Introduction to Evolution of Databases

By Farrokh Alemi, PhD

This material is based in part on "Data structures: Data Bases and Data Base Management Systems" in Austin C. Information Systems for Health Services Management.

Today computers deal with a great deal of data. They can do so because they are faster and better than before. But there is another reason why we can search and find what we want on our computers despite large amount of data being stored in them. Imagine if during a search we had to examine each data item to see if it is what we want. Then it would take a fraction of second to go through one item and over billions of item it may take hours and days. How is it that computers can search the entire world wide web of information and come back in a fraction of a second to provide us with the result?

The answer is simple. Our ability to use data is improved when we introduce more structure in how data is stored. Then it is no longer necessary to go through all of the data to reach a specific item. The more the structure, the easier to go through large amount of the data. Think of a structure as an address to a datum. The more precise the address we have, the easier it is to find a datum even when we do not knock at each neighbor's house. Likewise structure of the data tells the computer where to look for data and avoid unnecessary effort.

Ever wondered why an Internet search can return the addresses of home pages that contain a word so quickly. The answer is because the addresses are stored under a list of words arranged alphabetically. When you enter a word, the computer jumps to the appropriate row of the Table without any processing of information in other rows. Thus, it is possible to peruse very large databases in a fraction of a second.

In this section we introduce several approaches to putting structure into data.   The three main types of databases are flat, relational and hierarchical models.   We end the section with discussion of data-less databases, where only the structure of data is kept and no data.

Flat Databases

Flat data models have the least amount of structure. They typically take the form of one large Table, where the first row is the list of the variables and subsequent rows are data. Each case has a row of data. Many statisticians use flat models.

Here is an example of a flat file for students in a class:

Student ID	Name	Midterm grade	Final grade	Address	Zip code	...	...	..
4561	Ali Safaie	B	A	1311 Manor Park	22101	...	...	..
7878	Mike Smith	C	B	1619 Ozkan Street	44115	...	...	..
8954	Mike Smith Jr.	A	C	2121 Euclid 563	22101	...	...	..

Table 1: Example of a flat file

Advantages

Most software include free access to flat data files. For a small number of cases, flat databases do a reasonably fast job.

Disadvantages

Flat databases waste computer storage by requiring it to keep information on items that logically cannot be available. For example, if we are keeping information on zip codes, flat files require us to enter a missing information for zip codes in foreign countries. Hierarchical and structural databases avoid this problem by defining different tables for classes of countries in which zip code is not available. Thus flat files keep large sparse data full of missing information. When the size of database is large, search through the data takes a long time.

Flat databases are not conducive to complicated search queries that divide the database further. For example, in a flat file about students it would be difficult to find all students that live in a certain zip code and have received grade of C. Such a query will require repeated pass through the data. First pass may identify all students who live in a zip code, second pass may identify all students who have grade of C and the third pass may find students in both groups. Such multiple passes through data are inefficient and take long time periods. Since every simple if statement (if zip code is equal to 22101 then ...) in a computer program takes a fraction of a second, efficient search methods are important for large databases.

Relational Databases

In a relational data base, one stores a record with related fields as data. All items on the same record are said to be related to each other. Thus, one may store the information that patients have medical records in the following fashion:

- Data item: Patient ID number
- Data item: Medical record number
- Data item: First name

This information in essence says that the patient ID number, name and medical record number belong to the same person. To effectively store both information items and the relationships among these items, relational data are kept in table formats. The first row of the table shows the name of the variables and subsequent rows are data.   All items in the same row usually belong to the same case and are related.   One column of the table is treated as the key to the table. Numbers or characters in this column corresponds to items in other Tables. Thus it is possible to move from one table to another.

In a relational database, tables do not need to be of the same size but they all need to have a key column that connects them to each other and that uniquely defines the elements inside the table.

Example

Here is an example of a Table of grades for the example introduced under flat files:

Student ID Key	**Name**	**Mid-term**	**Final**
4561	Ali Gadiri	B	A
7878	Mike Smith	C	B
8954	Mike Smith Jr.	A	C

Table 2: Table for "Students grades"

This is an example of an additional table for contact information:

Student ID Key	**Address**	**Zip**
8954	2121 Euclid 563	22101

4561	1311 Manor Park	22101
7878	1619 Ozkan Street	44115

Table 3: "Students' contact information"

When a query is made, the relational database searches through its tables to find the answer. Often the answer involves information pooled together from different tables. For example, a query for names of people with grade of A, can be answered from the Table of Grades. The query for grades of people in zip code area 22101 must be answered from both Tables.

Can you try to answer this question: What is the average final grade of the persons in zip code area 22101? Did you notice how you move from one table to another to get your answer? You use the key shared between the two tables, to find the relevant information in the other table.

Advantages

A relational database makes life difficult for the designer of the database but easy for the user of the database. It is more difficult for the designer because many possible relationships must be anticipated. It is easier for the user because data can be examined from many different perspectives.

In addition, if Tables are appropriately defined it is possible to avoid having to enter missing information for variables that are not logically possible. This helps data entry and data processing speed.

A relational database is also easy to modify because adding new concepts involves adding new Tables, not altering old ones.

Hierarchical Databases

Hierarchical database models are one type of relational data models in which the relationship between any two adjacent item is similar to a father-child relationship. Hierarchical database models resemble a-cyclical tree structures (directional tree structures that do not include circular paths).

Example

The file directory in your desk top is an example of a hierarchical database system. A folder may contains other folders which may contain other files, which contain data.

Advantages

In hierarchical models, children inherit the relationships and characteristics of their fathers. An operation on the father affects the children; if you tell the computer to do an operation on a folder, the operation affects all the folders and files that it contains. For example, if you delete the top folder, you would delete all of the folders and files it contains. This feature saves time for the person maintaining the files.

Sometimes, when the real world relationship that is being modeled is not hierarchical, it is difficult to fit relationships into a hierarchical database model. In these circumstances, an operation on the father should not be executed on the children.

Distributed Databases

Most databases physically reside in one place. They may be backed up to another place but the elements of the database are not maintained in different places. In a distributed database, data are kept in different settings and on different computers. One or several central computers maintain indexes to where the data are.   Using the address of the data then computers can communicate and find the information needed.

Distributed databases need not only addresses for where the data are but also need an audit trail of who has updated data or retrieved it. Audit data are needed in order to pinpoint errors in the system and in order to understand where confidentiality of the system breaks down. When a computer requests data from another, an audit trail is created by storing who sent data where and when.   When this computer passes the data to another, the information needs to be updated in the original computer. As the number of computers receiving the data increases the task of auditing becomes more difficult. At some point (at least theoretically), it is necessary to cutoff the original computer from being updated about where the information has traveled.

Example

A good example of a distributed database is the World Wide Web pages.   These pages of data are kept on a different computers, often referred to as Web servers. The address of each file is the location address you enter when you want to see the page. This location address is an index to the Web pages. Centralized computers keep the beginning of these addresses, called domains. Subsequent detailed addresses are kept at the Web servers.

When searching a distributed database, two steps must occur. First a program, sometimes called crawler, must index the content of the databases, then another program, often called a search engine, would search the index for your request.   When a match is found, the index is used to find the address of the information.   This address is provided to the engine that assembles a list for you. When you click on the items identified by the search engine, you use the address to retrieve the data items it has found.

Advantages and disadvantages

The information system manager must make decisions on whether to use centralized or de-centralized databases based on a number of issues including the following:

- Decentralized databases exchange files and therefore may exchange corrupted files or viruses that may affect the entire system. Security of these databases are difficult to maintain. Though the section on Data-less information systems discusses how these systems can be used to create more confidential information exchanges.
- In decentralized databases the type of data to be exchanged, the process of addressing the data, and the protocol for updating the data must be agreed upon ahead of time and plans must be in place for updating the process.
- In centralized databases lack of backup or inadequate back up may result to complete loss of data while in distributed data systems data loss is limited to nodes affected.
- De-centralized databases are more flexible and allow different units to update and maintain their own data. At the same time, this increased flexibility runs the risk that some units may institute changes that may make them less accessible by others.
- When different groups and systems are involved in maintaining the data, then there is more opportunity for differences in quality of data to emerge. A decentralized database needs to have procedures for determining the quality (accuracy, recency, reliability, etc.) of the data.

Historically, the design of Information Systems, especially national information systems, has focused on creation of super-databases that contain all of the transactions about the patient, from which the clinicians can retrieve portion of the database to which they have legitimate access. These super-databases are expensive to construct and maintain, as they require 24 hour, seven days a week operation, a tight security, extensive and ongoing coordination of data elements across existing modern and legacy systems, as well as a cadre of trained personnel to maintain the database.

Furthermore, the design of national registries and databases create perplexing problems with patients’ privacy and confidentiality of medical records. Around the world, local and State laws require that patients’ consent be sought before disclosing the content of a patient records to others. In the United States, the Congress is debating Federal endorsement of already existing State laws that limit the nature of what is an appropriate consent. To be meaningful, patients’ consent should be for release to a specific organization, for a specific purpose and governed by a specific time frame. National registries make a mockery of what is a meaningful consent. The patient is asked to consent to release of information to an intermediary who may disclose it to others, who may in turn release it to others. The patient never knows how and when the information is actually used; therefore the consent provided is not an informed release of medical information. For example, data from immunization registries may show in court custody cases without the current custodian's consent. Few parents who agree to release information to a national or regional database have in mind that the data can be used at some time in the future to argue that they are not fit parents.

There are also other problems with the current process of obtaining consent. In many States, open-ended consent is illegal. Local and State law require that consent be time dependent. Thus, national and regional registries face a practical operational problem. They can warehouse information about the patient for a specific time period, but when that time period is expired, they have no legal authority to continue to disclose the information to others.

Finally, the problem of managing the data after release creates large and perhaps unrealistic problems. For example, to understand who is responsible for an illegal release of medical information, it is important to trace back the information to the source. Given that a series of agencies may have collected and released information to each other before the illegal release occurred, one has to track many pieces of information. Thus the person who creates a large medical database must not only create the database but also maintain an equally large database about patterns of release of information.

The problems with cost of operation and difficulties with maintaining privacy and confidentiality of the patients has encouraged us to suggest an alternative approach.

Components of a Data-less Information System

A data-less information system contains no centralized data and relies on distributed databases. The Data-less Information System relies on rapid communication to construct the data at the point of need for the data. A data-less information system also does not require any additional hardware; instead it relies on the hardware of existing clinics and health institutions. There are three components to a data-less information system:

- Decoder. This software resides in the database of participating clinics and medical centers. It reads and analyzes the structure of the clinic's medical records. It uses known characteristics of the data, national and international standard of data, and clinic's supplied format of the data to decipher the format and structure of the data. When it is finished with understanding the structure of the data, it sends a message to other units regarding its willingness to participate in Data-less Information Systems.

- Communicator. This software resides in the database of participating clinics and medical centers. It verifies that the patients' consent has been obtained, the nature of data to be collected, and the patient's and the system's suggestion about where to look for the data. This software contacts all sites that have a reporting decoder and collect the necessary information, including the possibility of different possible interpretation of the same data at different sites. The data collected by the software is for one time use and would be erased after the use.

- Analysis. This software component resides in the databases of participating clinics and medical centers. It allows a information system user to specify how the data should be presented after collection, whether sampling strategies should be used to collect more than one case for policy analysis, and how should the data be summarized and analyzed.

Using these three components, any clinic or government agency who has patients' consent can rapidly pool and analyze information about the patient from other machines. When a patient shows at a clinic, consent is obtained, then the networked is pooled, and the necessary data are put together. When the data is reviewed, the information is erased, reducing any possibility of accidental disclosure or any need to manage data post release.

When data is needed about a group of patients -- as when a policy analysis needs to be done -- it is difficult to obtain consent from the patients. Furthermore, group reports may be used in ways that could harm organizations that collect and store these data. For example, organizations may not wish to report where their customers and patients come from in fear that their competitors may use this information to gain advantage over them. Data-less Information Systems should be organized in a fashion that prohibits any reports that singles out an organization without the consent of that organization.

When group reports are obtained across organizations and across a large number of patients, the database can remove patient and organizational information before disclosing the data. For example, if a policy maker wants to know how many people in his/her region are immunized on time, he can ask the system to sample new births from the birth registry, request data on the sample of cases identified, pool data on the cases, remove the identifiers, calculate the summary statistics needed and present the results for a one time use of the policy maker. The data is provided without consent if and only if the sample includes multiple organizations and patients.

The key advantages of the Data-less Information Systems are:

- The system is substantially less expensive than centralized registries as it requires no new equipment and little personnel.
- The use of the system does not require vague and time-independent patients' consents. The patient gives consents for the network to pool the data for a specific use and for the immediate time period.
- The system does not require duplication of data in different databases. It uses what is available. National governments can then focus their attention on improving what is available and the quality of data kept by different clinics and medical centers.

An example

When using the World Wide Web, the information is cached, meaning that the information is downloaded for temporary use and discarded afterwards. A browser has many characteristics of a data-less information system. It relies on large distributed data, it relies on communication devices to collect the data and it discards the data after use. If a browser is further developed to add some of the features described above, then it will become a true data-less information system. It should for example have decoders for accessing legacy systems. It should not allow copying of information. It should allow group information without consent but remove organization specific or patient specific identifiers. And, it should allow more analysis of the data. Under these circumstances, existing browsers become what we have in mind when we talk of data-less information systems.

A data-less information system can radically improve the operation of United States immunization registries. Both the Center for Disease Control and Robert Wood Johnson Foundation have funded the operation of a number of immunization registries around the United States. These registries face a number of problems and limitations that are typical of traditional registries. A key problem is how to obtain patients' consent. Another problem is how to maintain these operations after grant funding for them expires. The proposed data-less information registries solves both the cost and the privacy concerns. One national or multiple regional networks can be created to pool data on demand.