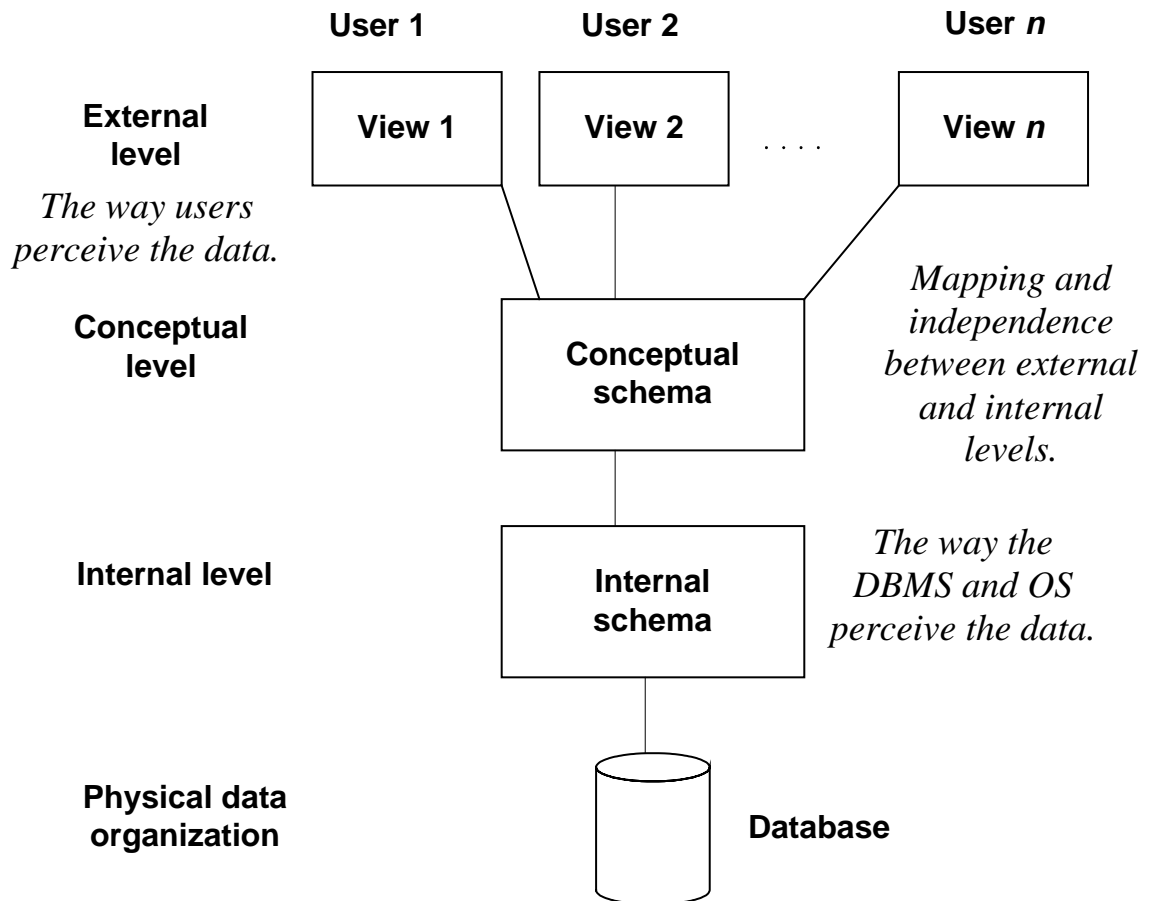


THREE-LEVEL ARCHITECTURE



Objective: separate each user's view of the database from the way it is physically represented.

EXTERNAL LEVEL (highest level)

- The user's view of the database.
- Consists of a number of different external views of the DB.
- Describes part of the DB for particular group of users.
- Provides a powerful and flexible security mechanism by hiding parts of the DB from certain users. The user is not aware of the existence of any attributes that are missing from the view.
- It permits users to access data in a way that is customized to their needs, so that the same data can be seen by different users in different ways, at the same time.

CONCEPTUAL LEVEL

- The logical structure of the entire database as seen by DBA.
- What data is stored in the database.
- The relationships among the data.
- Complete view of the data requirements of the organization, independent of any storage consideration.
- Represents:
 - Entities, attributes, relations
 - constraints on data
 - semantic information on data
 - security, integrity information

Supports each external view: any data available to a user must be contained in, or derivable from the conceptual level.

INTERNAL LEVEL

- Physical representation of the DB on the computer.
- How the data is stored in the database.
- Physical implementation of the DB to achieve optimal run–time performance and storage space utilization.
 - Storage space allocation for data and indexes
 - Record description for storage
 - Record placement
 - Data compression, encryption

PHYSICAL LEVEL

Managed by the OS under the direction of the DBMS.

SCHEMAS, MAPPINGS, INSTANCES

DB schema: overall description of the DB.

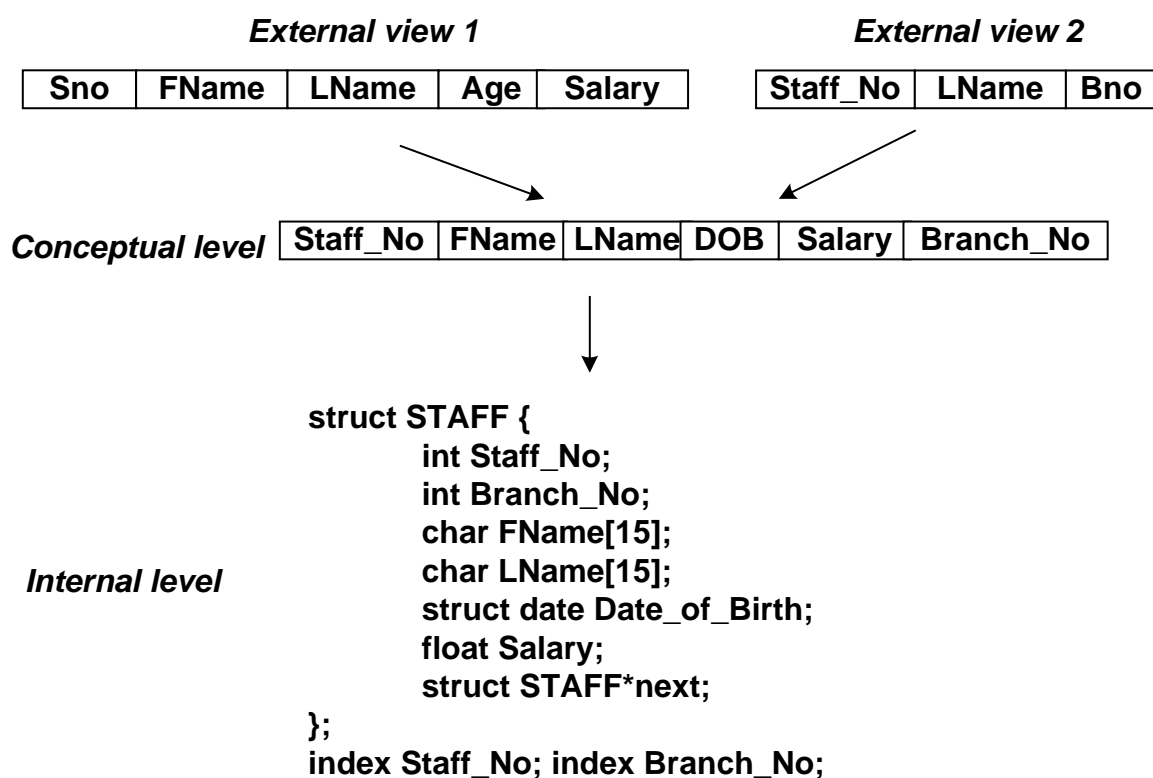
Three different schemas according to the level of abstraction.

DBMS: mapping between schemas

consistency of schemas

conceptual/internal mapping: to find the actual record (combinations) in physical storage that constitute a logical record in the conceptual schema.

external/conceptual mapping: map names in the user's view onto the relevant part of the conceptual schema.



Database instance: the data in the DB at any particular point in time.

DATA INDEPENDENCE

The ability to modify a scheme definition in one level without affecting a scheme definition in a higher level is called **data independence**.

1. There are two kinds:

Logical data independence

- The ability to modify the conceptual scheme without causing application programs to be rewritten.
- Immunity of external schemas to changes in the conceptual schema.
- Usually done when logical structure of database is altered

Physical data independence

- The ability to modify the internal scheme without having to change the conceptual or external schemas.
- Modifications at this level are usually to improve performance.

DATA MODELS

Data models are a collection of conceptual tools for describing data, data relationships, data semantics and data constraints.

Components: structural part
 manipulative part
 integrity rules

There are three different groups:

Object-based Data Models	}	Describe data at the conceptual and external levels
Record-based Data Models		
Physical Data Models	→	Describe data at the internal level

Object-based Data Models

- Entity-relationship model.
- Object-oriented model.
- Semantic data model.
- Functional data model

Record-based Data Models

- Named so because the database is structured in fixed-format records of several types.
- Each record type defines a fixed number of fields, or attributes.
- Each field is usually of a fixed length (this simplifies the implementation).
- The three most widely accepted models are the **relational, network, and hierarchical data model**.

Physical Data Models

1. Are used to describe data at the lowest level.
2. Very few models, e.g.
 - Unifying model.
 - Frame memory.