## **Indexed Sequential File Organization**

An indexed sequential file consists of records that can be accessed sequentially. Direct access is also possible. It consists of two parts −

* **Data File** contains records in sequential scheme.
* **Index File** contains the primary key and its address in the data file.

Following are the key attributes of sequential file organization −

* Records can be read in sequential order just like in sequential file organization.
* Records can be accessed randomly if the primary key is known. Index file is used to get the address of a record and then the record is fetched from the data file.
* Sorted index is maintained in this file system which relates the key value to the position of the record in the file.
* Alternate index can also be created to fetch the records.

**ADVANTAGES of INDEXED FILES**

1. Quite easy to process,
2. With proper selection of a key field, records in a large file can be searched and accessed in very quickly.
3. Any field of the records can be used as the key. The key field can be numerical or alphanumerical.

**DISADVANTAGES of INDEXED FILES**

1. Extra data structures have to be maintained (the COBOL run-time modules take care of these and it is **not** the programmers' concern). These extra data structures maintained on the disk can use up much disk space, especially for long key values.
2. The indexed files have to be reorganized from time time to get rid of deleted records and improve performance that gets gradually decreased with addition of n

• Heap file: which places records on disk in no particular order. 3

• Sorted sequential file: which holds records in a particular order based on the value of a specified field (i.e. attribute).

• Hashed file: which uses a hash function to decide where a record should be placed on disk. I

**Physical files** contain the actual data that is stored on the system, and a description of how data is to be presented to or received from a program. They contain only one record format, and one or more members. Records in database files can be externally or program-described.

A physical file can have a keyed sequence access path. This means that data is presented to a program in a sequence based on one or more key fields in the file.

**Physical files** contain the actual data that is stored on the system, and a description of how data is to be presented to or received from a program. They contain only one record format, and one or more members.

**Logical files** do not contain data. They contain a description of records found in one or more physical files. A logical file is a view or representation of one or more physical files. Logical files that contain more than one format are referred to as **multi-format** logical files.

|  |  |
| --- | --- |
| **Physical File** | **Logical File** |
| It occupies the portion of memory. It contains the original data. | It does not occupy memory space. It does not contain data. |
| A physical file contains one record format. | It can contain upto 32 record formats. |
| It can exist without logical file. | It cannot exist without physical file. |
| If there is a logical file for physical file, the physical file cannot be deleted until and unless we delete the logical file. | If there is a logical file for a physical file, the logical file can be deleted without deleting the physical file. |

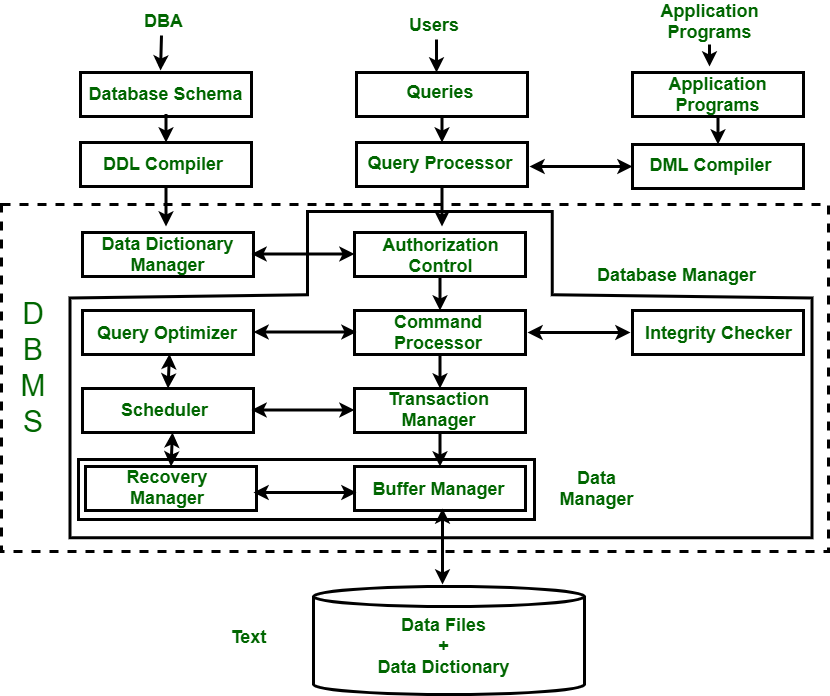
# **Structure of Database Management System**

It is the job of the database system to translate updates and queries written in a nonprocedural language, at the logical level, into an efficient sequence of operations at the physical level.

[Database Management System (DBMS)](https://www.geeksforgeeks.org/introduction-of-dbms-database-management-system-set-1/) is a software that allows access to data stored in a database and provides an easy and effective method of –

* Defining the information.
* Storing the information.
* Manipulating the information.
* Protecting the information from system crashes or data theft.
* Differentiating access permissions for different users.

The database system is divided into three components: Query Processor, Storage Manager, and Disk Storage. These are explained as following below.



***Figure –****Structure of DBMS*

**1. Query Processor :**   
It interprets the requests (queries) received from end user via an application program into instructions. It also executes the user request which is received from the DML compiler.   
Query Processor contains the following components – 

* **DML Compiler –**   
  It processes the DML statements into low level instruction (machine language), so that they can be executed.
* **DDL Interpreter –**   
  It processes the DDL statements into a set of table containing meta data (data about data).
* **Embedded DML Pre-compiler –**   
  It processes DML statements embedded in an application program into procedural calls.
* **Query Optimizer –**   
  It executes the instruction generated by DML Compiler.

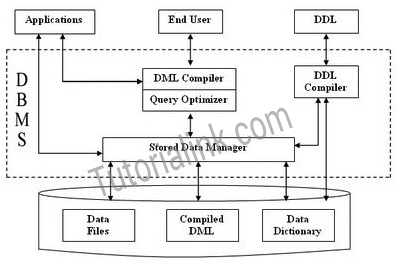
**2. Storage Manager :**   
Storage Manager is a program that provides an interface between the data stored in the database and the queries received. It is also known as Database Control System. It maintains the consistency and integrity of the database by applying the constraints and executes the [DCL](https://www.geeksforgeeks.org/sql-ddl-dql-dml-dcl-tcl-commands/) statements. It is responsible for updating, storing, deleting, and retrieving data in the database.   
It contains the following components – 

* **Authorization Manager –**   
  It ensures role-based access control, i.e,. checks whether the particular person is privileged to perform the requested operation or not.
* **Integrity Manager –**   
  It checks the integrity constraints when the database is modified.
* **Transaction Manager –**   
  It controls concurrent access by performing the operations in a scheduled way that it receives the transaction. Thus, it ensures that the database remains in the consistent state before and after the execution of a transaction.
* **File Manager –**   
  It manages the file space and the data structure used to represent information in the database.
* **Buffer Manager –**   
  It is responsible for cache memory and the transfer of data between the secondary storage and main memory.

**3. Disk Storage :**   
It contains the following components – 

* **Data Files –**   
  It stores the data.
* **Data Dictionary –**   
  It contains the information about the structure of any database object. It is the repository of information that governs the metadata.
* **Indices –**   
  It provides faster retrieval of data item.

## **Structure of DBMS:**

* DBMS (Database Management System) acts as an interface between the user and the database. The user requests the DBMS to perform various operations such as insert, delete, update and retrieval on the database.
* The components of DBMS perform these requested operations on the database and provide necessary data to the users.  
  
* The various components of DBMS are described below:

## **Components of a DBMS**

The components of DBMS can be divided into two parts:

### *Function and Services of DBMS*

1. DDL Compiler:
   * Data Description Language compiler processes schema definitions specified in the DDL.
   * It includes metadata information such as the name of the files, data items, storage details of each file, mapping information and constraints etc.
2. DML Compiler and Query optimizer:
   * The DML commands such as insert, update, delete, retrieve from the application program are sent to the DML compiler for compilation into object code for database access.
   * The object code is then optimized in the best way to execute a query by the query optimizer and then send to the data manager.
3. Data Manager:
   * The Data Manager is the central software component of the DBMS also knows as Database Control System.
   * The Main Functions Of Data Manager Are:
     1. Convert operations in user's Queries coming from the application programs or combination of DML Compiler and Query optimizer which is known as Query Processor from user's logical view to physical file system.
     2. Controls DBMS information access that is stored on disk.
     3. It also controls handling buffers in main memory.
     4. It also enforces constraints to maintain consistency and integrity of the data.
     5. It also synchronizes the simultaneous operations performed by the concurrent users.
     6. It also controls the backup and recovery operations.
4. Data Dictionary:
   * Data Dictionary, which stores metadata about the database, in particular the schema of the database.
   * names of the tables, names of attributes of each table, length of attributes, and number of rows in each table.
   * Detailed information on physical database design such as storage structure, access paths, files and record sizes.
   * Usage statistics such as frequency of query and transactions.
   * Data dictionary is used to actually control the data integrity, database operation and accuracy. It may be used as a important part of the DBMS
5. Data Files:
   * Which store the database itself.
6. Compiled DML:
   * The DML complier converts the high level Queries into low level file access commands known as compiled DML.
7. End Users:
   * The second class of users then is end user, who interacts with system from online workstation or terminals.
   * Use the interface provided as an integral part of the database system software.
   * User can request, in form of query, to access database either directly by using particular language, such as SQL, or by using some pre-developed application interface.
   * Such request are sent to query evaluation engine via DML pre-compiler and DML compiler
   * The query evaluation engine accepts the query and analyses it.
   * It finds the suitable way to execute the compiled SQL statements of the query.
   * Finally, the compiled SQL statements are executed to perform the specified operation
   * Query Processor Units:

Interprets DDL statements into a set of tables containing metadata.  
Translates DML statements into low level instructions that the query evaluation engine understands.  
Converts DML statements embedded in an application program into procedure calls int he host language.  
Executes low level instructions generated by DML compiler.

* + 1. DDL Interpreter
    2. DML Compiler
    3. Embedded DML Pre-compiler
    4. Query Evalution Engine
  + Storage Manager Units

Checks the authority of users to access data.

Checks for the satisfaction of the integrity constraints.

Preserves atomicity and controls concurrency.

Manages allocation of splace on disk.

#### **Fetches data from disk storage to memory for being used.**

* + 1. Authorization Manager
    2. Integrity Manager
    3. Transaction Manager
    4. File manager
    5. Buffer Manager
  + Functions of DBMS:
    1. DBMS free the programmers from the need to worry about the organization and location of the data i.e. it shields the users from complex hardware level details.
    2. DBMS can organize process and present data elements from the database. This capability enables decision makers to search and query database contents in order to extract answers that are not available in regular Reports.
    3. Programming is speeded up because programmer can concentrate on logic of the application.
    4. It includes special user friendly query languages which are easy to understand by non programming users of the system.
  + The service provided by the DBMS includes :-
    1. Authorization services like log on to the DBMS start the database stop the Database etc.
    2. Transaction supports like Recovery, Rollback etc,
    3. Import and Export of Data.
    4. Maintaining data dictionary
    5. User's Monitoring

log on to the DBMS :::::The **log** is a sequence of **log** records, recording all the update activities in the database. In a stable storage, **logs** for each transaction are maintained. Any operation which is performed on the database is recorded is on the **log**

Data is a collection of a distinct small unit of information. It can be used in a variety of forms like text, numbers, media, bytes, etc. it can be stored in pieces of paper or electronic memory, etc.

1. Query Processor :

(a) DML Compiler  
(b) Embedded DML pre-compiler  
(c) DDL Interpreter  
(d) Query Evaluation Engine

2. Storage Manager :

(a) Authorization and Integrity Manager  
(b) Transaction Manager  
(c) File Manager  
(d) Buffer Manager

3. Data Structure :

(a) Data Files  
(b) Data Dictionary  
(c) Indices  
(d) Statistical Data

#### 1. Query Processor Components :

**• DML Pre-compiler :** It translates DML statements in a query language into low level instructions that query evaluation engine understands. It also attempts to transform user's request into an equivalent but more efficient form.  
**• Embedded DML Pre-compiler :** It converts DML statements embedded in an application program to normal procedure calls in the host language. The Pre-compiler must interact with the DML compiler to generate the appropriate code.  
**• DDL Interpreter :** It interprets the DDL statements and records them in a set of tables containing meta data or data dictionary.  
**• Query Evaluation Engine :** It executes low-level instructions generated by the DML compiler.

2. Storage Manager Components :

They provide the interface between the low-level data stored in the database and application programs and queries submitted to the system.  
**• Authorization and Integrity Manager :** It tests for the satisfaction of integrity constraints checks the authority of users to access data.  
**• Transaction Manager :** It ensures that the database remains in a consistent state despite the system failures and that concurrent transaction execution proceeds without conflicting.  
**• File Manager :** It manages the allocation of space on disk storage and the data structures used to represent information stored on disk.  
**• Buffer Manager :** It is responsible for fetching data from disk storage into main memory and deciding what data to cache in memory.

3. Data Structures :

Following data structures are required as a part of the physical system implementation.  
**• Data Files :** It stores the database.  
**• Data Dictionary :** It stores meta data (data about data) about the structure of the database.  
**• Indices :** Provide fast access to data items that hold particular values.  
**• Statistical Data :** It stores statistical information about the data in the database. This information is used by query processor to select efficient ways to execute query.

* **Applications: –** It can be considered as a user-friendly web page where the user enters the requests. Here he simply enters the details that he needs and presses buttons to get the data.
* **End User: –** They are the real users of the database. They can be developers, designers, administrators, or the actual users of the database.
* [**DDL**](https://www.tutorialcup.com/dbms/data-definition-language.htm)**: –** Data Definition Language (DDL) is a query fired to create database, schema, tables, mappings, etc in the database. These are the commands used to create objects like tables, indexes in the database for the first time. In other words, they create the structure of the database.
* [**DDL**](https://www.tutorialcup.com/dbms/data-definition-language.htm)**Compiler: –** This part of the database is responsible for processing the DDL commands. That means this compiler actually breaks down the command into machine-understandable codes. It is also responsible for storing the metadata information like table name, space used by it, number of columns in it, mapping information, etc.
* [**DML**](https://www.tutorialcup.com/dbms/data-manipulation-language.htm)**Compiler: –** When the user inserts, deletes, updates or retrieves the record from the database, he will be sending requests which he understands by pressing some buttons. But for the database to work/understand the request, it should be broken down to object code. This is done by this compiler. One can imagine this as when a person is asked some question, how this is broken down into waves to reach the brain!
* [**Query Optimizer**](https://www.tutorialcup.com/dbms/query-optimization.htm)**: –** When a user fires some requests, he is least bothered how it will be fired on the database. He is not all aware of the database or its way of performance. But whatever be the request, it should be efficient enough to fetch, insert, update, or delete the data from the database. The query optimizer decides the best way to execute the user request which is received from the DML compiler. It is similar to selecting the best nerve to carry the waves to the brain!
* **Stored Data Manager: –** This is also known as Database Control System. It is one of the main central systems of the database. It is responsible for various tasks
  + It converts the requests received from query optimizer to machine-understandable form.  It makes actual requests inside the database. It is like fetching the exact part of the brain to answer.
  + It helps to maintain consistency and integrity by applying the constraints.  That means it does not allow inserting/updating / deleting any data if it has child entry. Similarly, it does not allow entering any duplicate value into database tables.
  + It controls concurrent access. If there are multiple users accessing the database at the same time, it makes sure, all of them see correct data. It guarantees that there is no data loss or data mismatch happens between the transactions of multiple users.
  + It helps to back up the database and recovers data whenever required. Since it is a huge database and when there is any unexpected exploit of the [transaction](https://www.tutorialcup.com/dbms/transaction.htm), and reverting the changes is not easy. It maintains the backup of all data so that it can be recovered.
* **Data Files: –** It has the real data stored in it. It can be stored as magnetic tapes, magnetic disks, or optical disks.
* **Compiled DML: –** Some of the processed DML statements ([insert](https://www.tutorialcup.com/sql/insert-query.htm), [update](https://www.tutorialcup.com/sql/update-query.htm), [delete](https://www.tutorialcup.com/sql/delete-query.htm)) are stored in it so that if there are similar requests, it will be re-used.
* [**Data Dictionary**](https://www.tutorialcup.com/dbms/data-dictionary.htm)**: –** It contains all the information about the database. As the name suggests, it is the dictionary of all the data items. It contains a description of all the [tables](https://www.tutorialcup.com/dbms/tables.htm), [view](https://www.tutorialcup.com/dbms/views.htm), [materialized views](https://www.tutorialcup.com/dbms/materialized-views-dbms.htm), [constraints](https://www.tutorialcup.com/dbms/constraints.htm), [indexes](https://www.tutorialcup.com/sql/indexes.htm), [triggers](https://www.tutorialcup.com/dbms/triggers.htm), etc.

**[READ  Tables in DBMS](https://www.tutorialcup.com/dbms/tables.htm" \t "_blank)**

Structure of Database Management System (DBMS)