

DATABASE MANAGEMENT SYSTEM



Outline

- Introduction to database
- Basic Definitions
- Typical DBMS Functionality
- Example of a Database (UNIVERSITY)
- Main Characteristics of the Database Approach
- Database Users
- Advantages of Using the Database Approach
- When Not to Use Databases

What do you mean by DBMS?

- **Database:**
 - A collection of related data.
- **Data:**
 - Known facts that can be recorded and have an implicit meaning
- **Database Management System (DBMS):**
 - A database management system (DBMS) is a collection of programs that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications
- **Database System:**
 - The DBMS software together with the data itself. Sometimes, the applications are also included

What is meta-data?

- It is the data about the data.
- The information stored in the system catalog is called meta-data, and it describes the structure of the primary database.

Example of a simple database

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	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	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.

Example of a simplified database catalog

RELATIONS

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

Figure 1.3

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

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

Note: Major_type is defined as an enumerated type with all known majors. XXXXNNNN is used to define a type with four alpha characters followed by four digits.

Simplified Database System

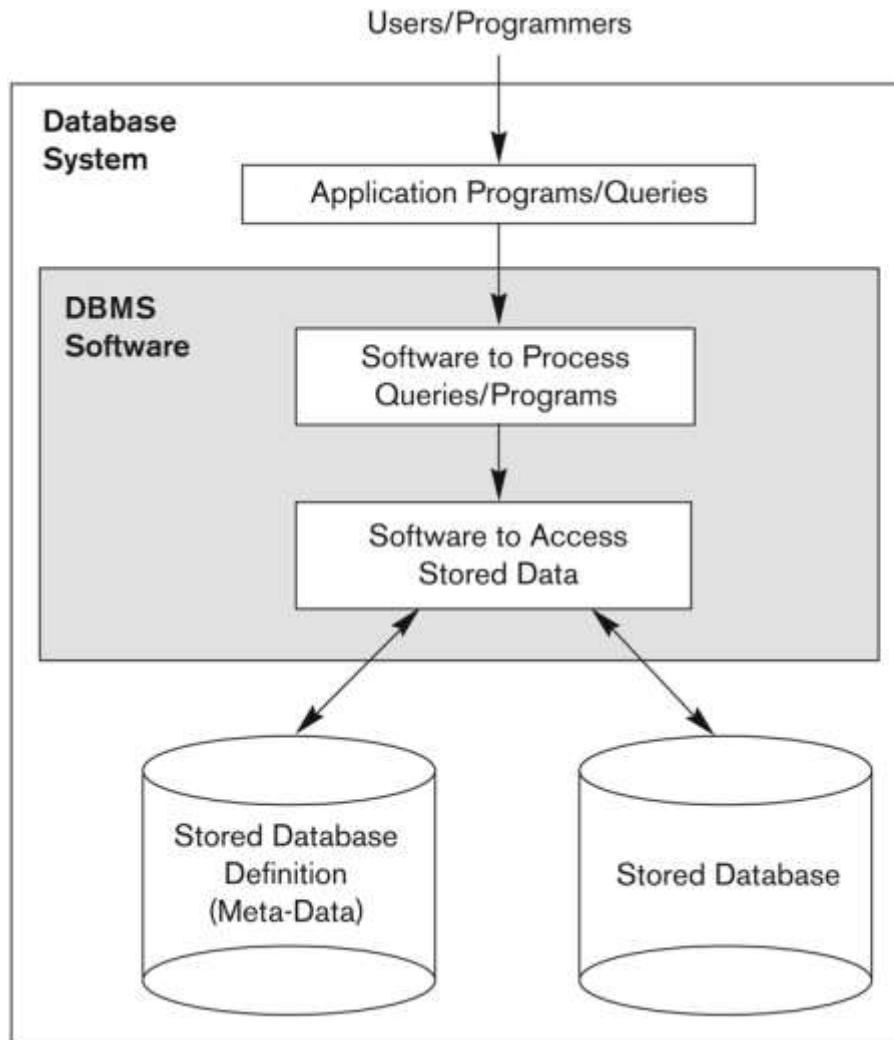


Figure 1.1
A simplified database system environment.

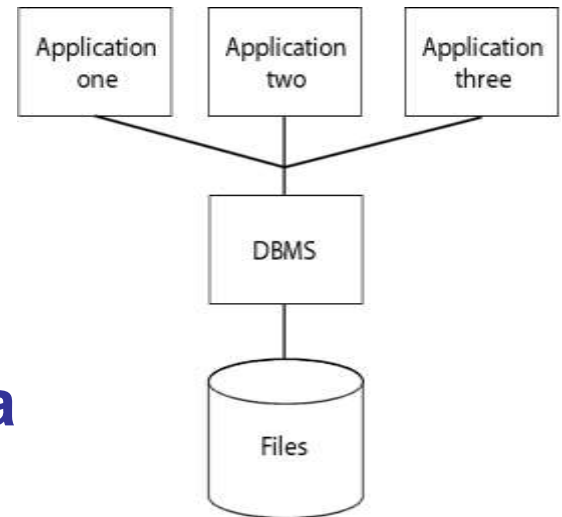
Typical DBMS Functionality

- **Define** a particular database in terms of its data types, structures, and constraints
- **Construct or Load** the initial database contents on a secondary storage medium
- **Manipulating** the database:
 - Retrieval: Querying, generating reports
 - Modification: Insertions, deletions and updates to its content
 - Accessing the database through Web applications
- **Processing and Sharing** by a set of concurrent users and application programs – yet, keeping all data valid and consistent

Main Characteristics of the Database Approach

- **Self-describing nature of a database system:**

- A DBMS **catalog** stores the description of a particular database (e.g. data structures and types)
- The description is called **meta-data**.
- This allows the DBMS software to work with different database applications.



- **Insulation between programs and data**

- Called **program-data independence**.
- Allows changing data structures and storage organization without having to change the DBMS access programs.

- **Program-operation independence:**

- User application programs can operate on the data by invoking these operations through their names and arguments, regardless of how the operations are implemented.

- **Data Abstraction:**

- 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.

- **Support of multiple views of the data:**
 - Each user may see a different view of the database, which describes **only** the data of interest to that user.
 - A view may be a subset of the database.

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

(a)

COURSE_PREREQUISITES

Course_name	Course_number	Prerequisites
Database	CS3380	CS3320
		MATH2410
Data Structures	CS3320	CS1310

(b)

Figure 1.5

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

- **Sharing of data and multi-user transaction processing:**

- Allowing a set of **concurrent users** to retrieve from and to update the database
- Care is needed to avoid interferences
- *Concurrency control* within the DBMS guarantees that each **transaction** is correctly executed or aborted
- *Recovery* subsystem ensures each completed transaction has its effect permanently recorded in the database

Database Users

- Users may be divided into
 - Those who actually use and control the database content, and those who design, develop and maintain database applications (called “Actors on the Scene”), and
 - Those who design and develop the DBMS software and related tools, and the computer systems operators (called “Workers Behind the Scene”).

Database Users

- Actors on the scene
 - **Database Designers:**
 - Responsible to define the content, the structure, the constraints, and functions or transactions against the database. They must communicate with the end-users and understand their needs.
 - **Database administrators:**
 - Responsible for authorizing access to the database, for coordinating and monitoring its use, acquiring software and hardware resources, controlling its use and monitoring efficiency of operations.

- **End-users:** They use the data for queries, reports and some of them update the database content.
- **System Analysts and Application Programmers (Software Engineers) :**
 - They implement these specifications as programs; then they test, debug, document, and maintain. 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.

Advantages of Using the Database Approach

- Controlling redundancy in data storage and in development and maintenance efforts.
- Restricting Unauthorized Access.
- Backup and recovery subsystem
- Sharing of data among multiple users.
- Ensuring consistency of data.
- Enforcing Integrity Constraints
- Providing storage structures (e.g. indexes) for efficient query processing

