GLA UNIVERSITY
MATHURA
Recognised by UGC Under Section 2(f)

Accredited with A Grade by NAAC

DATABASE MANAGEMENT SYSTEM BCSC0003

Dr. Neeraj Gupta
Assistant Professor, Dept. of CEA
neeraj.gupta@gla.ac.in

OUTLINE

❑ Three-Schema Architecture

❑ Data Independence

❑ Database Users

THREE-SCHEMA ARCHITECTURE

Proposed to support DBMS characteristics of:

- **Program-data independence.**
- Support of **multiple views** of the data.

Not explicitly used in commercial DBMS products, but has been useful in explaining database system organization

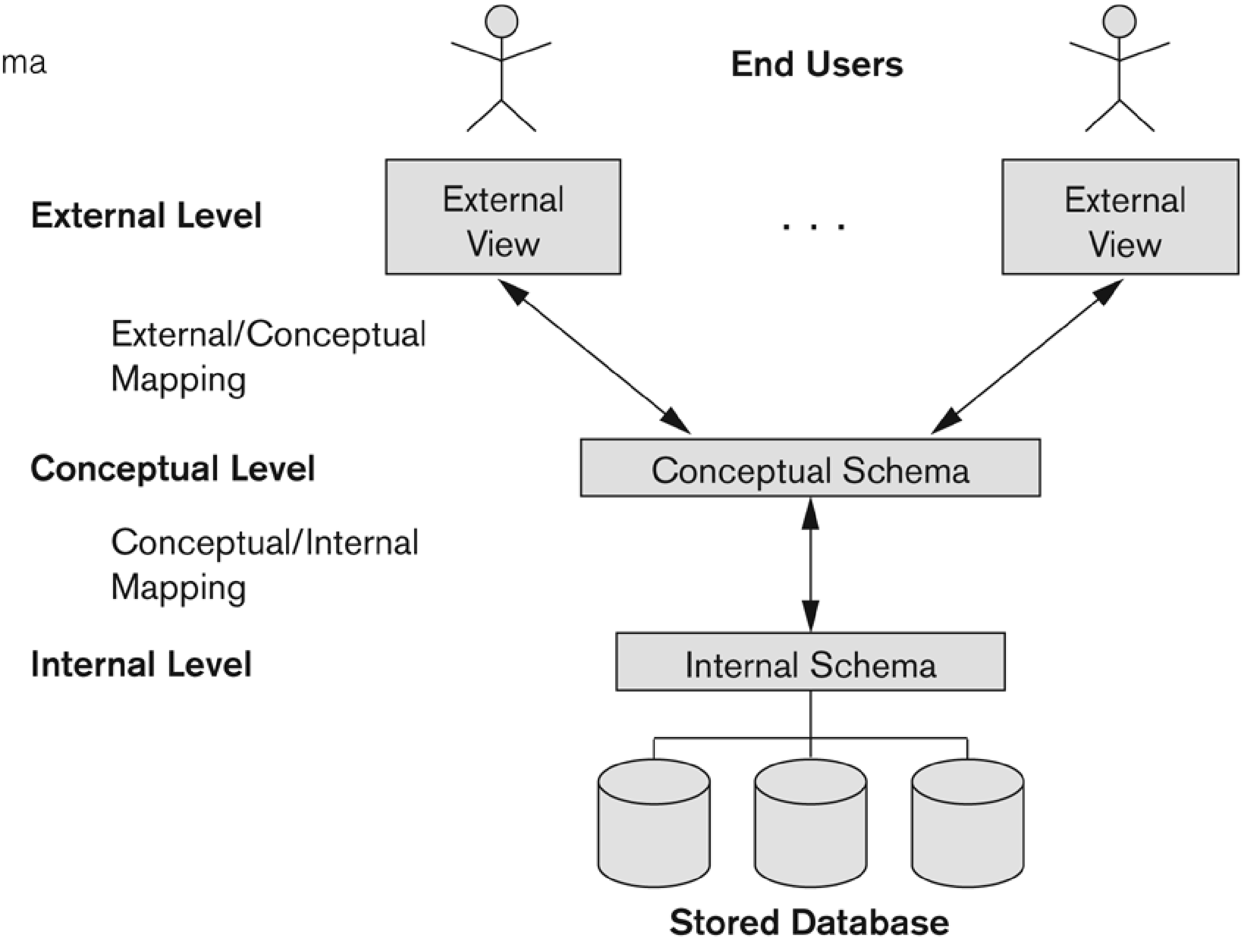
THREE-SCHEMA ARCHITECTURE

Defines DBMS schemas at ***three*** levels:

- **Internal schema** at the internal level to describe physical storage structures and access paths (e.g indexes).
  - Typically uses a **physical** data model.
- **Conceptual schema** at the conceptual level to describe the structure and constraints for the whole database for a community of users.
  - Uses a **conceptual** or an **implementation** data model.
- **External schemas** at the external level to describe the various user views.
  - Usually uses the same data model as the conceptual schema.

THE THREE-SCHEMA ARCHITECTURE

Figure 2.2
The three-schema architecture.



THREE-SCHEMA ARCHITECTURE

Mappings among schema levels are needed to transform requests and data.

Programs refer to an external schema are mapped by the DBMS to the internal schema for execution.

DATA INDEPENDENCE

- **Logical Data Independence:** The capacity to change the conceptual schema without having to change the external schemas and their application programs.

- **Physical Data Independence:** The capacity to change the internal schema without having to change the conceptual schema.

DATA INDEPENDENCE

When a schema at a lower level is changed, only the **mappings** between this schema and higher-level schemas need to be changed in a DBMS that fully supports data independence.

The higher-level schemas themselves are *unchanged*. Hence, the application programs need not be changed since they refer to the external schemas.

SIMPLIFIED DATABASE SYSTEM ENVIRONMENT

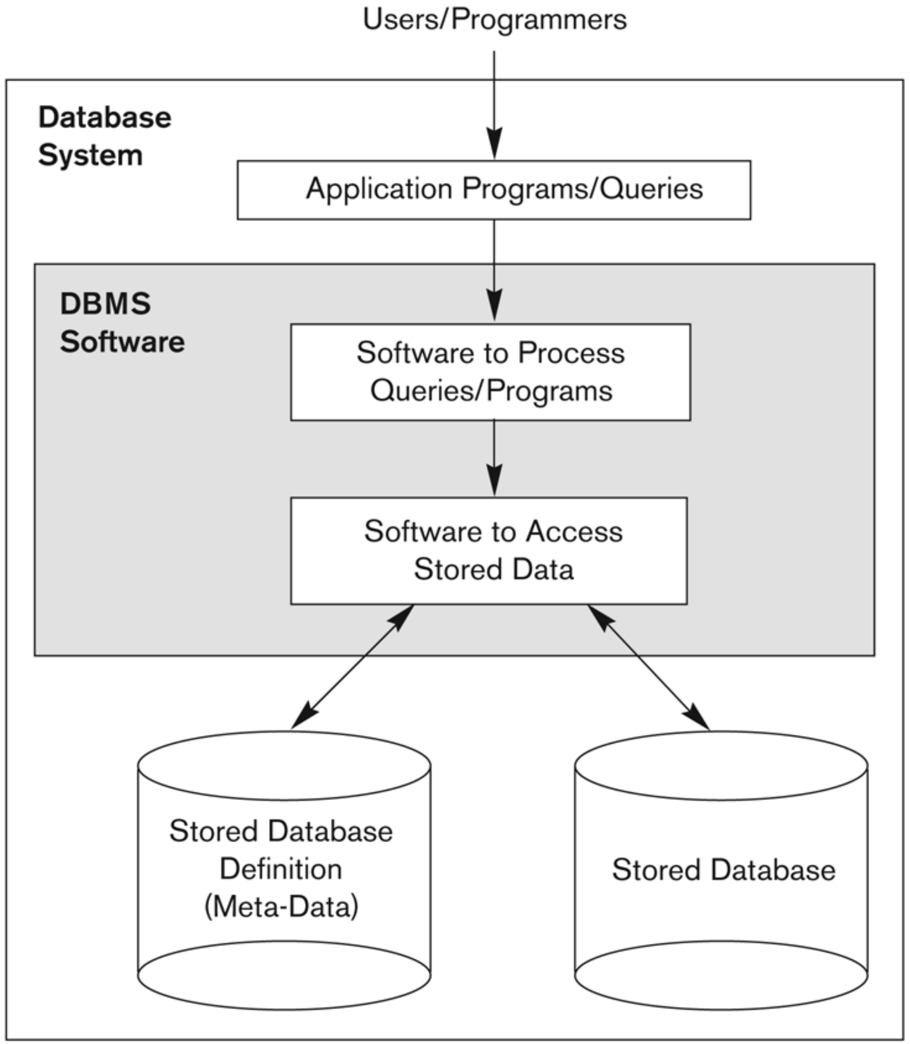

Users/Programmers
Database System
Application Programs/Queries
DBMS Software
Software to Process Queries/Programs
Software to Access Stored Data
Stored Database Definition (Meta-Data)
Stored Database

Figure 1.1
A simplified database system environment.

DATABASE USERS

Users may be divided into

- Those who 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 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.

- **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.

CATEGORIES OF END-USERS

Actors on the scene (continued)

- **End-users:** They use the data for queries, reports and some of them update the database content.
- **End-users can be categorized into:**
  - **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 (CONTINUED)

- **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 its own internal database.
  - Another example is a user that maintains an address book

ACTORS ON THE SCENE

System Analysts/Application Programmers:

System analysts determine the requirements of end user, especially naïve and parametric end user, and develop specification for canned transaction that meet these requirement.

Application programmers implement these specification as programs.

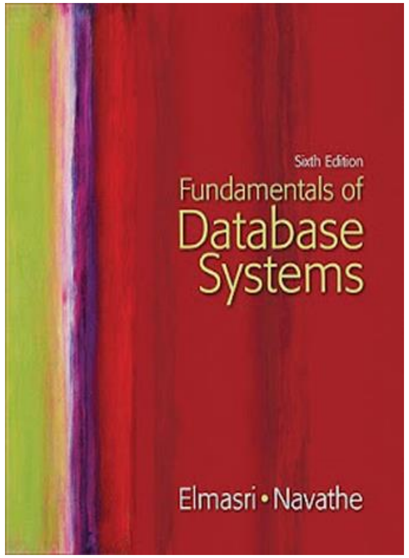
WORKERS BEHIND THE SCENE

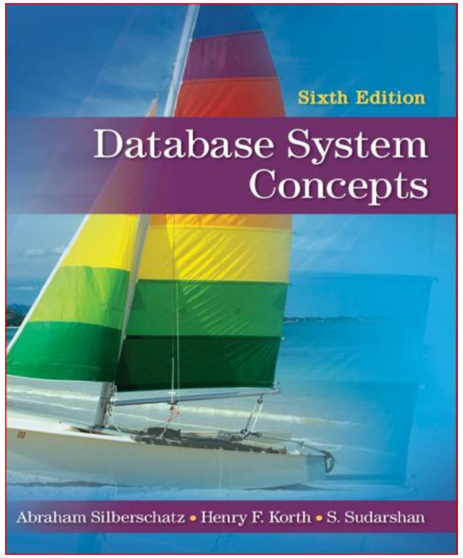
DBMS system designers and implementers are persons who design and implement the DBMS modules and interfaces as a software package.

Tool developers include persons who design and implement tools- the software packages that facilitate database system design and use and that help to improve performance.

Operators and maintenance personnel are the system administration personnel who are responsible for the actual running and maintenance of the hardware and software environment for the database system.

REFERENCE BOOKS





Keep Learning
Keep Growing