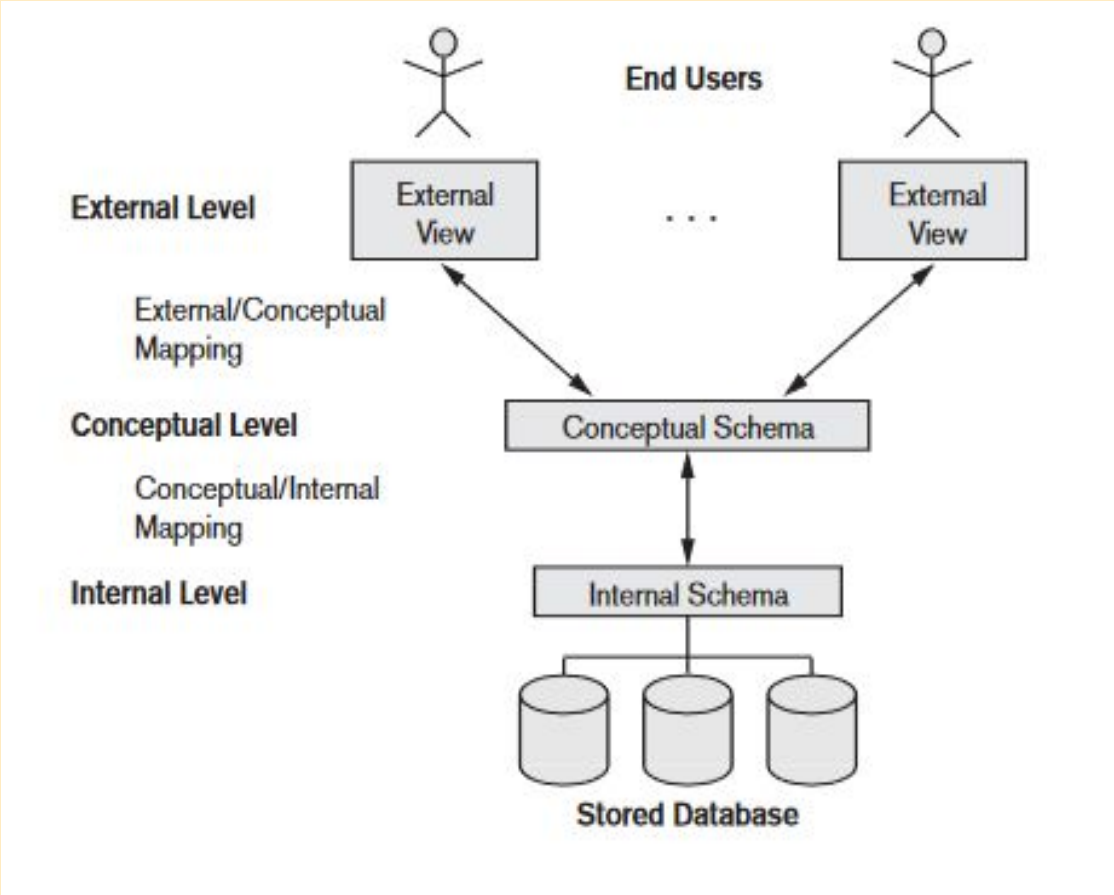
UNIT I

INTRODUCTION

Recap

- **Data Models**
- **Categories of data models**
- **Schema V/s Instances**
- **Database Schema Vs. Database State**
- **The Three Schema Architecture**

The Three Schema Architecture



- In this architecture, schemas can be defined at the following three levels:
  - Internal level.
  - Conceptual level and
  - External level

Mappings

- In a DBMS based on the three-schema architecture, each user group refers to its own external schema.
- Hence, the DBMS must transform a request specified on an external schema into a request against the conceptual schema, and then into a request on the internal schema for processing over the stored database.
- If the request is a database retrieval, the data extracted from the stored database must be reformatted to match the user's external view.
- The processes of transforming requests and results between levels are called **mappings.**
  - These mappings may be time-consuming.
  - Some DBMSs—especially those that are meant to support small databases—do not support external views

Data Independence

- The three-schema architecture can be used to further explain the concept of **data independence.**
- The capacity to change the schema at one level of a database system without having to change the schema at the next higher level.
- Two types of data independence:
  1) Logical data independence
  2) Physical data independence

Logical data independence:

- The capacity to change the conceptual schema without having to change external schemas or application

programs.

Data Independence

- We may change the conceptual schema:
  - to expand the database (by adding a record type or data item),
  - to change constraints, or
  - to reduce the database (by removing a record type or data item) - in this case, external schemas that refer only to the remaining data should not be affected.
- After the conceptual schema undergoes a logical reorganization, application programs that reference the external schema constructs must work as before.

Data Independence

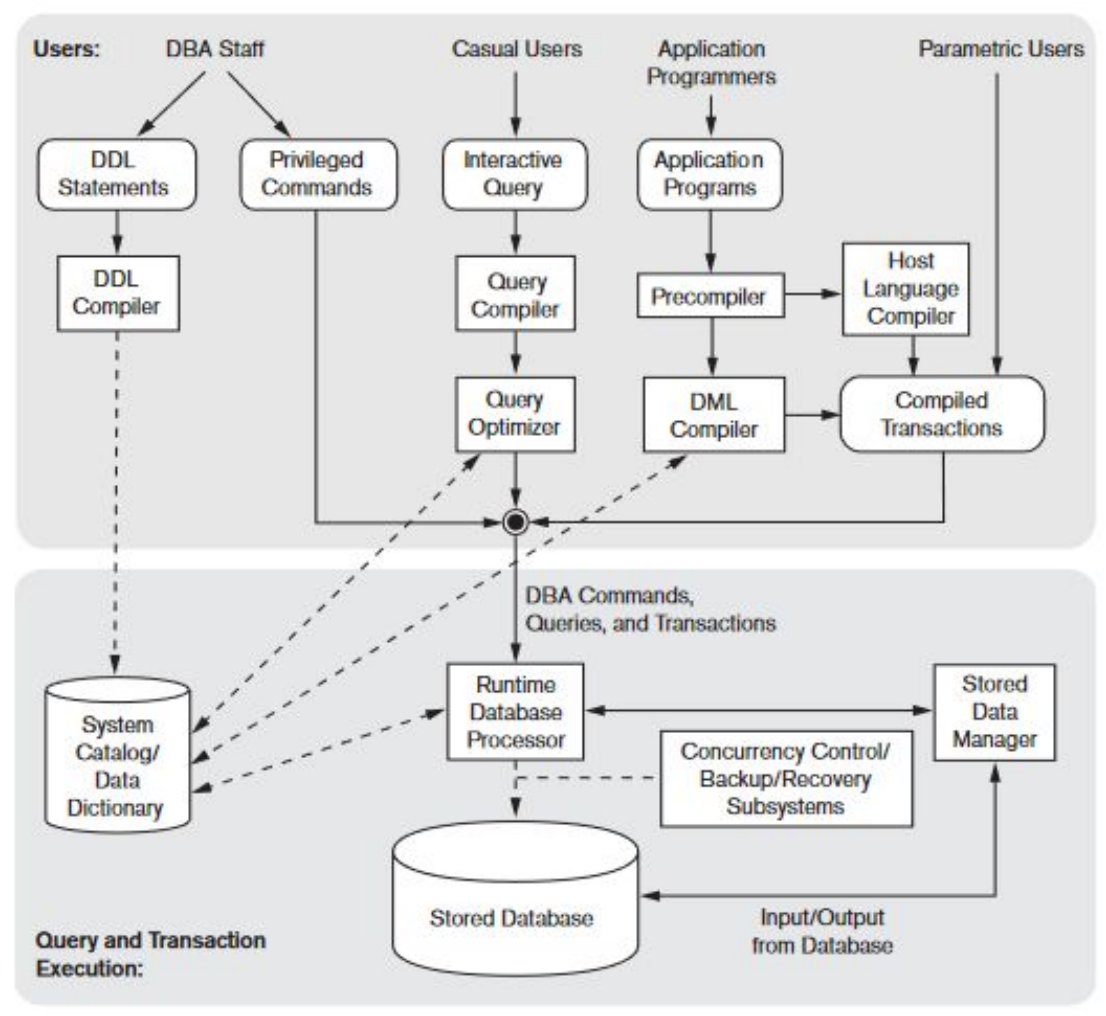
Physical data independence:

- The capacity to change the internal schema without having to change the conceptual schema. Hence, the external schemas need not be changed as well.
- Changes to the internal schema may be needed:

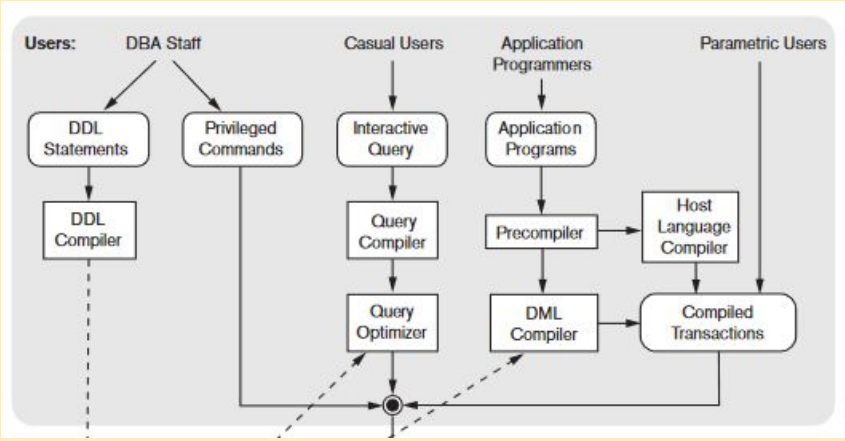
• because some physical files were reorganized—for example, by creating additional access structures—to improve the performance of retrieval or update.

- If the same data as before remains in the database, we should not have to change the conceptual schema.

The Database System Environment

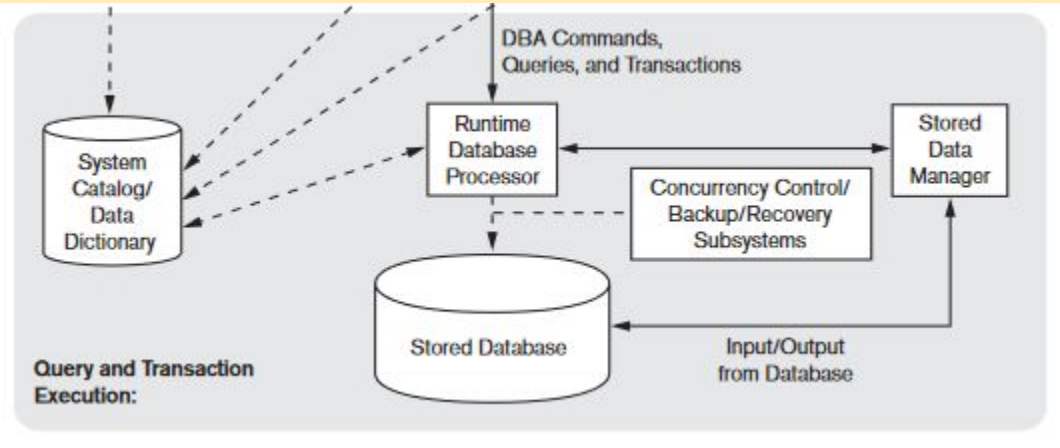


- A DBMS is a complex software system.
- The types of software components that constitute a DBMS and the types of computer system software with which the DBMS interacts.
- The figure is divided into two parts.
  - **The top part** - refers to the various users of the database environment and their interfaces.
  - **The lower part** – shows the internals of the DBMS responsible for storage of data and processing of transactions.



- The DBA staff works on defining the database and tuning it by making changes to its definition using the DDL and other privileged commands.
- The DDL compiler : processes schema definitions, specified in the DDL, and stores descriptions of the schemas (meta-data) in the DBMS catalog.
- Query Compiler :
- Casual users and persons with occasional need for information from the database interact using some form of interface, which we call the interactive query interface (used to generate the interactive query automatically)

- These queries are parsed and validated for correctness of the query syntax, by a query compiler that compiles them into an internal form. This internal query is subjected to query optimization.
- Pre-Complier:
- Application programmers write programs in host languages such as Java, C, or C++that are submitted to a precompiler. The precompiler extracts DML commands from an application program written in a host programming language. These commands are sent to the DML compiler for compilation into object code for database access. The rest of the program is sent to the host language compiler.
- The object codes for the DML commands and the rest of the program are linked, forming a canned transaction.
- Canned transactions are executed repeatedly by parametric users, who simply supply the parameters to the transactions. Each execution is considered to be a separate transaction.
- An example is a bank withdrawal transaction where the account number and the amount may be supplied as parameters



The runtime database processor executes:

- the privileged commands, the executable query plans, and the canned transactions with runtime parameters.
- It works with the system catalog and may update it with statistics.
- It also works with the stored data manager, which in turn uses basic operating system services for carrying out low-level input/output (read/write) operations between the disk and main memory.