

1. An Introduction to Databases and Database Management Systems.

- [Syllabus](#)
 - [End of Page](#)
-

1.1. What is a Database?

A brief definition might be:

- A STORE OF INFORMATION,
- HELD OVER A PERIOD OF TIME,
- IN COMPUTER-READABLE FORM.

Let us examine the parts of this definition in more detail.

1.1.1. A store of information

Typical examples of information stored for some practical purpose are:

- Information collected for the sake of making a statistical analysis, e.g. the national census, or a survey of cracks in a stretch of motorway.
- Textual material required for information retrieval e.g. technical abstracts, statutory or other regulations. Currently there is some interest in extending such data bases in the direction of intelligent knowledge-based systems (IKBS) where rules for interpretation by an expert user are included along with the information itself.
- Operational and administrative information required for running an organisation. In a commercial concern this will take the form of stock records, personnel records, customer records, among others.

The three main examples given here are of STATIC, GROWING and DYNAMIC databases respectively.

Note that nothing was said in the definition about the quantity of information being held. Although many of the benefits associated with using a database are due to economies of scale, a small database may be very worthwhile (for instance to the secretary of the local sports club) if the information is to be processed frequently and in a repetitive manner.

1.1.2. held over a period of time.

This part of the definition goes without saying in most people's minds but it is worth dwelling on it for a minute. Because of the investment involved in setting up a database, the expectation must be that it will continue to be useful, over years rather than months. But the relationship with time varies from one type of information to another.

- Census information is collected on a particular date and stored as a snapshot of the state of affairs when the survey was taken. Information from later observations will be kept quite separately, but appropriate comparisons may be made provided that the framework remains consistent.
- Bibliographic or other textual databases are accumulated over time - new material is added periodically but probably very little will be removed. When designing such a database it will be important to estimate and allow for the expected rate of growth, and perhaps to ensure that the more recent information is given some priority.
- An organisational database may not change very drastically in size, but it will be subject to frequent updating (deletions, amendments, insertions) following relevant actions within the organisation itself. Ensuring the accuracy, efficiency and security of this process is the main concern of many database designers and administrators.

1.1.3. in computer-readable form.

Information (often referred to in this context as data) has been processed by computer for over 30 years, using a variety of storage media. Some form of magnetic disc is likely to be used, since discs currently provide the most cost-effective way of holding large quantities of data while allowing fast access to any individual item. Other methods are obviously under development, notably optical storage - CD Rom - which as yet does not give enough scope for updating in most database applications.

Database handling techniques grew out of earlier and simpler file processing techniques. A file consists of an ordered collection of records; a database consists of two or more related files which we may wish to process together in various different ways. It will store not only the individual records containing the numbers or words needed for some application, but auxiliary information which will allow those records to be accessed more quickly, or which will link related records or data items together. A database designer may be required to choose how much and what sort of auxiliary information to store, using his knowledge of how the database will be used.

Computer storage and processing implies the use of software: in the current context a DATABASE MANAGEMENT SYSTEM (DBMS). The function of the DBMS is to store and retrieve information as required by applications programs or users sitting at terminals, using the facilities provided by the computer operating system. It is one of a number of software layers making computer facilities available to users with perhaps comparatively little technical expertise.

1.2. Summary of DBMS functions.

1.2.1. Data definition.

This includes describing:

- FILES

- RECORD STRUCTURES
- FIELD NAMES, TYPES and SIZES
- RELATIONSHIPS between records of different types
- Extra information to make searching efficient, e.g. INDEXES.

1.2.2. Data entry and validation.

Validation may include:

- TYPE CHECKING
- RANGE CHECKING
- CONSISTENCY CHECKING

In an interactive data entry system, errors should be detected immediately - some can be prevented altogether by keyboard monitoring - and recovery and re-entry permitted.

1.2.3. Updating.

Updating involves:

- Record INSERTION
- Record MODIFICATION
- Record DELETION.

At the same time any back-ground data such as indexes or pointers from one record to another must be changed to maintain consistency. Updating may take place interactively, or by submission of a file of transaction records; handling these may require a program of some kind to be written, either in a conventional programming language (a host language, e.g. COBOL or C) or in a language supplied by the DBMS for constructing command files.

1.2.4. Data retrieval on the basis of selection criteria.

For this purpose most systems provide a QUERY LANGUAGE with which the characteristics of the required records may be specified. Query languages differ enormously in power and sophistication but a standard which is becoming increasingly common is based on the so-called RELATIONAL operations. These allow:

- selection of records on the basis of particular field values.
- selection of particular fields from records to be displayed.
- linking together records from two different files on the basis of matching field values.

Arbitrary combinations of these operators on the files making up a database can answer a very large number of queries without requiring users to go into one record at a time processing.

1.2.5. Report definition.

Most systems provide facilities for describing how summary reports from the database are to be created and laid out on paper. These may include obtaining:

- COUNTS
- TOTALS
- AVERAGES
- MAXIMUM and MINIMUM values

over particular CONTROL FIELDS. Also specification of PAGE and LINE LAYOUT, HEADINGS, PAGE-NUMBERING, and other narrative to make the report comprehensible.

1.2.6. Security.

This has several aspects:

- Ensuring that only those authorised to do so can see and modify the data, generally by some extension of the password principle.
- Ensuring the consistency of the database where many users are accessing and up-dating it simultaneously.
- Ensuring the existence and INTEGRITY of the database after hardware or software failure. At the very least this involves making provision for back-up and re-loading.

1.3. Why have a database (and a DBMS)?

An organisation uses a computer to store and process information because it hopes for speed, accuracy, efficiency, economy etc. beyond what could be achieved using clerical methods. The objectives of using a DBMS must in essence be the same although the justifications may be more indirect.

Early computer applications were based on existing clerical methods and stored information was partitioned in much the same way as manual files. But the computer's processing speed gave a potential for RELATING data from different sources to produce valuable management information, provided that some standardisation could be imposed over departmental boundaries. The idea emerged of the integrated database as a central resource. Data is captured as close as possible to its point of origin and transmitted to the database, then extracted by anyone within the organisation who requires it. However many provisos have become attached to this idea in practice, it still provides possibly the strongest motivation for the introduction of a DBMS in large organisations. The idea is that any piece of information is entered and stored just once, eliminating duplications of effort and the possibility of inconsistency between different departmental records.

Other advantages relate to the task of running a conventional Data Processing (DP) department. Organisational requirements change over time, and applications programs laboriously developed need to be periodically adjusted. A DBMS gives some protection against change by taking care of basic storage and retrieval functions in a standard way, leaving the applications developer to concentrate on specific organisational requirements. Changes in one of these areas need not have repercussions elsewhere. In general a DBMS is a substantial piece of software, the result of many man-years of effort. Because its development costs are spread over a number of purchasers it can probably provide more facilities than would be economic in a one-off product.

The points discussed above are probably most relevant to the larger organisation using a DBMS for its administrative functions - the environment in which the idea of databases first

originated. In other contexts the convenience of a DBMS may be the primary consideration. The purchaser of a small business computer needs all the software to run it in package form, written so that the minimum of expertise is required to use it. The same applies to departments (e.g. Research & Development) with special needs which cannot be satisfied by a large centralised system. When comparing database management systems it is obvious that some are designed in the expectation that professional DP staff will be available to run them, while others are aimed at the total novice.

There are of course costs associated with adopting a DBMS. Actual monetary costs vary widely from, for instance, a large multi-user Oracle system to a small PC-based filing system. In the first case the charge will cover support, some training, extensive documentation and the provision of periodical upgrades to the software; in the second case the purchaser will be on his own with the manual. But there is also a tendency for the cost of software to reflect the cost of the hardware on which it is run!

Probably the main cost associated with acquiring a DBMS is due to the work involved in designing and implementing systems to use it. In order to provide a general and powerful set of facilities for its users any DBMS imposes restraints on the way information can be described and accessed, and demands familiarity with the DATA MODEL which it supports and the command language which it provides to define and manipulate data. Data models still in use are HIERARCHICAL (tree-structured), NETWORK and RELATIONAL (tabular). Of these the last is the current favourite, providing a good basis for high-level query languages and giving scope for the exploitation of special-purpose hardware in efficient large-scale data handling.

This course will concentrate on the RELATIONAL model.

1.4. Data Base Project Development.

The conventional SYSTEMS LIFE CYCLE consists of:

1.4.1. ANALYSIS 1.4.2. DESIGN 1.4.3. DEVELOPMENT 1.4.4. IMPLEMENTATION
1.4.5. MAINTENANCE

In practice these phases are not always sharply distinguished; for small projects it may not be necessary to go formally through every one. The move from one phase to the next is essentially a move from the general to the specific. At each stage, particularly where a DBMS is involved, we shall be concerned both with information and with processes to be performed using that information.

1.4.1. Analysis

The outputs from this stage should be:

- A CONCEPTUAL DATA MODEL describing the information which is used within the organisation but not in computer-related terms. This level of data analysis will be considered in more detail later. One of the problems with any systems design in a large organisation is that it must proceed in a piecemeal manner - it is impossible to create a totally new GLOBAL system in one fell swoop, and each sub-system must dovetail with others which may be at quite a different stage of development. The conceptual

data model provides a context within which more detailed design specifications can be produced, and should help in maintaining consistency from one application area to another.

- A CONCEPTUAL PROCESS MODEL describing the functions of the organisation in terms of events (e.g. a purchase, a payment, a booking) and the processes which must be performed within the organisation to handle them. This may lead to a more detailed functional specification - describing the organisational requirements which must be satisfied, but not how they are to be achieved.

1.4.2. Design

This stage should produce:

- A LOGICAL DATA MODEL: a description of the data to be stored in the database, using the conventions prescribed by the particular DBMS to be used. This is sometimes referred to as a SCHEMA and some DBMSs also give facilities for defining SUB-SCHEMA or partitions of the overall schema. Logical data models supported by present day DBMSs will be considered later.
- A SYSTEM SPECIFICATION, describing in some detail what the proposed system should do. This will now refer to COMPUTER PROCESSES, but probably in terms of INPUT and OUTPUT MESSAGES rather than internal logic, describing, for instance, the effect of selecting an item from a menu, or any option within a command driven system. Program modules are defined in terms of the screen displays and/or reports which they generate. Note that the data referred to here has a temporary existence, in contrast with what is stored in the database itself.

1.4.3. Development.

Specification of the database itself must now come down another level, to decisions about PHYSICAL DATA STORAGE in particular files on particular devices. For this a knowledge of the computer operating system, as well as the DBMS, is required. Conventional program development - coding, testing, debugging etc. may also be done. If a totally packaged system has been purchased this may not be necessary - it will simply be a matter of discovering how to use the command and query language already supplied to store and retrieve data, generate reports and other outputs. Even here an element of testing and debugging may be involved, since it is unlikely that the new user of a system will get it exactly right the first time. It is certainly inadvisable for this sort of experimentation to take place using a live database!

1.4.4. Implementation.

This puts the work of the previous three phases into everyday use. It involves such things as loading the database with live rather than test data, staff training, probably the introduction of new working practices. It is not unusual to have an old and a new system running side by side for a while so that some back-up is available if the new system fails unexpectedly.

1.4.5. Maintenance.

Systems once implemented generally require further work done on them as time goes by, either to correct original design faults or to accommodate changes in user requirements or operating constraints. One of the objectives of using a DBMS is to reduce the impact of such changes - for example the data can be physically re-arranged without affecting the logic of the programs which use it. Some DBMSs provide utility programs to re-organise the data when either its physical or logical design must be altered.

- [Syllabus](#)
 - Next: [The Conceptual Data Model](#)
 - [Top of page.](#)
-

Mike Gatford and Tony Valsamidis
October 1998

tony@soi.city.ac.uk