Module 1:

Introduction to Database Concepts

Topics:

- Introduction
- Characteristics of Databases
- File System v/s Database System
- Data Abstraction and Data Independence
- Schemas and Instance
- Users of Database System
- Three level Schema Architecture
- Database Architecture

• Introduction

Data

• Data is nothing but facts and statistics stored or free flowing over a network, generally it's raw and unprocessed.

• Data becomes information when it is processed, turning it into something meaningful.

Database:

- The database is a collection of inter-related data which is used to retrieve, insert and delete the data efficiently.
- It is also used to organize the data in the form of a table, schema, views, and reports, etc.
- Using the database, you can easily retrieve, insert, and delete the information.

For example: The college Database organizes the data about the admin, staff, students and faculty etc.

Database Management System (DBMS):

• A DBMS is software that allows creation, definition and manipulation of database, allowing users to store, process and analyse data easily.

• DBMS provides us with an interface or a tool, to perform various operations like creating database, storing data in it, updating data, creating tables in the database and a lot more.

- DBMS also provides protection and security to the databases.

- It also maintains data consistency in case of multiple users. Here are some examples of popular DBMS used:

MySql

Oracle

SQL Server

IBM DB2

DATABASE APPLICATIONS:

- Telecom: There is a database to keeps track of the information regarding calls made, network usage, customer details etc.
- Industry: Where it is a manufacturing unit, warehouse or distribution centre, each one needs a database to keep the records of ins and outs
- Banking System: For storing customer info, tracking day to day credit and debit transactions, generating bank statements etc.
- Sales: To store customer information, production information and invoice details.
- Airlines: To travel though airlines, we make early reservations; this reservation information along with flight schedule is stored in database.
- Education sector: Database systems are frequently used in schools and colleges to store and retrieve the data regarding student details, staff details, course details, exam details, payroll data, attendance details, fees details etc.

Advantages of DBMS

- Controls database redundancy: It can control data redundancy because it stores all the data in one single database file and that recorded data is placed in the database.

- Data sharing: In DBMS, the authorized users of an organization can share the data among multiple users.

- Easily Maintenance: It can be easily maintainable due to the centralized nature of the database system.

- Reduce time: It reduces development time and maintenance need.

- Backup: It provides backup and recovery subsystems which create automatic backup of data from hardware and software failures and restores the data if required.

- multiple user interface: It provides different types of user interfaces like graphical user interfaces, application program interfaces

Disadvantages of DBMS

- Cost of Hardware and Software: It requires a high speed of data processor and large memory size to run DBMS software.

• Size: It occupies a large space of disks and large memory to run them efficiently.

• Complexity: Database system creates additional complexity and requirements.

• Higher impact of failure: Failure is highly impacted the database because in most of the organization, all the data stored in a single database and if the database is damaged due to electric failure or database corruption then the data may be lost forever.

• Characteristics of Databases

• Data stored into Tables: Data is never directly stored into the database. Data is stored into tables, created inside the database.

• Reduced Redundancy: In the modern world hard drives are very cheap, but earlier when hard drives were too expensive, unnecessary repetition of data in database was a big problem. But DBMS follows Normalisation which divides the data in such a way that repetition is minimum.

• Data Consistency: On Live data, i.e. data that is being continuously updated and added, maintaining the consistency of data can become a challenge. But DBMS handles it all by itself.

• Support Multiple user and Concurrent Access: DBMS allows multiple users to work on it(update, insert, delete data) at the same time and still manages to maintain the data consistency.

- Query Language: DBMS provides users with a simple Query language, using which data can be easily fetched, inserted, deleted and updated in a database.

- **File System v/s Database System**

DBMS	File System
DBMS is a collection of data. In DBMS, the user is not required to write the procedures.	File system is a collection of data. In this system, the user has to write the procedures for managing the database.
Searching data is easy in Dbms	Searching is difficult in File System
Dbms is structured data	Files are unstructured data
No data redundancy in Dbms	Data redundancy is there in file system
Memory utilisation well in dbms	Memory utilisation poor in file system
No data inconsistency in dbms	Inconsistency in file system

DBMS gives an abstract view of data that hides the details.	File system provides the detail of the data representation and storage of data.
DBMS provides a crash recovery mechanism, i.e., DBMS protects the user from the system failure.	File system doesn't have a crash mechanism, i.e., if the system crashes while entering some data, then the content of the file will lost.
DBMS provides a good protection mechanism.	It is very difficult to protect a file under the file system.
DBMS contains a wide variety of sophisticated techniques to store and retrieve the data.	File system can't efficiently store and retrieve the data.
DBMS takes care of Concurrent access of data using some form of locking.	In the File system, concurrent access has many problems like redirecting the file while other deleting some information or updating some information.

- # Data Abstraction and Data Independence

View of Data in DBMS

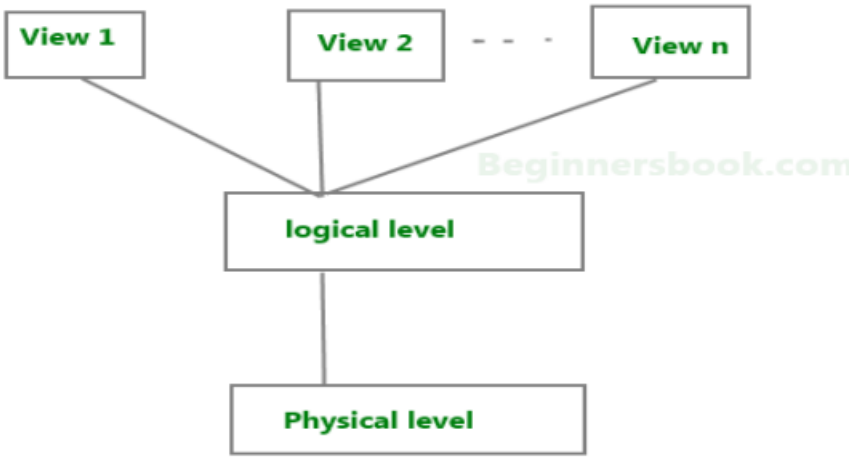
• Abstraction is one of the main features of database systems.

• Hiding irrelevant details from user and providing abstract view of data to users, helps in easy and efficient user-database interaction.

• the three level of DBMS architecture, The top level of that architecture is “view level”. The view level provides the “view of data” to the users and hides the irrelevant details such as data relationship, database schema, constraints, security etc from the user.

Data Abstraction in DBMS

Database systems are made-up of complex data structures. To ease the user interaction with database, the developers hide internal irrelevant details from users. This process of hiding irrelevant details from user is called data abstraction.



Three Levels of data abstraction

We have three levels of abstraction:

Physical level: This is the lowest level of data abstraction. It describes how data is actually stored in database. You can get the complex data structure details at this level.

Logical level: This is the middle level of 3-level data abstraction architecture. It describes what data is stored in database.

View level: Highest level of data abstraction. This level describes the user interaction with database system.

Data Independence in DBMS:

Data Independence is defined as a property of DBMS that helps to change the Database schema at one level of a database system without requiring to change the schema at the next higher level. Data independence helps you to keep data separated from all programs that make use of it. The main purpose of the three levels of data abstraction is to achieve data independence. If the database changes and expands over time, it is very important that the changes in one level should not affect

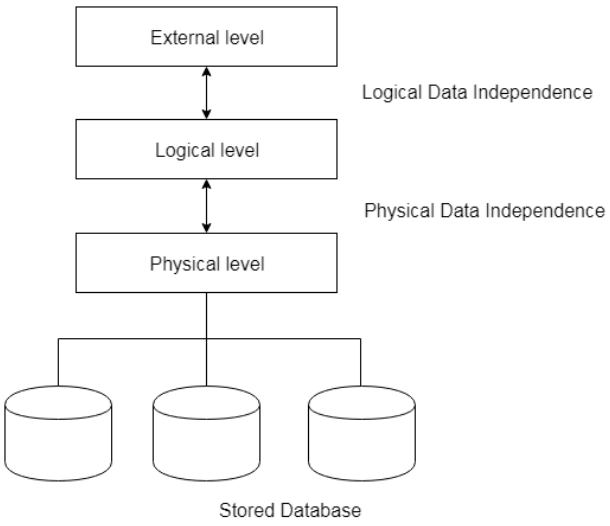
the data at other levels of the database. This would save time and cost required when changing the database.

There are two levels of data independence based on three levels of abstraction.

- Physical Data Independence
- Logical Data Independence

1. *Physical Data Independence*

- Physical data independence can be defined as the capacity to change the internal schema without having to change the conceptual schema.
- If we do any changes in the storage size of the database system server, then the Conceptual structure of the database will not be affected.
- Physical data independence is used to separate conceptual levels from the internal levels.
- Physical data independence occurs at the logical interface level.



2. *Logical Data Independence*

- Logical data independence refers characteristic of being able to change the conceptual schema without having to change the external schema.
- Logical data independence is used to separate the external level from the conceptual view.
- If we do any changes in the conceptual view of the data, then the user view of the data would not be affected.
- Logical data independence occurs at the user interface level.

- ## Schemas and Instance

Schema in DBMS

- Design of a database is called the schema.
- Schema is of three types: Physical schema, logical schema and view schema.
- The design of a database at physical level is called physical schema, how the data stored in blocks of storage is described at this level.
- Design of database at logical level is called logical schema, programmers and database administrators work at this level, at this level data can be described as certain types of data records gets stored in data structures, however the internal details such as implementation of data structure is hidden at this level (available at physical level).
- Design of database at view level is called view schema. This generally describes end user interaction with database systems.

Instance in DBMS

- The data stored in database at a particular moment of time is called instance of database.
- Database schema defines the variable declarations in tables that belong to a particular database; the value of these variables at a moment of time is called the instance of that database.

Difference Between Schema and Instance in DBMS

Parameters	Schema in DBMS	Instance in DBMS
Meaning	Schema refers to the overall description of any given database.	Instance basically refers to a collection of data and information that the database stores at any particular moment.
Alterations	The schema remains the same for the entire database as a whole.	One can change the instances of data and information in a database using updation, deletion, and addition.
Frequency of Change	It does not change very frequently.	It changes very frequently.
Uses	We use Schema for defining the basic structure of any given database. It defines how the available needs to get stored.	We use Instance for referring to a set of information at any given instance/ time.

- ## Users of Database System

Database users are the one who really use and take the benefits of database. There will be different types of users depending on their need and way of accessing the database.

- **Application Programmers -** They are the developers who interact with the database by means of DML queries. These DML queries are written in the application programs like C,

C++, JAVA, Pascal etc. These queries are converted into object code to communicate with the database.

For example, writing a C program to generate the report of employees who are working in particular department will involve a query to fetch the data from database. It will include a embedded SQL query in the C Program.

- **Sophisticated Users -** They are database developers, who write SQL queries to select/insert/delete/update data. They do not use any application or programs to request the database. They directly interact with the database by means of query language like SQL. These users will be scientists, engineers, analysts who thoroughly study SQL and DBMS to apply the concepts in their requirement. In short, we can say this category includes designers and developers of DBMS and SQL.
- **Specialized Users -** These are also sophisticated users, but they write special database application programs. They are the developers who develop the complex programs to the requirement.
- **Stand-alone Users -** These users will have stand –alone database for their personal use. These kinds of database will have readymade database packages which will have menus and graphical interfaces.
- **Native Users -** these are the users who use the existing application to interact with the database. For example, online library system, ticket booking systems, ATMs etc. which has existing application and users use them to interact with the database to fulfil their requests.

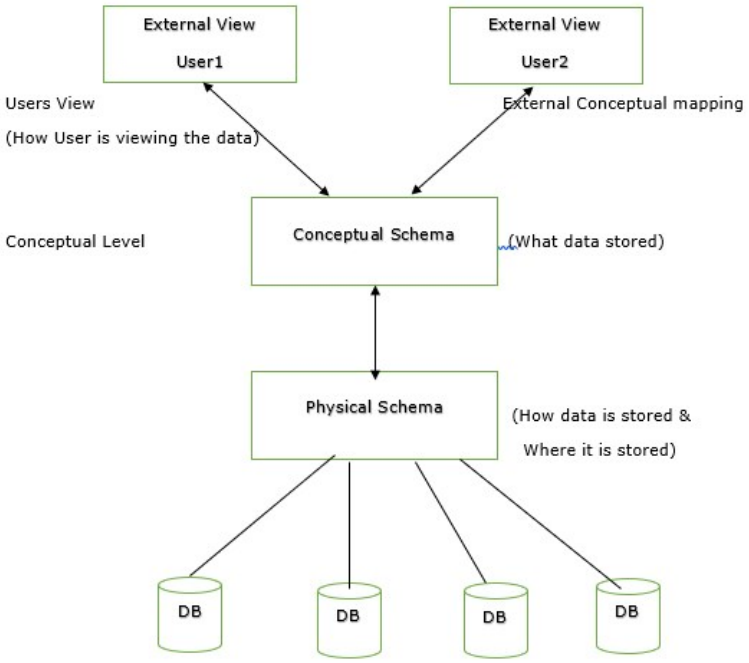
• Three level Schema Architecture

The three-schema architecture divides the database into three-level used to create a separation between the physical database and the user application. In simple terms, this architecture hides the details of physical storage from the user.

The database administrator (DBA) responsible is to change the structure of database storage without affecting the user's view. It deals with the data, the relationship between them and the different access methods implemented on the database. The logical design of database is called a schema

This architecture contains three layers of database management system, which are as follows −

- External level
- Conceptual level
- Internal level



External/ View level

This is the highest level of database abstraction. It includes a number of external schemas or user views. This level provides different views of the same database for a specific user or a group of users. An external view provides a powerful and flexible security mechanism by hiding the parts of the database from a particular user.

Conceptual or Logical level

This level describes the structure of the whole database. It acts as a middle layer between the physical storage and user view. It explains what data to be stored in the database, what the data types are, and what relationship exists among those data. There is only one conceptual schema per database.

This level describes the structure of the whole database. It acts as a middle layer between the physical storage and user view. It explains what data to be stored in the database, what the data types are, and what relationship exists among those data. There is only one conceptual schema per database.

Internal or Physical level

This is the lowest level of database abstraction. It describes how the data is stored in the database and provides the methods to access data from the database. It allows viewing the physical representation of the database on the computer system.

The interface between the conceptual and internal schema identifies how an element in the conceptual schema is stored and how it may be accessed. It is one which is closest to physical storage. The internal schema not only defines different stored record types, but also specifies what indices exist, how stored fields are represented.

- # Database Architecture

• Database management systems architecture will help us understand the components of database system and the relation among them.

• The architecture of DBMS depends on the computer system on which it runs.

• the basic client/server architecture is used to deal with a large number of PCs, web servers, database servers and other components that are connected with networks.

• The client/server architecture consists of many PCs and a workstation which are connected via the network.

• DBMS architecture depends upon how users are connected to the database to get their request done.

TYPES OF DBMS ARCHITECTURE

There are three types of DBMS architecture:

1. Single tier architecture

2. Two tier architecture

3.Three tier architecture

1-Tier Architecture

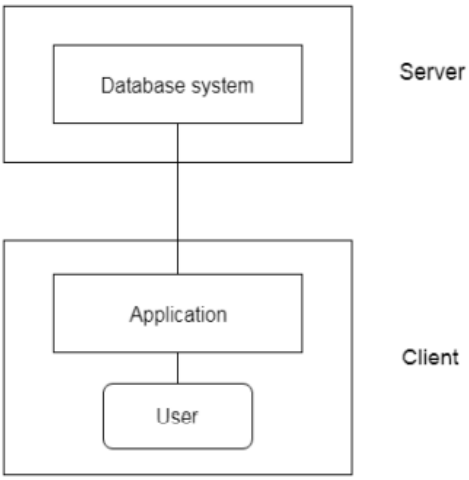
• In this type of architecture, the database is readily available on the client machine, any request made by client doesn't require a network connection to perform the action on the database.

• Any changes done here will directly be done on the database itself. It doesn't provide a handy tool for end users.

• The 1-Tier architecture is used for development of the local application, where programmers can directly communicate with the database for the quick response.

Two tier architecture

• In two-tier architecture, the Database system is present at the server machine and the DBMS application is present at the client machine, these two machines are connected with each other through a reliable network.
• Whenever client machine makes a request to access the database present at server using a query language like sql, the server perform the request on the database and returns the result back to the client.
• The application connection interface such as JDBC, ODBC are used for the interaction between server and client.



3-Tier Architecture

- In three-tier architecture, another layer is present between the client machine and server machine.
- In this architecture, the client application doesn't communicate directly with the database systems present at the server machine, rather the client application communicates with server application and the server application internally communicates with the database system present at the server.