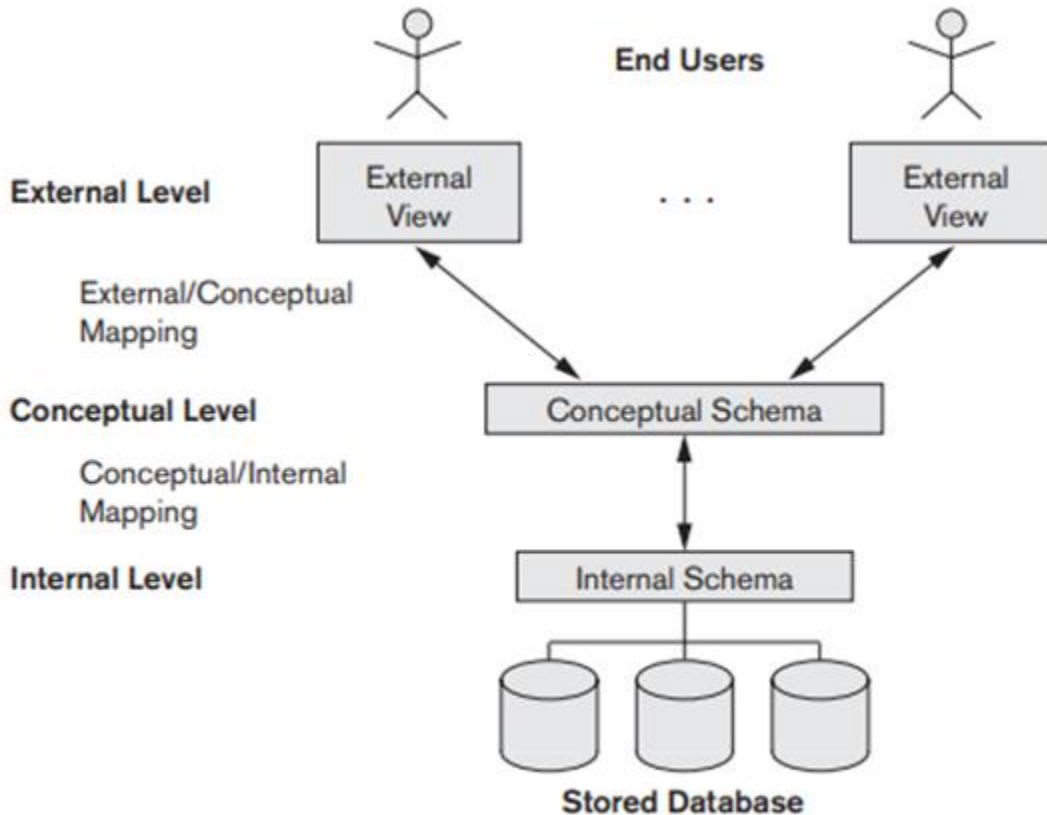


## Three Schema Architecture



Database approach had the following characteristics

- (1) self-describing nature of database
- (2) insulation of programs and data
- (3) support of multiple user views.

To help achieve and visualize these characteristics Three schema architecture was proposed. To separate the user applications from the physical database, schemas 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. Usually, a representational data model is used to describe the conceptual schema when a database system is implemented.
3. The **external or view level** includes a number of external schemas 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 that user group.

In 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.

## **Data Independence**

It is 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:

- a) **Logical data independence** is the capacity to change the conceptual schema without having to change external schemas or application programs. 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). External schemas that refer only to the remaining data should not be affected. Only the view definition and the mappings need to be changed in a DBMS that supports logical data independence.

b) **Physical data independence** is 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— by creating additional access structures—to improve the performance of retrieval or update.

<b>Logical Data independence</b>	<b>Physical Data independence</b>
is the capacity to change the conceptual schema without having to change external schemas or application programs.	is the capacity to change the internal schema without having to change the conceptual schema.
it is difficult to achieve logical data independence.	it is easier to achieve logical 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).	Changes to the internal schema may be needed because some physical files were reorganized— by creating additional access structures—to improve the performance of retrieval or update.