Date .....

Chapter - 2
Database System Concepts and Architecture:

- In a basic client / server DBMS architecture, the system functionality is distributed between 2 types of modules.
  → Client module
  It is designed so that it will run on user workstation or personal computer.
  → Server module
  It typically, handles data storage, access, search and other functions.

* Data Abstraction
Data abstraction is referred to the suppression of details of data organization and storage and the highlighting of the essential features for an improved understanding of data.

* Data Model
- A data-model is a collection of concepts that can be used to describe the structure of a database — provides the necessary means to achieve this abstraction.

* Data Model Operations
Operations for specifying database retrievals and updates by referring to the concepts of the data model. Operations on the data model may include basic operations and user-defined operations.

Date .....

* Dynamic aspect or behavior of database application.
  - Allows the database designer to specify a set of valid operations allowed on the database objects
  - Example : COMPUTE_GPA, which can be applied to a STUDENT object.
  - Generic operations to insert, delete, modify or retrieve any kind of object are often included in the basic data model operations.

* Categories of Data Model
  - On the basis of types of concepts used to describe the database structure. The categories are:

1. High-level or conceptual data model
2. Low-level or physical data model
3. Representational or implementational data model

1. High-level or conceptual data model
   - provides concepts that are close to the way many users perceive data.
   - uses concepts in modeling data.
   - It uses concepts such as Entities, attributes and relationships.
   - Entity-Relationship model — a popular high-level conceptual data model.

2. Low-level or physical data model.
   - provides concepts that describe the details of how data is stored on the computer storage media, typically magnetic disk.
   - meant for computer specialists, not for end users.

Date .....

3. Representational or implementational data model
- Easily understood by end users.
- Also similar to how data is organized in computer storage.
- Hides many details of data storage on disk but can be implemented on a computer system directly.
- It includes relational, network and hierarchical data models.
- It represents data using record structures and hence sometimes called record-based data models.

Object data model
- An example of new family of higher-level implementation data models that are closer to conceptual data model.
- A standard for object databases called the ODMG object model has been proposed by the Object Data Management Group. (ODMG)

* Schema, Instances, and Database State

Database Schema - It is the description of a database. Includes descriptions of the database structure and the constraints that should hold on the database

Schema diagram - A diagrammatic display of a database schema.

Date .....

Schema Construct - A component of the schema or an object within the schema. Eg: student, society.

Database Instance - The actual data stored in a database at a particular moment in time. Also called database state (or occurence).

Meta-data - Description of schema constructs and constraints stored in file called catelogue.

Database State - Refers to the content of a database at a moment of time.

Initial database State - Refers to the database when it is loaded.

Valid State - A state that satisfies the structure and constraints of the database.

- The database schema changes very infrequently. The database state changes every time the database is updated. Known as Schema Evolution. Schema is called intension and state is also called extension.

* Three-Schema Architecture.

This supports 3 main characteristics of DBMS

1. Use of a catalog to store the database description so as to make it self-describing.
2. insulation of programs and data (Program-data indepence).
3. Support of multiple user views.

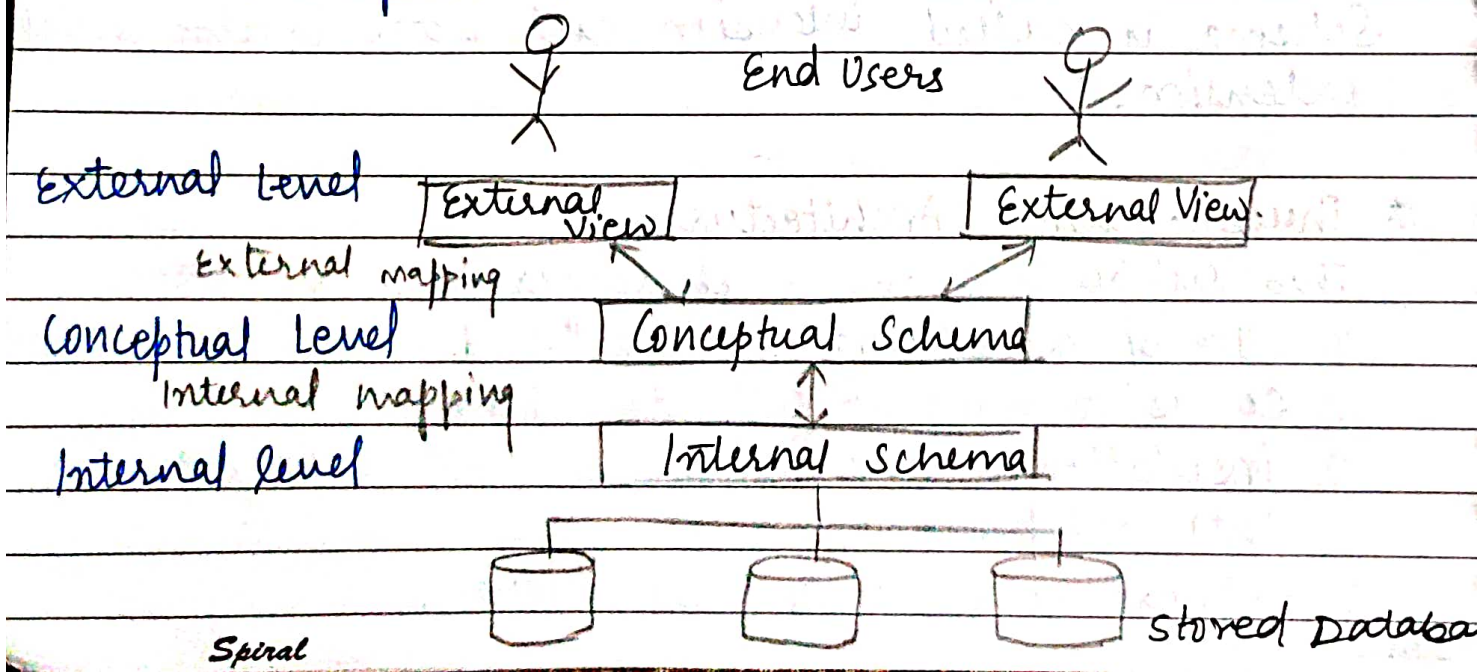
Date .....

- The goal of the 3-schema architecture is to separate the user application from the physical database.
- Defines DBMS & schema at 3 levels:

1. Internal Schema.
   At the internal level.
   Describes the physical storage structures and access paths. Uses physical data model.

2. Conceptual Schema.
   At the conceptual level.
   Describes the structure and constraints for the whole database for a community of users.
   Uses a conceptual or implementational model

3. External Schema.
   At the external level.
   Describes the various user views.
   Usually uses the same data model as the conceptual level.

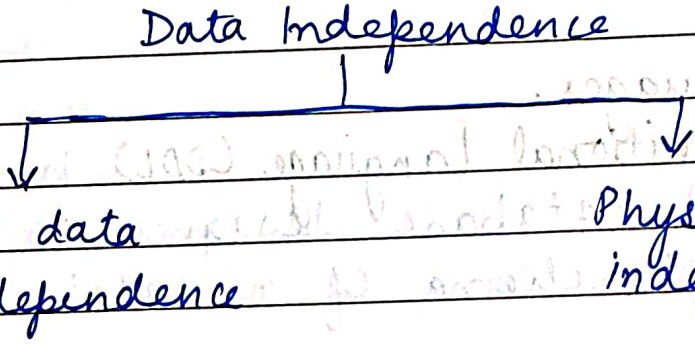


Date .....

- The processes of transforming requests and results between levels are called Mappings.
- Mappings among schema levels are needed to transform requests and data.
- Programs refer to an external schema, and are mapped by the DBMS to the internal schema for execution.

* Data Independence.

- 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



Logical Data Independence

The capacity to change the conceptual schema without having to change the external schema and their application programs.

Physical Data Independence

The capacity to change the internal schema without having to change the conceptual schema. Hence, the external schemas need not to be changed as well.

- Physical data independence exists in most databases and file environments where

Date .....

physical details such as exact location are hidden from the user.

- Logical data independence is harder to achieve because it allows structural and constraint changes without affecting application programs — a much stricter requirement.

- When a schema at a lower level is changed only the mappings between this schema and higher level schemas need to be changed in a DBMS that fully supports data independence.
- No change in application programs since they refer to the external schemas.

* DBMS Languages.
  - Data Definitional Language (DDL) is used by the DBA and database designers to specify the conceptual schema of a database, ~~where there is no~~
  - In some DBMSs, separate storage definition language (SDL) and view definition language (VDL) are used to define internal and external schemas.
  - Data Manipulation Language (DML) used to specify database retrievals and updates.
  - DML Commands (sublanguage) can be embedded in a general purpose programming language (host language) such as COBOL, C.
  - Alternatively, stand alone DML commands can be applied directly (query language).

Date .....

- High-level or Non-Procedural Languages like SQL are set-oriented and specify what data to retrieve than how to retrieve. Also called declarative languages.

- Low level or Procedural languages: record at a time. They specify how to retrieve data and include constructs such as looping.

* DBMS Interfaces

- User friendly interfaces provided by a DBMS includes:
  - → Menu based, popular for browsing on the web.
  - → Forms based, designed for naive users.
  - → Graphics-based.
  - → Natural language.
  - → Speech Input and Output.
- Programmer interfaces for embedding DML in programming languages:
  - → Pre-compiler Approach.
  - → Procedure (subroutine) Call Approach.

* Database System Utilities.

DBMS have database utilities that help the DBA manage the database systems. Common utilities have following functions:

- → Loading data stored in files into a database. Includes data conversion tools.
- → Backing up the database periodically on tape
- → Reorganizing database file structures.
- → Report generation utilities.
- → Performance monitoring utilities.

Date .....

* Data dictionary / repository.
  - Used to store schema descriptions and other information such as design decisions, application program descriptions, user information, usage standards, etc.
  - Active data dictionary is accessed by DBMS software and users / DBA.
  - Passive data dictionary is accessed by users / DBA only for performance evaluation.

* Application Development Environments and CASE (Computer-aided software engineering) tools:
  Examples: Power builder (Sybase), Builder (Borland) Visual Studio (Mysql, oracle, etc.)

* Examples of DBMS

  Single User: Typically used with micro-computers MS Access, Mysql.

  Multi User: Most DBMS's.
  Oracle, Ingres, Mongodb.
  Nosql - non-relational, distributed, open-source and horizontally scalable.