Database Systems

Lecture # 1,2,3

Chapter # 1

Databases and Database Users

eman.shahid@nu.edu.pk

Subject's Marks Distribution

Class Activities	Max. Marks
Mid Term Examination	30
Assignments	10
Class Participation	1
Project	9
Final Examination	50

Chapter Outlines

1. Introduction
2. Characteristics of Database Approach
3. Files Vs. Databases
4. Advantages of using DBMS
5. When not to use DBMS

General Idea

- **Essential component of life:** most of us encounter several activities every day that involve some interaction with a database.
  - Banking(money deposit and withdraw)
  - Hotel Reservation
  - Airline Reservation
  - Accessing online Library
  - Shopping (Daraz, AliExpress, Amazon etc.)
  - Car Booking (Careem, Uber etc.)

General Idea

- Traditional databases: Most of the information that is stored and accessed is either textual or numeric.
- Non-traditional databases/Bigdata storage systems/ NOSQL systems: created to manage data for social media applications.
  - Facebook (Posts, images and video clips)
  - Twitter (Tweets, images and video clips)
  - Google
  - Amazon
  - Yahoo
- A large amount of data now resides on the “cloud”. which means it is in huge data centers using thousands of machines.

Introduction

- **Data:** Known facts that can be recorded and have an implicit meaning;
- **Database:** a highly organized, interrelated, and structured set of data.

Introduction

- A database can be of any size and complexity.
- An example of a large commercial database is Amazon.com. It contains:
  - Data for over 60 million active users, and millions of books, CDs, videos, DVDs, games, electronics, apparel, and other items.
  - The database occupies over 42 terabytes.

Properties of Database

- A database has the following implicit properties:
  - A database represents some aspect of the real world, sometimes called the miniworld or the universe of discourse (UoD). Changes to the miniworld are reflected in the database.
  - A database is a logically coherent collection of data with some inherent 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 preconceived applications in which these users are interested.

Database management system (DBMS)

- Computerized system that enables users to create and maintain a database.
  - For example: MySQL, Oracle, etc. are a very popular commercial database which is used in different applications.
- **General-purpose software system:** facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications.

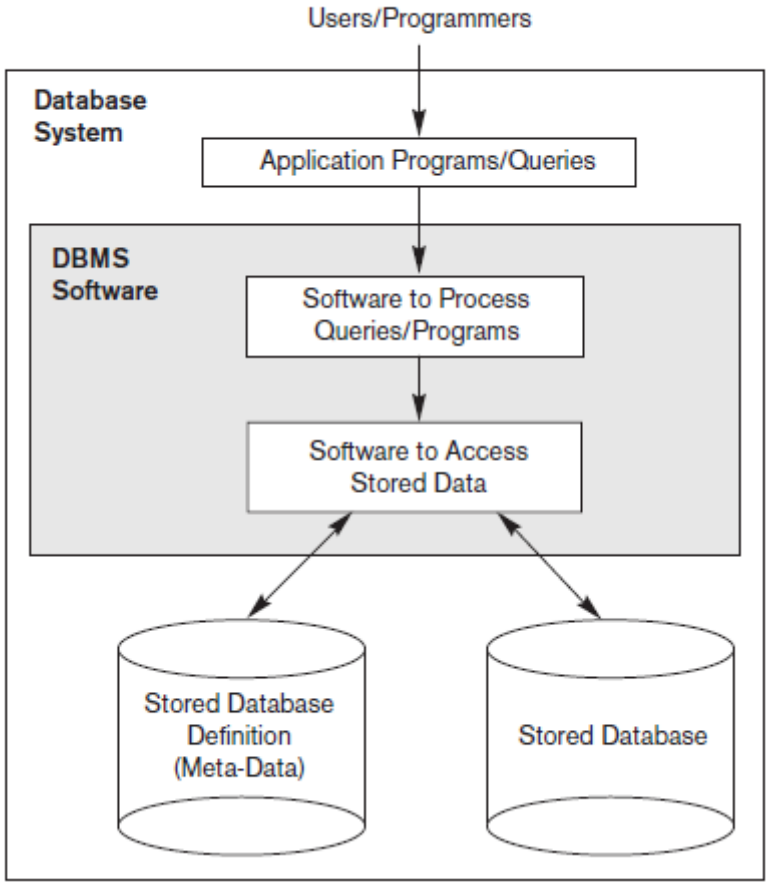
DBMS Functionality

- **Define a database** : in terms of data types, structures and constraints
- **Construct or Load:** storing data on a secondary storage medium.
- **Manipulating the database :** querying, generating reports, insertions, deletions and modifications to its content
- Concurrent Processing and Sharing by a set of users and programs – yet, keeping all data valid and consistent.

DBMS FUNCTIONALITY

- An application program accesses the database by sending queries for data to the DBMS.
  - **Query:** to retrieve and manipulate data.
  - Transaction: that reads and write data into the database.
- Other important functions provided by the DBMS include:
  - **Protection** against hardware or software malfunction and unauthorized or malicious access.
  - **Maintenance:** database can be maintained and updated for a long period of time.

Simplified Database System Environment



UNIVERSITY Database Example

- Some mini-world entities:
  - Students
  - Courses
  - Sections (of courses)
  - Departments
  - Instructors
- Some mini-world relationships:
  - Sections are of specific Courses
  - Students take Sections
  - Courses have Prerequisite Courses
  - Instructors teach Sections
  - Courses are offered by departments
  - Students major in Departments

UNIVERSITY Database Example

STUDENT

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

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

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

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2
A database that stores student and course information.

Characteristics of Database Approach

- **Self-describing nature of a database system.**
  - Database system contains not only the database itself but a complete definition or description of the database structure and constraints.
  - This definition is stored in the DBMS catalog.
    - Structure of each file
    - Type
    - Storage
    - Format of each data item
    - Various constraints on the data.
- The information stored in the catalog is called meta-data, and it describes the structure of the primary database.

Example of a **DATABASE CATALOG**

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

Figure 1.3
An example of a database catalog for the database in Figure 1.2.

Characteristics of Database Approach

- **Insulation between programs and data, and data abstraction**
  - **Program-data independence:** Allows changing data storage structures and operations without having to change the DBMS access programs.
  - **Program-operation independence:** User application programs can operate on the data by invoking operations through their names and arguments, regardless of how the operations are implemented.
- The characteristic that allows program-data independence and program-operation independence is called data abstraction.

Internal storage format for a STUDENT record, based on the database catalog

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

- **Support of multiple views of the data**
  - A database typically has many types of users, each of whom may require a different view of the database.
  - View: subset of the database or it may contain virtual data that is derived from the database files but is not explicitly stored.

UNIVERSITY Database Example

STUDENT

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

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

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

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2
A database that stores student and course information.

Characteristics of Database Approach

(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

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

Characteristics of Database Approach

- **Sharing of data and multiuser transaction processing.**
  - Multiple users to access the database at the same time.
  - **Concurrency control:** ensures that several users trying to update or retrieve the same data in so controlled manner.
    - i.e., Airline Reservation
    - A fundamental role of multiuser DBMS software is to ensure that concurrent transactions operate correctly and efficiently.

Files vs Databases

- Program data dependence.
- No Data Security.
- Data Inconsistency and lack of data Integrity.
- Not Flexible.
- Separation and isolation of data
- Duplication of data.
- Fixed queries

Advantages of using DBMS

- **Controlling Redundancy**
  - Duplication of efforts.
  - Storage space is wasted when the same data is stored repeatedly.
  - **Inconsistency:** 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
- **Restricting Unauthorized Access**
  - Using particular accounts protected by password
  - For example, financial data such as salaries and bonuses is often considered confidential, and only authorized persons are allowed to access such data.

Advantages of using DBMS

- **Providing Persistent Storage for Program Objects.**
- **Providing Storage Structures and Search Techniques for Efficient Query Processing.**
  - **Data buffering or caching:** DBMSs do their own data buffering.
  - **Query processing and optimization module:** responsible for choosing an efficient query execution plan for each query based on the existing storage structures.

Advantages of using DBMS

- **Providing Backup and Recovery**
- **Providing Multiple User Interfaces**
  - Query languages for casual users
  - Programming language interfaces for application programmers
  - Forms and command codes for parametric users
  - Menu-driven interfaces
  - Natural language interfaces for standalone users.
- **Representing Complex Relationships among Data**
- **Enforcing Integrity Constraints.**

Additional Implications of Using the Database Approach

- **Potential for Enforcing Standards.**
  - Standards can be defined for names and formats of data elements, display formats, report structures, terminology, and so on.
- **Reduced Application Development Time**
  - once a database is up and running, substantially less time is generally required to create new applications.
- **Flexibility to change the structure of a database as requirements change.**
- **Availability of Up-to-Date Information**
  - Reservation systems
  - Banking databases, and it is made possible by the concurrency control and recovery subsystems of a DBMS
- **Economies of Scale**
  - Combining all the organization's operational data into one database and creating a set of applications that work on this one source of data can result in cost savings.

When Not To Use DBMS

- **Situations in which a DBMS may involve unnecessary overhead costs.**
  - High initial investment in hardware, software, and training.
  - The basic properties that a DBMS provides for defining and processing data
  - Overhead for providing security, concurrency control, recovery, and integrity functions.

When Not To Use DBMS

- No multiple-user access to data.
- Embedded systems with limited storage capacity, where a general-purpose DBMS would not fit.
- Stringent, real-time requirements for some application programs that may not be met because of DBMS overhead.
  - i.e., telephone switching systems
- Simple, well-defined database applications that are not expected to change at all.