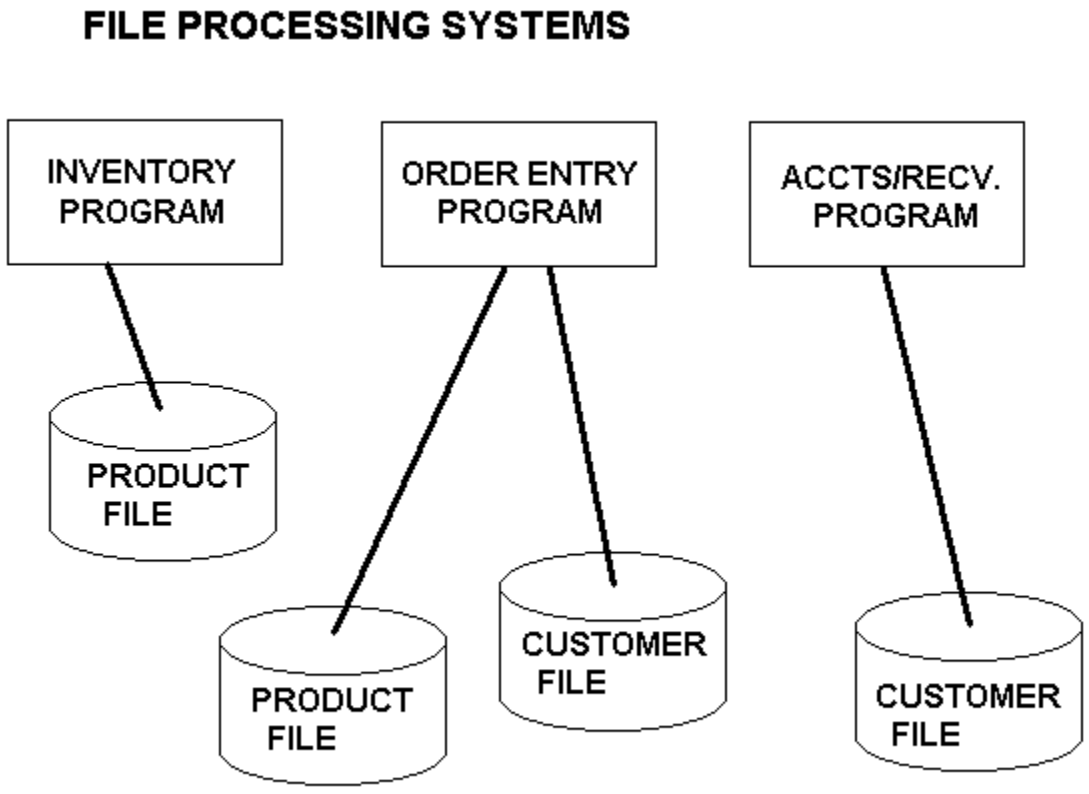
Disadvantages of File processing systems

Hi! The journey has started, today you will learn about the various flaws in the conventional file processing systems. The actual reason for the introduction of DBMS.

Data are stored in files and database in all information systems. Files are collections of similar records. Data storage is build around the corresponding application that uses the files.

File Processing Systems

- Where data are stored to individual files is a very old, but often used approach to system development.
- Each program (system) often had its own unique set of files.



Diagrammatic representation of conventional file systems

Users of file processing systems are almost always at the mercy of the Information Systems department to write programs that manipulate stored data and produce needed information such as printed reports and screen displays.

What is a file, then?

A File is a collection of data about a single entity.

Files are typically designed to meet needs of a particular department or user group.

Files are also typically designed to be part of a particular computer application

Advantages:

- are relatively easy to design and implement since they are normally based on a single application or information system.
- The processing speed is faster than other ways of storing data.

Disadvantages:

- Program-data dependence.
- Duplication of data.
- Limited data sharing.
- Lengthy program and system development time.
- Excessive program maintenance when the system changed.
- Duplication of data items in multiple files. Duplication can affect on input, maintenance, storage and possibly data integrity problems.
- Inflexibility and non-scalability. Since the conventional files are designed to support single application, the original file structure cannot support the new requirements.

Today, the trend is in favor of replacing file-based systems and applications with database systems and applications.

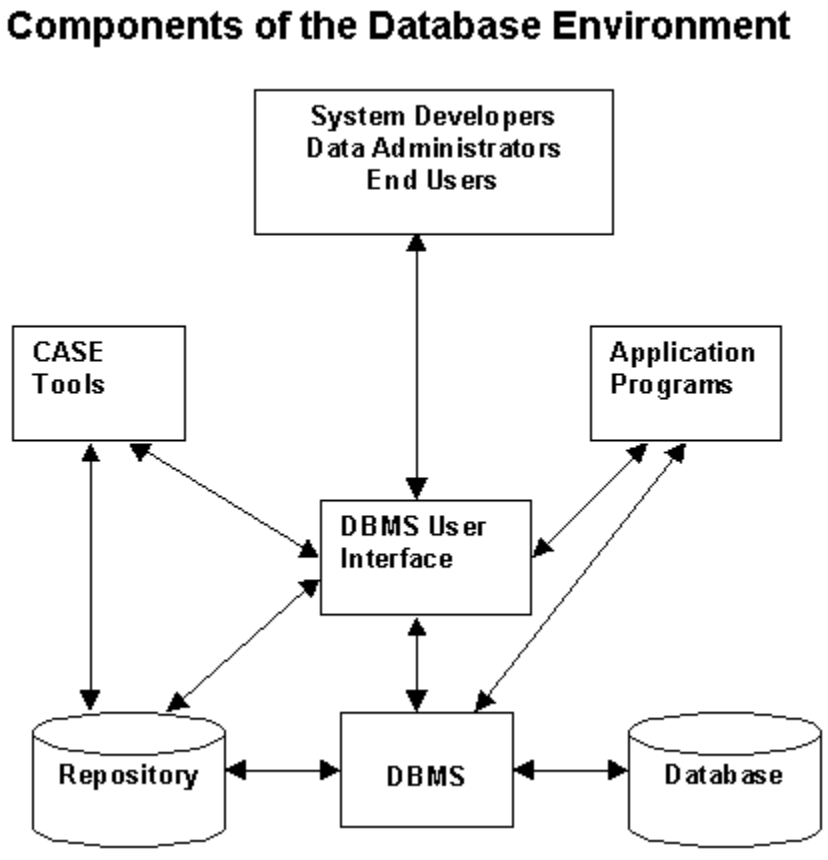
Database Approach

A database is more than a file – it contains information about more than one entity and information about relationships among the entities.

Data about a single entity (e.g., Product, Customer, Customer Order, Department) are each stored to a "table" in the database.

Databases are designed to meet the needs of multiple users and to be used in multiple applications.

One significant development in the more user-friendly relational DBMS products is that users can sometimes get their own answers from the stored data by learning to use data querying methods.



Advantages:

- Program-data independence.
- Minimal data redundancy, improved data consistency, enforcement of standards, improved data quality.
- Improved data sharing, improved data accessibility and responsiveness.
- Increased productivity of application development.
- Reduced program maintenance Data can be shared by many applications and systems.
- Data are stored in flexible formats. Data independence. If the data are well designed, the user can access different combinations of same data for query and report purposes.
- Reduce redundancy.

Database Application Size

Personal Computer Database:

- Supports a single-user.
- Stand-alone.
- May purchase such an application from a vendor.
- Can't integrate data with other applications.

Workgroup Database:

- Example would be a small team using the same set of applications such as in a physician's office.
- Includes numerous workstations and a single server typically.

Department Database:

- A functional area (such as production) in a firm.

- Same hardware/software as Workgroup database, but is specialized for the department.

Enterprise Database:

- Databases or set of databases to serve an entire organization.
- May be distributed over several different physical locations.
- Requires organizational standards for system development and maintenance.

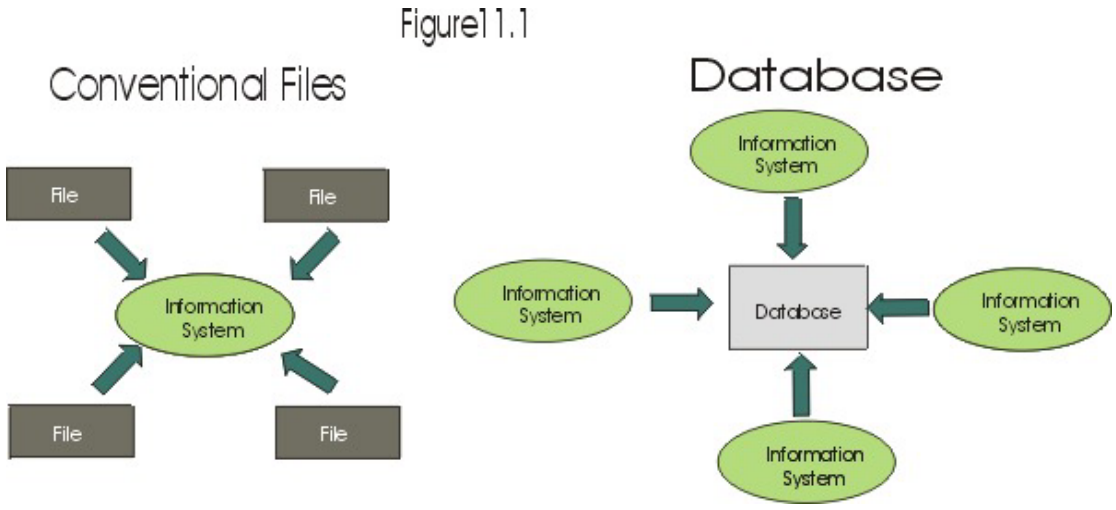


Figure 11.1 Conventional files versus the database

Database design in perspective

The focus is on the data from the perspective of the system designer. The output of database design is database schema. Data models were developed during the definition phase. Then, data structures supporting database technology are produced during the database design.

Advantages of using a DBMS

There are three main features of a database management system that make it attractive to use a DBMS in preference to more conventional software. These features are *centralized data management*, *data independence*, and *systems integration*.

In a database system, the data is managed by the DBMS and all access to the data is through the DBMS providing a key to effective data processing. This contrasts with conventional data processing systems where each application program has direct access to the data it reads or manipulates. In a conventional DP system, an organization is likely to have several files of related data that are processed by several different application programs.

In the conventional data processing application programs, the programs usually are based on a considerable knowledge of data structure and format. In such environment any change of data structure or format would require appropriate changes to the application programs. These changes could be as small as the following:

1. Coding of some field is changed. For example, a null value that was coded as -1 is now coded as -9999.
2. A new field is added to the records.
3. The length of one of the fields is changed. For example, the maximum number of digits in a telephone number field or a postcode field needs to be changed.
4. The field on which the file is sorted is changed.

If some major changes were to be made to the data, the application programs may need to be rewritten. In a database system, the database management system provides the interface between the application programs and the data. When changes are made to the data representation, the metadata maintained by the DBMS is changed but the DBMS continues to provide data to application programs in the previously used way. The DBMS handles the task of transformation of data wherever necessary.

This independence between the programs and the data is called *data independence*. Data independence is important because every time some change needs to be made to the data structure, the programs that were being used before the change would continue to work. To provide a high degree of data independence, a DBMS must include a sophisticated metadata management system.

In DBMS, all files are integrated into one system thus reducing redundancies and making data management more efficient. In addition, DBMS provides centralized control of the operational data. Some of the advantages of data independence, integration and centralized control are:

Redundancies and inconsistencies can be reduced

In conventional data systems, an organization often builds a collection of application programs often created by different programmers and requiring different components of the operational data of the organization. The data in conventional data systems is often not centralized. Some applications may require data to be combined from several systems. These several systems could well have data that is redundant as well as inconsistent (that is, different copies of the same data may have different values). Data inconsistencies are often encountered in everyday life. For example, we have all come across situations when a new address is communicated to an organization that we deal with (e.g. a bank, or Telecom, or a gas company), we find that some of the communications from that organization are received at the new address while others continue to be mailed to the old address. Combining all the data in a database would involve reduction in redundancy as well as inconsistency. It also is likely to reduce the costs for collection, storage and updating of data.

Better service to the Users

A DBMS is often used to provide better service to the users. In conventional systems, availability of information is often poor since it normally is difficult to obtain information that the existing systems were not designed for. Once several conventional systems are

combined to form one centralized data base, the availability of information and its up-to-datedness is likely to improve since the data can now be shared and the DBMS makes it easy to respond to unforeseen information requests.

Centralizing the data in a database also often means that users can obtain new and combined information that would have been impossible to obtain otherwise. Also, use of a DBMS should allow users that do not know programming to interact with the data more easily.

The ability to quickly obtain new and combined information is becoming increasingly important in an environment where various levels of governments are requiring organizations to provide more and more information about their activities. An organization running a conventional data processing system would require new programs to be written (or the information compiled manually) to meet every new demand.

Flexibility of the system is improved

Changes are often necessary to the contents of data stored in any system. These changes are more easily made in a database than in a conventional system in that these changes do not need to have any impact on application programs.

Cost of developing and maintaining systems is lower

As noted earlier, it is much easier to respond to unforeseen requests when the data is centralized in a database than when it is stored in conventional file systems. Although the initial cost of setting up of a database can be large, one normally expects the overall cost of setting up a database and developing and maintaining application programs to be lower than for similar service using conventional systems since the productivity of programmers can be substantially higher in using non-procedural languages that have been developed with modern DBMS than using procedural languages.

Standards can be enforced

Since all access to the database must be through the DBMS, standards are easier to enforce. Standards may relate to the naming of the data, the format of the data, the structure of the data etc.

Security can be improved

In conventional systems, applications are developed in an ad hoc manner. Often different system of an organization would access different components of the operational data. In such an environment, enforcing security can be quite difficult.

Setting up of a database makes it easier to enforce security restrictions since the data is now centralized. It is easier to control that has access to what parts of the database. However, setting up a database can also make it easier for a determined person to breach security. We will discuss this in the next section.

Integrity can be improved

Since the data of the organization using a database approach is centralized and would be used by a number of users at a time, it is essential to enforce integrity controls.

Integrity may be compromised in many ways. For example, someone may make a mistake in data input and the salary of a full-time employee may be input as $4,000 rather than $40,000. A student may be shown to have borrowed books but has no enrolment. Salary of a staff member in one department may be coming out of the budget of another department.

If a number of users are allowed to update the same data item at the same time, there is a possibility that the result of the updates is not quite what was intended. For example, in an airline DBMS we could have a situation where the number of bookings made is larger than the capacity of the aircraft that is to be used for the flight. Controls therefore must be introduced to prevent such errors to occur because of concurrent updating activities.

However, since all data is stored only once, it is often easier to maintain integrity than in conventional systems.

Enterprise requirements can be identified

All enterprises have sections and departments and each of these units often consider the work of their unit as the most important and therefore consider their needs as the most important. Once a database has been set up with □entralized control, it will be necessary to identify enterprise requirements and to balance the needs of competing units. It may become necessary to ignore some requests for information if they conflict with higher priority needs of the enterprise.

Data model must be developed

Perhaps the most important advantage of setting up a database system is the requirement that an overall data model for the enterprise be built. In conventional systems, it is more likely that files will be designed as needs of particular applications demand. The overall view is often not considered. Building an overall view of the enterprise data, although often an expensive exercise is usually very cost-effective in the long term.

DBMS ARCHITECTURE

We now discuss a conceptual framework for a DBMS. Several different frameworks have been suggested over the last several years. For example, a framework may be developed based on the functions that the various components of a DBMS must provide to its users. It may also be based on different views of data that are possible within a DBMS. We consider the latter approach.

A commonly used views of data approach is the three-level architecture suggested by ANSI/SPARC (American National Standards Institute/Standards Planning and Requirements Committee). ANSI/SPARC produced an interim report in 1972 followed by a final report in 1977. The reports proposed an architectural framework for databases.

Under this approach, a database is considered as containing data about an *enterprise*. The three levels of the architecture are three different views of the data:

1. *External* – individual user view
2. *Conceptual* – community user view
3. *Internal* – physical or storage view

The three level database architecture allows a clear separation of the information meaning (conceptual view) from the external data representation and from the physical data structure layout. A database system that is able to separate the three different views of data is likely to be flexible and adaptable. This flexibility and adaptability is data independence that we have discussed earlier.

We now briefly discuss the three different views.

The external level is the view that the individual user of the database has. This view is often a restricted view of the database and the same database may provide a number of different views for different classes of users. In general, the end users and even the applications programmers are only interested in a subset of the database. For example, a department head may only be interested in the departmental finances and student enrolments but not the library information. The librarian would not be expected to have any interest in the information about academic staff. The payroll office would have no interest in student enrolments.

The conceptual view is the information model of the enterprise and contains the view of the whole enterprise without any concern for the physical implementation. This view is normally more stable than the other two views. In a database, it may be desirable to change the internal view to improve performance while there has been no change in the conceptual view of the database. The conceptual view is the overall community view of the database and it includes all the information that is going to be represented in the database. The conceptual view is defined by the conceptual schema which includes definitions of each of the various types of data.

The internal view is the view about the actual physical storage of data. It tells us what data is stored in the database and how. At least the following aspects are considered at this level:

1. Storage allocation e.g. B-trees, hashing etc.
2. Access paths e.g. specification of primary and secondary keys, indexes and pointers and sequencing.
3. Miscellaneous e.g. data compression and encryption techniques, optimization of the internal structures.

Efficiency considerations are the most important at this level and the data structures are chosen to provide an efficient database. The internal view does not deal with the physical devices directly. Instead it views a physical device as a collection of physical pages and allocates space in terms of logical pages.

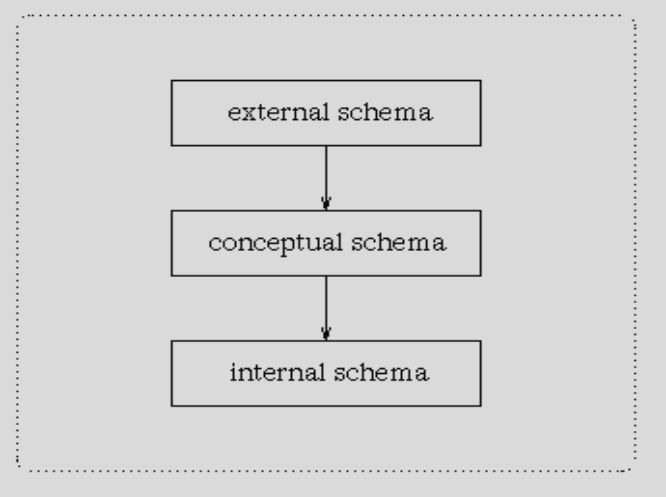
The separation of the conceptual view from the internal view enables us to provide a logical description of the database without the need to specify physical structures. This is often called *physical data independence*. Separating the external views from the conceptual view enables us to change the conceptual view without affecting the external views. This separation is sometimes called *logical data independence*.

Assuming the three level view of the database, a number of mappings are needed to enable the users working with one of the external views. For example, the payroll office may have an external view of the database that consists of the following information only:

1. Staff number, name and address.
2. Staff tax information e.g. number of dependents.
3. Staff bank information where salary is deposited.
4. Staff employment status, salary level, leaves information etc.

The conceptual view of the database may contain academic staff, general staff, casual staff etc. A mapping will need to be created where all the staff in the different categories

are combined into one category for the payroll office. The conceptual view would include information about each staff's position, the date employment started, full-time or part-time, etc. This will need to be mapped to the salary level for the salary office. Also, if there is some change in the conceptual view, the external view can stay the same if the mapping is changed.



Summary

Now we are coming to the end of this lecture, but before parting we will revise the things. Files are collections of similar records. Data storage is build around the corresponding application that uses the files. Duplication of data items in multiple files. Duplication can affect on input, maintenance, storage and possibly data integrity problems. Inflexibility and non-scalability. Since the conventional files are designed to support single application, the original file structure cannot support the new requirements.

Today, the trend is in favor of replacing file-based systems and applications with database systems and applications.

Questions

1. Explain a file with its advantages and disadvantages
2. Define centralized data management, data independence and systems integration
3. Explain DBMS architecture
4. Explain the 3 different views or architecture of the data

Selected Bibliography