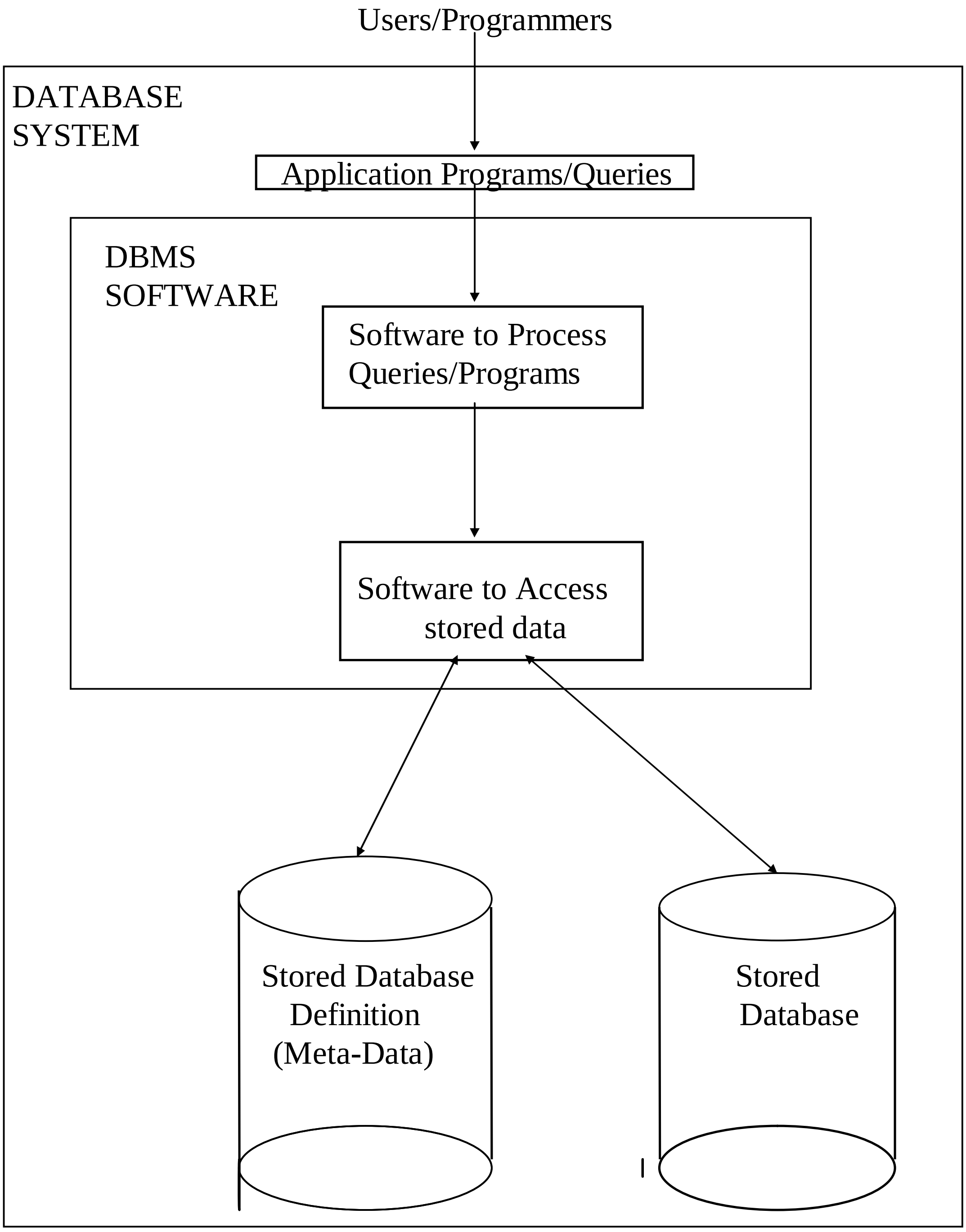
Features of DBMS

Security : Not every user should have access to all the data. For example, if personnel records are kept, only key personnel with the right and need to know salaries should be able to access this data.

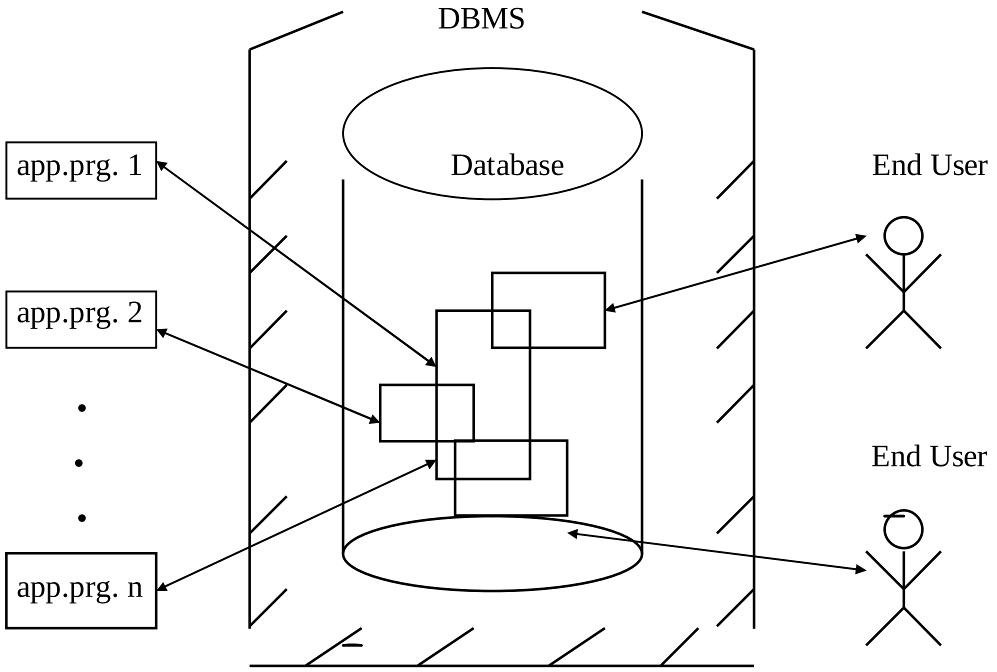
Integrity : Certain kinds of consistency constraints can be checked by the DBMS if it is told to do so. It is useful to have such checks made whenever a user gives a command in the data manipulation language to insert, delete or change some data.

Synchronization : Often many users are running programs that access the database at the same time. The DBMS should provide protection against inconsistencies that result from two approximately simultaneous operations on a data item.

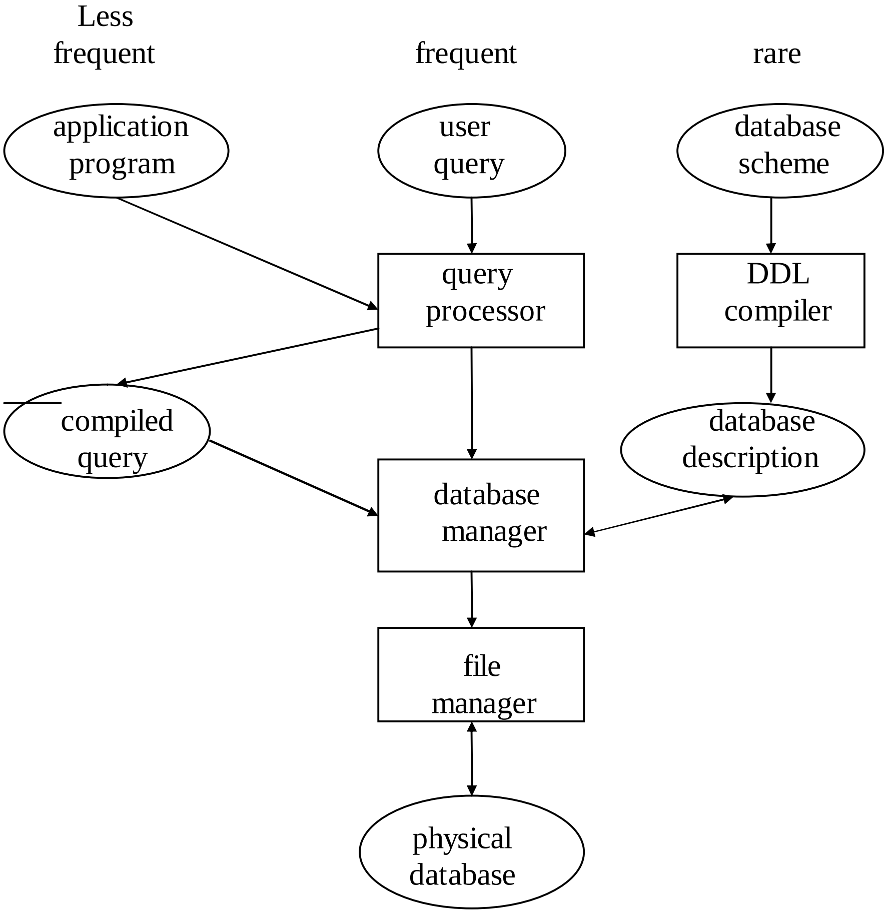
Other than these features, there are many features namely ‘data independence’, support to ‘query language’ etc. which will be discussed later.


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

A simplified database system environment



Simplified Picture of a database System



Schematic diagram of a database system

The three-schema architecture

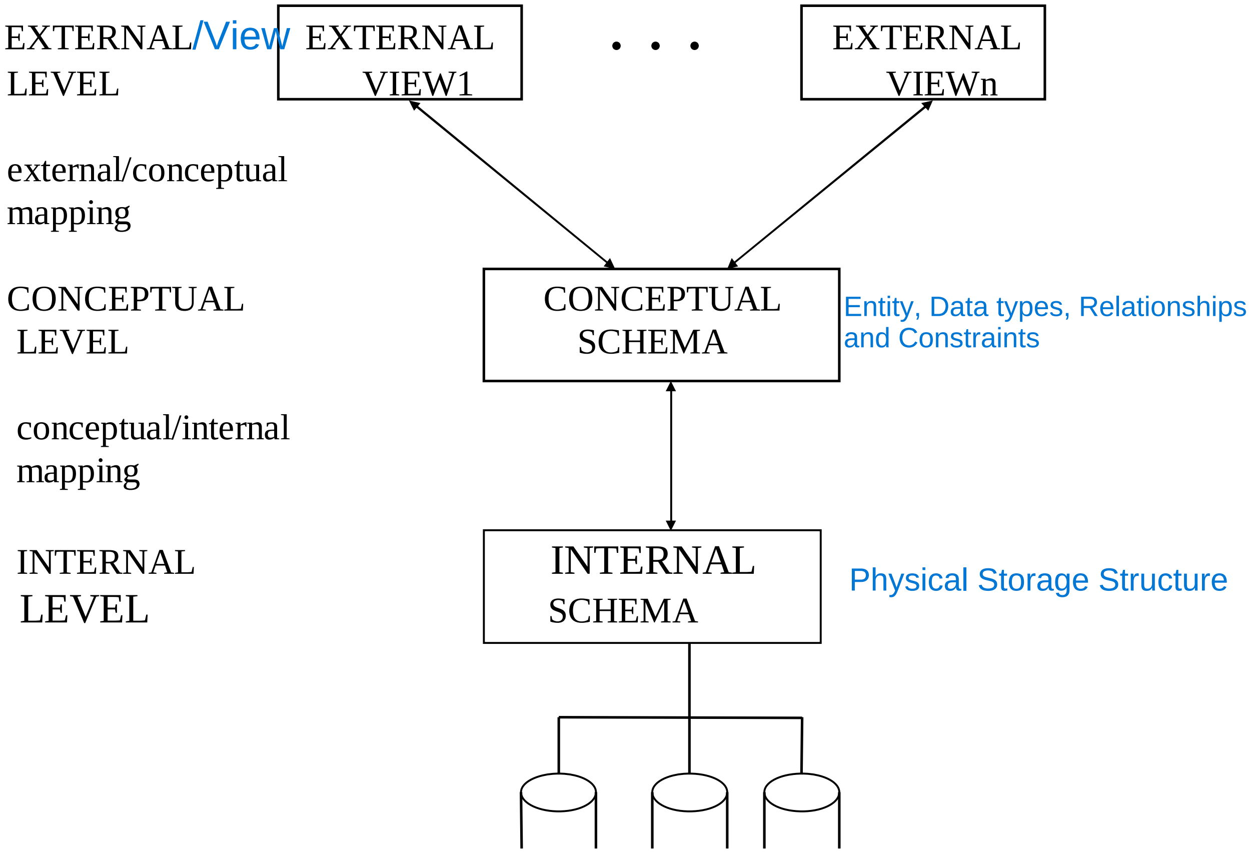
The goal of the three-schema architecture, illustrated in the following figure, is to separate the user application and the physical database. In this architecture, schema can be defined at the following three levels :

1. The internal level has an internal schema which describes the physical storage structure of the database. The internal schema uses a physical data model and describes the complete details of data storage and access paths for the database.

2. The conceptual level has a conceptual schema which describes the structure of the whole database for a community of users. The conceptual schema hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations and constraints.

3. The external or view level includes a number of external schema or user views. Each external schema describes the part of the database that a particular user group is

interested in and hides the rest of the database from the user group.


EXTERNAL/View LEVEL
EXTERNAL VIEW1
. . .
EXTERNAL VIEWn
external/conceptual mapping
CONCEPTUAL LEVEL
CONCEPTUAL SCHEMA
Entity, Data types, Relationships and Constraints
conceptual/internal mapping
INTERNAL LEVEL
INTERNAL SCHEMA
Physical Storage Structure

STORED DATABASE

The three-schema architecture

Most DBMSs do not separate the three levels completely, but several of them support the three-schema architecture to some extent.

Data Independence

The three-schema architecture can be used to explain the concept of data independence which can be defined as the capacity to change the schema at one level of a database system without having to change the schema at the next higher level. There are two types of data independence as follows :

Logical data independence is the capacity to change the conceptual schema without having to change external schema or application programs. We may change the conceptual schema to expand the database or to reduce the database. Only the view definition and the mappings need to be changed in a DBMS that supports logical data independence.

Application programs that reference the external schema constructs must work as before after the conceptual schema undergoes a logical reorganisation. Changes to constraints can also be applied to the conceptual schema without affecting the external schema.

Physical data independence is the capacity to change the internal schema without having to change the conceptual or external schema. Changes to the internal schema may be needed because some physical files had to be reorganised. Because physical data independence refers to the insulation of an application from the physical storage structures only, it is easier to achieve than logical data independence.

Users

There are three types of users. They are as follows :

1. End User
2. Application Programmer
3. Database Administrator

End User : An end user may interact with external/view level. A reservation clerk may be a typical example of an end user.

Application Programmer : Responsible for writing application programs that use the database interacting with the conceptual level.

Database Administrator : Responsible for any system dependent policy decision.