DATABASE MANAGEMENT SYSTEMS

Fundamentals of Database Systems

FIFTH EDITION

VTU

Includes Solved University Model Question Papers

Ramez Elmasri

Shamkant B. Navathe

ALWAYS LEARNING

PEARSON

Unit 1

Introduction: Characteristics of Database approach, Actors on the Scene, Workers behind the scene, Advantages of using DBMS approach, Data models, schemas and instances, Three schema architecture and data independence, Database languages and interfaces, the database system environment, Centralized and client-server architectures, Classification of Database Management systems

Entity-Relationship Model: Conceptual Database using high level conceptual data models for Database Design, A Sample Database Application, Entity types, Entity sets Attributes and Keys Relationship types, Relationship Sets, Roles and Structural Constraints Weak Entity Types.

Basic Definitions

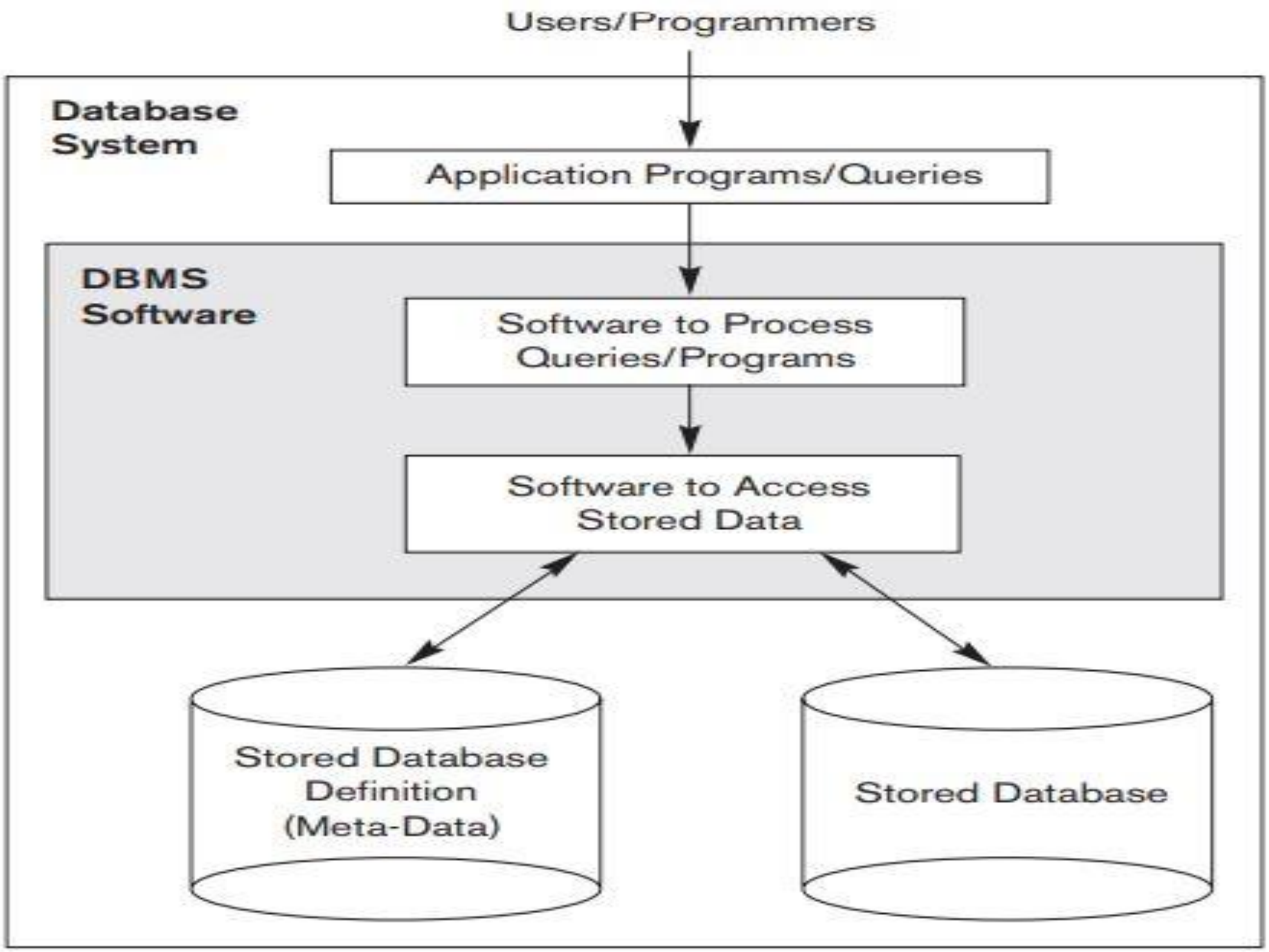
Database: A collection of related data.

Data: Known facts that can be recorded and have an implicit meaning.

Mini-world: Some aspect of the real world about which data is stored in a database. For example, student grades and transcripts at a university.

Database Management System (DBMS): A software/collection of programs that enables users to create and maintain a database.

Database System: The DBMS software together with the data itself. Sometimes, the applications are also included.


Users/Programmers
Database System
Application Programs/Queries
DBMS Software
Software to Process Queries/Programs
Software to Access Stored Data
Stored Database Definition (Meta-Data)
Stored Database

Typical DBMS Functionality

Define a database : in terms of data types, structures and constraints

Construct or Load the Database 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

Typical DBMS Functionality

Other features:

- Protection or Security measures to prevent unauthorized access.
- “Active” processing to take internal actions on data.
- Presentation and Visualization of data.

Example of a Database (with a Conceptual Data Model)

Mini-world for the example: Part of a UNIVERSITY environment.

Some mini-world *entities*:

- STUDENT
- COURSE
- SECTION
- GRADE REPORT
- PREREQUUISITE

Note: The above could be expressed in the ENTITY-RELATIONSHIP data model.

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.

Example of a Database (with a Conceptual Data Model)

Some mini-world *relationships*:

- SECTIONs *are of* specific COURSEs
- STUDENTs *take* SECTIONs
- COURSEs *have* prerequisite COURSEs
- STUDENTs *secured GRADE*

Note: The above could be expressed in the *ENTITY-RELATIONSHIP* data model.

Main Characteristics of the Database Approach

Self-describing nature of a database system: A DBMS **catalog** stores the *description* of the database. The description is called **meta-data**). This allows the DBMS software to work with different databases.

Insulation between programs and data, and Data Abstraction: Called **program-data independence**. Allows changing data storage structures and operations **program-operation independence** without having to change the DBMS access programs.

Main Characteristics of the Database Approach

Data Abstraction: A **data model** is used to hide storage details and present the users with a *conceptual view* of the database.

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.

Main Characteristics of the Database Approach

Sharing of data and multiuser transaction processing: allowing a set of concurrent users to retrieve and to update the database. Concurrency control within the DBMS guarantees that each **transaction** is correctly executed or completely aborted. OLTP (Online Transaction Processing) is a major part of database applications.

Database Users

- Users may be divided into those who actually use and control the content called “Actors on the Scene” and
- Those who enable the database to be developed and the DBMS software to be designed and implemented called “Workers Behind the Scene”.

Database Users

Actors on the scene

- **Database administrators:** responsible for authorizing access to the database, for co-ordinating and monitoring its use, acquiring software, and hardware resources, controlling its use and monitoring efficiency of operations.
- **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.
- **End-users:** they use the data for queries, reports and some of them actually update the database content.

Categories of End-users

Casual : access database occasionally when needed

Naïve or Parametric : they make up a large section of the end-user population. They use previously well-defined functions in the form of "canned transactions" against the database. Examples are bank-tellers or reservation clerks who do this activity for an entire shift of operations.

Categories of End-users

Sophisticated : these include business analysts, scientists, engineers, others thoroughly familiar with the system capabilities. Many use tools in the form of software packages that work closely with the stored database.

Stand-alone : mostly maintain personal databases using ready-to-use packaged applications. An example is a tax program user that creates his or her own internal database.

Database Users

workers behind the scene

- **DBMS system designers and implementers:** design and implement the DBMS modules and interfaces as a software package.
- **Tool developers:** s design and implement tools—the software packages that facilitate database modeling and design, database system design, and improved performance.
- **Operators and maintenance personnel:** are responsible for the actual running and maintenance of the hardware and software environment for the database system.

Advantages of Using the Database Approach

- Controlling Redundancy
- Restricting Unauthorized Access
- Providing persistent storage for program Objects
- Providing Storage Structures and Search Techniques for Efficient Query Processing
- Providing Backup and Recovery
- Providing Multiple User Interfaces
- Representing Complex Relationships among Data
- Enforcing Integrity Constraints
- Permitting Inferencing and Actions Using Rules

Additional Implications of Using the Database Approach

- **Potential for enforcing standards**: this is very crucial for the success of database applications in large organizations Standards refer to data item names, display formats, screens, report structures, meta-data (description of data) etc.
- **Reduced application development time**: incremental time to add each new application is reduced.

Additional Implications of Using the Database Approach

- **Flexibility to change data structures**: database structure may evolve as new requirements are defined.
- **Availability of up-to-date information:** very important for on-line transaction systems such as airline, hotel, car reservations.
- **Economies of scale**: by consolidating data and applications across departments wasteful overlap of resources and personnel can be avoided.