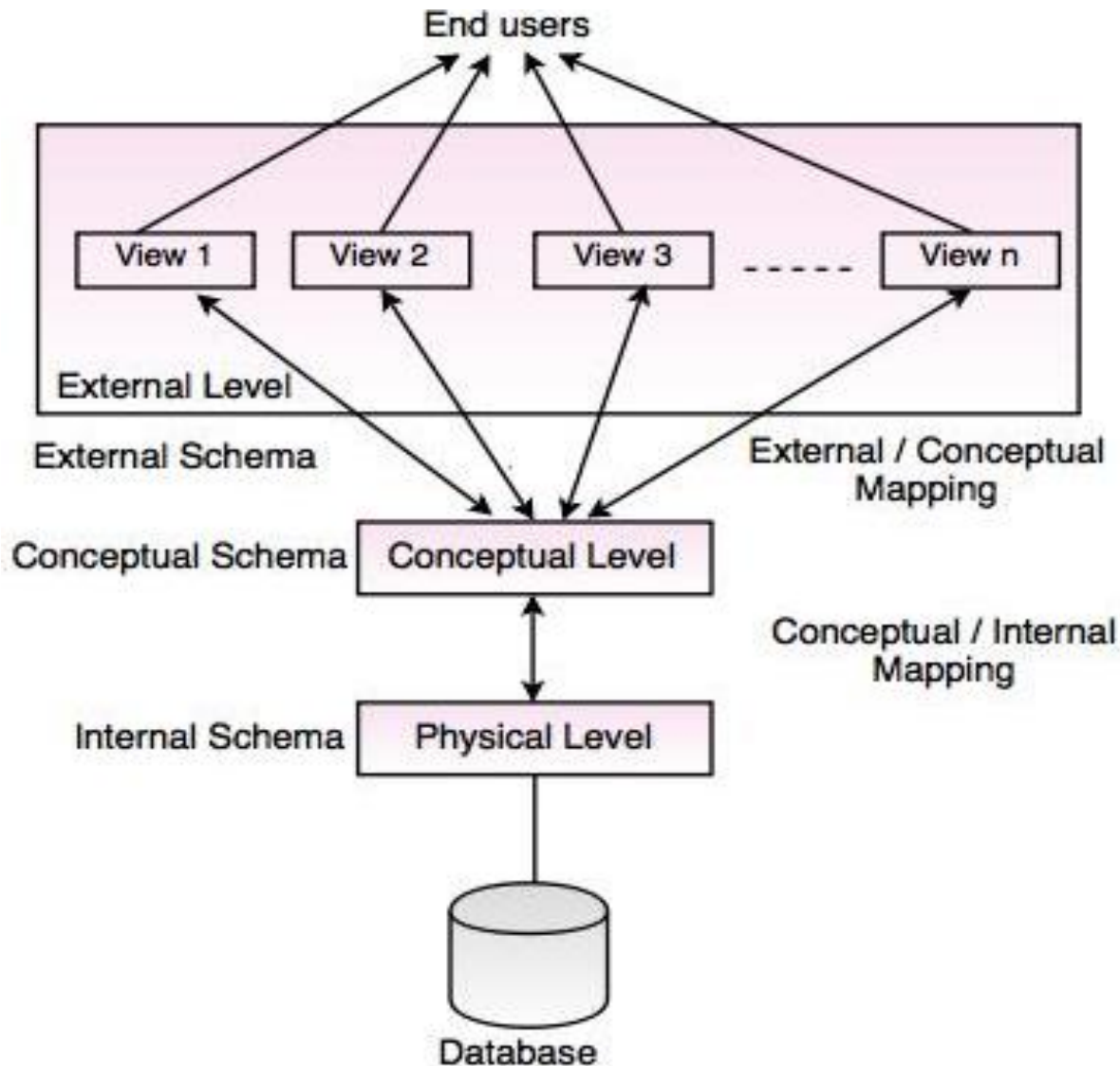




Three Tier Architecture of DBMS

Three-Level Architecture of DBMS



Physical Level

- Describes the physical storage structure of data in database.
- Also known as **Internal Level**
- At this level data is stored in the form of bits with the physical addresses on the secondary storage device.
- Describes details of how data is stored on disk
 - files, indices, etc.
- Describes the record layout of files and type of files
- Early applications worked at this level



Conceptual Level

- Describes the structure of the whole database
- A representation of the entire content of the database
- It is also called as the data model.
- It hides the internal details of physical storage.



External Level

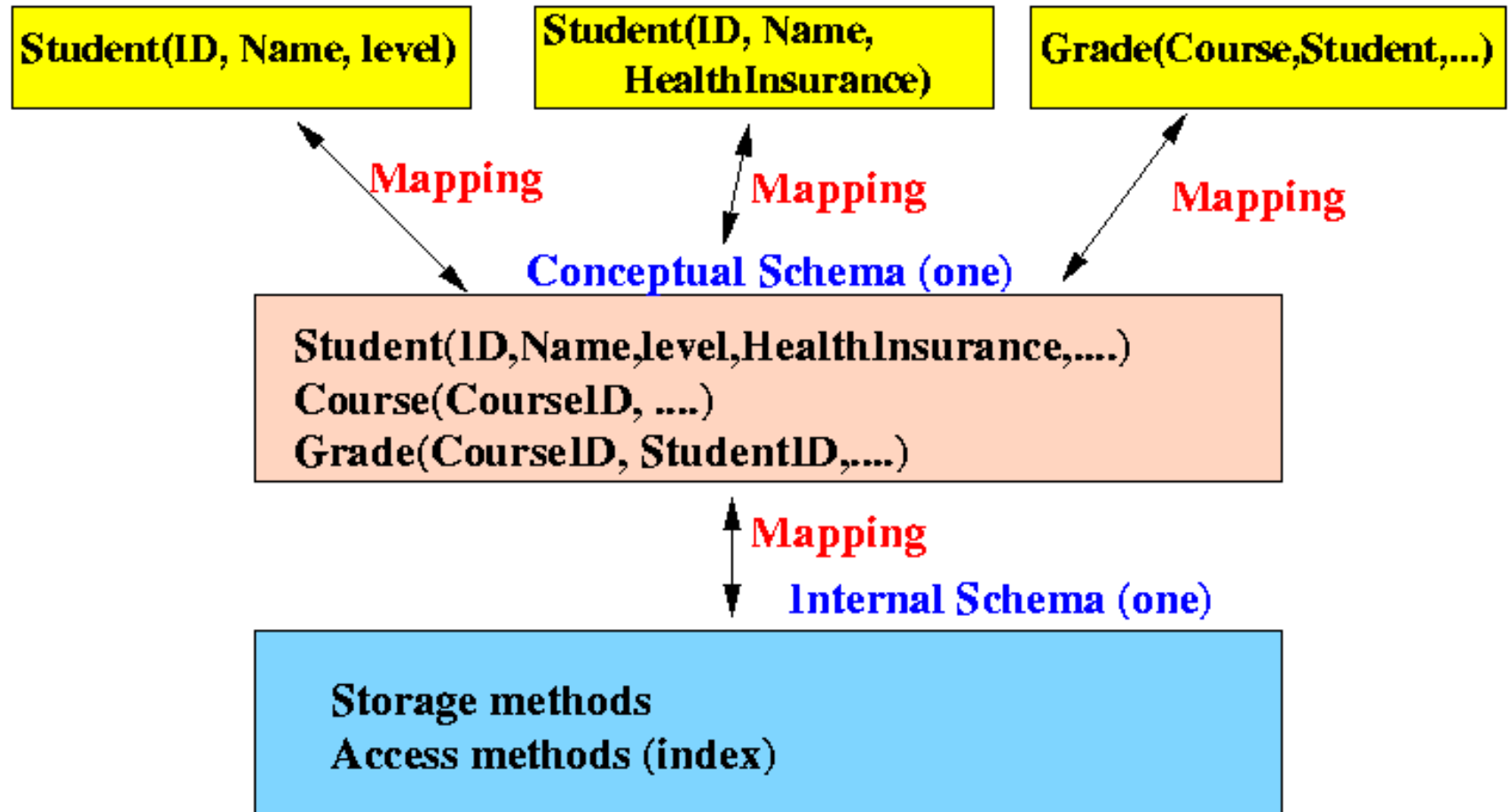
- Related to the data as is viewed by individual end users.
- This level is closest to the user.
- Describes the part of the database that is required for a particular user and hides the rest of the database

Mapping

- Process of transforming request and results between three level is called mapping
- Conceptual/Internal Mapping:
 - Relates conceptual schema with internal schema
 - If change is made to the storage structure definition-then the conceptual/internal mapping must be changed accordingly
- External/Conceptual Mapping:
 - Relates each external schema with conceptual schema
 - Example: several conceptual fields can be combined into a single external field
 - There could be several mappings between external and conceptual levels

Mapping 2

External Schemas

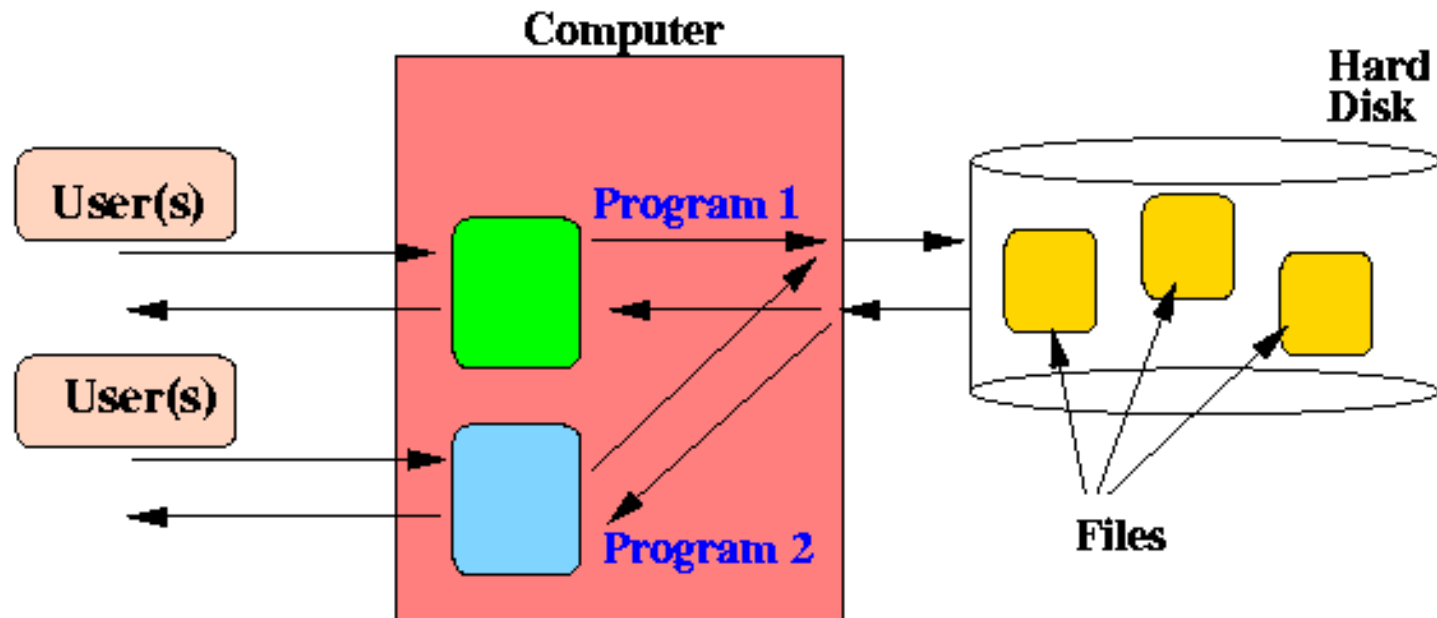


Physical Data Independence

- The ability to change the physical schema without changing the logical schema
- Physical data independence deals with hiding the details of the storage structure from user applications.
- Changing the internal schema should be possible without having to change the conceptual or external schemas.
- For example
 - Changing the file organization or storage structures
 - Changing storage devices
 - Format of Data file can change
 - Using different indexing strategy

Logical Data Independence

- Ability to present information in different ways to different users
- Different users have different requirements



Logical Data Independence 2

- Conceptual Schema
 - Store data in consolidated form
 - One schema for each database application
- Each user will have its own external schema
- The ability to change the logical (conceptual) schema without changing the External schema (User View)
- The way you view the data is **INDEPENDENT** from the way you store the data
- Example:
 - the addition or removal of new entities, attributes, or relationships to the conceptual schema will not affect existing application programs

Logical Data Independence Eg.

3. This structure can remain UNCHANGED !!!

