CHAPTER 1

INTRODUCTION TO DATABASE MANAGEMENT SYSTEM

1.1 What is Database Management System?

A Database Management System (DBMS) is essentially a collection of interrelated data, and a set of programs to access this data.

- Collection of data is usually called the database
- These collected data kept in a electronic filing cabinet.
- Database systems are designed to maintain large volume of data.

1.2 Management of Data Involves

- Defining the structures for storage of data.
- Providing the mechanisms for the manipulation of the data
- Providing the security of the data against unauthorized access.

1.3 Primary Objective of DBMS

To provide an environment that is convenient and efficient to use in retrieving information from and storing information into the database.

1.4 Users of DBMS

- **The Application Programmer:** Writes application programs that use the database. These programs operates on the data in the database. These operations include retrieving information inserting data, deleting or changing data.
- **The End User:** Interacts with system either by involving an application programs or by writing their queries in a data based query language. The database query language allows the end user to perform all the basic operations (retrieval, deletion, insertion, updation) on the data.
- **The Database Administrator (DBA):The DBA** has to coordinate the function of collecting information about the data to be stored, designing and maintaining the database and its security. The database provide the right information at the right time to the authorized people. These responsibilities belong to the DBA and his or her staff.

1.5 Facilities Provided to DBMS Users:

- Adding empty files to the database.
- Inserting new data into the existing files.
- Retrieving data from the files,
- Updating data in the files.
- Removing existing files from the database.

1.6 Why Use a Database System?

The conventional data processing approach is to develop a program (or many programs) for each application. This results:-

- One or more data files for each application.
- Some of the data may be common between files
- One application require the file to be arrange on a particular field, while another application may require the file to be arranged on another field.
- The same data may be required by two application, the data will have to be stored in two different places because each application depends on the way that data is stored.

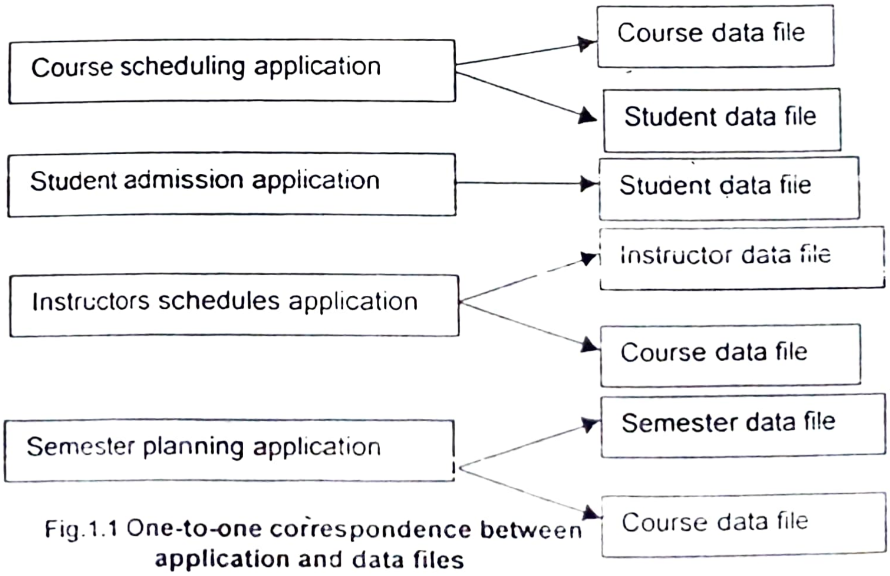
Course scheduling application
Course data file
Student data file
Student admission application
Student data file
Instructors schedules application
Instructor data file
Course data file
Semester planning application
Semester data file
Course data file

Fig.1.1 One-to-one correspondence between application and data files

1.7 Drawbacks of the Conventional Data File Processing Environment

Data Redundancy : Since data is required by multiple applications, it is stored in multiple data file. In most cases, there is a repetition of data. This is referred to as data redundancy.

Data integrity Problems: Data redundancy is one reason for problems of data integrity. Since the same data is stored in different places, it is inevitable that some inconsistency will creep. For Example: If any instructor of industrial psychology begins to take a course in child psychology, this change needs to be reflected in more than one place. If the change is not made in all the places, the university will have different information in different places about the same instructor.

Data Availability Constraints: When data is scattered in different files, the availability of information from a different files is difficult to some extent.

1.8 Benefits of the Database approach

A major advantage the database approach has over the conventional approach is that a database system provides centralized control of data.

Redundancy can be controlled:

- Centralized control of data by the DBA avoids unnecessary duplication of data
- Reduces the total amount of data storage required.
- Eliminates the extra processing to trace the required data in a large mass of data.

Inconsistency can be avoided: The DBMS guarantee that database is never inconsistent, by ensuring that a change made to any entry, automatically applies to the other entries as well. This process is known as propagating updates.

The data can be shared: A databases allows the sharing of data under its control by any number of application programs or users.

Standards can be enforced: Since there is centralized control of data, the database administration can ensure that standards are maintained in the representation of the stored data format. This is particularly useful for data interchange or migration of data between two systems.

Security restrictions can be applied: The DBMS has to guarantee that only authorized persons have access to the database.

- The DBMS defines the security checks to be carried out.
- Different checks can be applied to different operations on the same data.

For Example: A person may have the access right to query on a file, but may not have the right to delete or update that file.

Integrity can be maintained: Data integrity means that the data contained in the database is both accurate and consistent. Inconsistency between two entries can lead to integrity problems. Even if there is no redundancy - the data can still be inconsistent. For example: in a semester, when the maximum number of course one can enroll is 7. Another example could be that of a student enrolling in a course that is not being offered that semester. Above problems can be sorted out in a DBMS by establishing certain integrity checks to be carried out whenever any update operation is done.

Data Independence: In non-database systems, the requirements of the application dictate the way in which the data is stored, and the access techniques. Besides, the knowledge of the organization of the data and the access techniques are built into the logic and code of the application. These systems are data dependent. For example, suppose the university has an application that process the student file. For performance reasons, the file is indexed on the roll number. The application would be aware of the existing index, and internal structure of the application would be built around this knowledge. Now consider that for some reason the file is to be indexed on the registration date. In this case it is impossible to change the structure of the stored data without affecting the application too. Such an application is a data dependent one.

It is desirable to have data independent application. Suppose two applications X and Y need to access the same file. However, both applications require a particular filed to be stored in different format. Application X requires the field to be stored in decimal format, while application Y requires it to be stored in binary format. This would pose a problem in the old systems. In DBMS, different may exist in the way that data is actually stored and the way that is seen and used by a given application.

To conform the changing requirements of the enterprise, the DBA may need to change the storage structure or access techniques. The DBA should be able to do this without giving to modify the existing application. If application are data dependent, the programmer effort, that could otherwise be available for the creation of new application, would be necessary to modify existing application to match the changes made.