**RDBMS**

UNIT – 1

**What is a database?**

A database is information that is set up for easy access, management and updating. Computer databases typically store aggregations of [data](https://www.techtarget.com/searchdatamanagement/definition/data) records or files that contain information, such as sales transactions, customer data, financials and product information.

Databases are used for storing, maintaining and accessing any sort of data. They collect information on people, places or things. That information is gathered in one place so that it can be observed and analyzed. Databases can be thought of as an organized collection of information.

**Overview of DATA base?**

A Database Management System (DBMS) is a well-known term in data analysis. It refers to a collection of programs that enable users to access databases and manipulate, maintain, report, and relate data. A DBMS is often used to reduce data redundancy, share data in a controlled way, and reduce data integrity problems. DBMS is not an information system but is simply software.

**Database Terminology and Concepts**

**Criteria** – the conditions that control which records to display in a query.

**Database** – a collection of information related to a particular topic or purpose.

There are two types of databases: Nonrelational and relational.

**Database management system** – a program such as Access, that stores,

retrieves, arranges, and formats information contained in a database.

**Database model** – the structure of the information stored in the database. This

model should included how each individual piece of information relates to all the

other information in the database.

Proper planning, even in the initial pencil-and-paper stage, ensures that the

database you create and maintain is efficient and provides easy access to the

information you need most. A well-designed database should eliminate the need

to enter the same data repeatedly and prevent duplication of information, thereby

maintaining the integrity of the data.

**Database modeling** – the process of strategically planning where to store each

piece of information you wish to include in your database.

**Datasheet** – a format of columns and rows displaying information.

**display formats (format)** - Specifies how data is displayed and printed. An

Access database provides standard formats for specific data types, as does an

Access project for the equivalent SQL data types. You can also create custom

formats.

**Field** – a specific item of information containing a homogenous set of values

throughout the table. Fields appear as columns in a table and as cells in a form.

field data types - a characteristic of a field that determines what kind of data it

can store. For example, a field whose data type is Text can store data consisting

of either text or number characters, but a Number field can store only numerical

data.

**Field list** – a small window that lists the fields of a selected table or data source.

**Form –** a structured document with specific areas for viewing or entering data

one record at a time. Forms can be constructed in columnar, tabular, datasheet,

or a simple justified format.

**Join line** – the line between two tables identifying the common field between

them.

**Nonrelational database** – also called a flat file, stores information in one table.

Nonrelational databases are useful for information stored in a single list, such as

a list of student names, addresses, and phone numbers.

**Object –** a component of a database, such as a table, query, form, or report.

**One-to-many relationship** – a relationship in which a record in the primary table

can be related to one or more records in the related table.

**One-to-one relationship** – a relationship between two tables in which for each

record in the first table, there is only one corresponding record in the related

table.

**Primary Key** – a field in a table whose value is uniquely identifies each record in

the table.

**Query** – a request for a particular collection of data in a database.

**Query By Example (QBE) grid** – the portion of the Query Design window used

for selecting fields, setting criteria, and setting sort order in a select query.

QBE grid rows

**ROW DESCRIPTION**

Field Displays the name of the field used in the query.

Table Displays the name of the table from which a field is selected.

Sort Determines the order in which to display the records in the recordset.

Show Determines whether the field used appears in the recordset.

Criteria Displays selective criteria used to filter the query.

Or Displays additional criteria for the query.

**Record** – an individual listing of related information consisting of a number of

related fields stored in a table. A record is also called a row in the datasheet.

**Recordset** – the set of records and fields that result from running a query.

Related table – a table with a common field that uses values stored in a primary

table.

**Relational database** – is useful for maintaining and analyzing complex

information stored in a number of tables. For example, in addition to storing

student names, you can list TAKS test scores, or demographic information

related to the students in other tables in the same database.

**Relationship** – the direct or indirect association between any two tables in a

database.

**Report** – a formatted collection of information organized to provide printed data

on a specific subject.

**Select Query** – a query that answers a question about one or more tables by

limiting the number of records and fields displayed.

**Table** – an arrangement of related information stored in columns and rows.

# What Is a Data Model?

Data models are visual representations of an enterprise’s data elements and the connections between them. By helping to define and structure data in the context of relevant business processes, models support the development of effective information systems. They enable business and technical resources to collaboratively decide how data will be stored, accessed, shared, updated and leveraged across an organization.

Data models play a key role in bringing together all segments of an enterprise – IT, business analysts, management and others – to cooperatively design information systems (and the databases they rely on).

These systems require properly defined and formatted data, and models shine a clear light on what data is required and how it must be structured to support the desired business processes.

By explicitly determining the structure of your data, these models support a variety of use cases, including database modeling, information system design, and process development in support of a consistent, clean exchange of data.

It’s also important to understand the three different types of data models. Each will serve a different purpose as you work through the data modeling process.

## What are the different types of data models?

Depending on where you are in the data modeling process, you are likely to use one of three types of data models. We typically find that enterprises benefit from all three, depending on the problems they are trying to solve. Therefore, we like to view each as a “stage” of the data modeling process, rather than a distinct “type” of data model.

### Conceptual data models

Also known as domain models, conceptual data models explore and detail your high-level, static business structures and concepts. They are most frequently used during the beginning of a new project, when high-level concepts and initial requirements are hashed out. Often, they are created as precursors or alternatives to the next stage: logical data models.

### Logical data models

After your problem domain and initial concepts become more clear through conceptual data modeling, it’s time to get more specific with a logical data model. Whether you’re looking through the lens of a single project or your entire enterprise, these models clarify the various logical entities (types or classes of data) you’ll be working with, the data attributes that define those entities, and the relationships between them.

### Physical data models

When you get to the physical data modeling stage, it’s truly time to get down to the nitty-gritty. These models are used to design the internal schema of a database. That includes all of the various tables, the columns on those tables and the relationships between them. These models will be directly translated into production database design, which will support further development of information systems. Physical data models generally are used to design three types of databases: relational for traditional operational databases, document for NoSQL and JSON databases, and dimensional for aggregation and business intelligence data stores such as data warehouses and data marts.

Ultimately, all three models can and should work independently of each other. But as your project matures, the best results will come from a natural progression through all three models. Of course, consistency must be maintained across the models on a structural level. Adjusting the table/column format on a physical model, for example, should not change the initial conceptual model in any meaningful way.

By leveraging all three models, organizations can ensure their projects do not lose sight of initial objectives – but still maintain the flexibility to address unexpected changes in requirements or parameters.

### INTRODUCTION:

A Database Management System (DBMS) is a software system that is designed to manage and organize data in a structured manner. It allows users to create, modify, and query a database, as well as manage the security and access controls for that database.

### Some key features of a DBMS include:

1. Data modeling: A DBMS provides tools for creating and modifying data models, which define the structure and relationships of the data in a database.
2. Data storage and retrieval: A DBMS is responsible for storing and retrieving data from the database, and can provide various methods for searching and querying the data.
3. Concurrency control: A DBMS provides mechanisms for controlling concurrent access to the database, to ensure that multiple users can access the data without conflicting with each other.
4. Data integrity and security: A DBMS provides tools for enforcing data integrity and security constraints, such as constraints on the values of data and access controls that restrict who can access the data.
5. Backup and recovery: A DBMS provides mechanisms for backing up and recovering the data in the event of a system failure.
6. DBMS can be classified into two types: Relational Database Management System (RDBMS) and Non-Relational Database Management System (NoSQL or Non-SQL)
7. RDBMS: Data is organized in the form of tables and each table has a set of rows and columns. The data is related to each other through primary and foreign keys.
8. NoSQL: Data is organized in the form of key-value pairs, document, graph, or column-based. These are designed to handle large-scale, high-performance scenarios.

Database is a collection of interrelated data which helps in the efficient retrieval, insertion, and deletion of data from the database and organizes the data in the form of tables, views, schemas, reports, etc. For Example, a university database organizes the data about students, faculty, admin staff, etc. which helps in the efficient retrieval, insertion, and deletion of data from it.

**There are four types of Data Languages**

1. Data Definition Language (**DDL**)
2. Data Manipulation Language(**DML**)
3. Data Control Language(**DCL**)
4. Transactional Control Language(**TCL**)

**DDL** is the short name for Data Definition Language, which deals with database schemas and descriptions, of how the data should reside in the database.

* CREATE: to create a database and its objects like (table, index, views, store procedure, function, and triggers)
* ALTER: alters the structure of the existing database
* DROP: delete objects from the database
* TRUNCATE: remove all records from a table, including all spaces allocated for the records are removed
* COMMENT: add comments to the data dictionary
* RENAME: rename an object

**DML** is the short name for Data Manipulation Language which deals with data manipulation and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE, etc., and it is used to store, modify, retrieve, delete and update data in a database.

* SELECT: retrieve data from a database
* INSERT: insert data into a table
* UPDATE: updates existing data within a table
* DELETE: Delete all records from a database table
* MERGE: UPSERT operation (insert or update)
* CALL: call a PL/SQL or Java subprogram
* EXPLAIN PLAN: interpretation of the data access path
* LOCK TABLE: concurrency Control

**DCL**is short for Data Control Language which acts as an access specifier to the database.(basically to grant and revoke permissions to users in the database

* GRANT: grant permissions to the user for running DML(SELECT, INSERT, DELETE,…) commands on the table
* REVOKE: revoke permissions to the user for running DML(SELECT, INSERT, DELETE,…) command on the specified table

**TCL** is short for Transactional Control Language which acts as an manager for all types of transactional data and all transactions.Some of the command of TCL are

* Role Back: Used to cancel  or Undo changes made in the database
* Commit: It is used to apply or save changes in the database
* Save Point: It is used to save the data on the temporary basis in the database

**Database Management System:**The software which is used to manage databases is called Database Management System (DBMS). For Example, MySQL, Oracle, etc. are popular commercial DBMS used in different applications. DBMS allows users the following tasks:

* **Data Definition:** It helps in the creation, modification, and removal of definitions that define the organization of data in the database.
* **Data Updation:** It helps in the insertion, modification, and deletion of the actual data in the database.
* **Data Retrieval:** It helps in the retrieval of data from the database which can be used by applications for various purposes.
* **User Administration:** It helps in registering and monitoring users, enforcing data security, monitoring performance, maintaining data integrity, dealing with concurrency control, and recovering information corrupted by unexpected failure.

**Paradigm Shift from File System to DBMS**

File System manages data using files on a hard disk. Users are allowed to create, delete, and update the files according to their requirements. Let us consider the example of file-based University Management System. Data of students is available to their respective Departments, Academics Section, Result Section, Accounts Section, Hostel Office, etc. Some of the data is common for all sections like Roll No, Name, Father Name, Address, and Phone number of students but some data is available to a particular section only like Hostel allotment number which is a part of the hostel office. Let us discuss the issues with this system:

* **Redundancy of data:** Data is said to be redundant if the same data is copied at many places. If a student wants to change their Phone number, he or she has to get it updated in various sections. Similarly, old records must be deleted from all sections representing that student.
* **Inconsistency of Data:**Data is said to be inconsistent if multiple copies of the same data do not match each other. If the Phone number is different in Accounts Section and Academics Section, it will be inconsistent. Inconsistency may be because of typing errors or not updating all copies of the same data.
* **Difficult Data Access:** A user should know the exact location of the file to access data, so the process is very cumbersome and tedious. If the user wants to search the student hostel allotment number of a student from 10000 unsorted students’ records, how difficult it can be.
* **Unauthorized Access:** File Systems may lead to unauthorized access to data. If a student gets access to a file having his marks, he can change it in an unauthorized way.
* **No Concurrent Access:**The access of the same data by multiple users at the same time is known as concurrency. The file system does not allow concurrency as data can be accessed by only one user at a time.
* **No Backup and Recovery:** The file system does not incorporate any backup and recovery of data if a file is lost or corrupted.

### ADVANTAGES OR DISADVANTAGES:

### Advantages of using a DBMS:

1. Data organization: A DBMS allows for the organization and storage of data in a structured manner, making it easy to retrieve and query the data as needed.
2. Data integrity: A DBMS provides mechanisms for enforcing data integrity constraints, such as constraints on the values of data and access controls that restrict who can access the data.
3. Concurrent access: A DBMS provides mechanisms for controlling concurrent access to the database, to ensure that multiple users can access the data without conflicting with each other.
4. Data security: A DBMS provides tools for managing the security of the data, such as controlling access to the data and encrypting sensitive data.
5. Backup and recovery: A DBMS provides mechanisms for backing up and recovering the data in the event of a system failure.
6. Data sharing: A DBMS allows multiple users to access and share the same data, which can be useful in a collaborative work environment.

### Disadvantages of using a DBMS:

1. Complexity: DBMS can be complex to set up and maintain, requiring specialized knowledge and skills.
2. Performance overhead: The use of a DBMS can add overhead to the performance of an application, especially in cases where high levels of concurrency are required.
3. Scalability: The use of a DBMS can limit the scalability of an application, since it requires the use of locking and other synchronization mechanisms to ensure data consistency.
4. Cost: The cost of purchasing, maintaining and upgrading a DBMS can be high, especially for large or complex systems.
5. Limited use cases: Not all use cases are suitable for a DBMS, some solutions don’t need high reliability, consistency or security and may be better served by other types of data storage.

**What are the features of DBMS?**

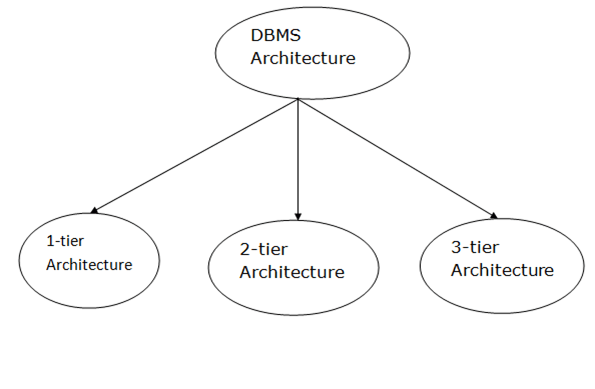
Features Of DataBase Management Systems (DBMS):

* Minimum Redundancy and Duplication. ...
* Reduced amount of space and money spent on storage. ...
* Data Organization. ...
* Customization of the Database. ...
* Data Retrieval. ...
* Usage Of Query Languages. ...
* Multi User Access. ...
* Data Integrity is Maintained.

# DBMS Architecture

* The DBMS design depends upon its architecture. 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



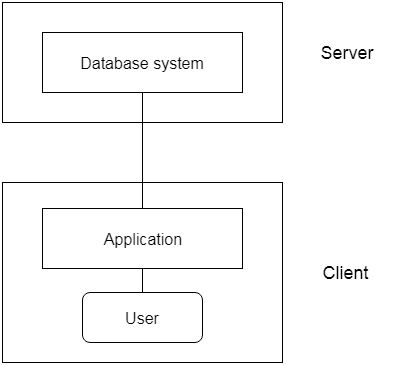
Database architecture can be seen as a single tier or multi-tier. But logically, database architecture is of two types like: **2-tier architecture** and **3-tier architecture**.

### 1-Tier Architecture

* In this architecture, the database is directly available to the user. It means the user can directly sit on the DBMS and uses it.
* 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.

### 2-Tier Architecture

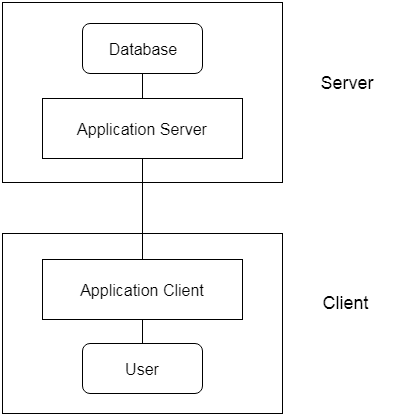
* The 2-Tier architecture is same as basic client-server. In the two-tier architecture, applications on the client end can directly communicate with the database at the server side. For this interaction, API's like: **ODBC**, **JDBC** are used.
* The user interfaces and application programs are run on the client-side.
* The server side is responsible to provide the functionalities like: query processing and transaction management.
* To communicate with the DBMS, client-side application establishes a connection with the server side.



**Fig: 2-tier Architecture**

### 3-Tier Architecture

* The 3-Tier architecture contains another layer between the client and server. In this architecture, client can't directly communicate with the server.
* The application on the client-end interacts with an application server which further communicates with the database system.
* End user has no idea about the existence of the database beyond the application server. The database also has no idea about any other user beyond the application.
* The 3-Tier architecture is used in case of large web application.



UNIT – 2

What is physical data organization?

Databases are stored physically on storage devices and organised as files and records. The overall performance of a database system is determined by the physical database organisation. Therefore, it is important that the physical organisation of data is efficiently managed.

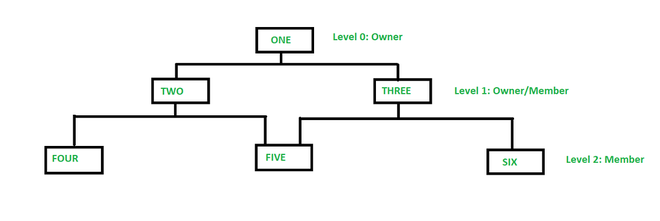
What are the 3 types of file organization?

The term "file organization" refers to the way in which data is stored in a file and, consequently, the method(s) by which it can be accessed. This COBOL system supports three file organizations: sequential, relative and indexed.

**Network Model :**  
This model was formalized by the Database Task group in the 1960s. This model is the generalization of the hierarchical model. This model can consist of multiple parent segments and these segments are grouped as levels but there exists a logical association between the segments belonging to any level. Mostly, there exists a many-to-many logical association between any of the two segments. We called **graphs**the logical associations between the segments. Therefore, this model replaces the hierarchical tree with a graph-like structure, and with that, there can more general connections among different nodes. It can have M: N relations i.e, many-to-many which allows a record to have more than one parent segment.  
Here, a relationship is called a set, and each set is made up of at least 2 types of record which are given below:

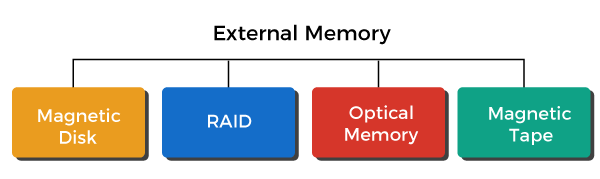
* An owner record that is the same as of parent in the hierarchical model.
* A member record that is the same as of child in the hierarchical model.

**Structure of a Network Model :**



**EXternal memory** can also be known as secondary memory or backing store. It is used to store a huge amount of data because it has a huge capacity. At present, it can measure the data in hundreds of megabytes or even in gigabytes. The important property of external memory is that whenever the computer switches off, then stored information will not be lost. The external memory can be categorized into four parts:

1. Magnetic disk
2. Raid
3. Optical memory
4. Magnetic Tape



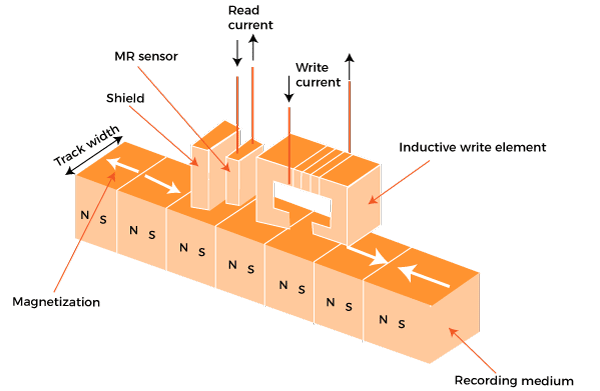
## Magnetic Disks

A disk is a type of circular platter constructed by a nonmagnetic material, which is known as a **substrate.** It is covered with a magnetic coating used to hold the information. The substrate is traditionally constructed by aluminium or aluminium alloy material. But recently, another material has been introduced, which is known as **glass substrates.** There are various benefits of glass substrates, which are described as follows:

* It has the ability to increase disk reliability by improving the uniformity of a magnetic film surface.
* It is used to reduce the errors of read-write by doing a significant reduction in overall surface defects.
* It has better stiffness, which will help to reduce disk dynamics. It has the great ability that it can withstand against shock and damage.

### Magnetic Read and Write Memory

The most important component of external memory is still magnetic disks. Many systems, such as supercomputers, personal computers, and mainframes computers, contain both removable and fixed hard disks. We can conduct a coil named as the **head** so that we can recover the data on and later and then retrieve it from the disk. A lot of systems contain two heads that are **read head** and **write head.** While the operation of reading and writing, the platter is rotating while the head is stationary.

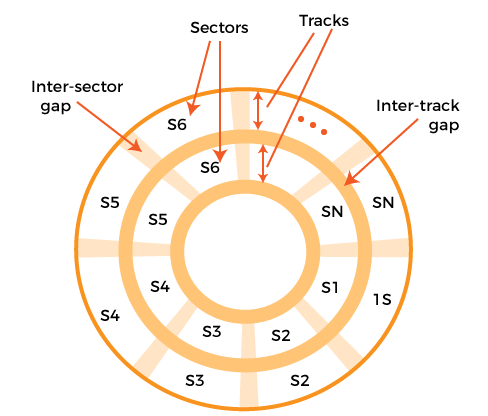


If the electricity is flowing through the **coil,** the **write mechanism** will exploit the fact that the coil will generate a magnetic field. The write head will receive the electric pulses, and the below surface will record the resulting magnetic pattern. It will be recorded into different patterns for negative and positive currents. If the electricity is flowing through the **coil,** the **read mechanism** will exploit the fact that it will generate an electric current in the coil. When the disk's surface passes under the head, it will produce a current with the same polarity as the already recorded one.

In this case, the structure of head is the same for reading and writing. Therefore, we can use the same head for both. These types of single heads can be used in older rigid disk systems and in floppy disk systems. A type of partially shielded magneto-resistive **(MR)** sensor consists in the **read head.** The electric resistance is contained in the MR material, which depends on the direction of magnetization of the medium moving under it.

### Data Organization and formatting

The head is known as a small device, which is able to read from or write to the portion of the platter rotating beneath it. The width of each track is the same as head. We have thousands of tracks per surface. The **gaps** are used to show the separation of adjacent tracks. This can prevent or minimize the error which is generated because of the interference of magnetic fields or misalignment of the head. The **sectors** are used to transfer the data from and to the disks.



The fixed-length sectors will be used in the most contemporary systems with 512 bytes, which is nearly a universal sector size. Intersector gaps separate the adjacent sectors so that we can avoid imposing unreasonable precision requirements on the systems. At the same rate, we can scan the information with the help of rotating the disk at a fixed speed, which is called **constant angular velocity (CAV).**

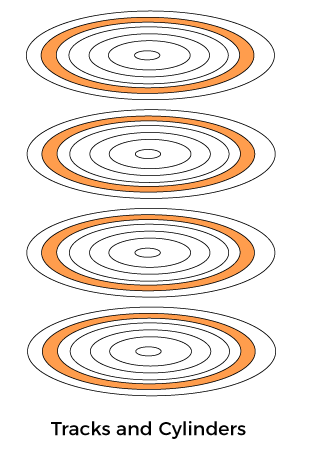
There are various things in which disks can be divided. So it can divide into a series of concentric tracks and into a number of pie-shaped sectors. The CAV has an advantage in that the tracks and sectors are able to directly address the data with the help of CAV. The CAV also has a disadvantage in that the amount of data that is stored on the short inner tracks and the long outer tracks are the same.

The modern hard disks introduce a technique to increase the density, which is called **Multiple zone recording.** Using this technique, the surface is able to divide into a number of concentric zones, which is typically equal to 16, which means 16 zones. The number of bits per track is constant within a zone. The zones which are closer to the centre have fewer amounts of bits or sectors as compared to the zones which are farther from the centre.

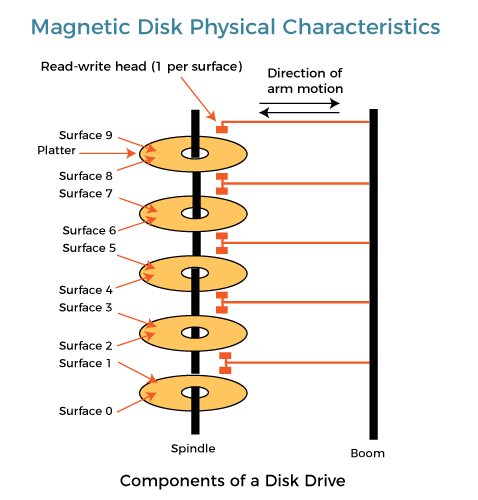
### Physical characteristics

If there is a fixed head disk, then it will contain one read-write head per track. All of these heads are mounted on a rigid arm, which has the ability to extend across all tracks. If there is a movable head disk, then it will contain only one read-write head. Here the head is also mounted on the arm. The head can position above any track. Due to this purpose, the arm can be retracted or extended.

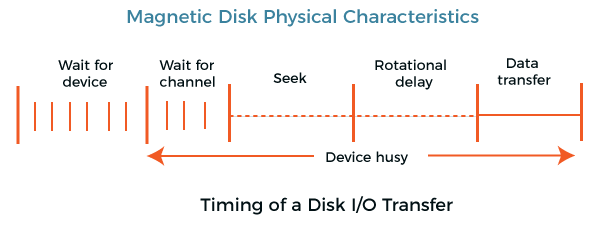
The disk drive always or permanently contains a non-removable disk. For example, in the personal computer, the **hard disk** can never be removed, or we can say that it is **a non-removable disk.** The **removable disk** is a type of disk that can be removed and replaced with other disks. Both sides of the platter contain the magnetizable coating for most of the disks, which will also be referred to as the double side. The single side disks are used in some less expensive disk systems.



A movable head is employed by the multiple platter disks with one head of read-write per platter surface. Form the centre of the disk, all the heads contain the same distance and move together because all the heads are mechanically fixed. In the platter, a set of all tracks in the same relative position will be known as a **cylinder.**



This type of mechanism is mostly used in a **floppy disk.** This type of disk is the least expensive, small, also contains a flexible platter. The sealed drive assemblies are almost free of contaminants, and it contains the Winchester's heads. IBM uses the term Winchester as a code name and it was used for the 3340 disk model prior to its announcement in IBM. The workstations and personal computers commonly contain a built-in disk, which is known as **Winchester disk.** This disk is also referred to as a **hard disk.**



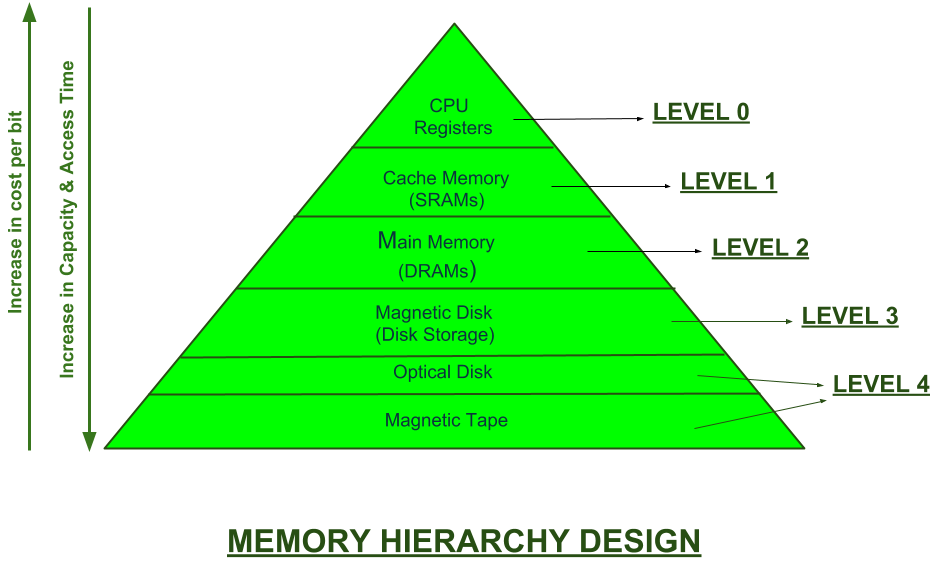
On a movable system, there will be a **seek time** which can be defined as the time taken to position the head at the track. There will also be a **rotation latency** or **rotation delay,** which can be defined as the time taken from the starting of the sector to reach the head. The time it takes to get into a position to write or read is known as access time which is equal to the sum of rotational delay and seeks time, if any.

Once the head gets its position, we are able to perform the read or write operation as the sector moves under the head. This process can be called the data transfer portion of the operation, and the time taken while transferring the data will be known as the **transfer time.**

## RAID

The RAID is also known as a **redundant array of independent disks.** It is a type of data virtualization technology, which is used to combine components of multiple disks into a logical unit so that they can improve the performance or create redundancy. If there are multiple disks/drives, it will allow the employment of various techniques such as disk mirroring, parity, and disk striping. We cannot consider RAID as a replacement for data backup. If RAID is going through the critical data, it will be backed up to a logical set of disks or other physical disks. When we make a connection with RAID, we will normally use the following terms:

**Striping:** In this, data will be split between more than one disk.

Memory Hierarchy is an enhancement to organize the memory such that it can minimize the access time. The Memory Hierarchy was developed based on a program behavior known as locality of references.The figure below clearly demonstrates the different levels of memory hierarchy :  This Memory Hierarchy Design is divided into 2 main types:

1. **External Memory or Secondary Memory –** Comprising of Magnetic Disk, Optical Disk, Magnetic Tape i.e. peripheral storage devices which are accessible by the processor via I/O Module.
2. **Internal Memory or Primary Memory –** Comprising of Main Memory, Cache Memory & CPU registers. This is directly accessible by the processor.

#### There are typically four levels of memory in a memory hierarchy:

**Registers**: Registers are small, high-speed memory units located in the CPU. They are used to store the most frequently used data and instructions. Registers have the fastest access time and the smallest storage capacity, typically ranging from 16 to 64 bits.

**Cache Memory**: Cache memory is a small, fast memory unit located close to the CPU. It stores frequently used data and instructions that have been recently accessed from the main memory. Cache memory is designed to minimize the time it takes to access data by providing the CPU with quick access to frequently used data.

**Main Memory**: Main memory, also known as RAM (Random Access Memory), is the primary memory of a computer system. It has a larger storage capacity than cache memory, but it is slower. Main memory is used to store data and instructions that are currently in use by the CPU.

**Secondary Storage**: Secondary storage, such as hard disk drives (HDD) and solid-state drives (SSD), is a non-volatile memory unit that has a larger storage capacity than main memory. It is used to store data and instructions that are not currently in use by the CPU. Secondary storage has the slowest access time and is typically the least expensive type of memory in the memory hierarchy.

We can infer the following characteristics of Memory Hierarchy Design from above figure:

1. **Capacity:** It is the global volume of information the memory can store. As we move from top to bottom in the Hierarchy, the capacity increases.
2. **Access Time:** It is the time interval between the read/write request and the availability of the data. As we move from top to bottom in the Hierarchy, the access time increases.
3. **Performance:** Earlier when the computer system was designed without Memory Hierarchy design, the speed gap increases between the CPU registers and Main Memory due to large difference in access time. This results in lower performance of the system and thus, enhancement was required. This enhancement was made in the form of Memory Hierarchy Design because of which the performance of the system increases. One of the most significant ways to increase system performance is minimizing how far down the memory hierarchy one has to go to manipulate data.
4. **Cost per bit:** As we move from bottom to top in the Hierarchy, the cost per bit increases i.e. Internal Memory is costlier than External Memory.

**B-Trees** maintain balance by ensuring that each node has a minimum number of keys, so the tree is always balanced. This balance guarantees that the time complexity for operations such as insertion, deletion, and searching is always O(log n), regardless of the initial shape of the tree.

## ****Properties of B-Tree:****

* All leaves are at the same level.
* B-Tree is defined by the term minimum degree ‘**t**‘. The value of ‘**t**‘ depends upon disk block size.
* Every node except the root must contain at least t-1 keys. The root may contain a minimum of **1** key.
* All nodes (including root) may contain at most (**2\*t – 1**) keys.
* Number of children of a node is equal to the number of keys in it plus **1**.
* All keys of a node are sorted in increasing order. The child between two keys **k1** and **k2** contains all keys in the range from **k1** and **k2**.
* B-Tree grows and shrinks from the root which is unlike Binary Search Tree. Binary Search Trees grow downward and also shrink from downward.
* Like other balanced Binary Search Trees, the time complexity to search, insert and delete is O(log n).
* Insertion of a Node in B-Tree happens only at Leaf Node.

## Variable-Length Records

Variable-length records are the records that vary in size. It requires the creation of multiple blocks of multiple sizes to store them. These variable-length records are kept in the following ways in the database system:

1. Storage of multiple record types in a file.
2. It is kept as Record types that enable repeating fields like multisets or arrays.
3. It is kept as Record types that enable variable lengths either for one field or more.

In variable-length records, there exist the following two problems:

1. Defining the way of representing a single record so as to extract the individual attributes easily.
2. Defining the way of storing variable-length records within a block so as to extract that record in a block easily.

Thus, the representation of a variable-length record can be divided into two parts:

1. An initial part of the record with fixed-length attributes such as numeric values, dates, fixed-length character attributes for storing their value.
2. The data for variable-length attributes such as varchar type is represented in the initial part of the record by (offset, length) pair. The offset refers to the place where that record begins, and length refers to the length of the variable-size attribute. Thus, the initial part stores fixed-size information about each attribute, i.e., whether it is the fixed-length or variable-length attribute.

# What is DBTG? Architecture of DBTG Model.

**DBTG** refers to the **Data Base Task Group** of the Conference on Data Systems Languages (CODASYL), the group responsible for standardization of the programming language COBOL. The DBTG final report appeared in Apri1971, it introduced a new distinct and self-contained language. The DBTG is intended to meet the requirements of many distinct programming languages, not just COBOL, the user in a DBTG system is considered to be an ordinary application programmer and the language therefore is not biased toward any single specific programming language.

It is based on network model. In addition to proposing a formal notation for networks (the Data Definition Language or DDL), the DBTG has proposed a Subschema Data Definition Language (Subschema DDL) for defining views of conceptual scheme that was itself defined using the Data Definition Language. It also proposed a Data Manipulation Language (DML) suitable for writing applications programs that manipulate the conceptual scheme or a view.

**Architecture of**DBTG**Model**

The architecture of a DBTG system is illustrated in Figure.

The architecture of DBTG model can be divided in three different levels as the architecture of a’ [database](https://ecomputernotes.com/fundamental/what-is-a-database/advantages-and-disadvantages-of-dbms) system. These are:

• Storage Schema (corresponds to Internal View of [database](https://ecomputernotes.com/fundamental/what-is-a-database/advantages-and-disadvantages-of-dbms))

• Schema (corresponds to Conceptual View of database)

• Subschema (corresponds to External View of database)

**Storage Schema**

The storage structure (Internal· View) of the database is described by the storage schema, written in a Data Storage Description Language (DSDL).

**Schema**

In DBTG, the Conceptual View is defined by the schema. The schema consists essentially of definitions of the various type of record in the database, the data-items they contain, and the sets into which they are grouped. (Here, logical record types are· referred to as record types, the fields in a logical record format are called data items)

**Subschema**

The External view (not a DBTG term) is defined by a *subschema.*A subschema consists essentially of a specification of which schema record types the user is interested in, which schema data-items he or she wishes to see in those records, and which schema relationships (sets) linking those records he or she wishes to consider. By default, all other types of record, data-item, and set are excluded.

In DBTG model, the users are application programmers, writing in an ordinary programming language, such as COBOL that has been extended to include the DBTG data manipulation language. Each application program “invokes” the corresponding sub-schema; using the COBOL Data Base Facility, for example, the programmer simply specifies the name of the required sub-schema in the Data Division of the program. This invocation provides the definition of the “user work area” (UWA) for that program. The UWA contains a distinct location for each type of record (and hence for each type (data-item) defined in the subschema. The program may refer to these data-item and record locations by the names defined in the subschema.