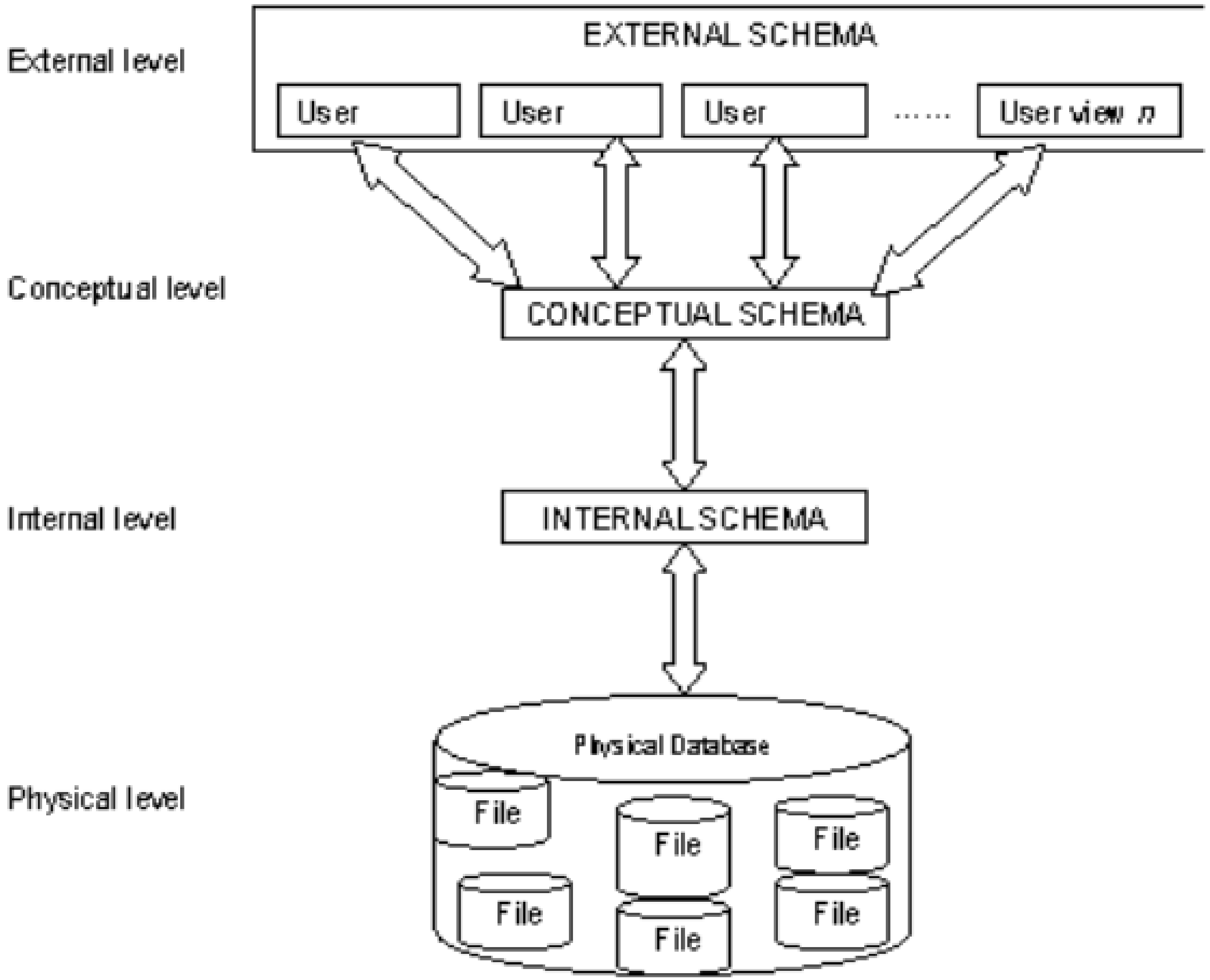
Relational Database Management System (RDBMS)

240110102

Three Level ANSI-SPARC Database Architecture



Internal level or Storage level

- It is the physical representation of the database on the computer
- This level describes how the data is stored in the database
- It covers the data structures and file organizations used to store data on storage devices.
- It interfaces with the operating system access methods to place the data on the storage devices, build the indexes, retrieve the data and so on.
- Defined by the DBA
- The internal level is concerned with such things as:
  - Storage space allocation for data and indexes;
  - Record descriptions for storage (with stored sizes for data items)
  - Data compression and data encryption techniques

Conceptual Level or Logical level

- This level describes what data is stored in the database and the relationships among the data.
- This middle level in the three-tier architecture is the conceptual level which contains the logical structure of the entire database as seen by the DBA.
- It is a complete view of the data requirements of the organization that is independent of any storage considerations.
- The conceptual level represents:
  - All entities, their attributes, and their relationships;
  - Constraint on the data
  - Semantic information about the data
  - Checks to retain data consistency and integrity
  - Security information
- For instance, the description of an entity should contain only data types of attributes (for example, integer, real, character) and their length (such as the maximum number of digits or characters), but not any storage considerations, such as the number of bytes occupied

External Level or View level

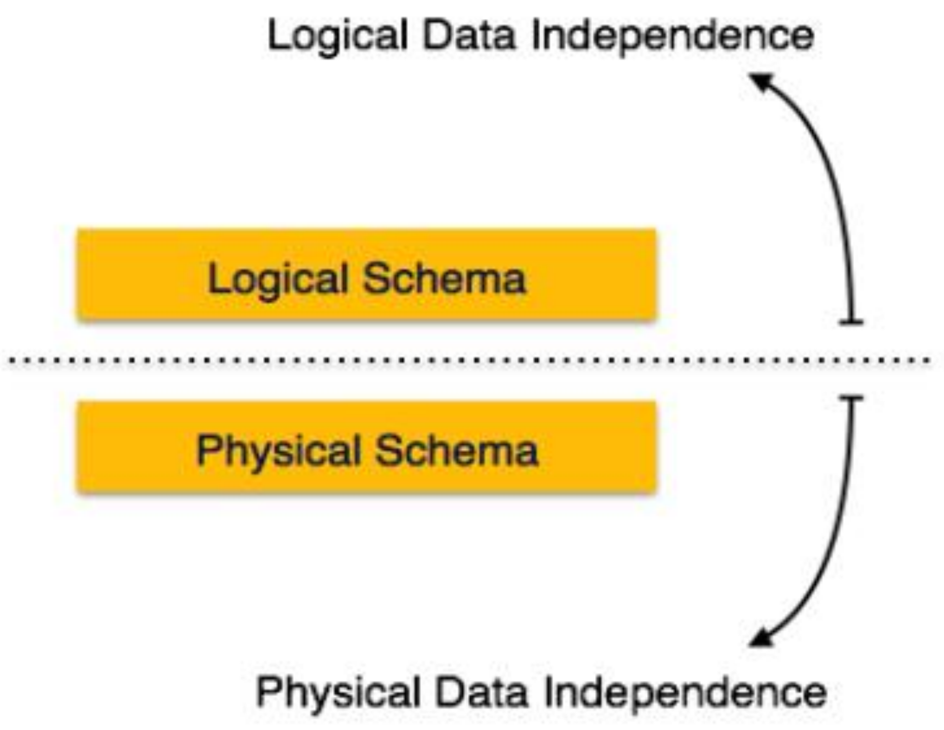
- It is the users' view of the database. This level describes that part of the database that is relevant to each user.
- External level is the one which is closest to the end users. .
- This level deals with the way in which individual users view data. Individual users are given different views according to the user's requirement.
- A view involves only those portions of a database which are of concern to a user. So, the same database can have different views for different users
- Example:
  - one user may view dates in the form (day, month, year),
  - while another may view dates as (year, month, day).

Advantages of Using Three-Tier Architecture

- It makes the logical separation between business layer and presentation layer and database layer.
- Migration to new graphical environments is faster.
- As each tier is independent it is possible to enable parallel development of each tier by using different sets of developers.
- Easy to maintain and understand large project and complex project.
- Since application layer is between the database layer and presentation layer so the database layer will be more secured and client will not have direct access to the database.
- Posted data from presentation layer can be verified or validated at application layer before updating it to the database.
- Database Security can be provided at application layer.
- Application layer or middle layer or business layer can be a protection shield to the database.
- New rules or new validation rules can be defined any time and changes made to middle layer will not affect presentation layer.
- We can hide unnecessary methods from business layer in the presentation layer.
- Easy to apply object oriented concept
- Easy to update data provider queries.

Data Independence

- Denotes the property that higher levels of abstraction and are not influenced by changes in the lower levels.
- The ability to change the schema at one level of a database system without affecting a schema definition in the next higher level is called data independence.
- There are three levels of abstractions so there are two types of data independence.
- There are two kinds of data independence:
  - Logical data independence
  - Physical data independence



Logical data independence

- 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), or to reduce the database (by removing a record type or data item) – here external schemas that refer only to the remaining data should not be affected
- Changes to constraints can be applied also to the conceptual schema without affecting the external schemas or application programs

Physical Data Independence

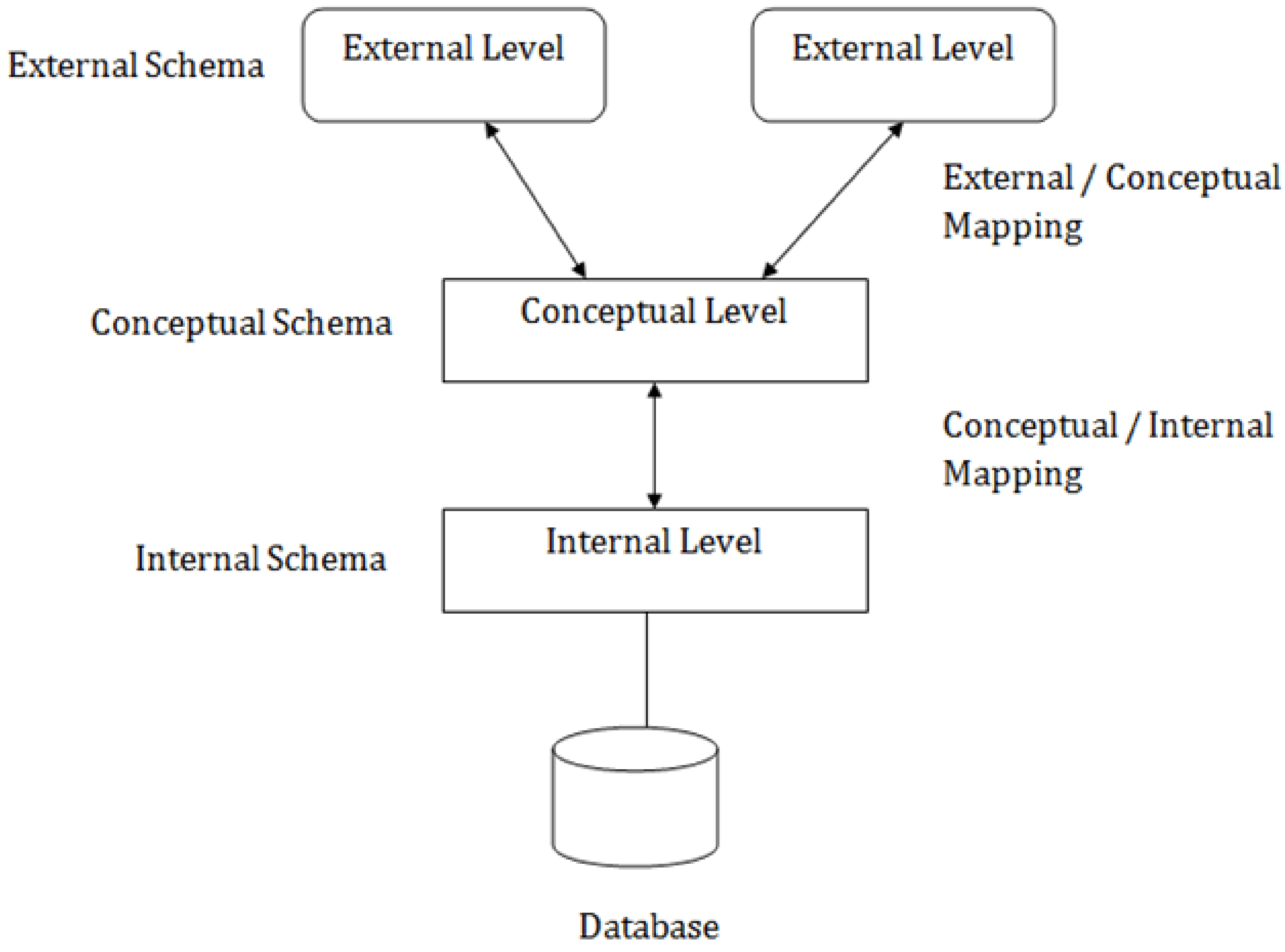
- Physical data independence is the capacity to change the internal schema without having to change the conceptual (or external) schemas
- For example, by creating additional access structures—to improve the performance of retrieval or update. If the data remains same in the database as before, we should not have to change the conceptual schema.

Data Abstraction

- The system hides certain details of how the data are stored and maintained is called data abstraction
- Since database system users are not computer trained, developers hide the complexity from users through 3 levels of abstraction, to simplify user’s interaction with the system.

1. Physical level of data abstraction: The lowest level of abstraction which describes *how data are actually stored*.
2. Logical level of data abstraction: This level hides *what data are actually stored* in the database and what relationship exists among them.
3. View Level of data abstraction: View provides security mechanism to *prevent user from accessing certain parts of database*.

Mapping


External Schema
External Level
External Level
External / Conceptual Mapping
Conceptual Schema
Conceptual Level
Conceptual / Internal Mapping
Internal Schema
Internal Level
Database