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

Accredited with A Grade by NAAC

APPLIED DATABASE MANAGEMENT SYSTEM
BCSC0014

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

OUTLINE

- Three-Schema Architecture
- Data Independence
- Types of DBMS Languages

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

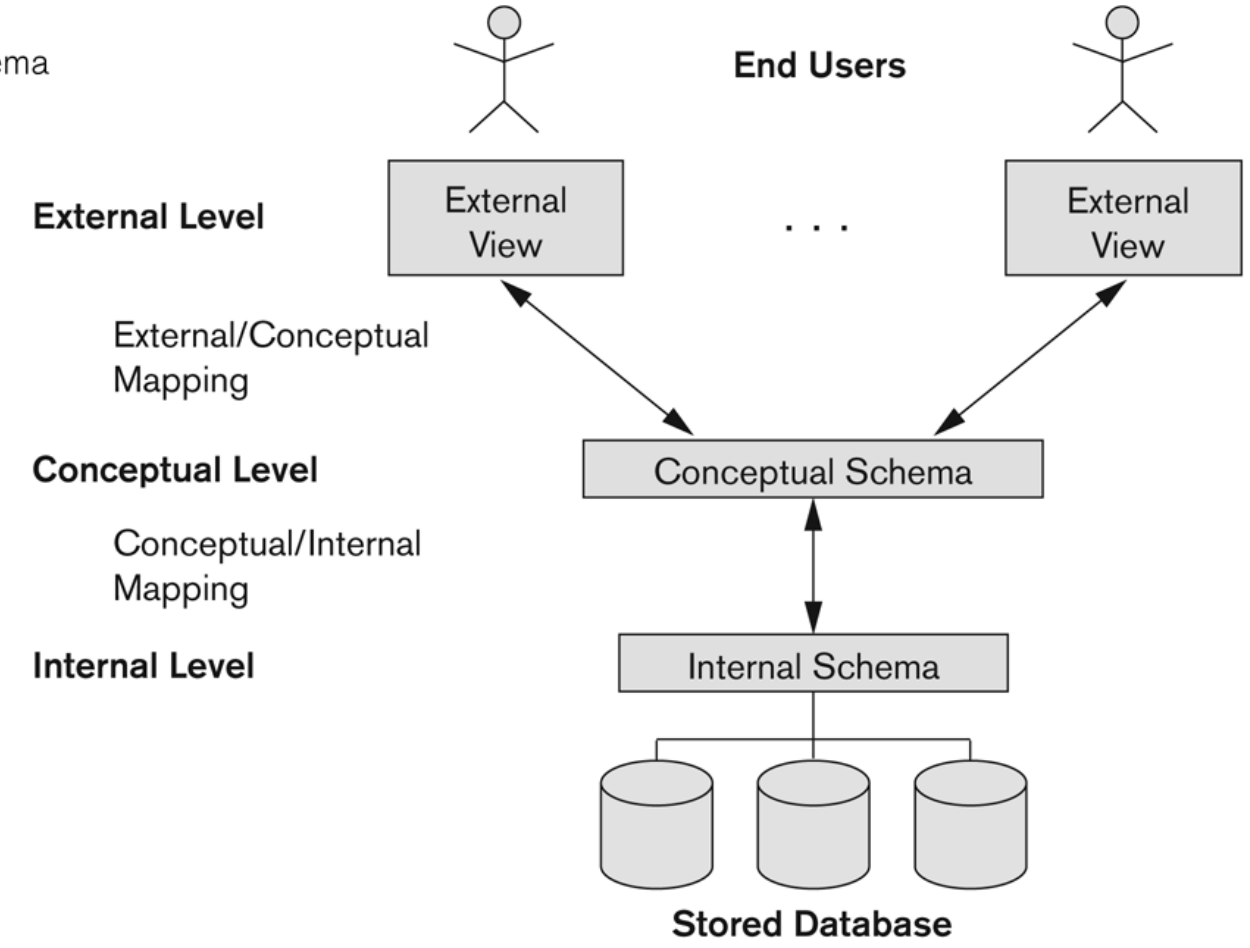
THREE-SCHEMA ARCHITECTURE

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, and 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.

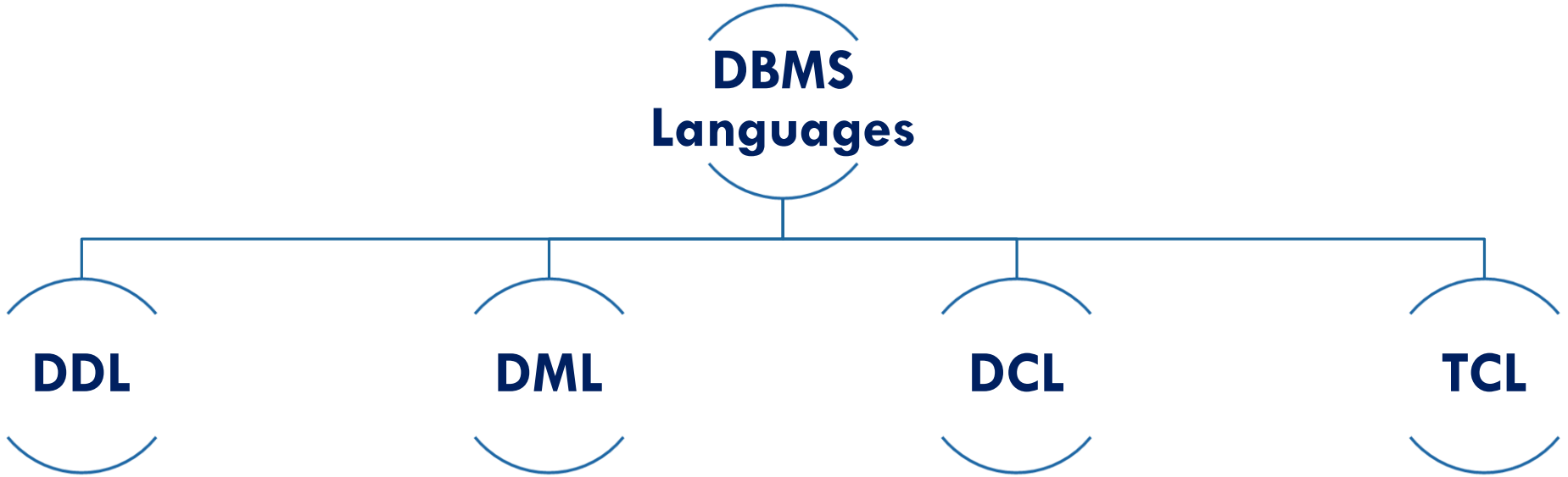
- **Physical Data Independence:** The capacity to change the internal schema without having to change the conceptual schema.

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.

TYPES OF DBMS LANGUAGES



DATA DEFINITION LANGUAGE (DDL)

DDL is used for specifying the database schema. It is used for creating tables, schema, indexes, constraints etc. in database. Lets see the operations that we can perform on database using DDL:

- To create the database instance – **CREATE**
- To alter the structure of database – **ALTER**
- To delete tables in a database instance – **TRUNCATE**
- To rename database instances – **RENAME**
- To drop objects from database such as tables – **DROP**

All of these commands either defines or update the database schema that's why they come under Data Definition language.

DATA MANIPULATION LANGUAGE (DML)

DML is used for accessing and manipulating data in a database. The following operations on database comes under DML:

- To read records from table(s) – **SELECT**
- To insert record(s) into the table(s) – **INSERT**
- Update the data in table(s) – **UPDATE**
- Delete all the records from the table – **DELETE**

DATA MANIPULATION LANGUAGE (DML)

There are basically two types of DML

- **High Level** or **Non-procedural Languages**: which requires a user to specify what data is needed with out specifying how to get it.
- **Low Level** or **Procedural Languages**: which requires a user to specify what data is needed and how to get it.

DATA CONTROL LANGUAGE (DCL)

DCL is used for granting and revoking user access on a database –

- To grant access to user – **GRANT**
- To revoke access from user – **REVOKE**

TRANSACTION CONTROL LANGUAGE(TCL)

The changes in the database that we made using DML commands are either performed or roll backed using TCL.

- To persist the changes made by DML commands in database – **COMMIT**
- To rollback the changes made to the database – **ROLLBACK**

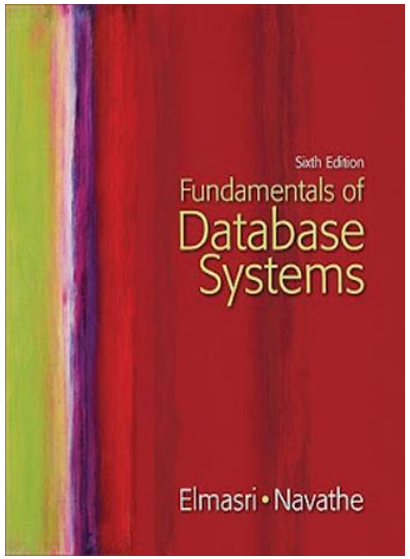
DBMS INTERFACES

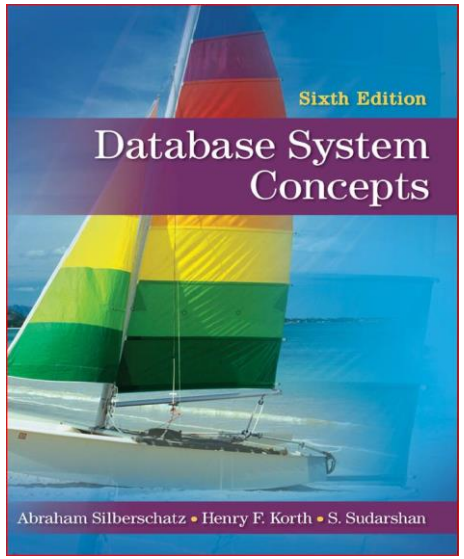
- Stand-alone query language interfaces.
- **Programmer interfaces for embedding DML in programming languages:**
  - Pre-compiler Approach
  - Procedure (Subroutine) Call Approach
- **User-friendly interfaces:**
  - Menu-based, popular for browsing on the web
  - Forms-based, designed for naïve users
  - Graphics-based (Point and Click, Drag and Drop etc.)
  - Natural language: requests in written English
  - Combinations of the above

OTHER DBMS INTERFACES

- Speech as Input and Output
- Web Browser as an interface
- Parametric interfaces (e.g., bank tellers) using function keys.
- Interfaces for the DBA:
  - Creating accounts, granting authorizations
  - Setting system parameters
  - Changing schemas or access path

REFERENCE BOOKS






Keep Learning
Keep Growing