BCA Part II
Paper-XI: DBMS Using MS-ACCESS
Topic: DBMS Architecture

Prepared by: Dr. Kiran Pandey
(School of Computer Science)

Email-id: kiranpandey.nou@gmail.com

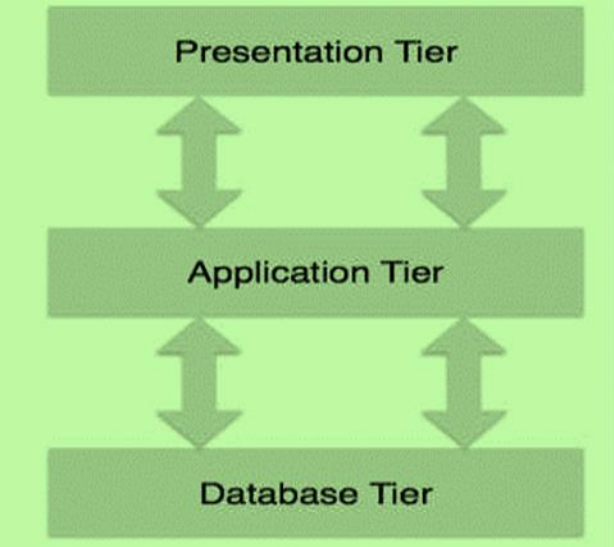
The design of a DBMS depends on its architecture. It can be centralized or decentralized or hierarchical. The architecture of a DBMS can be seen as either single tier or multi-tier. An n-tier architecture divides the whole system into related but independent **n** modules, which can be independently modified, altered, changed, or replaced.

In 1-tier architecture, the DBMS is the only entity where the user directly sits on the DBMS and uses it. Any changes done here will directly be done on the DBMS itself. It does not provide handy tools for end-users. Database designers and programmers normally prefer to use single-tier architecture.

If the architecture of DBMS is 2-tier, then it must have an application through which the DBMS can be accessed. Programmers use 2-tier architecture where they access the DBMS by means of an application. Here the application tier is entirely independent of the database in terms of operation, design, and programming.

3-TIER ARCHITECTURE

A 3-tier architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database. It is the most widely used architecture to design a DBMS.



- **Database (Data) Tier**: At this tier, the database resides along with its query processing languages. We also have the relations that define the data and their constraints at this level.

- **Application (Middle) Tier**: At this tier reside the application server and the programs that access the database. For a user, this application tier presents an abstracted view of the database. End-users are unaware of any existence of the database beyond the application. At the other end, the database tier is not aware of any other user beyond the application tier. Hence, the application layer sits in the middle and acts as a mediator between the end-user and the database.
- **User (Presentation) Tier**: End-users operate on this tier and they know nothing about any existence of the database beyond this layer. At this layer, multiple views of the database can be provided by the application. All views are generated by applications that reside in the application tier.

The need for three level architecture

The objective of the three level architecture is to separate each user's view of the database from the way the database is physically represented.

- **Support of multiple user views:** Each user is able to access the same data, but have a different customized view of the data. Each user should be able to change the way he or she views the data and this change should not affect other users.

- **Insulation between user programs and data that does not concern them:** Users should not directly deal with physical storage details, such as indexing or hashing. The user's interactions with the database should be independent of storage considerations. It can be more clearly defined as:

  (i) The Database Administrator should be able to change the storage structures without affecting users' views.

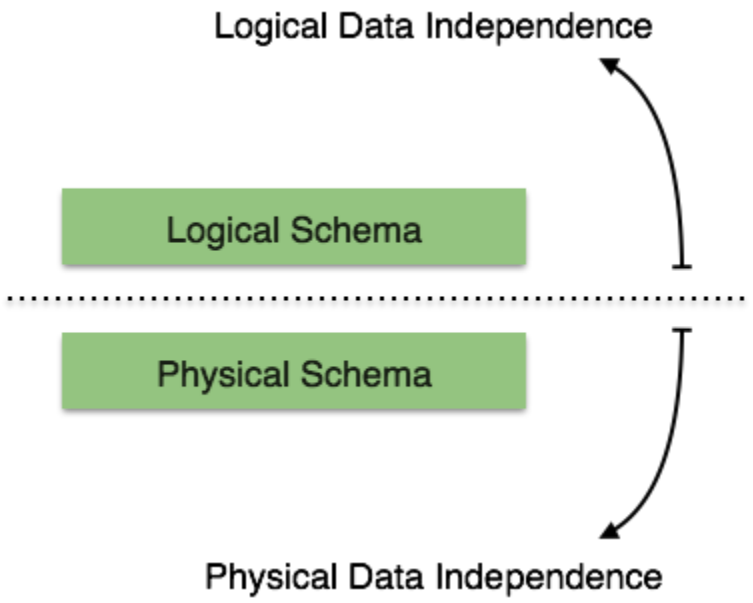
  (ii) The internal structure of the database should be unaffected by the changes to the physical aspects of the storage, such as changing to a new storage device.

Multiple-tier database architecture is highly modifiable, as almost all its components are independent and can be changed independently.

Data Independence

If a database system is not multi-layered, then it becomes difficult to make any changes in the database system. Database systems are designed in multi-layers as we learnt earlier.

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.



Metadata itself follows a layered architecture, so that when we change data at one layer, it does not affect the data at another level. This data is independent but mapped to each other.

Logical Data Independence

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.

Physical Data Independence

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.

COMPONENTS OF DBMS

A DBMS software is partitioned into several modules or components and each module or component is assigned a specific operation to perform. While designing a DBMS, its interface must be taken into account with the OS as some of the functions of DBMS are supported by OS to provide basic services and DBMS is built on top of it. The functional components of a database system can be broadly divided into –

- **Query Processor Component:** It simplify and facilitate access to data (convenient and efficient)
- **Storage Manager Component:** It minimize the need to move data between disk and main memory
- **Transaction Manager Component:** It handle atomicity and concurrency of transactions and consistency and durability of the databases

Query Processor

Query processor is used to interpret the online user query and converts it into an efficient series of operation in a form capable of being send to the data manager for execution. The query processor use the data dictionary to find the structure of the relevant portion of the data base and use this information in modifying the query and prepare an optimal plan to access the database. It is a program module that provides the interface between the database and the application programs/queries. The Query Processor Components include –

(i) **Data Definition Language(DDL) Compiler-** DDL compiler takes the data definition statement that is the source form & convert them into the object form (or) interprets DDL commands and records them in the data dictionary

(ii) **Data Modelling Language (DML) compiler –** translates DML commands into query evaluation plans

(iii) **Query evaluation engine –** executes queries according to the plans

Storage Manager

A Storage Manager is a component or program module that provides the interface between the low-level data stored in the database and the application programs/queries submitted to the system. The Storage Manager Components include:

(i) File Manager- File manager manages the file space and it takes care of the structure of the file. It manages the allocation space on disk storage and the data structures used to represent info stored on other media. . It is also responsible for locating the block containing the required record, requesting this block from the disk manager, and transmitting the required record to the data manager. The file manager can be implemented using an interface to the existing file subsystem provided by the operating system of the host computer or it can include a file subsystem written especially for the DBMS.

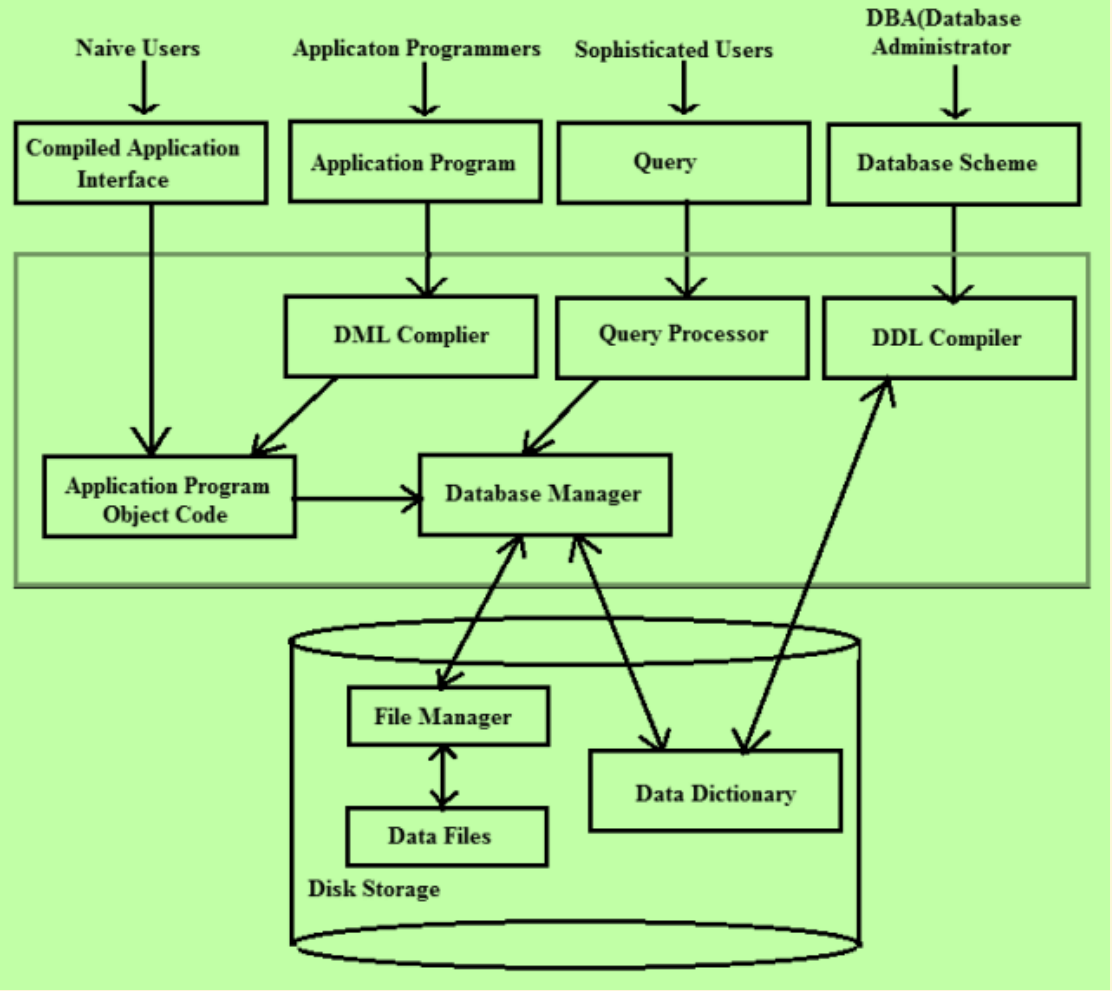
(ii) Buffer Manager – It transfers blocks between disk (or other devices) and Main Memory. A DMA (Direct Memory Access) is a form of Input/output that controls the exchange of blocks process. When a processor receives a request for a transfer of a block, it sends it to the DMA Controller which transfers the block uninterrupted.

(iii) Disk Manager- The block requested by the file manager is transferred by the Disk Manager. The disk manager is part of the operating system of the host computer and all physical input and output operations are performed by it. The disk manager transfers the block or page requested by the file manager so that the latter need not be concerned with the physical characteristics of the underlying storage media.

(iv) **Data Manager-**The data manager is the central software component of the DBMS. It is sometimes referred to as the database control system. One of the functions of the data manager is to convert operations in the user's queries coming directly via the query processor or\ indirectly via an application program from the user's logical view to a physical file system. The data manager is responsible for interfacing with the file system as show. In addition, the tasks of enforcing constraints to maintain the consistency and integrity of the data, as well as its security, are also performed by the data manager. It is also the responsibility of the Data. Manager to provide the synchronization in the simultaneous operations performed by concurrent users and to maintain the backup and recovery operations.

The Structures maintained by Storage manager are-

1. **Data Files-** Data files contains the data portion of the data base.
2. **Data Dictionary-** DBMS must a data dictionary function. The dictionary contains the data about the data. Rather than just raw data. The information about attributes, entity, mapping & cross reference information is contained in the data dictionary.
3. **Indices or Indexing and Access Aids –** An index is a small table having two columns in which the first column contains a copy of the primary or candidate key of a table and the second column contains a set of pointers holding the address of the disk block where that particular key value can be found. The advantage of using indices is that index makes search operation perform very fast. In a data base system, a set of access aids in the form of indexes are usually provided to improve the performance of a database system.



Transaction Manager /Data Manager –

A Transaction is a collection of operations that performs as a single logical function in a database application. All Transactions must follow transaction properties, which are called ACID Properties. Data Manager converts the user queries from the user logical view to a physical file system.

The ACID Properties are –

- **Atomicity**: either all operations succeed or all of them fail
- **Consistency**: the database is changed from one consistent state to another consistent state
- **Isolation**: no transaction interfere other transactions in the middle
- **Durability**: operations of successful transactions must persist

The Components included by Transaction Manager is –

1. **Transaction Manager –** Transaction Manager controls the execution of transactions.
2. **Lock Manager –** Access of items in DBMS is controlled by LOCKS. And the part of DBMS that keeps a record of locks issued to transactions, is done by the Lock Manager. It maintains a LOCK table which is a hash table, with data object identifier as the key
3. **Recovery Manager –** Recovery manager is responsible for atomicity and durability. It allows DBMS to restore the database to a consistent state following a failure.

Tele-Communication System- Online user of the computer system whether remote or local communicate with it by sending and receiving message over communication line. These messages are routed by communication line.

ADVANTAGES OF DBMS

Traditionally, data was organized in file formats. DBMS was a new concept then, and all the research was done to make it overcome the deficiencies in traditional style of data management. A modern DBMS has the following characteristics:

- **Real-world entity**: A modern DBMS is more realistic and uses real-world entities to design its architecture. It uses the behavior and attributes too. For example, a school database may use students as an entity and their age as an attribute.
- **Relation-based tables**: DBMS allows entities and relations among them to form tables. A user can understand the architecture of a database just by looking at the table names.
- **Isolation of data and application**: A database system is entirely different than its data. A database is an active entity, whereas data is said to be passive, on which the database works and organizes. DBMS also stores metadata, which is data about data, to ease its own process.
- **Less redundancy**: DBMS follows the rules of normalization, which splits a relation when any of its attributes is having redundancy in values. Normalization is a mathematically rich and scientific process that reduces data redundancy.

- **Consistency**: Consistency is a state where every relation in a database remains consistent. There exist methods and techniques, which can detect attempt of leaving database in inconsistent state. A DBMS can provide greater consistency as compared to earlier forms of data storing applications like file-processing systems.

- **Query Language**: DBMS is equipped with query language, which makes it more efficient to retrieve and manipulate data. A user can apply as many and as different filtering options as required to retrieve a set of data. Traditionally it was not possible where file-processing system was used.
- **ACID Properties**: DBMS follows the concepts of **A**tomicity, **C**onsistency, **I**solation, and **D**urability (normally shortened as ACID). These concepts are applied on transactions, which manipulate data in a database. ACID properties help the database stay healthy in multi-transactional environments and in case of failure.
- **Multiuser and Concurrent Access**: DBMS supports multi-user environment and allows them to access and manipulate data in parallel. Though there are restrictions on transactions when users attempt to handle the same data item, but users are always unaware of them.
- **Multiple views**: DBMS offers multiple views for different users. A user who is in the Sales department will have a different view of database than a person working in the Production department. This feature enables the users to have a concentrate view of the database according to their requirements.
- **Security**: Features like multiple views offer security to some extent where users are unable to access data of other users and departments. DBMS offers methods to impose constraints while entering data into the database and retrieving the same at a later stage. DBMS offers many different levels of security features, which enables multiple users to have different views with different features. For example, a user in the Sales department cannot see the data that belongs to the Purchase department. Additionally, it can also be managed how much data of the Sales department should be displayed to the user. Since a DBMS is not saved on the disk as traditional file systems, it is very hard for miscreants to break the code.

TYPES OF DATABASE USERS

A typical DBMS has users with different rights and permissions who use it for different purposes. Some users retrieve data and some back it up. The users of a DBMS users can be broadly categorized as follows:

- **Administrators**: **Database Administrator (DBA) –** is a person or group of persons responsible for overall control of database DBA maintains the DBMS and are responsible for administrating the database. They are responsible to look after its usage and by whom it should be used. They create access profiles for users and apply limitations to maintain isolation and force security. Administrators also look after DBMS resources like system license, required tools, and other software and hardware related maintenance.
- **Designers**: Designers are the group of people who actually work on the designing part of the database. They keep a close watch on what data should be kept and in what format. They identify and design the whole set of entities, relations, constraints, and views.
- **End Users**: End users are those who actually reap the benefits of having a DBMS. End users can range from simple viewers who pay attention to the logs or market rates to sophisticated users such as business analysts. There are two types of end users-

    - **Casual users –** These users are trained in the use of the on-line query language and access data by entering queries at terminals.
    - **Naive Users –** These users access database through application programs. They do not need to know the structure or language of database systems.

Role of Database Administrator

The main responsibilities of DBA are:

• **Makes decisions concerning the content of the database:** It is the DBA's job to decide exactly what information is to be held in the database-in other words, to identify the' entities of interest to the enterprise and to identify information to be recorded about those entities .

• **Plans storage structures and access strategies:** The DBA must also decide how the data is to be represented in the database, and must specify the representation by writing the storage structure definition (using the internal data definition language).

In addition, the associated mapping between the storage structure definition and the conceptual schema must also be specified.

• **Provides support to users:** It is the responsibility of the DBA to provide support to the users, to ensure that the data they require is available, and to write the\ necessary external schemas (using the appropriate external data definition language).

In addition, the mapping between any given eA1ernal schema and the conceptual' schema must also be specified.

• **Defines security and integrity checks:** DBA is responsible for providing the authorization and authentication checks such that no malicious users can access database and it must remain protected. DBA must also ensure the integrity of the database.

• **Interprets backup and recovery strategies:** In the event of damage to any portion\ of the database-caused by human error, say, or a failure in the hardware or supporting operating system-it is essential to be able to repair the data concerned witl1 a minimum of delay and with as little effect as possible on the rest of the system.

The DBA must define and implement an appropriate recovery strategy to recover he database from all types of failures.

• **Monitoring performance and responding to changes in requirements:** The

DBA is responsible for so organizing the system as to get the performance that is "best for the enterprise," and for making the appropriate adjustments as requirements change. Apart from these regular and official duties, the Database Administrator has to maintain confidentiality as per the ethics. Any breaching of the rules, regulations, and ethics is a serious violation of acceptable conduct and the Local Authority Freedom of Information and Protection of Privacy Act.

NOTE: A Database Administrator must possess a sound knowledge of the operating and understand the business. A graduation in computer science and an associated degree in database administration or its equivalent is desirable. He should be ready to work as part of the team and provide support **365*24*7**. In addition, he must have strong communication skills to work with management, development team, system administrator, vendor, and other service