Name Of Subject : Database Management System

Unit-1:-Introductory Concepts of DBMS

Ravirajsinh Chauhan
Assistant Professor
School of Engineering, P P Savani University

Introduction

- **DATA-** those could be recorded & stored on computer media.
- **Information**- Data that could be used to increases the knowledge of the user.
- **Database-** It is a collection of a data contains information about one particular enterprise
- **DBMS-** It is a collection of interrelated data and a set of program to access those data

DBMS

Introduction

- A database-management system (DBMS) is a collection of interrelated data and a set of programs to access those data.
- Database, contains information relevant to an enterprise
- Goal of a DBMS is to provide a way to store and retrieve database information that is both *convenient* and *efficient*.

Database System Applications

- **Banking**
- **Airlines**
- **Universities**
- **Credit and Transactions**
- **Telecommunications**
- **Finance**
- **Sales**
- **Human Resources**
- **Manufacturing**

Database Systems versus File Systems

- **Data redundancy and inconsistency**
  - Multiple file formats, duplication of information in different files
- **Difficulty in accessing data**
  - Need to write a new program to carry out each new task
- **Data isolation-**
- data are scattered in various files, and files may be in different formats, writing new application programs to retrieve the appropriate data is difficult.
- **Integrity problems-** when new constraints are added, it is difficult to change the programs to enforce them.

Database Systems versus File Systems

- **Atomicity problems**
  - Failures may leave database in an inconsistent state with partial updates carried out
  - Example: Transfer of funds from one account to another should either complete or not happen at all
- **Concurrent-access by multiple users**
  - Example: Two people reading a balance (say 100) and updating it by withdrawing money (say 50 each) at the same time
- Security problems
- Poor data control
- Limited data sharing
- Excessive programming effort

Application of DBMS

- Providing Application Flexibility with Relational Databases.
- Object oriented Applications and the need for more complex Databases.
- Early DB Applications
- Extending DB capabilities for new applications

Advantages of DBMS

- Minimal data redundancy
- Program data independence
- Efficient data access
- Improved data sharing
- Improved security
- Economy of scale
- Reduced program maintenance
- Improved Backup
- Improved data quality

Purpose of DBMS

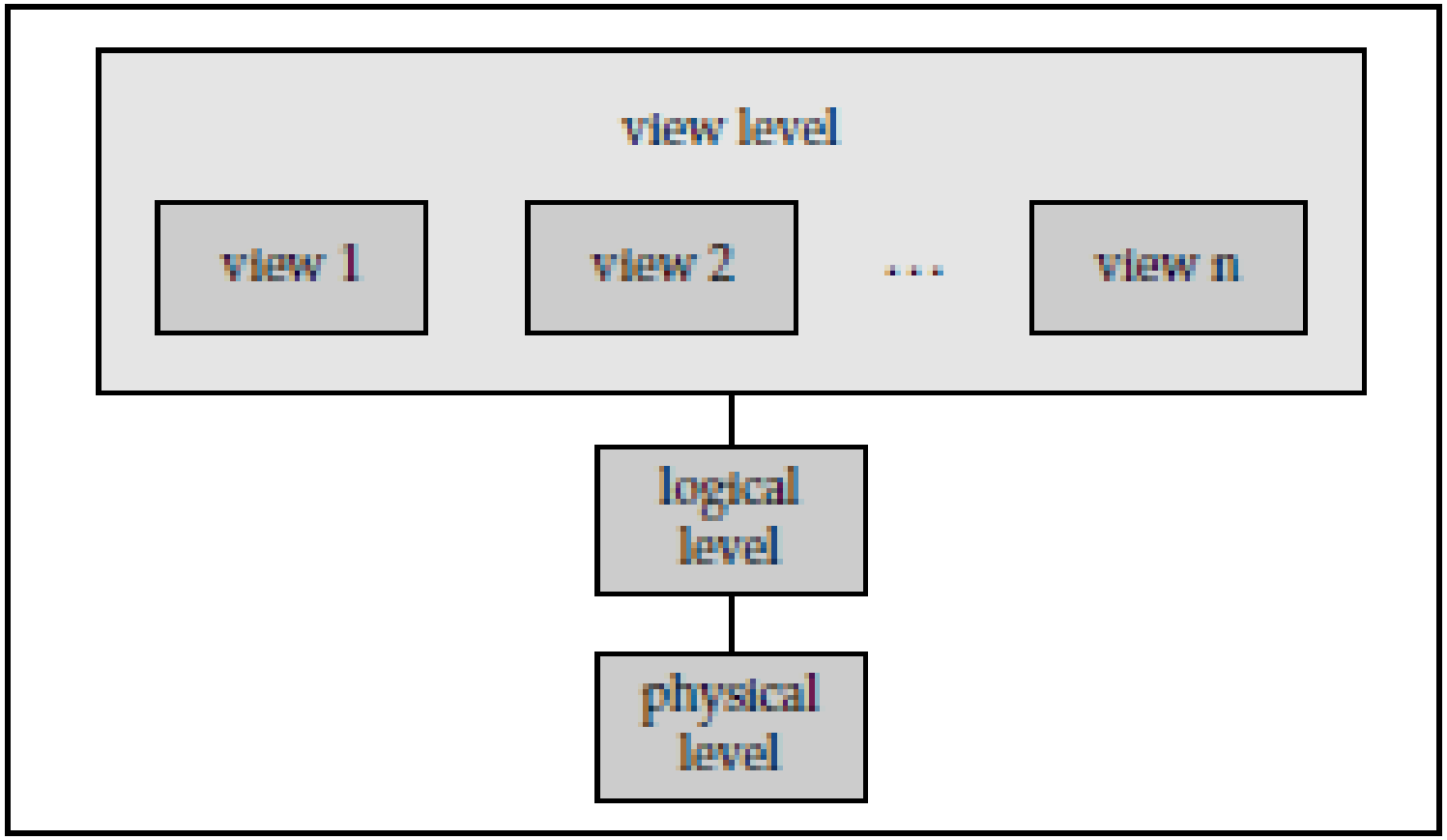
- Compactness- no need of paper work
- Speed
- Accuracy
- Protection

Benefits of DB Approach

- Data can be Shared
- Redundancy can be reduced
- Inconsistency can be avoided
- Security can be enforced
- Conflicting requirements can be balanced
- Integrity can be maintained

Database System Architecture Levels (3-tier Architecture)

- Physical level (Internal Level)
- Logical level (Conceptual Level)
- View Level (External Level)



Physical Level(Internal Level)

- Internal Level (Storage Level)- concerned with the way the data is stored inside the system

- This schema pertains to the actual storage of data and its form of storage like files, indices, etc. It defines how the data will be stored in a secondary storage.

- the database system hides lowest-level storage details from database programmers

- Database administrators, on the other hand, may be aware of certain details of the physical organization of the data.

Logical level

- This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.

- Record is described by a type definition

- database administrators usually work at this level of abstraction.

External Level

- Closest to the Users
- At the external level, several views of the database are defined
- Database users see these views
- This level also provide a security mechanism to prevent users from accessing certain parts of the database.
- Eg. tellers in a bank see only that part of the database that has information on customer accounts; they cannot access information about salaries of employees.

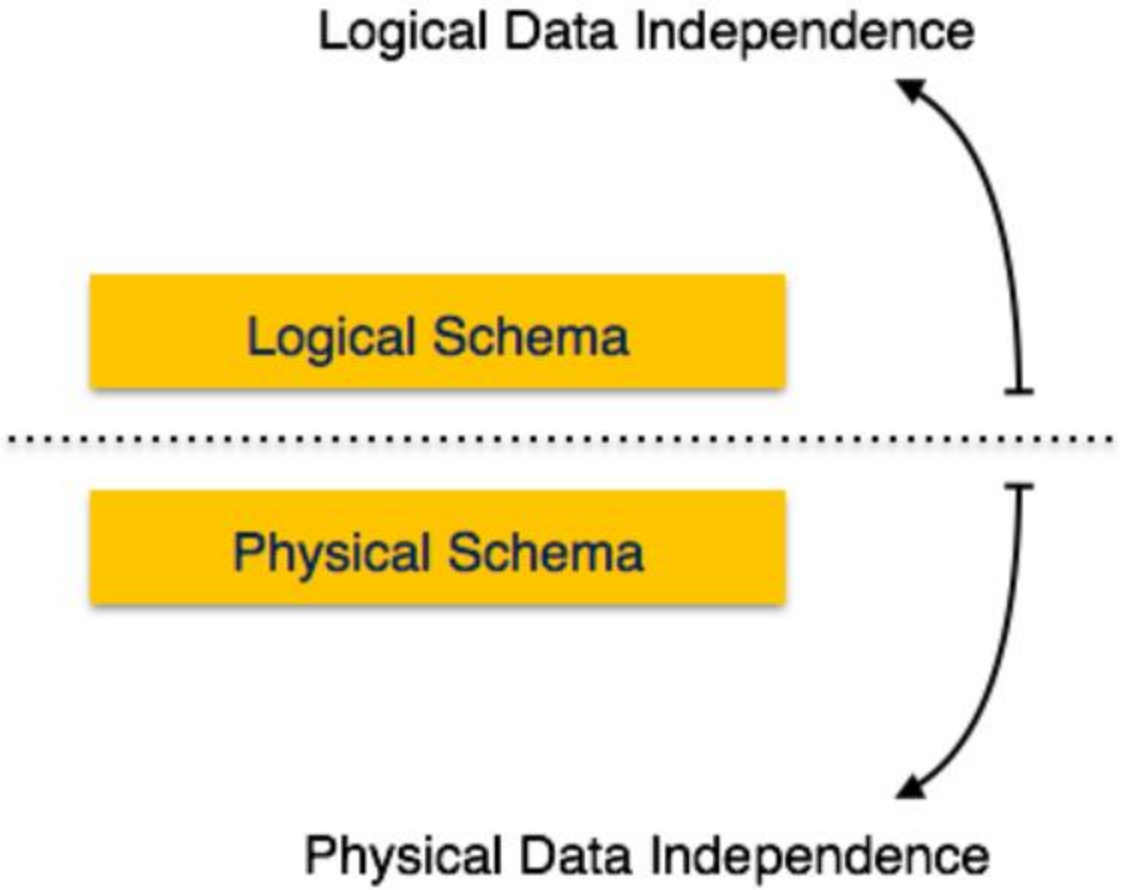
Data Independence

- Ability to modify a schema definition in one level without affecting a schema definition in the next higher level .

- A database system normally contains a lot of data in addition to users' data. For example, it stores data about data, known as metadata, to locate and retrieve data easily.

- It is rather difficult to modify or update a set of metadata once it is stored in the database.
- But as a DBMS expands, it needs to change over time to satisfy the requirements of the users.
- If the entire data is dependent, it would become a tedious and highly complex job.

Data Independence



Physical Data Independence

- Ability to modify the physical scheme without causing application program to be rewritten.

- All the schemas are logical, and the actual data is stored in bit format on the disk. Physical data independence is the power to change the physical data without impacting the schema or logical data.

- For example, in case we want to change or upgrade the storage system itself − suppose we want to replace hard-disks with SSD − it should not have any impact on the logical data or schemas.

- It is generally necessary in order to improve performance

Logical Data Independence

- Ability to modify the conceptual scheme without causing application program to be rewritten.

- Logical data is data about database, that is, it stores information about how data is managed inside. For example, a table (relation) stored in the database and all its constraints, applied on that relation.

- Logical data independence is a kind of mechanism, which liberalizes itself from actual data stored on the disk. If we do some changes on table format, it should not change the data residing on the disk
- It is generally necessary whenever the logical structure of the database is altered.

Database Users

- **Application Programmer**
  - The one who writes application programs that uses the database.
  - They interact with DBMS through DML (Data manipulation language) calls.

- **End Users**

  - **Casual end users-** they may need different information each time

  These users have great knowledge of query language. Casual users access data by entering different queries from the terminal end. They do not write programs but they can interact with the system by writing queries.

  - **Naïve (Parametric) end users-**
    - Bank tellers
    - Reservation clerks

Database Users

- **DBA (Database Administrator)**

DBA can be a single person or it can be a group of person. Database Administrator is responsible for everything that is related to database. He makes the policies, strategies and provides technical supports.

- **System Analyst (Database designer)**

System analyst is responsible for the design, structure and properties of database. All the requirements of the end users are handled by system analyst. Feasibility, economic and technical aspects of DBMS is the main concern of system analyst.

Standalone users – they maintain personal database by using ready made program packages that provide graphical based interface.

DBA

- Data Administrator is the person who make the strategic and policy decision
- Database Administrator is the person who provides the necessary technical support for implementing those decisions.
- Responsible for the overall control of the system at a technical level.

DBA Tasks

- Defining the conceptual schema
- Defining the internal schema
- Liaising with users
- Defining security and integrity constraints
- Defining dump/restore schemes
- Monitoring performance and responding to changing requirements.