

DBMS note 1

Manik Chand Patnaik (M.Tech.)
Lecturer, (C.S.E), Roland Institute of Technology.
Sunday, January 07 – 2007.

Advantages of a DBMS over traditional File Processing Systems

Why would you choose a database system instead of using a file processing system?

Answer

A collection of data which is known as database, needs to be placed in permanent storage, either as data-files or stored inside a database management system. There are several limitations of storing data in data-files. These limitations along-with the benefits of using a database is explained here:-

1. Difficulty

File processing system is a customised solution, so we need to define the data-structures and write the appropriate programs to access, manipulate and maintain the data stored in the files.

DBMS provides a generic solution so you just need to logically define the structure of data to store. All the programs to access, manipulate and maintenance of data are provided by the DBMS vendor.

2. Lack of Independence of data

File processing system provides a specific solution for a particular problem for which it is implemented. For example: a student information system can not cater for the library even if the same students go to the library to borrow books. Even if there are two exactly similar student information systems exist in two different colleges both would have data-structures and student records incompatible with each other because the data in data-files and the programs acting on them are very closely tied with each other. You cannot even change the order of individual values inside the record. If the roll number comes before the registration number it has to be that way all through the application. You cannot keep the registration number before the roll number in any case.

In DBMS there is no connection of the user or programmer with the storage of data. Both the programmer and the end-user need not worry about how the data is stored or accessed. They have to use the libraries/interfaces provided by the DBMS vendor to access the data. So data is portable and a single source of data can serve several users around a variety of applications.

3. Inconsistency in data

In a file processing system the data-files are not protected in any way so they can be changed in any manner without any restrictions. At most the user account may be password protected. Any one able to use the program to handle the data-files can change anything on the data file so there is every threat of data becoming inconsistent due to improper handling.

The example of a non existing student with borrowed books from library is the classic example of inconsistency in data in a file-processing system.

4. Duplication of data

This is another problem closely related with the previously discussed point. In a file processing system the process of getting inter-related data is difficult and so mostly data is duplicated across data-files to ease the data access. Duplicated data is a cause of inconsistency because data updated at a single place is not available instantly to all the places where data is duplicated.

DBMS effectively addresses this problem by allowing tables to have references with other tables so as to minimize or complete eliminate duplication of data. In this way the inconsistency problem is eliminated.

5. Absence of constraints

In a file processing system all the programs accessing data are independent so how to control the access of data is a big issue. There is absolutely no constraints at the file level. Any constraints required need be written as a program and it is going to be very complex.

DBMS have the ability not only to enforce constraints but also change them without affecting the program in any way.

6. Concurrency

When many users want to access a single record at a time this results in concurrency problem. Only the last successful insertion is stored.

A DBMS can effectively solve this issue because updates in their true sense is supported. As an update can be relative to the value stored, no illogical data will get inserted because of concurrency problem.

7. Atomicity

This is a non-existent concept in file processing systems. Partial data manipulation can result in illogical data being inserted into the data-files. For example transfer of funds from one account to another account. If the program fails after withdrawal and before deposit to another account, this would result in illogical transaction.

DBMS provides for transaction support so multiple operations on database can be taken as a single unit. If all operations succeed then the whole unit become successful in updating the data else all operations will fail (even those operations which can succeed are made to roll-back their operations as if nothing has happened).

8. Security

See the point 3 and 5. The security in a file processing system is limited to setting up of passwords for the user. A part of data cannot be exposed while restrictions are provided for other parts of data. This results in whole-scale exposure of the data to the user to manipulate and this is a bad practice as any user having access to the file processing system can modify any data.

DBMS allows for securing data with authorization by a chain of authority. A super user provides authorization on specific tables to other users and the users having the authorization can further grant the authority to other users. Reverse is also true. If the super user revokes the authorization, all subsequent authorizations become nullified. This allows

for greater control over data and its use.

9. Flexibility

See point 2. As the file processing system is a customised solution, it cannot be adopted for another problem or can be marketed. For each and every query even you have to write a program and it is a lot of work and there is very less re-use of code as code is tightly bound with data.

10. Hardware Constraints

Generally data tends to grow with use. The file processing system suffers from one serious bottleneck, that is availability of system resources. A 32 bit system gives 4Gb of addressing space so taking the OS and application footprint into consideration, a big data file of ~3.5Gb won't load up in the memory even if you have the quantity of RAM. (Currently very high end systems ship with such a large quantity of RAM, generally systems shipped today have 512-1024Mb of RAM).

A DBMS frees the user from the storage and retrieval details, so the user is not bothered about storage space. Though there is a limit to the size of the database any commercial DBMS can handle, generally it is quite large.

All these benefits of using a DBMS results in a reduced application development time because only application-specific code needs to be written. There is independence of data and most importantly the data is logical, secure and protected from unauthorized access.