff may be allowed to use certain privileged software, such as the software for creating new accounts.

Advantages Of Using DBMS

- **Providing Persistent Storage for Program Objects**
- Databases can be used to provide persistent storage for program objects and data structures.

- Programming languages typically have complex data structures, such as structs or class definitions in C++ or Java. The values of program variables or objects are discarded once a program terminates, unless stored in permanent files. storing the items require saving structures in to suitable file formats.

- A complex object in C++ can be stored permanently in an object-oriented DBMS. Such an object is said to be **persistent**, since it survives the termination of program execution. Object oriented DBMS allows you to use the persistent objects later on.

- OODBMS offers data structure compatibility with one or more OOP languages

Advantages Of Using DBMS

- Providing Storage Structures and Search Techniques for Efficient Query Processing
- The DBMS must provide specialized data structures and search techniques to speed up disk search for the desired records.
- Indexes are often used for this purpose.
  - Indexes are typically based on tree data structures or hash data structures that are highly effective for disk search.
  - **Choosing which index to speedup searching is done by DBA**
- An efficient query execution plan for each query based on the existing storage structures is selected by DBMS.
  - Done by query processing & optimization module in DBMS.
  - We will discuss that in later chapters

Advantages Of Using DBMS

- **Providing Backup and Recovery**
- The **backup and recovery subsystem** of the DBMS is responsible for recovery.
  - For example, if the computer system fails in the middle of a complex update transaction, the recovery subsystem is responsible for making sure that the database is restored to the state it was in before the transaction started executing.
  - Creating save points, checkpoints and transaction rollback in case of any failure
  - We will cover that in detail while studying transaction processing systems

Advantages Of Using DBMS

- **Providing Multiple User Interfaces**
- Many types of users with different levels of technical knowledge use a database, so a DBMS should provide a variety of user interfaces that include:
  - apps for mobile users,
  - query languages for casual users,
  - programming language interfaces for application programmers,
  - Forms/menu-driven interfaces also called GUI for standalone users.

Advantages Of Using DBMS

- Representing Complex Relationships among Data
- A DBMS must handle many varieties of data that are interrelated in many ways.
- The record for ‘Brown’ in the STUDENT file is related to four records in the GRADE_REPORT file.

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

Advantages Of Using DBMS

- **Enforcing Integrity Constraints**
- Most database applications have integrity constraints that must hold.
- The simplest type of integrity constraint involves specifying a **data type for each data item.**
  - For example, we specified that the value of the Class data item within each STUDENT record must be a one-digit integer and that the value of Name must be a string of no more than 30 alphabetic characters.
- A more complex type of constraint that involves specifying that a record in one file must be related to records in other files known as **a referential integrity constraint.**
  - For example, in Figure 1.2, we can specify that every section record must be related to a course record.
- Another type of constraint specifies uniqueness on data item values, such as every course record must have a unique value for Course_number. This is known as a **key or uniqueness constraint.**

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

Advantages Of Using DBMS

- **Permitting Inferencing and Actions Using Rules and Triggers**
- **Deductive database systems deduce new information from the stored database facts** Because it contains the deduction rules for inferring from the data.
  - **For example, there may be complex rules for determining when an employee is on probation.** These can be specified as rules, which when compiled and maintained by the DBMS can determine all employees on probation.
- A **trigger** is a form of a rule activated by updates to the table, which results in performing some additional operations to some other tables, sending messages, and so on.
- More involved procedures to enforce rules are popularly called **stored procedures**, they become a part of the overall database definition and are invoked when certain conditions are met.

When Not to Use 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
    - DBMS needs a high-speed processor and also a large memory size leads to costly hardware resources and Educated staff (database administrator, application programmers, data entry operations).
  - The generality that a DBMS provides for defining and processing data
  - Overhead for providing security, concurrency control, recovery, and integrity functions
- Therefore, it may be more desirable to develop customized database applications under the following circumstances:
  - **Realtime requirements** for some application programs **that may not be met because of DBMS overhead**
  - **Embedded systems with limited storage capacity**, where a general-purpose DBMS would not fit
  - **No multiple-user access to data**
- Certain industries and applications have elected not to use general-purpose DBMSs.
  - For example, many computer-aided design (CAD) tools used by mechanical and civil engineers have proprietary file and data management software that is geared for the internal manipulations of drawings and 3D objects.