DATABASES

A database is a logically related group of data elements that may be structured in a number of ways to meet the multiple processing and retrieval needs of organisations and individuals.

Why a Database?

The various aspects of data processing in organisations are traditionally grouped by department and application.

This approach can create a number of problems:

Data Redundancy (Integration)
The same basic data fields included in many different files.

File Updating (Integrity)
When changes occur in a data item, every file which contains that filed must be updated to reflect the change. Confusion occurs when this does not happen.

Lack of Data/Program Independence.
The programs used with file-oriented applications usually contain 'Picture', 'Format' or 'Data' statements. When a format changes the application program must be changed. A significant revision to a program may require a re-structuring of the data file.

Solution

A centralised and integrated shared data file that emphasises the independence of programs and data. Transactions introduced to the system once, and all database records are effectively updated.

Database Management System (DBMS)

A Database Management System can organise, process and present selected data elements from the database. The DBMS will manage the stored data items and assemble the needed items from the common database.

The part of the database management system (DBMS) that handles the organisation, storage and retrieval of the data is the ‘database manager’. A database manager may work with traditional programming languages, such as COBOL and BASIC, or may work only with its proprietary programming language. The terms database manager and database management system are used interchangeably.

A database manager links two or more files together and is the foundation for developing routine business systems. Contrast with file manager, which works with only one file at a time and is typically used interactively on a personal computer for managing personal, independent files, such as name and address lists.

Data Security and Integrity

Data security prevents unauthorised users from viewing or updating the database. Using passwords, users are allowed access to the entire database or subsets of the database, called subschemas (pronounced "sub-skeema"). For example, an employee database can contain all the data about an individual employee, but one group of users may be authorised to view only payroll data, while others are allowed access to only work history and medical data.

The DBMS can maintain the integrity of the database by not allowing more than one user to update the same record at the same time. The DBMS can keep duplicate records out of the database; for example, no two customers with the same customer numbers (key fields) can be entered into the database.

The Schema

The schema is a description of the organisation of the database. Defined in terms of syntax and semantics, there are commonly conceptual (or logical) schemas, physical schemas, external schemas and internal schemas. An example of a physical would be the field names in a relational table.

Interactive Query

Query languages and report writers allow users to interactively interrogate the database and analyse its data, see below for SQL.

Interactive Data Entry and Updating

If the DBMS provides a way to interactively enter and update the database, as well as interrogate it, this capability allows for managing personal databases. However, it does not automatically leave an audit trail of actions and does not provide the kinds of controls necessary in a multi-user organisation. These controls are only available when a set of application programs are customised for each data entry and updating function.

This is one of the most common misconceptions about database management systems that are used in personal computers. Thoroughly comprehensive and sophisticated business systems can be developed in dBASE, Paradox and other DBMSs; however, they are created by experienced programmers using the DBMS's own programming language.

Database Design

A business information system is made up of subjects (customers, employees, vendors, etc.) and activities (orders, payments, purchases, etc.). Database design is the process of deciding how to organize this data into record types and how the record types will relate to each other. The DBMS that is chosen is the one that can mirror the organisation's data structure properly and process the transaction volume efficiently.

Organisations may use one kind of DBMS for daily transaction processing and then move the detail

onto another computer that uses another DBMS better suited for random inquiries and analysis. Overall systems design decisions are performed by data administrators and systems analysts. Detailed database design is performed by database administrators.

DBMS structures

List Database
Hierarchial Database
Network Database
Relational Database

List, hierarchical, network and relational databases are the four most commonly methods of organizing data. A database management system may provide one, two, three or all four methods. Inverted files using secondary indexing, partial inversion, full inversion and bit map indexing are other methods also used.

The structure that best suits an organisation depends on the organisation's particular requirements, the volume of daily transactions and the estimated number of ad hoc inquiries that will be made.

List Database

Records are linked by use of pointers. A pointer is a data item in one record that identifies the storage location of another logically related record. This can also be known as a flat database and can be equated to the manual card index system or an electronic equivalent to a spreadsheet.

Hierarchical or Tree Database

Hierarchical databases link records together like an organisation chart, each data item has a superior-subordinate relationship. Each node can own one or more elements, but each element can only have a single owner. The nodes or owner are not connected. For example, orders are owned by only customer. Hierarchical structures were widely used in the first mainframe database management systems. However, due to their restrictions, they often cannot be used to relate structures that exist in the real world.

Network Database

A network database structure similar to that of a tree database but each element may have several owners and each node may own several elements. In the order processing example, orders are owned by both customers and products, since that's the way they relate in the business.

Relational Database

Relational databases don't link records together physically, records are grouped in tables with each table having a relationship with the other via a linking field. Therefore, the design of the records must provide a common field, such as account number, to allow for matching. Quite often, the fields used for matching are indexed in order to speed up the process. In the order processing example,

customers, orders and products are linked by comparing data fields and/or indexes when information from more than one record type is needed. This method is the most flexible for ad hoc inquiries, but may be too slow for heavy transaction processing environments. Although relational database is often considered a new concept, database management systems have traditionally provided relational capability with their hierarchical or network designs.

Intelligent Databases

All DBMSs provide some data validation; for example, they will reject invalid dates entered into date fields, alphabetic data entered into money fields. But the real processing is left up to the application programs.

Intelligent databases provide more validation; for example, table lookups will reject incorrect spelling or coding of items. There's no limit to the amount of processing that can be placed into an intelligent database as long as the process is a standardised function for that data. For example, the correct sales tax can be computed by the database and applied to all orders for the customer based on the customer's billing address.

When the validation process is left up to the individual application program, one program can allow one set of codes to be entered into a field, while another program can allow a totally different, and erroneous, set of codes. Data integrity is best served when there's one controlling source for the validation of data.

Mainframe databases have increasingly become more intelligent, and personal computer database systems are rapidly following suit. In time, all database management systems will be "intelligent."

Advantages of a DBMS

* Fewer application programs and lengthy regular reports containing reference data when users can access database.
* Better integration (and less duplication) of data originating at different points.
* Faster preparation of information to support nonrecurring tasks and changing conditions.
* Savings in the cost of developing new applications, and in data entry and data storage.
* Fewer errors (thus increase in data integrity) when several records are updated at once.

Disadvantages of a DBMS

* More complex hardware and software.
* Lengthy conversion period, higher training costs and more sophisticated skills needed.
* Sensitive data available on-line.
* Hardware and software failures could be disastrous.

Examples of DBMS packages include:

Mainframe - ADABAS, IDMS, IMS, TOTAL
Micros - dBASE IV, Paradox, Foxbase, DataEase

Database Server

A stand-alone computer in a local area network that holds and manages the database. It implies that database management functions, such as locating the actual record being requested, is performed in the server computer. Contrast with file server, which acts as a remote disk drive and requires that large parts of the database, for example, entire indexes, be transmitted to the user's computer where the real database management tasks are performed.

First-generation personal computer database software was not designed for a network; thus, modified versions of the software released by the vendors employed the file server concept. Second-generation products, designed for local area networks, perform the management tasks in the server where they should be done, and consequently are turning the file server into a database server.

Structured Query Language (SQL)

A language in which users of a database can (interactively) formulate requests and generate reports. The best known is SQL.

A language which provides a user interface to relational database management systems, developed by IBM in the 1970s for use in System R. SQL is the de facto standard, as well as being an ISO and ANSI standard. It is often embedded in other programming languages.

The first SQL standard, in 1986, provided basic language constructs for defining and manipulating tables of data; a revision in 1989 added language extensions for referential integrity and generalised integrity constraints. Another revision in 1992 provided facilities for schema manipulation and data administration, as well as substantial enhancements for data definition and data manipulation.

Development is currently underway to enhance SQL into a computationally complete language for the definition and management of persistent, complex objects. This includes: generalisation and specialisation hierarchies, multiple inheritance, user defined data types, triggers and assertions, support for knowledge based systems, recursive query expressions, and additional data administration tools. It also includes the specification of abstract data types (ADTs), object identifiers, methods, inheritance, polymorphism, encapsulation, and all of the other facilities normally associated with object data management.

The SQL3 standard was expected in 1998.