File Access Methods

Various files acccess methods are as follows :

(1) Sequential : It is the simplest method to store and retrieve data from a file. Sequential organization simply means storing and sorting in physical form on tape or disk. In a sequential organization a record can be added only at the end of the file. That is, in a sequential file, records are stored one after the other without concern for the actual value of the data in the records. It is not possible to insert a record in the middle of the file without re-writing the file. In a sequential file update, transaction records are in the same sequence as in the master file. Records from both files are matched, one record at a time, resulting in an updated master file. It is a characteristic of sequential files that all records are stored by position, the first one is at the first position, the second one occupies the second position and third is at third and so on, there are no addressed or location assignments in sequential.

To read a sequential file, the system always starts at the beginning of the file. If the record sought is somewhere in the file, the system reads its ways upto it, one record at a time. For example, if a particular record happens to be the fifteenth one in a file, the system starts from the first one and reads ahead one record at a time until the fifteenth one is reached. It cannot jump directly to the fifteenth one in a sequential file without starting from the beginning.

(2) Random or Direct : For a proposed system, when the sequential files are assumed as a disadvantages, another file organisation called direct organization is used. As with a sequential file, each record in a direct file must contain a key field. However, the records need not appear on the file in key field sequence. In addition any record stored in a direct file can be accessed if its location or address is known. All the previous records needs not to be accessed. The problem, however, is to determine how to store the data records so that, given the key field of the desired record, its storage location on the file can be determined. In other words, if the program knows the record key, it can determine the location address of a record and retrieve it independently of any other records in the file. It would be ideal if the key field could also be the location of the record on the file. This method is known as direct addressing method. This is quite simple method but the requirements of this method often prevents its use. Because of many other factors, this method could not become popular. Hence, it is rarely used.

Therefore, before a direct organized file can be created, a formula or method must be devised to convert the key field value for a record to the address or location of the record on the file. This formula or method is generally known as an algorithm. Otherwise known as hashing addressing.

(3) Indexed : The third way of accessing records in the system is through an index. The basic form of an index includes a record key and the storage address for a record.

To find a record, when the storage address is unknown it is necessary to scan the records. However, if an index is used, the search will be faster since it takes less time to search an index than an entire file of data.

Index file offers the simplicity of sequential file while at the same time offering a capability for direct access. The records must be initially stored on the file in sequential order according to a key field. In addition, as the records are being recorded on the file, one or more indices are established by the system to associate the key field value with the storage location of the record on the file. These indices are then used by the system to allow a record to be directly accessed. To find a specific record when the file is

stored under an indexed organization, the index is searched first to find the key of the record wanted. When it is found, the corresponding storage address is noted and then the program can access the record directly. This method uses a sequential scan of index, followed by direct access to the appropriate record. The index helps to speed up the search compared with a sequential file, but it is slower as compared it to the direct addressing.

✔ Database Design

Database design is the process of producing a detailed data model of a database. The logical data model contains all the needed logical and physical design choices and physical storage parameters that are used to generate a design in a Data Definition Language, which can then be used to create a database. A fully attribute data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data.

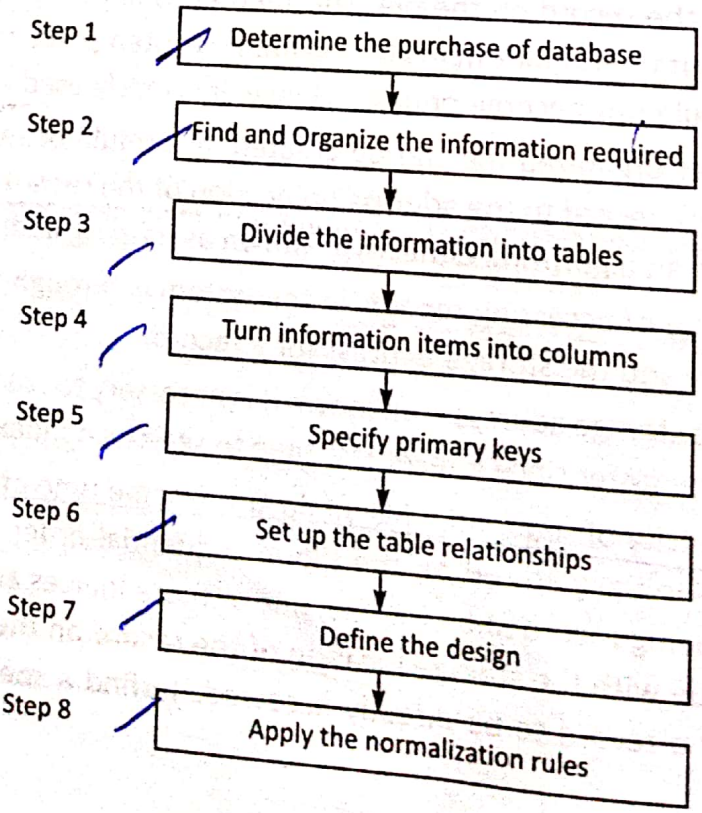
The process of database design generally consists of a number of steps which will be carried out by a database designer. Usually the designer must—

(a) Determine the relationships between the different data elements.

(b) Superimpose a logical structure upon data on the basis of their relationships.

In a majority of cases, a person who is doing the design of a database is a person with expertise in the area of database design, rather than expertise in the domain from which the data to be stored is darawn *e.g.,* financial information, biological information etc. Therefore, the data to be stored in the database must be determined incorporation with a person who does have expertise in that domain, and who is aware of what data must be stored with in the system.

✔ Process of Database Design



✔ Objectives of Database

1. Controlled Redundancy : Redundant data occupies space and therefore, is wasteful. If version of the same data are in different phases of updating, the system often gives conflicting information. A unique aspect of database design is storing data only once, which controls redundancy and improves system performance.

2. Ease of Learning and Use : A major feature of a user friendly database package is how easy it is to learn and use. Related to this point is that a database can be modified without interfering with established ways of using the data.

3. Data Independence : An important database objective is changing hardware and storage procedures or adding new data without having to rewrite application programs. The database should be 'tunable' to improve performance without rewriting programs.

4. More Information at Low Cost : Using, storing and modifying data at low cost are important. Although hardware prices are falling, software and programming costs are on the rise. This means that programming and software enhancements should be kept simple and easy to update.

5. Accuracy and Integrity : The accuracy of a database ensures that data quality and content remains constant. Integrity controls detect data inaccuracies where they occur.

6. Recovery from Failure : With multiuser access to a database, the system must recover quickly after it is down with no loss of transactions. This objective also helps maintain data accuracy and integrity.

7. Privacy and Security : For data to remain private, security measures must be taken to prevent unauthorized access. Database security means that data are protected from various forms of destruction, users must be positively identified and their actions monitered.

People Who Deals with Databases

1. Database Manager :

(a) The 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.

(b) Databases typically requires lots of storage space. This must be stored on disks. Data is moved between disk and main memory as needed.

(c) The goal of the database system is to simplify and facilitates access to data. Performance is important. Views provide simplification.

(d) So, a database manager is responsible for—

—Interation with the file manager.

—Integrity enforcement.

—Security enforcement.

—Backup and recovery

—Concurrency Control

(e) Some small database systems may miss some of these features, resulting in simpler database managers (for example, noconcurrency is required on a PC (running MS-DOS). These features are necessary on larger systems.

2. Database Administrator : The database administrator is a person having central control over data and programs accessing that data.

Duties of Database Administrator include.

(a) **Scheme Definition :** The creation of the original database scheme. This involves writing a set of definitions in a DDL, compiled by the DDL compiler into a set of tables stored in the data dictionary.

(b) **Storage a Structure and Access Method Definition :** Writing a set of definition translated by the data storage and definition language compiler.

(c) **Scheme and Physical Organization Modification :** Writing a set of definition used by the DDL compiler to generate modifications to appropriate internal system tables (*e.g.*, data dictionary). This is done rarely, but sometimes the data base scheme or physical organization must be modified.

(d) **Integrity Constraint Specification :** These are consulted by the database manager module whenever updates occur.

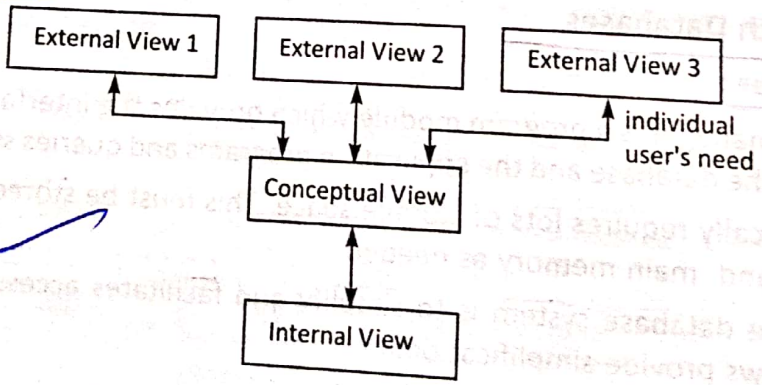
3. Database Users :

(a) **Application Programmers :** Application programmers are computer professionals interacting with the system through DML calls embedded in a program written in a host language (*e.g.*, C, PL/I, Pascal).

(b) **Specialized Users :** Specialized users are sophisticated users writing special database application programs. These may be CAD system, expert system, knowledge based systems.

(c) **Naive Users :** Naive users are unsophisticated users who interact with the system by using permanent application programs.

Database System Architecture



(1) Physical Level (Internal Level) : This is the lowest level of data which describes how the data are actually stored and organized on the medium. It describes the complete details of data storage and access paths for the database. In other words, it has the definition of the records stored and methods of representing the data fields.

(2) Conceptual Level (Global Level) : It deals with data actually stored in the database. It describes the structure of the whole database for users in terms of relatively small structures. There is one conceptual scheme per database, which also contains the method for driving the objects in the conceptual view from the objects in the internal view. It describes all the records and relationships included in the conceptual view.