unit I

DBMS: Basic Concepts

1. Introduction
2. Disadvantages of file oriented approach
3. Database
4. Why Database
5. Database Management System(DBMS)
6. Function of DBMS
7. Advantages of DBMS and disadvantage of DBMS
8. Database Basics
9. Three level architecture of DBMS
10. Database users
11. Database language
12. Database structure

Introduction:

In computerized information system data is the basic resource of the organization. So, proper organization and management for data is required fro organization to run smoothly. Database management system deals the knowledge of how data stored and managed on a computerized information system. In any organization, it requires accurate and reliable data for better decision making, ensuring privacy of data and controlling data efficiently.

The examples include deposit and/or withdrawal from a bank,hotel,airline or railway reservation, purchase items from supermarkets in all cases, a database is accessed.

What is data:

Data is the known facts or figures that have implicit meaning. It can also be defined as it is the representation of facts ,concepts or instruction in a formal manner, which is suitable for understanding and processing. Data can be represented in alphabets(A-Z, a-z),in digits(0-9) and using special characters(+,-.#,$, etc) e.g: 25, “ajit” etc.

Information:

Information is the processed data on which decisions and actions are based. Information can be defined as the organized and classified data to provide meaningful values.

Eg: “The age of Ravi is 25”

File:

File is a collection of related data stored in secondary memory.

File Oriented approach:

The traditional file oriented approach to information processing has for each application a separate master file and its own set of personal file. In file oriented approach the program dependent on the files and files become dependent on the files and files become dependents upon the programs

Disadvantages of file oriented approach:

1) **Data redundancy and inconsistency:**
   The same information may be written in several files. This redundancy leads to higher storage and access cost. It may lead data inconsistency that is the various copies of the same data may longer agree for example a changed customer address may be reflected in single file but not else where in the system.

2) **Difficulty in accessing data :**
   The conventional file processing system do not allow data to retrieved in a convenient and efficient manner according to user choice.

3) **Data isolation :**
   Because data are scattered in various file and files may be in different formats with new application programs to retrieve the appropriate data is difficult.

4) **Integrity Problems:**
   Developers enforce data validation in the system by adding appropriate code in the various application program. How ever when new constraints are added, it is difficult to change the programs to enforce them.

5) **Atomicity:**

   It is difficult to ensure atomicity in a file processing system when transaction failure occurs due to power failure, networking problems etc. (atomicity: either all operations of the transaction are reflected properly in the database or non are)

6) **Concurrent access:**
   In the file processing system it is not possible to access a same file for transaction at same time

7) **Security problems:**
   There is no security provided in file processing system to secure the data from unauthorized user access.

Database:

A database is organized collection of related data of an organization stored in formatted way which is shared by multiple users. The main feature of data in a database are:

1. It must be well organized
2. it is related
3. It is accessible in a logical order without any difficulty
4. It is stored only once

for example:
consider the roll no, name, address of a student stored in a student file. It is collection of related data with an implicit meaning.
Data in the database may be persistent, integrated and shared.

Persistent:
If data is removed from database due to some explicit request from user to remove.
Integrated:
A database can be a collection of data from different files and when any redundancy among those files are removed from database is said to be integrated data.
Sharing Data:
The data stored in the database can be shared by multiple users simultaneously with out affecting the correctness of data.

Why Database:

In order to overcome the limitation of a file system, a new approach was required. Hence a database approach emerged. A database is a persistent collection of logically related data. The initial attempts were to provide a centralized collection of data. A database has a self describing nature. It contains not only the data sharing and integration of data of an organization in a single database.

A small database can be handled manually but for a large database and having multiple users it is difficult to maintain it, In that case a computerized database is useful. The advantages of database system over traditional, paper based methods of record keeping are:

- **compactness**:
  No need for large amount of paper files
- **speed**:
  The machine can retrieve and modify the data more faster way then human being
- **Less drudgery**: Much of the maintenance of files by hand is eliminated
- **Accuracy:** Accurate,up-to-date information is fetched as per requirement of the user at any time.

Database Management System (DBMS):
A database management system consists of collection of related data and refers to a set of programs for defining, creation, maintenance and manipulation of a database.

Function of DBMS:

1. **Defining database schema**: it must give facility for defining the database structure also specifies access rights to authorized users.
2. **Manipulation of the database:** The dbms must have functions like insertion of record into database updation of data, deletion of data, retrieval of data
3. **Sharing of database:** The DBMS must share data items for multiple users by maintaining consistency of data.
4. **Protection of database:** It must protect the database against unauthorized users.
5. **Database recovery:** If for any reason the system fails DBMS must facilitate data base recovery.

Advantages of dbms:

Reduction of redundancies:

Centralized control of data by the DBA avoids unnecessary duplication of data and effectively reduces the total amount of data storage required avoiding duplication in the elimination of the inconsistencies that tend to be present in redundant data files.

Sharing of data:

A database allows the sharing of data under its control by any number of application programs or users.

Data Integrity:

Data integrity means that the data contained in the database is both accurate and consistent. Therefore data values being entered for storage could be checked to ensure that they fall with in a specified range and are of the correct format.

Data Security:

The DBA who has the ultimate responsibility for the data in the dbms can ensure that proper access procedures are followed including proper authentication schemas for access to the DBS and additional check before permitting access to sensitive data.

Conflict resolution:

DBA resolve the conflict on requirements of various user and applications. The DBA chooses the best file structure and access method to get optional performance for the application.

Data Independence:

Data independence is usually considered from two points of views; physically data independence and logical data independence.
Physical data Independence allows changes in the physical storage devices or organization of the files to be made without requiring changes in the conceptual view or any of the external views and hence in the application programs using the data base.
Logical data independence indicates that the conceptual schema can be changed without affecting the existing external schema or any application program.

Disadvantage of DBMS:

1. DBMS software and hardware (networking installation) cost is high
2. The processing overhead by the dbms for implementation of security, integrity and sharing of the data.
3. centralized database control
4. Setup of the database system requires more knowledge, money, skills, and time.
5. The complexity of the database may result in poor performance.

Database Basics:

Data item:
The data item is also called as field in data processing and is the smallest unit of data that has meaning to its users.
Eg: “e101”,”sumit”

Entities and attributes:

An entity is a thing or object in the real world that is distinguishable from all other objects
Eg:
Bank,employee,student
Attributes are properties are properties of an entity.
Eg:
Empcode,ename,rolno,name

Logical data and physical data :

Logical data are the data for the table created by user in primary memory.
Physical data refers to the data stored in the secondary memory.

Schema and sub-schema :

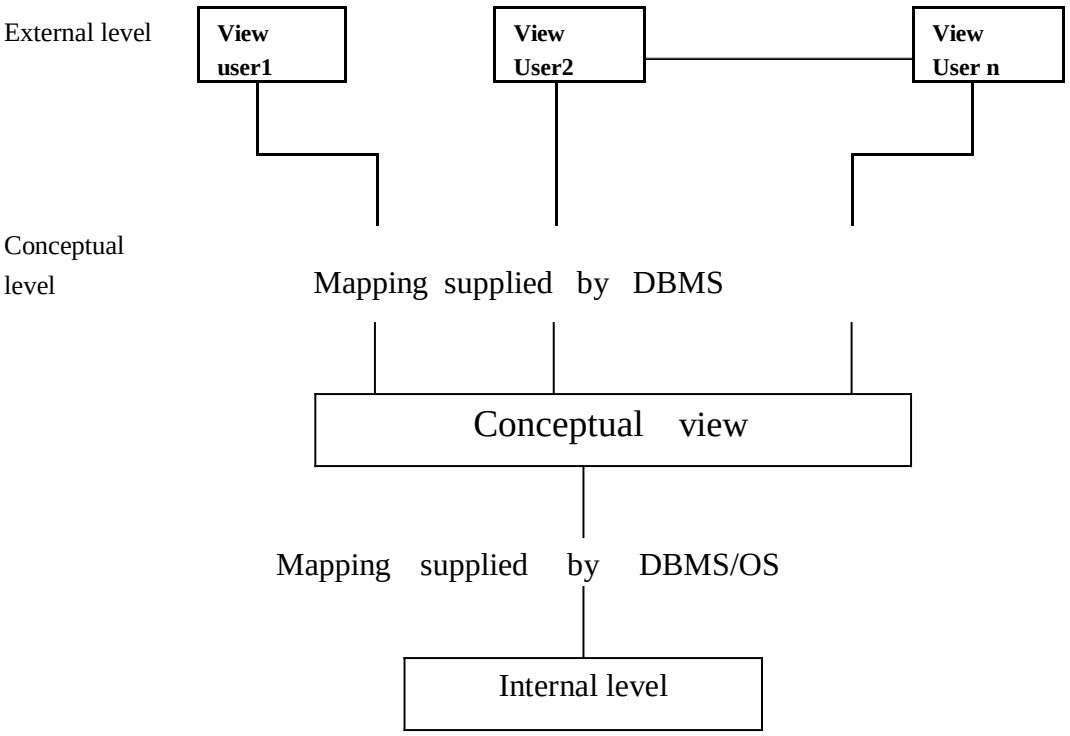
A schema is a logical data base description and is drawn as a chart of the types of data that are used . It gives the names of the entities and attributes and specify the relationships between them.

A database schema includes such information as :

- Characteristics of data items such as entities and attributes .
- Logical structures and relationships among these data items .
- Format for storage representation.
- Integrity parameters such as physical authorization and back up policies.

A *subschema* is derived schema derived from existing schema as per the user requirement. There may be more then one subschema create for a single conceptual schema.

Three level architecture of DBMS :



A database management system that provides three level of data is said to follow three-level architecture .

- External level
- Conceptual level
- Internal level

External level :

The external level is at the highest level of database abstraction . At this level, there will be many views define for different users requirement. A view will describe only a subset of the database. Any number of user views may exist for a given global or subschema.

for example , each student has different view of the time table. the view of a student of Btech (CSE) is different from the view of the student of Btech(ECE).Thus this level of abstraction is concerned with different categories of users.

Each external view is described by means of a schema called schema or schema.

Conceptual level :

At this level of database abstraction all the database entities and the relationships among them are included . One conceptual view represents the entire database . This conceptual view is defined by the conceptual schema.

The conceptual schema hides the details of physical storage structures and concentrate on describing entities , data types, relationships, user operations and constraints.

It describes all the records and relationships included in the conceptual view . There is only one conceptual schema per database . It includes feature that specify the checks to relation data consistency and integrity.

Internal level :

It is the lowest level of abstraction closest to the physical storage method used . It indicates how the data will be stored and describes the data structures and access methods to be used by the database . The internal view is expressed by internal schema. The following aspects are considered at this level:

1. Storage allocation e.g: B-tree,hashing
2. access paths eg. specification of primary and secondary keys,indexes etc
3. Miscellaneous eg. Data compression and encryption techniques,optimization of the internal structures.

Database users :

Naive users :

Users who need not be aware of the presence of the database system or any other system supporting their usage are considered naïve users . A user of an automatic teller machine falls on this category.

Online users :

These are users who may communicate with the database directly via an online terminal or indirectly via a user interface and application program. These users are aware of the database system and also know the data manipulation language system.

Application programmers :

Professional programmers who are responsible for developing application programs or user interfaces utilized by the naïve and online user falls into this category.

Database Administration :

A person who has central control over the system is called database administrator . The function of DBA are :

1. creation and modification of conceptual Schema definition
2. Implementation of storage structure and access method.
3. schema and physical organization modifications .
4. granting of authorization for data access.
5. Integrity constraints specification.
6. Execute immediate recovery procedure in case of failures
7. ensure physical security to database

Database language :

1) Data definition language(DDL) :

DDL is used to define database objects .The conceptual schema is specified by a set of definitions expressed by this language. It also give some details about how to implement this schema in the physical devices used to store the data. This definition includes all the entity sets and their associated attributes and their relation ships. The result of DDL statements will be a set of tables that are stored in special file called data dictionary.

2) Data manipulation language(DML) :

A DML is a language that enables users to access or manipulate data stored in the database. Data manipulation involves retrieval of data from the database, insertion of new data into the database and deletion of data or modification of existing data.

There are basically two types of DML:

- **procedural**: Which requires a user to specify what data is needed and how to get it.
- **non-rocedural:** which requires a user to specify what data is needed with out specifying how to get it.

3) Data control language(DCL):

This language enables user to grant authorization and canceling authorization of database objects.

Elements of DBMS:

DML pre-compiler:

It converts DML statement embedded in an application program to normal procedure calls in the host language. The pre-complier must interact with the query processor in order to generate the appropriate code.

DDL compiler:

The DDL compiler converts the data definition statements into a set of tables. These tables contains information concerning the database and are in a form that can be used by other components of the dbms.

File manager:

File manager manages the allocation of space on disk storage and the data structure used to represent information stored on disk.

Database manager:

A database manager is a program module which provides the interface between the low level data stored in the database and the application programs and queries submitted to the system.

The responsibilities of database manager are:

1. **Interaction with file manager**: The data is stored on the disk using the file system which is provided by operating system. The database manager translate the the different DML statements into low-level file system commands. so The database manager is responsible for the actual storing,retrieving and updating of data in the database.
2. **Integrity enforcement:**The data values stored in the database must satisfy certain constraints(eg: the age of a person can't be less then zero).These constraints are specified by DBA. Data manager checks the constraints and if it satisfies then it stores the data in the database.
3. **Security enforcement:**Data manager checks the security measures for database from unauthorized users.
4. **Backup and recovery:**Database manager detects the failures occurs due to different causes (like disk failure, power failure,deadlock,s/w error) and restores the database to original state of the database.
5. **Concurrency control:**When several users access the same database file simultaneously, there may be possibilities of data inconsistency. It is

responsible of database manager to control the problems occurs for concurrent transactions.

query processor:

The query processor used to interpret to online user's query and convert it into an efficient series of operations in a form capable of being sent to the data manager for execution. The query processor uses the data dictionary to find the details of data file and using this information it create query plan/access plan to execute the query.

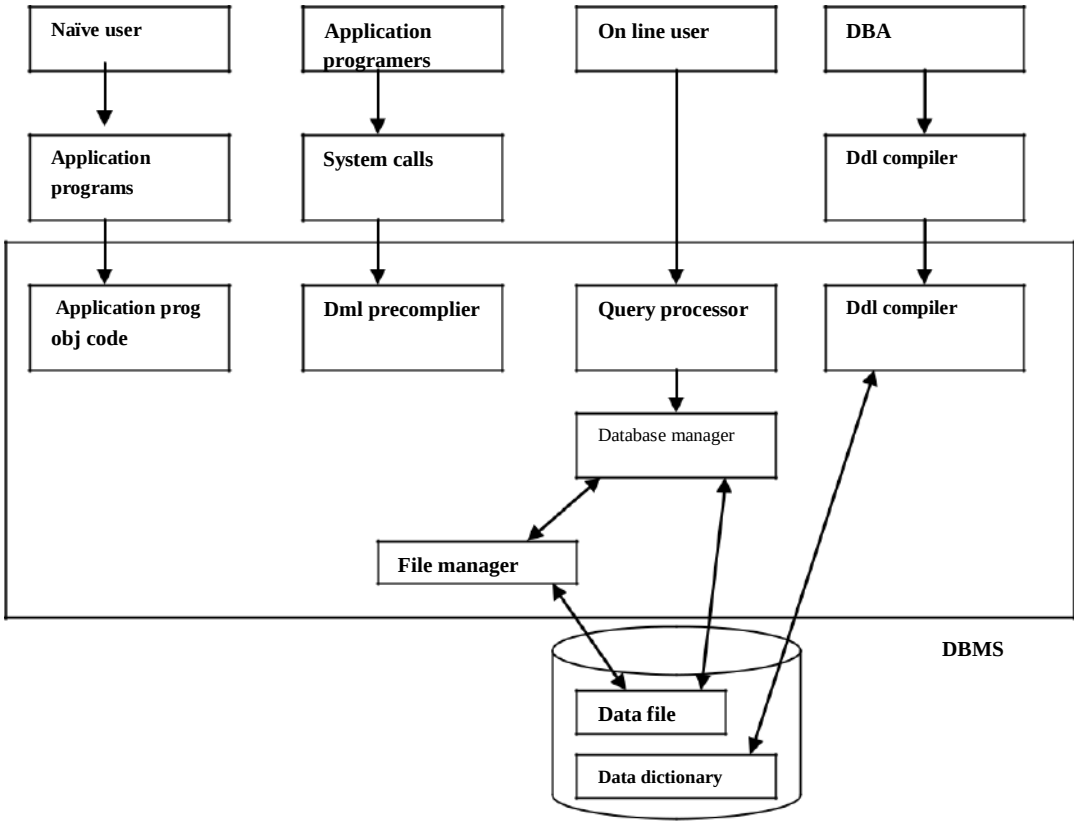
Data Dictionary:

Data dictionary is the table which contains the information about database objects. It contains information like

1. external, conceptual and internal database description
2. description of entities , attributes as well as meaning of data elements
3. synonyms, authorization and security codes
4. database authorization

The data stored in the data dictionary is called *meta data.*

DBMS STRUCTURE:



Q. List four significant differences between a file-processing system and a DBMS.

Answer: Some main differences between a database management system and a file-processing system are:

- Both systems contain a collection of data and a set of programs which access that data. A database management system coordinates both the physical and the logical

access to the data, whereas a file-processing system coordinates only the physical access.

- A database management system reduces the amount of data duplication by ensuring that a physical piece of data is available to all programs authorized to have access to it, where as data written by one program in a file-processing system may not be readable by another program.
- A database management system is designed to allow flexible access to data (i.e., queries), whereas a file-processing system is designed to allow predetermined access to data (i.e., compiled programs).
- A database management system is designed to coordinate multiple users accessing the same data at the same time. A file-processing system is usually designed to allow one or more programs to access different data files at the same time. In a file-processing system, a file can be accessed by two programs concurrently only if both programs have read-only access to the file.

Q.Explain the difference between physical and logical data independence.

Answer:

- Physical data independence is the ability to modify the physical scheme without making it necessary to rewrite application programs. Such modifications include changing from unblocked to blocked record storage, or from sequential to random access files.
- Logical data independence is the ability to modify the conceptual scheme without making it necessary to rewrite application programs. Such a modification might be adding a field to a record; an application program's view hides this change from the program.

Q. List five responsibilities of a database management system. For each responsibility, explain the problems that would arise if the responsibility were not discharged.

Answer: A general purpose database manager (DBM) has five responsibilities:

a. interaction with the file manager.
b. integrity enforcement.
c. security enforcement.
d. backup and recovery.
e. concurrency control.

If these responsibilities were not met by a given DBM (and the text points out that sometimes a responsibility is omitted by design, such as concurrency control on a single-user DBM for a micro computer) the following problems can occur, respectively:

a. No DBM can do without this, if there is no file manager interaction then nothing stored in the files can be retrieved.

b. Consistency constraints may not be satisfied, account balances could go below the minimum allowed, employees could earn too much overtime (e.g.,hours > 80) or, airline pilots may fly more hours than allowed by law.
c. Unauthorized users may access the database, or users authorized to access part of the database may be able to access parts of the database for which they lack authority. For example, a high school student could get access to national defense secret codes, or employees could find out what their supervisors earn.
d. Data could be lost permanently, rather than at least being available in a consistent state that existed prior to a failure.
e. Consistency constraints may be violated despite proper integrity enforcement in each transaction. For example, incorrect bank balances might be reflected due to simultaneous withdrawals and deposits, and so on.

Q. What are five main functions of a database administrator?

Answer: Five main functions of a database administrator are:

- To create the scheme definition
- To define the storage structure and access methods
- To modify the scheme and/or physical organization when necessary
- To grant authorization for data access
- To specify integrity constraints

Q. List six major steps that you would take in setting up a database for a particular enterprise.

Answer: Six major steps in setting up a database for a particular enterprise are:

- Define the high level requirements of the enterprise (this step generates a document known as the system requirements specification.)
- Define a model containing all appropriate types of data and data relationships.
- Define the integrity constraints on the data.
- Define the physical level.
- For each known problem to be solved on a regular basis (e.g., tasks to be carried out by clerks or Web users) define a user interface to carry out the task, and write the necessary application programs to implement the user interface.
- Create/initialize the database.

EXERCISES:

1. What is database management system
2. What are the disadvantage of file processing system

3. State advantage and disadvantage of database management system
4. What ate different types of database users
5. What is data dictionary and what are its contents
6. What are the function of DBA
7. What are the different database languages explain with example.
8. Explain the three layer architecture of DBMS.
9. Differentiate between physical data independence and logical data independence
10. Explain the function of data base manager
11. Explain meta data

Extra content for unit 1:

Client Server Architecture

Client-Server architecture is an architectural deployment style that describe the separation of functionality into layers with each segment being a tier that can be located on a physically separate computer. They evolved through the component-oriented approach, generally using platform specific methods for communication instead of a message-based approach.

This architecture has different usages with different applications. It can be used in web applications and distributed applications. The strength in particular is when using this architecture over distributed systems. In this course work, I will furthermore invest this through the example of three-tier architecture in web applications.

Structure

Using this architecture the software is divided into 3 different tiers: ***Presentation tier, Logic tier, and Data tier***. Each tier is developed and maintained as an independent tier.

*1-**Presentation tier***

This is the topmost level of the application. The presentation layer provides the application's user interface (UI). Typically, this involves the use of Graphical User Interface for smart client interaction, and Web based technologies for browser-based interaction. The presentation tier displays information related to such services as browsing merchandise, purchasing, and shopping cart contents. It communicates with other tiers by outputting results to the browser/client tier and all other tiers in the network.

2-Logic tier **(called also business logic, data access tier, or middle tier)**

The logic tier is pulled out from the presentation tier and, as its own layer; it controls an application's functionality by performing detailed processing. Logic tier is where mission-critical business problems are solved. The components that make up this layer can exist on a server machine, to assist in resource sharing. These components can be used to enforce business rules, such as business algorithms and legal or governmental regulations, and data rules, which are designed to keep the data structures consistent within either specific or multiple databases. Because these

middle-tier components are not tied to a specific client, they can be used by all applications and can be moved to different locations, as response time and other rules require. For example, simple edits can be placed on the client side to minimize network round-trips, or data rules can be placed in stored procedures.

3-Data tier

This tier consists of database servers, is the actual DBMS access layer. It can be accessed through the business services layer and on occasion by the user services layer. Here information is stored and retrieved. This tier keeps data neutral and independent from application servers or business logic. Giving data its own tier also improves scalability and performance. This layer consists of data access components (rather than raw DBMS connections) to aid in resource sharing and to allow clients to be configured without installing the DBMS libraries and ODBC drivers on each client. An example would be a computer hosting a database management system (DBMS), such as a Microsoft SQL Server database.

Components Interconnections:

3 tier application architecture is characterized by the functional decomposition of applications, service components, and their distributed deployment, providing improved scalability, availability, manageability, and resource utilization. During an application's life cycle, the three-tier approach provides benefits such as reusability, flexibility, manageability, maintainability, and scalability. Each tier is completely independent from all other tiers, except for those immediately above and below it. You can share and reuse the components and services you create, and you can distribute them across a network of computers as needed. You can divide large and complex projects into simpler projects and assign them to different programmers or programming teams. You can also deploy components and services on a server to help keep up with changes, and you can redeploy them as growth of the application's user base, data, and transaction volume increases.

Logic layer is moved outside the presentation layer and into the business layer as it enhances reuse. As applications grow, applications often grow into other realms. Applications may start out as a web application, but some of the functionality may later be moved to a smart client application. Portions of an application may be split between a web site and a web or windows service that runs on a server. In addition, keeping logic helps aid in developing a good design (sometimes code can get sloppier in the UI).

The main benefits of the 3-tier architectural style are:

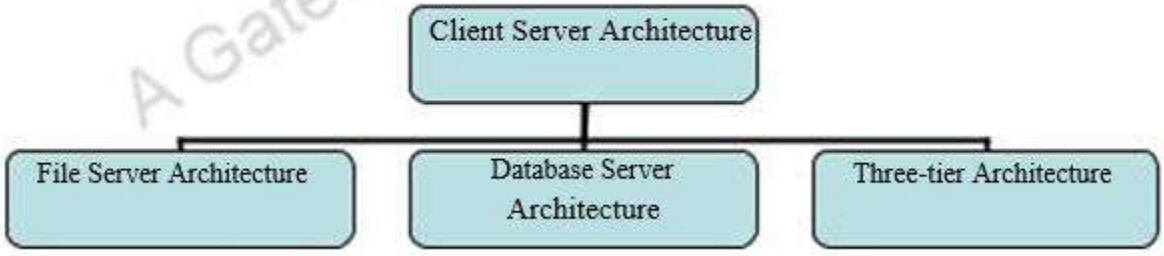
- Maintainability. Because each tier is independent of the other tiers, updates or changes can be carried out without affecting the application as a whole.

- Scalability. Because tiers are based on the deployment of layers, scaling out an application is reasonably straightforward.
- Flexibility. Because each tier can be managed or scaled independently, flexibility is increased.
- Availability. Applications can exploit the modular architecture of enabling systems using easily scalable components, which increases availability.

Consider the 3-tier architectural style if the processing requirements of the layers in the application differ such that processing in one layer could absorb sufficient resources to slow the processing in other layers, or if the security requirements of the layers in the application differ. For example, the presentation layer should not store sensitive data, while this may be stored in the business and data layers. The 3-tier architectural style is also appropriate if you want to be able to share business logic between applications, and you have sufficient hardware to allocate the required number of servers to each tier.

The client server architectures are of three types

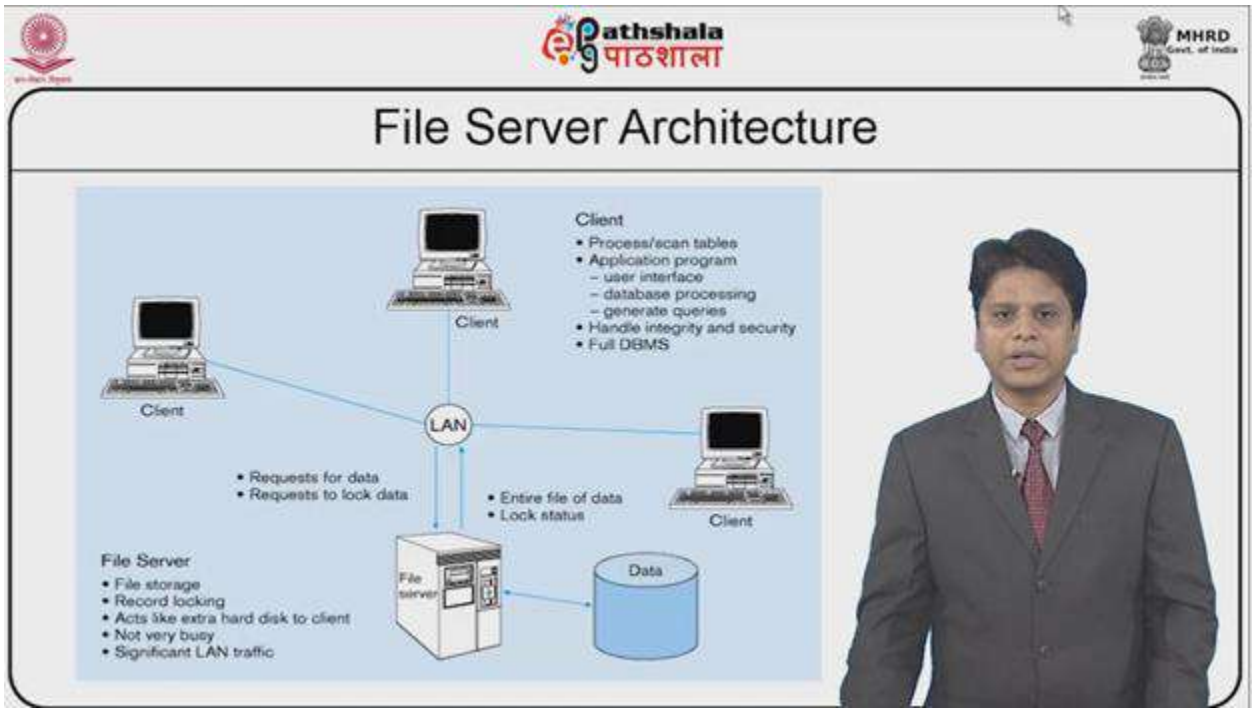
1. Filer Server Architecture
2. Database Server Architecture
3. Three-tier Architecture



FILE SERVER ARCHITECTURE

The first client server architecture being developed is the file server architecture., all processing will be done at the PC that requested the data that is client handles the presentation logic, the processing logic and much of the storage logic. That means out of t he hundred percent work that need to be done every logics that we have seen application logics that we have seen like presentation ;logic, processing logic and storage logic more than 80% of the job it will be carried out at the client side itself. So that is what I am trying to say here. All processing is done at the PC that requested the data that is client handles the presentation ;logic processing logic and much of the storage logic. A file server is a device that manage sonly the file operation and shared by each of the client PCs attached to the local area network. Each filer server acts as a n additional hard disk so wherever I am talking about a file server every file server

will be ac5ring as an extra hard disk which will be present not only at the server level and which will also be present at the client side. For each of the client PC. Each PC maybe called a fat client because this client does an extensive work based on that I am calling this client as a fat client so most of the processing since it happens at the client side it is called fat client. Entire file are transferred from the server to the client for processing. So anything that the client has requested that query will be taken by the server and the server will respond all the possible states back to the server without any processing and client after the client receives all the service or the files requested for processing the client executes everything and it will be called a process.



So this is the complete architecture of a file se rver system where if you look at the machine all the clients are connected by a local area network to a file server this file server in turn manages all the data. So if you look at the major responsibility of a client and the responsibilities of the server, client does processing and scanning of tables, client does application program user interface not only that client does data processing, client generates queries, handles integrity that means it checks of the validation of the data and it also checks for the identification of from which node it has come from so it also have to handle security aspects of it. So that means almost all the processes taking at the client level so it maintains only a copy of DBMS that is available in the server. At the same side, if you look at the file server architecture the file server looks after the storage of all the files and locking of records acts like an extra hard disk to the client and this will not be very busy because many a times I am not transferring all the information from the server's end I am transferring the

information to the client and I don't process the information. So it is not very busy and it has got significant LAN traffic. So when you look at the e ntire architecture although many machines are connected to the server the responsibility of the serve r will any request that I get form different client machine is push all the data from my end to the client's side. It is a responsibility of a client to take up all the data that is available in the server p rocess whatever which is required for the client to execute those will be done at the client by itself. So if you look at the networking system the client can request for a data, request to log the data and from the server it will be given back where the entire file of the data will be transferred to the client and the lock status will also be given to the client. Say lock like read, write locks once the client request it means it is trying to read from the server so it locks. so whatever information that is to be read, maybe it could be put as a shared lock or a n exclusive lock where in case of a shared lock many clients can request to a server, what are the information that the client request to read from the server and if it is to be an exclusive lock if it is from the client's end it will be locked at the server then the client after transferring all the information to the client then the lock will be released. So those information will be handled at the server's end. So as I have already told you this filer server architecture since most of the work is carried out at the client end, that is out of 100% of the job 80% of the work is being done at the client's end, so I call these type of machine as fat client machines.

Every system will have its own drawbacks. There are a few drawbacks wherein case because of transfer of information from the server to the client as requested there will be a huge amount of data transfer on the network because when an client wants to access a whole data table, in case a client wants a whole data table to processing a simple information then the whole data will be transferred from the server to the client. So this has to be transferred to the PC so the server is doing very little work that means serve locate where the table is transfers the entire table to the entire table to the client machine. So server does only little work, network is transferring large blocks of data and the client is being busy processing all the data with an extensive data manipulations. The second issue that raises because of the file server architecture, each client is authorized to use database management system, when a DB program runs particularly on that PC. Thus there is one database but many concurrently running copies of DBMS. The DBMS copy I each client PC must manage the shared database integrity. So programmers must be sophisticated to recognize various subtle conditions that can arise in a multiple user database environment.

DATABASE SERVER ARCHITECTURE:

It is similar to data warehouse where the website store or maintain their data and information. A Database Server is a computer in a LAN that is dedicated to database storage and retrieval. The database server holds the Database Management System (DBMS) and the databases. Upon requests from the client machines, it searches the database for selected records and passes them back over the network**.**

A database server can be defined as a server dedicated to providing database services. Such a server runs the database software. A database server can typically be seen in a client-server environment where it provides information sought by the client systems.

A database server is useful for organizations that have a lot of data to deal with on a regular basis. If you have client-server architecture where the clients need process data too frequently, it is better to work with a database server. Some organizations use the file server to store and process data. A database server is much more efficient than a file server.

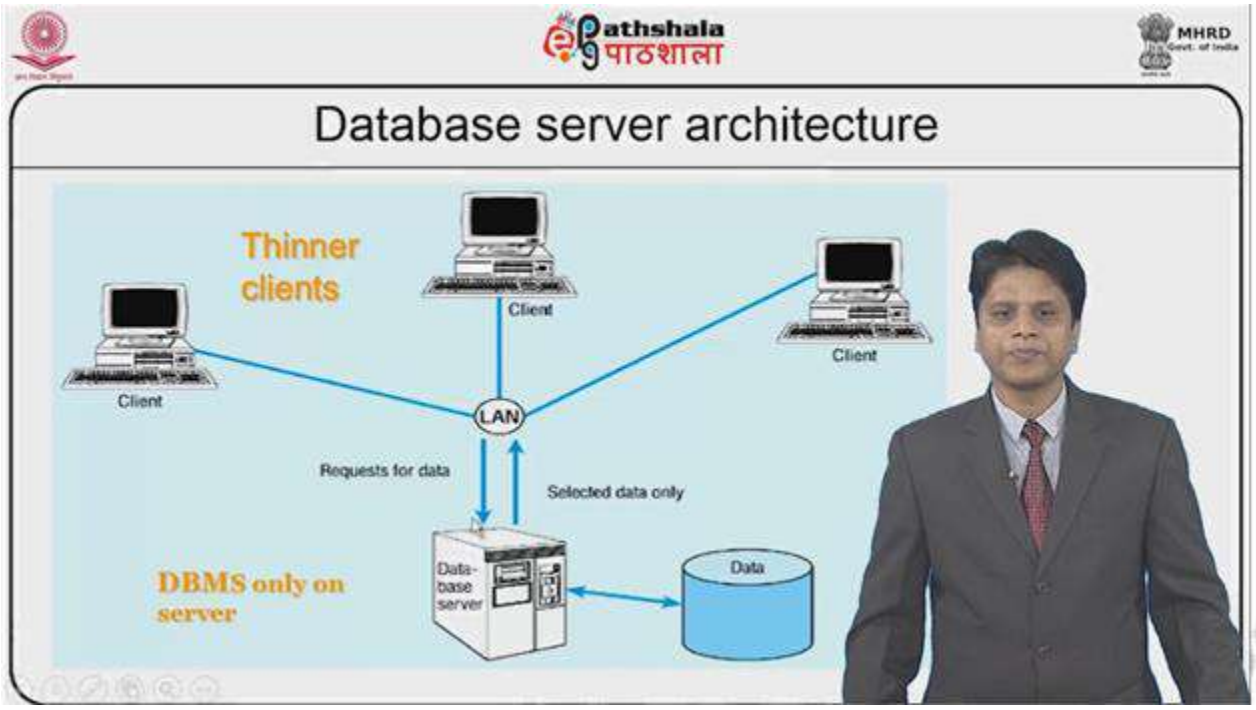
In Database Network the client execute SQL requests to the database server. The Network Database Server Process the client database request and the executed answers of SQL command are come back over the network computer. In the whole concept Database server serves its own power to process the request or search the requested result. The Database server some time also known as SQL engine.

All database functions are controlled by the database server. Any type of computer can be used as database server. It may be microcomputer, minicomputer or mainframe computer. In large organization networks, the mainframe computers are used as server.

Some people refer to the central DBMS functions as the back-end functions, whereas the application programs on the client computer as front -end programs. You can say that client is the application, which is used to interface with the DBMS, while database server is a DBMS.

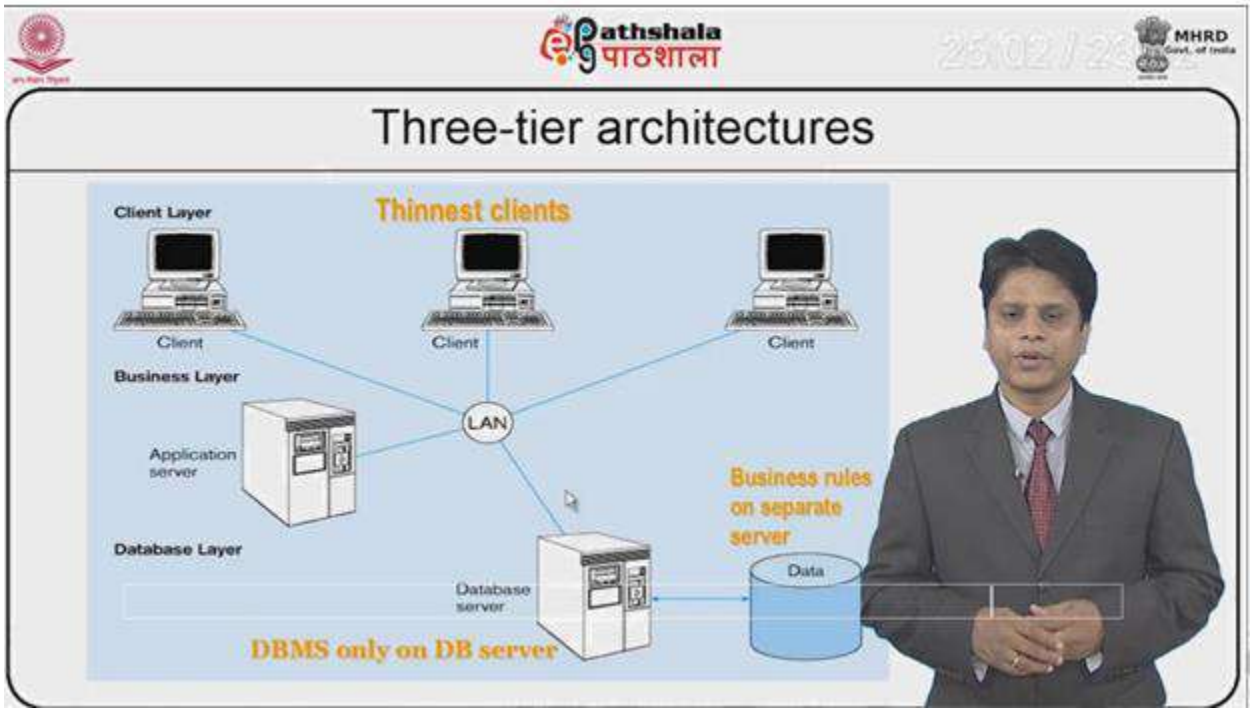
The Database server manages the recovery security services of the DBMS. It enforces the constraints that are specified inside the DBMS. It controls and manages all the clients that are connected to it. It handles all database access and control functions.

It provides concurrent access control. It provides better security and server hides the DBMS from clients. It provides the multi-user environment. Several users can access the database simultaneously. All the data is stored on the data server therefore, the DBA can easily create the backup of the database.



A standard called ***ODBC*** (Open ***Database Connectivity)*** provides an application programming interface (API), which allows client side programs to call the DBMS on the server side. For this purpose, necessary software should be installed on both sides (i.e., client and server). Hence, a client program connects to the Database server and sends requests (queries) using the ODBC Application Programming Interface (API). The-server processes the queries and sends back the results of queries to the client program, which are processed by the client computer

THREE TIER ARCHITECTURE



1. Presentation Layer

In the presentation layer the user interaction takes place as defined before. The user enters the address in the web browser and in the browser the URL is decoded into protocol/host/file, i.e. host name converted to IP address. Then an issue request is sent to remote server using appropriate protocol (usually HTTP).

Also a returned HTML from the logic tier might be accepted. In the presentation layer interaction with client side scripts (e.g. using DHTML) is supported and user inputer of variety controls on the form are accepted.

2. Logic Tier

In the logic Tier the application's functionality is done by performing detailed processing of data from presentation layer. Server such as Appache (or IIS) or Server Script (such as PHP) can be used to support this.

With Server (Apache or IIS) the appropriate action to be taken is identified, such as fetching a file, or passing request to an interpreter. Also it sends an output back to caller in MIME package. As such support for thousands for concurrent users, multithreading (allow multiple processor to run concurrently) and caching (holding results in a temporary store to reduce recalculation) is achieved).

With Server script (example in PHP) interacting with server such as accessing input or generating input is done. It interprets the requests according to business rules and past transactions from this client, and requests appropriate data from the persistence layer. It also computes the derived data and creates HTML (or GIF…) for the page.

3. Data Layer

This tier consists of database servers. The interaction with the database is done using standard languages such as SQL queries using database specific protocol over TCP/IP. The data structures (for example tables) are defined and modified themselves, that insertion, updating and deleting of data for example. Data maintenance should be maintained with backup and recovered. Access to compilation of queries should be optimized, with indexing or replication of tables. An example of technology using this would be .NET that is built into the .NET framework, as ADO.NET contains a mechanism to query data out of the database and return it to the caller in a connected or disconnected fashion.

Real life example of a web system explained above would be in Emails done using 3 Tier Architecture. Reading e-mail using a Web-based interface, such as Hotmail, uses a three-tier architecture. The three tiers are:

1. Presentation Layer: The client's web browser that sends HTTP requests to the
2. Logic Layer: The Web server:

a. sends HTTP responses to the Web client

b. Translates the client's HTTP requests into SMTP packets which are then sent to the Mail server.

3. Data Layer: The Mail server performs the following functionality, and when performed it is transformed to the Logic Layer.

a. When completed, an e-mail message is sent by the sender's e-mail client as an SMTP packet to the local mail server.

b. The mail server's message transfer agent next reads the packet's destination address and sends it over the Internet to the receiver's mail server.

c. The destination mail transfer agent then stores the message in the receiver's mail box.

d. When the receiver next accesses e-mail, his or her user agent contacts the local mail server which then downloads the message to the receiver 's client computer.