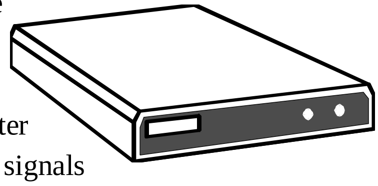
- *Routers* and *bridges* are computers dedicated to the translation of network protocols and standards between different networks. They are becoming important as more and more organisations are linking their own networks to those of other organisations. They may be using
  - different operating systems (Novell, Unix or Windows NT),
  - other technologies (coax or fibre-optic),
  - or different protocols (proprietary or public standards set for computer communications).

- Finally, the *modem* allows a computer to communicate with another computer by means of the public voice telephone network, rather than by using digital cabling. This requires the conversion of digital computer signals (used inside the computer) into analogue sound signals (that can travel over the voice telephone lines) – this process is called *modulation*. At the other end of the line, these sound signals are converted back into digital signals – or *de-modulated*. The word *modem* refers to this modulation / demodulation process. You may have heard this "modulated signal" when listening to a fax machine, which is really a scanner/printer/modem in one. Since the modem replaces the network card, it usually carries out similar error-checking functions to ensure the correct transmission of data.



Trivial fact: More than 5000 satellites are orbiting the earth and most of them are involved in telecommunications.

6.3 SA Public Telecommunications Services

Because telecommunication services are a critical part of any country's infrastructure, most governments have been very protective towards their telecoms. Paradoxically, this protectionism often resulted in high tariffs (monopolies!), thus reducing the overall competitiveness of local businesses. Recently more and more countries have started to privatise these services and allowed competition to drive prices down. The South African public telecommunication services are controlled by Telkom, although its legal monopoly is being phased out. The following are the main data network services provided by Telkom.

6.3.1 Public Switched Telephone Network (PSTN) Services

The oldest data service provided by Telkom is the *Datel* service, which provides a connection between computers by means of the standard *Public Switched Telephone Network (PSTN)* i.e. the same as the normal voice telephone traffic. This requires the use of built-in or external modem equipment that modulates the digital signal into an analogue audio signal (and de-modulates it back at the receiving end). This service is quick and easy to set up since it is available anywhere where there is an ordinary voice telephone point. The main drawbacks are the limited transmission speed, high error rate and the lack of security. Customers may choose between a *dial-up* or *leased line* connection.

6.3.2 Diginet

Diginet is a dedicated digital data service from Telkom that provides reliable and efficient point-to-point (i.e. not switched) data connections. It differs from the Datel network in that the transmission path is entirely digital: a combination of fibre-optic, microwave and coaxial cable. Because the signal does not have to be translated into analogue form, no modem is required, resulting in a cost saving. However, its main advantages are the higher transmission rates and a substantial reduction in transmission errors. The standard Diginet service allows for 64 kpbs (*kilobits per second*) though an enhanced service called *Diginet-Plus* has been designed to transfer up to 1920 kpbs, which allows slow-scan TV and video conferencing signals to be transmitted in real-time.

6.3.3 Public Switched Data Network (PSDN) Services

Saponet is Telkom’s *Public Switched Data Network (PSDN)*. The *Saponet-P* service relies on a *packet-switching* mechanism whereby all data transmissions are broken up into smaller, standard-size units or *data packets*. Each of these packets is then routed independently to their destination. The path travelled by the packet depends on the available capacity and bottlenecks. At the destination, the original transmission is reassembled out of the constituent packets. A *Packet Assembler & Disassembler (PAD)* is responsible for the breaking up of a message into packets and the opposite process of reassembling packets into a message at the destination. This PAD can be a separate hardware device or a software program.

6.3.4 X.400 and Telkom400

Telkom400 is a VAN on top of the X.400 infrastructure. It supports electronic message handling and *electronic data interchange (EDI)*. *EDI* is the automated computer to computer application exchange of structured business data between different organisations. An international standard defines common business documents such as order forms, invoices or electronic funds transfer documents that are exchanged directly between the computers of the respective business partners.

6.3.5 ISDN and ADSL

Now that most of Telkom’s telephone exchanges have become digital, Telkom is able to provide new functions and services. One all-digital connection that has sufficient capacity (*bandwidth*) to support speech, video conferencing, facsimile, data and image transfer. This connection is called an *ISDN line (Integrated Services Digital Network)* and is currently available in selected metropolitan areas. Much more popular is the newer Asymmetric Digital Subscriber Line (ADSL) which allows a broadband connection (at least several hundred kilobits/second) over your standard telephone line while keeping the line available for voice telephone calls. It is called “asymmetrical” because the standard allows for much greater “download” than “upload” speeds; this reflects the typical home user pattern. Higher volumes and transmission speeds of up to 150mbs – typically needed by mid-size and larger businesses – are available through Telkom’s *ATM Express* service.

6.4 The Internet

The Internet is probably the most exciting, the most popular, most visible and definitely the "coolest" information systems development of the decade.

6.4.1 What is the Internet?

The origins of the Internet can be found in the early sixties, when the US Department of Defence sponsored a project to develop a telecommunications network that would survive a nuclear attack. It had to link together a diverse set of computers and work in a decentralised manner so that, if any part of the network were not functioning, network traffic would automatically be re-routed via other network nodes. This project quickly grew into a popular academic network linking virtually all major research institutions and US universities. Soon other countries jumped onto the bandwagon, thus linking academics and researchers across the globe. True to the academic ethos, it quickly became a means for global information sharing. By now, businesses also got a piece of the action. This was spurred on by the trend to network the personal computers in home and business environments and the development of more user-friendly, graphical interfaces: the web-browser and the Windows operating system.

The *Internet* (or, more colloquially, *the Net*) consists of a huge and fast-growing number (hundreds of thousands) of *inter*connected *net*works linked together. Currently more than 100 million users are connected to the Internet. The popularity of the Internet can be explained by the amount of information it makes available: the equivalent of many libraries of information is stored on millions of computers (*Internet hosts*), much of it free of charge to all Internet users. This information is provided by educational institutions, governmental agencies and organisations, individuals, and increasingly by businesses. Hence, the Internet is frequently referred to as the *Information Highway* or the *Infobahn*.

But the Internet is more than just a huge information resource. Its initial purpose was to act as a communications network and it fulfils that role well. It is the transport mechanism for *electronic mail,* the transfer of computer files, remote computer access and even allows for voice calls. Businesses quickly realised the potential of the multimedia-enabled Internet for marketing purposes. Of late, more and more business transactions are being conducted via the Internet: *electronic commerce (e-commerce)* is the latest revolution to be embraced by the Internet community.

6.4.2 Electronic mail

Probably the most popular Internet service is *electronic mail,* more commonly known as *e-mail*. This consists of the sending of messages composed on the computer, via a network, directly to the computer of the recipient who reads the message on his/her computer. Knowledge workers with access to e-mail write five to ten times as many e-mail messages as hand-written notes. The following are just some of the advantages of e-mail.

- *Reliability*: although there is no guarantee, you will normally receive quick feedback if the address does not exist or there is a similar delivery problem.
- *Efficiency*: many short-cut tools exist to increase your efficiency when composing messages. You can use your computer's cut-and-paste function, you can have managed address books and lists, when replying to another message you can automatically incorporate any part of the message to which you are, etc. And it is just

as easy to send a message to one as to a whole list of addressees. (Admittedly, this results in a lot of abuse and information overload on the recipient's side.)

- *Digital*: e-mail is composed on a computer and remains in computer-readable format all the way to its destination. Thus one can also easily incorporate other computer data such as graphics or document files.
- *Cheap*: because the capacity of the Internet and disk storage is increasing all the time, the cost of a sending and storing a one-page e-mail message is negligible.
- *Speed*: messages are generally delivered across the world in a matter of seconds.

The e-mail address

Just like with ordinary postal mail (now usually referred to as *snail-mail*), you need to know the recipient's address before you can send your message. Internet *e-mail addresses* have a standard format: username@domain. The *username* is often the name that your addressee uses to connect to the network, e.g. "jvanbelle" or sometimes a long number. This username is allocated by the LAN administrator. The *domain* identifies the file server, which acts as the local post office for your recipient's e-mail. The domain consists of several parts, separated by full stops or *dots*. The international standard for domain identification is <name of LAN server>.<name of organisation>.<type of organisation>.<country code>.

- The country code is the international two-letter code for the country e.g. *au* for Australia, *za* for South Africa, *sa* for Saudi-Arabia, *uk* for Great Britain, etc.
- The two most common types of organisations are *co* for a commercial organisation and *ac* for an academic institution . Less frequent are *org* for (not-for-profit) organisations, *mil* for military, *net* for networks and *gov* for government agencies.
- Each country has a national Internet naming body that allows its organisations to chose their own name, as long as no one has claimed the same name before. Examples of South African domain names are anc.org.za, uct.ac.za, fnb.co.za.
- Large organisations often refine the domain further by adding the name of their LAN servers, e.g mail.uct.ac.za.

Examples of possible e-mail addresses are: JaneDoe@stats.uct.ac.za (Jane working in the statistics department at the University of Cape Town in South Africa); info@anc.org.za (information department at the ANC, a political party) or SoapJoe@marketing.bt.co.uk (Joe Soap in the marketing department of British Telekom in the U.K.).

The US Americans, having "invented" the Internet, use a slightly different way for their addresses. They leave off their country code (*us*) and use *com* for commercial organisation or *edu* for educational institution. Since the majority of Internet users hail from the US, you will encounter many addresses such as JWood@mit.edu or Bill@microsoft.com.

Netiquette

Just as in any other social interaction environment, there are some rules and guidelines for appropriate social behaviour on the Net: *Netiquette* (etiquette on the Internet). The following are some illustrative examples pertaining primarily to e-mail.

- *Shouting*¸ THE PRACTICE OF TYPING ENTIRE SENTENCES IN UPPER CASE is generally seen as novice (*newbie*) behaviour and frowned upon. Perhaps it stems from the disgust with old teletypes and mainframe terminals that did not have lower-case characters.

- The use of *emoticons* to indicate emotive content of a sentence is highly recommended. Typed text does not reveal any body language and a joking remark can easily be interpreted the wrong way. Whenever one writes something in jest or with humorous intent, it is advisable to add an emoticon. An *emoticon* (an icon indicating emotional content) consists of a series of text characters which are meant to be rotated a quarter turn and represent a laughing :-) (i.e. equivalent to J or the *smiley*) winking ;-) or sad face :-(L.

:-)	Happy face
:-(	Sad or sorrow
;-)	Wink
:-0	Shock
:-\	Sarcasm
:^]	Wide grin
:-x	Blowing a kiss
:'(	Teary-eyed
:-P	Sticking tongue out
8-)=	Beard & glasses
:--)%	Boy on skateboard
<g>	Grin
<w>	Whisper
{{}}	Hug you

BTW	By the way
IMHO	In my humble opinion
FWIW	For what it's worth
SO	Significant other (partner)
BRB	Be right back
FAQ	Frequently Asked Questions
RTFM	Read the f*** manual
TTYL	Talk to you later
IRL	In real life
F2F	Face to face
LOL	Laughing out loud

- *Flaming* is the carrying on of a heated personal emotional debate between two or more individuals on a public Internet forum. A *flame war* is generally a sign of immature behaviour by individuals who cannot take perspective and should really take the discussion off-line.

- *Netizens* (inhabitants of the Internet i.e. frequent net surfers) often use standard but, to the non-initiated, cryptic abbreviations. Examples are: BTW = by the way ; ROFL = rolling on the floor with laughter ; TPTB = the powers that be ; BRB = Be Right Back. This vocabulary has been adopted and expanded with the growth of Short Message Service (SMS) use on cellular phones.

6.4.3 The Web

The Internet service that has received the most attention from the public media is the *World-Wide Web* or *the Web* for short (sometimes also called *WWW* or *W3*). The Web is a vast collection of multimedia information located on Web servers attached to the Internet.

Its popularity is due to a number of reasons.

- Information links are transparent. Links to any other piece of information located anywhere on the Internet can be inserted in a web document. A simple click of the mouse takes the reader completely automatically from one Web server to another, quite possibly in another country.

- Information can be presented in a *hypertext link* format whereby one can jump immediately from one concept to a related concept or explanation. No need to read text in the traditional top-to-bottom sequential way.

- It allows for *multimedia* information. A Web document can incorporate rich and colourful graphics, animation, video clips, sound etc. Just think of the marketing opportunities!

- The Web supports interactive applications. Web applications can request information from visiting users and documents can include programming instructions. Users can even download small programs (often written in *Java*) that could perform some processing on the user's computer or display special visual effects.

Reading or accessing information on the Web is called *surfing the Net* because one jumps from one hypertext link to another following whatever takes your fancy. In order to *surf* the Net you need some special *browser* program that understands the Web protocols and formats and presents the information to suit your computer monitor. You also need an access point or connection to the Internet. Your Internet connection may be automatic if your computer is connected to a (corporate) LAN that connects directly to the Internet, or it may be by means of a special subscription to a business that specialises in providing Internet access for others: the *Internet Service Provider (ISP)*. Access to the ISP for individual users is usually via a dial-up connection i.e. using a modem and telephone.

Once a *newbie* (new user) is connected to the Internet (*online*), she faces the daunting task of finding her way amongst the huge variety of information offered. The easiest way in is usually by means of a *search engine:* a Web site that tries to catalogue the information available on the Internet. By entering one or more search words, the engine will provide you with a couple of adverts and a list of documents that contain the word(s) for which you are looking.

All information on the Web is uniquely identified by the *URL (Uniform Resource Location)*, which is really the full Internet address of a Web document. The URL consists again of the Web server's domain address, followed by the access path and file name on the server. Examples of URLs are www.hotbot.com/sports/main.html (the main page on the sports section of the HotBot search engine) or http://www.commerce.uct.ac.za/informationsystems/ (containing details about UCT's department of information systems). Note the similarities and differences between an URL and an e-mail address.

6.4.4 Other Internet services

A number of other services are available on the Internet. The *Usenet* consists of ongoing discussion fora (or *newsgroups*) on an extremely wide variety of topics, from forensic psychology to Douglas Adams, from Star Trek to cryptography. The discussion happens entirely by means of e-mail and, when you subscribe to a given newsgroup, you can browse through the contributions of the last few days and reply with your own contribution.

More specialised services exist, such as *ftp (file transfer protocol)* for the transfer of large computer files, and *telnet*, the remote access of computers elsewhere, but they are used less frequently. In any way, these services are now being performed transparently by most Web browsers. Similarly, older services such as *Gopher* and *Veronica* have really been replaced almost entirely by the Web.

6.4.5 Internet protocols and standards

Different computers and networks can communicate via the Internet because a number of basic Internet communication standards have been defined. Any network connected to the Internet will translate its own standards and protocols into those used on the Internet by means of a bridge.

The most fundamental and "lowest level" protocol is the *TCP/IP (Transmission Control Protocol/Internet Protocol)*. This protocol is also the *native* protocol of computers using the Unix operating system, which explains why Unix computers are so popular as Internet servers.

On top of TCP/IP are the "mid-level" protocols defined for the various Internet services. Perhaps the best known of these is *http (Hypertext Transmission Protocol)*, which specifies how the Web information is made available and transmitted across the Internet. Other protocols and standards are *STMP* and *MIME* (for e-mail) or *ftp*.

Information made available via the Web is usually formatted using a special standard: the Hypertext Markup Language (HTML), which actually consists of plain text files with visual formatting commands inserted between the text. Most desktop productivity software allows you to save your document directly in the HTML format. Special HTML editors allow much finer control over the final layout of your Web document. A later development is Extensible Markup Language (XML), which increases the flexibility of web documents by allowing them to be viewed not only using a web browser, but also on different platforms such as a PDA or cellular telephone.

6.5 South African Perspective

The internet is being used in South Africa to help in the prevention of child abuse. Ordinary citizens who are outraged by the high levels of child abuse and wish to make a positive contribution towards combating and preventing the problem, can now volunteer their assistance. A website has been launched at www.volunteerchildnetwork.org.za, which offers volunteers the opportunity to identify the skills they have that could be useful to organisations, and to connect them with organisations that need their assistance.

A similar type of business operating in Gauteng gives tourists the opportunity for hands-on work in wildlife conservation, by placing volunteers with various rehabilitation centres, conservancies and nature reserves.

6.6 Beyond the Basics

Your cellphone may be all you need to gain entry to World Cup 2006 soccer matches. Two Austrian companies have combined their expertise to develop a paperless ticket system which allows users to pay for tickets via the Internet, and a few seconds later receive an encrypted entry code via SMS to their cellphone. The user then gains access to the event by passing the display of the cellphone, containing the encrypted code, over a special scanning device at the entry gate.

In order to allow for the various standards of different cellphone manufacturers, users must enter the name of their cellphone model when ordering tickets. The developers claim that 95% of all existing handsets are able to use their system. So far it has been successfully tested at the Football Expo 2002 in Cannes, and at a nightclub in Austria. However, they don't say what will happen if your cellphone is stolen, or if you accidentally erase that vital SMS!

6.7 Exercises

6.7.1 Search Engines

- The Internet is vast and information is usually located using search engines such as Google, Yahoo!, Lycos, WebCrawler or Ananzi. Give the name and URL of two other Internet *search engines*, of which at least one must be South African.

- Take your date of birth ("DOB") and search the Internet to see if you can find a home (personal) page of another person who shares your date of birth, or a page reporting on an event that happened on your DOB. Give the URL of the home page that you have located, as well as the name of the search engine you used.

- Think of your favourite hobby or sport. Locate a web-site in South Africa (use www.ananzi.co.za) that is devoted to your interest and give the name of your sport/hobby and the URL of the main (index) page of the site you have found.

- Go to Independent On-Line (www.iol.co.za). Search the site for the most recent article that mentions your place of birth and give the full TITLE and date of the article. [If your birth place is not found, search for the nearest big town to your place of birth.]

6.7.2 Cellular communications

In South Africa, more people have cellphones than have access to computers. What implications does this have for the development of our telecommunications infrastructure? Do you think this will have any significant impact in reducing the "digital divide" and improving access to local and global information resources?

7. Databases

Often one hears about the purchase of a company with few physical assets for large sums of money. Why are investors often prepared to pay an enormous premium over the asset value of a company instead of just buying the same fixed assets and starting up a competing company themselves? There are three main reasons: companies may have extremely efficient business processes, they may have valuable intangible property such as patents or trademarks, or they may have a very valuable information database of customers. Company executives are increasingly realising the value of the information that is contained in the databases of their information systems. This chapter explores the database component of information systems in more detail.

7.1 From File-based Systems to the Database Approach

7.1.1 The File-based Approach

Many small (personal computer) applications, as well as a large number of larger *legacy* (i.e. older) corporate information systems take a file-based approach to data storage. The function-oriented information system stores all its data into its own files. The structure and integrity of these files is maintained within the application.

As a typical example of the file-based approach, let us take a look at the information systems of a commercial bank whose identity shall remain anonymous. The bank has developed a sophisticated system to process its cheque accounts. This system maintains, *inter alia*, its own customer information. The bank has another, entirely separate system for its credit card accounts. This system obviously also keeps customer details in its own files. There are other, stand-alone systems for mortgage bonds, savings accounts, notice deposits etc. Each of these systems was developed independently and probably even runs on different computer systems. None of the information is shared between the applications. What are the problems?

Imagine the typical scenario of a bank customer who moves house. She notifies the bank's help desk of her change of address, and the next month (with luck) her cheque account statement arrives at the new address. However, the savings account statement is still sent to the old address. On enquiry, she is told that the bank was unaware of the fact that she also had savings account and changed only her information on the cheque system; so now they have to enter the change of address separately on the savings account system. Surprise! The next month, her *mortgage bond* statement is still sent to the old address … This is, however, not the only problem. Imagine what would happen if the customer accidentally issues a cheque for an amount exceeding her overdraft. A bank clerk bounces the cheque, resulting in an embarrassed customer, only to discover later that this customer has in fact a large amount invested in various deposit accounts. The irate customer may well decide to change bank and the bank loses a profitable customer! In addition, the bank probably loses many marketing opportunities (such as cross-selling) by its inability to identify multiple account-holding customers.

7.1.2 The Database Approach

What is the solution? The *database* approach. All customer information should reside centrally in one database, and only account-specific information should be stored separately. All the system applications that require customer information should access the same customer database. In fact, no individual application will be allowed to access this database directly. A spcialised system, the *database management system (DBMS),* will control all access to the database and be responsible for security, integrity, data formats, backups and other technical issues. Any application requiring or updating data sends a request to the DBMS, which will then extract data from, or update, the database.

7.1.3 Advantages and Disadvantages of the Database Approach

The database approach has some *disadvantages:*

- It introduces an additional layer of software between the application and the database, often resulting in a slower access to the data.
- System designers and functional users loose some freedom and control over their data, since they can no longer decide exactly what and how data will be stored. All changes or additions to the database have to be cleared by the Database Administrator (DBA).
- The reliance on one single database increases the vulnerability of the organisation: the lack of redundancy, industrial espionage, a higher level of complexity and size.
- A DBMS introduces additional costs: a typical DBMS licence is very expensive and requires specialised, highly-paid staff.

On the other hand, the *benefits* of the DBMS usually outweigh the disadvantages by far.

- The performance penalty exists only if a file-based application is extremely well designed. In most cases, the designers of a DBMS have done a lot of research and spent much time in optimising the data access routines. As an example: sorting of data using a very non-intuitive but highly effective method is often hundreds of times faster than the obvious ways generally adopted by the non-specialist application designer.
- The centralisation of data facilitates risk management: it is much easier to back-up, monitor or audit one central database than thousands of separate files.
- The database is independent of the individual applications. The DBA can easily say, add a new field (or change an existing field) in the database without affecting the applications. Where the same file was accessed or shared by different applications, each application had to be individually modified to accommodate a change in file layout. When the database structure changes, this will generally not affect the DBMS calls programmed into the existing applications.
- The integration of data in a single logical DBMS view (although it may physically be stored in many different databases) allows for better management decision support. It is now possible to have the complete picture or profile of a customer; or to relate employee performance to sales.
- There is no more, or much more controlled, data redundancy and hence data should be more consistent.

- A DBMS offers many additional tools to access and manipulate the data. It will typically include powerful security features, backup and disaster recovery tools, database programming and query language interfaces.
- It is usually easier to upgrade or adopt new technologies. When the DBMS vendor upgrades the DBMS to include the new developments, these become available automatically to all applications without major redevelopment efforts. Thus an organisation may be able to implement object-orientation, e-commerce, client/server or enterprise-wide systems without costly system conversions.

7.2 Data Structures

In chapter 2, we saw that the smallest data element is the *bit*: a yes/no, true/false or any possible choice between two options. We also found that, in practice, a *character* is really the smallest practical data element: a character can be a letter, digit or punctuation symbol. Most textual and numerical data can be represented using characters. However, there are larger units or groupings of data.

Business information systems currently deal mostly with large amounts of structured textual (and numerical) data: customers have a name, address, account balance, and place orders that in turn again contain standard elements such as order date, product item descriptions and amounts etc. The efficiency of storing and retrieving this data can be increased by structuring it into a standard format and grouping similar data items together.

7.2.1 Database.

The totality of the data collected by an information system is called the *database*. Although the ideal is to have a single database for the entire organisation, in practice organisations may have a number of functional databases that may not be entirely compatible. The customer order system may have a database of customers, orders and shipments while the human resources system may have a completely separate database with employee information, job positions, training and salaries.

7.2.2 Table and file

Normally, the database will be structured into smaller logical or physical units: tables or files respectively. E.g. the order database may use one file to keep customer information, another to keep track of orders, yet another with account details and probably a file with product details. As we will see later, the *table* is really a *logical* concept applicable mainly to the popular network and relational databases. The *file* is a *physical* way in which data is stored on the computer.

7.2.3 Record

Within a table (or file), information will be organised in the form of *records*. Each record represents one instance of a real-world entity or transaction. E.g. in the customer database, there may be a record containing all the data about Joe Smith (including his address, contact numbers, credit status), there will be another record with the data for Jane Doe and so on, one record for each of the customers.

7.2.4 Field

The information within a record is structured into separate *fields*. In the above example, Joe Smith's record would have at least four fields: his name, his address, his contact telephone number and his credit status. In practice, his name may be split into at least three separate fields: one for his surname "Smith", one for his first name "Joe", one for his designation "Mr" and probably one for his initial(s). Similarly, the address will usually be split into several fields to separate e.g. the postal code, the town and the street. This separation will make it easier to extract subsets of the data, such as all the sales for a given area, or to print out customer address labels sorted according to postal code.

7.2.5 Characters and Bytes

Finally, the data inside a field will be encoded in a number of characters (for textual data) or bytes (for numerical and multimedia data).

When designing information systems, a lot of attention must usually be given to the logical design of the database. As an illustration, we will look at some of the considerations involved with fields.

7.2.6 Some Field Design Considerations

Fields can have a *fixed length* (such as the postal code) or a *variable length* (such as general text comments, or a sound track in a multimedia database). Because fixed length fields are much easier to work with from a technical perspective, system designers often will try to impose a fixed length on fields that are really variable length fields. E.g. the information system will reserve 30 characters of space for the "city" address field. Shorter names such as "Cape Town" will then automatically be "padded" with extra empty spaces (resulting in some wasted database storage space), but the simpler design (resulting in performance improvements) will usually offset the marginal cost increase. On the other hand, a problem results when a city name exceeds the sample 30 characters space limit. To take an extreme case: not many information systems will cope with the place called "Taumatawhakatangi-hangakoauauotamateaturipukakapikimaungahoronukupokaiwhenuakitanatahu" in New Zealand, or, to take a slightly less extreme example, one of the authors often has problems fitting his first names "Jean-Paul Willy Georges Dominique" on standard forms!

Another problem is often to identify uniquely each record. How would one identify a customer record? A first thought is to use the *name* of the customer. However, any large organisation is likely to have several customers with the same name: in Cape Town alone there are more than 5000 persons with the surname Abrahams or Smith (and, contrary to popular belief, only about half as many Mohammeds or van der Merwes). A common solution is to look for some guaranteed unique field such as an identity number, a combination of fields (e.g. name and date of birth or address) or by artificially generating a unique code for each customer: the dreaded "customer code" (or, in the case of student records: your student number). The field that uniquely identifies a record is called the ***primary key*** field and is very important in database design, not only when searching for a record, but also to link one data table to another (e.g. when linking customers to their orders).

7.3 Database Models

Just as computer hardware and software development languages have seen different technological generations, database and DBMS technology have also undergone major changes. This manifests itself primarily in the conceptual database model i.e. the type and nature of the relationships that are allowed between different data tables.

7.3.1 The Hierarchical Data Model.

The *hierarchical data(base) model* is the oldest and conceptually simplest model. This structure allows only for one type of relationship, the "parent-child" (or one-to-many) relationship. In addition, there can be only one single relationship between tables. This structure can also be visualised or represented using the image of a tree.

For instance, a customer (the "parent" entity or *root*) can place one or several orders (= "child"). Each order, in turn, will consist of one or several line items (the different products or services that are ordered). In this model, it is very simple to trace the various orders placed by a customer. However, it is more difficult to investigate the correlation between, e.g. the sales of different products (line items) according to geographical area (address of customer) or sales person.

7.3.2 The Network Model

The *network data model* accommodates the "many-to-many" relationships often found in the real world. It allows the explicit linking of sets of entities by means of network relationships.

In a university environment, the relationships between courses and students could not (or with great difficulty) be accommodated in a hierarchical database. Each student can enrol for one or several courses, whilst each course will have an enrolment of zero, one or, hopefully, many students. Student Xannie Du Toit may enrol in the 3 courses Quantitative Philosophy, Exobiology and Data Warehousing; the course Exobiology may have students Jim Mbeki, Xannie Du Toit, Yusuf Arafat and Wang Ching.

However, although the network model allows more complex relationships to be defined than the hierarchical model, it is significantly more complicated than the other models to design and maintain.

7.3.3 The Relational Model

The *relational data model*, pioneered by Codd, is founded on a sound logical set-theoretical basis. Although it took well over a decade for his ideas to find widespread acceptance, almost all major DBMS in use today support the relational model almost in its entirety. Although a detailed discussion of the relational model is outside the scope of this course, a simplified example below will illustrate some of the basic concepts and features for a partial university administration system.

The relational model requires that all data is stored in *data tables*, with unique key fields (the *primary* key) and all other data fields in the table entirely and fully dependent on the key fields. In the example below, there are a large number of tables (the relational model tends to generate a large number of tables). The student table has the student number as the key field.

The order or sequence of individual records is of no concern. Students do not have to be stored in alphabetical or any other specific order, though they may be sorted for reporting purposes.

Tables can be linked, by reference to their primary key fields, in any possible way. Links between tables do not have to be predefined when the database is designed, providing structural flexibility and efficient information retrieval. Each link between tables is a *relationship* and consists of a separate table; e.g. the course enrolment table is really the "enrolment" relationship. There could be other relationships between the same two tables, e.g. a "tutor" relationship whereby some postgraduate students have been appointed as tutors for certain undergraduate course.

The inclusion of a primary key field from another table is called a *foreign key*. E.g. the current degree code in the student table is an example of a foreign key. (This model assumes that an active student can be enrolled for only one single degree.)

Normalisation is a step-by-step process of ensuring that a database structure conforms to the requirements of the standard relational database models. One might be tempted to include course prerequisites in the course table by means of a course prerequisite field. However, some courses may have more than one prerequisite (e.g. Econometrics could require Statistics 1A and Macro-economics 2 as prerequisites). This would require two or more course prerequisite fields. The latter contradicts the first normalisation rule - no repeating fields are allowed.

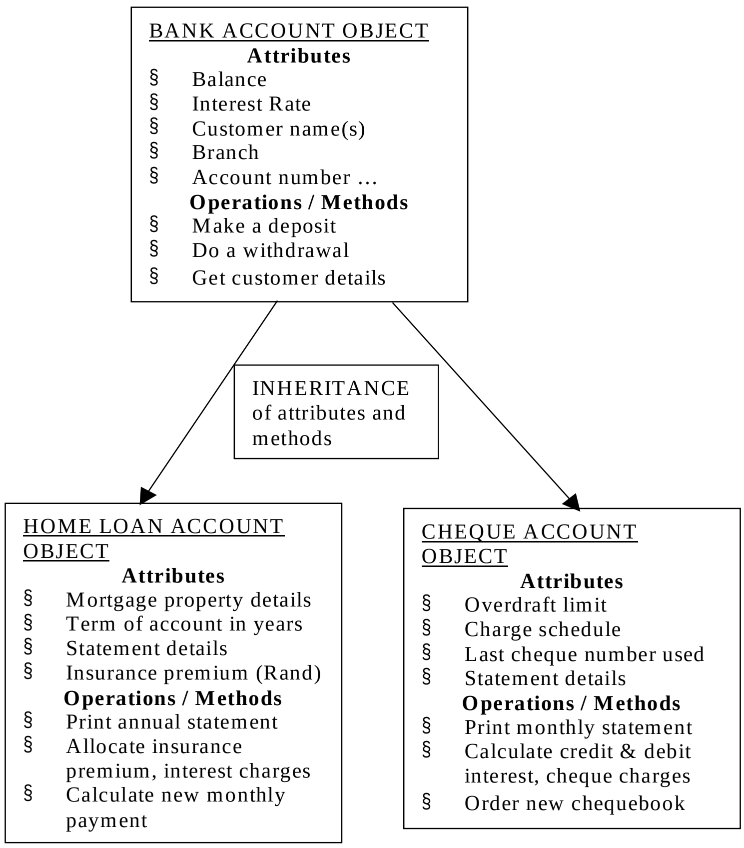
A typical example to illustrate the use of a relational database is that of student enrolments at the university. There will be a *student* table containing the student number (the key field), surname, name, year of first registration, activity status (currently registered or not) and current degree code (foreign key). Similarly, there will be a *course* table which contains course codes (key field), course names, level of course (1^{st} year, 2^{nd} year etc.), type of course (first semester, second semester, full-year, summer term), responsible department (code), number of credits, etc. There will be many other "primary" or entity tables such as lecturers, venues, lecture and tutorial groups etc. The power of the relational model is that it allows *any* type of relationship. E.g. in a small university, different students could be assigned individual tutors, who are senior students, for the various courses. This assignment (in effect, a relationship) could be stored in a "tutor assignment table" containing at least the following fields "tutored student" (foreign key = student number), "tutor" (foreign key = student number), "tutored course" (foreign key = course code). This would be very difficult to achieve in the network model. Note that it is quite possible, indeed very likely, that there will be many relationship tables with exactly the same fields (e.g. "students enrolled in course" and "tutors assigned to course"; or "required courses for degree" and "optional courses for degree").

7.3.4 The Object-Oriented Model

The *object-oriented (OO)* approach was introduced in the chapter on software. Remember that an object encapsulates both the data and the methods or operations that can be performed on the data. Although business databases were not initially considered to be good candidates for OO, more and more applications are benefiting from the use of *object-oriented database management systems (OODMBS).*

Two particular strengths of the OO model are the capability to handle *multimedia* more easily and the feature of *inheritance*. Multimedia, such as pictures or sound, are more easily handled by OODBMS because it is much easier to store the methods, or procedures, to access and edit the data within the object than it is to include those methods permanently as part of the DBMS. As such, OODBMS are popular for engineering and scientific applications. However, product and human resource databases can equally benefit from picture and photograph fields.

The other advantage is that the OO model allows specialised objects to inherit object attributes and methods from their *parent*. A banking example can illustrate this. A generic "bank account" object would have the attributes of "owner details", "account number", "balance", "interest" etc. Specific types of accounts, such as home loans, cheque and saving accounts, inherit the common account attributes and methods. In addition, each account type would have its own specific attributes and methods.



7.3.5 Free-form databases

Another attempt to deal with unstructured data and information is the *free-form database*. Textual (and sometimes multimedia) information can be entered in logical chunks and related to other pieces of information, often using hypertext principles. These databases feature powerful indexing and searching tools. Most free-form databases are stand-alone end-user applications but, in a way, one could consider the entire web to be one huge distributed free-form database. A free-form database differs from a knowledge-based database in that the latter must be able to interpret the knowledge to allow for limited reasoning, inference, specialisation or inheritance.

7.4 Database management

In order for a database to provide and secure efficient data storage and retrieval, there must be careful control of what data is to be included and how it will be maintained, who has access to the database, and how errors can best be prevented or resolved.

Normally, a specialist job function will be created for a *Database Administrator (DBA)*, the person who is responsible for looking after the computerised corporate database(s). The main responsibility of this person is to define the standard data elements and structure of the database (i.e. table and field structure). She will also look after the integrity and safety of the *contents* of the database, and may be responsible for the maintenance and fine-tuning of the DBMS software.

An important component of the DBMS is the *data dictionary* or *data encyclopaedia*. This describes in detail all data elements in the database, i.e. all data tables, field names (and synonyms or *aliases*), descriptions/definitions, field types (text, numeric values, monetary units, boolean, dates), field lengths, key fields, relationships and dependencies between fields and tables, etc.

The database approach distinguishes a number of different *views* or perspectives of the organisational data.

§ The *physical* view (or *data description*) of the data is concerned with where and how files, tables and records are stored. A large database may be split or *distributed* physically over different computers or storage devices; what may seem to be fields of the same record may actually be stored separately; often used records or fields may be stored differently to less frequently used data; various different indexing methods may be used to assist in locating records more quickly etc. The physical data view is the concern of the DBMS (and the DBA) only.

§ The *logical* data view or *conceptual schema* represents the conceptual structure of the data, as seen from an organisation-wide point of view. This would entail all the data fields that are required for each table, their data dictionary names and descriptions, their key fields and the relations between tables. System developers need to know what data is available in order to design their system, and users need it in order to define their information requirements. Most of this information is available in the DBMS's data dictionary.

§ The *subschema* represents the partial or limited view of the data by an individual user or application. A student marks processing application (or lecturer) should not be concerned with a student's personal or financial details, hence the DBMS will only enable access to the academic record components of the student database.

7.5 Database Architectures

Various models or *architectures* can be used to design an information system. This architecture is a decision that must be taken up front about the overall functioning of an information system. It affects the choice of hardware, software and the network configuration.

Although there are many ways to look at database architectures – indeed, the above database models are also an architectural decision – we will look at the decision which is related to *where* most of the data processing should take place

Most business systems have multi-user, networked databases. The user operates a workstation that requires access to the corporate database. The latter is normally located on a separate, larger computer. With the evolution in technology, specifically the increasing power of the user workstations, a decision has to be made about which computer will perform the bulk of the data processing.

7.5.1 Centralised processing

In a *centralised processing* environment, the entire processing load is performed by the central computer, usually a mainframe, to which a number of terminals are connected. This central and powerful computer is therefore usually referred to as the *host* computer. The host controls the data storage and the communication. The application program itself (e.g. a booking system) also runs on the host computer, which performs the various checks and does all or most of the data formatting. The terminal itself does almost no data processing, apart from the most basic screen display formatting instructions.

Although this model was very popular in the early days of mainframe computing, it is now very much on the wane. It is still found in travel agents (airline booking systems), libraries and Computicket offices.

7.5.2 Distributed processing

In the *distributed computing* environment, there is no single host computer that controls all of the organisation's data. All data resides on individual workstations or, more often, on the file servers of the LANs of the functional departments or local branches that actually need the data. These departments' computers are then responsible for *all* the data processing. This is the exact opposite of centralised computing. Distributed processing is made possible by the development of powerful microcomputers with high-capacity hard disks.

In reality, there are two versions of distributed processing. The more common approach is to *divide* the entire corporate database in smaller partitions or sections. These partitions are then distributed on the local network or data servers across the organisations. An alternative approach is to keep *local copies of the entire database* on the different local servers. On a regular basis, usually during the night shift, the servers contact head office and co-ordinate their changes so that, by the next morning, all distributed databases reflect the changes made throughout the organisation.

As an example, a national bank could split its customer account database into regional databases, perhaps one for each province. The transaction details for each account would be stored in the database of the province where each customer's regular branch is located. Only when a customer performs a banking transaction at a branch outside her home province, would there be a need for one regional database to make telecommunications contact with another.

7.5.3 The client-server model

The *client-server* model is a compromise between the two extremes of centralised and distributed forms of processing. Here the processing workload is shared between a central computer and the local workstation.

Generally, the central computer is responsible for maintaining the database. It runs the database management system and performs all related functions such as backup, data integrity, access control and security. This computer issues the required data to the various workstations that ask for it, and is therefore called the *server*. The local workstation, the *client*, receives the data from the server and is responsible for the presentation of the data. The term *client* can also refer to the software that resides on the local workstation i.e. the software responsible for communicating with the server and formatting the data for on-screen display and editing.

An ideal example of the client-server model is the World-wide Web on the Internet. Various *web servers* around the world hold data available for Internet users to see. Anyone connecting to the Internet gains access to the web by means of a *web browser*, which constitutes the client software. The web browser initiates the contact with the web server, who sends the requested data (a multimedia document in html format) to the client. The client's web browser then interprets the various formatting instructions found in the document to present it in the manner most appropriate for the client computer.

Thin and fat clients

Depending on the amount of processing done by the client, we can distinguish between a *thin* and a *fat* client. A *thin client* does little else than the formatting and display of the data; the real business application software, as well as the data management function, is performed by the server. This is quite close to the old centralised processing model.

At the opposite end of the scale is the *fat client*, where the application software resides on the client computer. All data is processed and checked locally and only the final data records are transmitted back to the data server. The server runs only the database management system software; all application software is run on the client.

Two-tier versus three-tier

The client-server model described above is the traditional *two-tier* (or two-level) model, with one large server connected to many clients. In a large organisation, it may not be feasible to split the processing load between the server and the client. What can be done if the database is very large and needs to be accessed by many users but needs to be kept centralised? If the organisation's applications are too complex to be delegated to the client, none of the above models will work. There is too much work for both the server and the client, but the need for one central database precludes the option of distributed processing.

This is where the case for a *three-tier* client-server model can be made. In the *three tier* model, an extra level (or tier) is inserted. At the highest level is the enterprise or central database server, which runs the data management system to perform all the pure data-related processing tasks. This server communicates with a number of intermediate, local servers that run the specific business applications. These servers are usually organised by functional department or geographic business unit. Finally, the client is responsible for the presentation and local editing of the data. (In some cases, more than three computers may be involved in data processing, in which case this is known as an *n-tier* model.)

A simplistic example is the situation of the inventory database of a large, national organisation. Because this data is vital to most functional units within the organisation, it makes sense for the database to reside on one central data server. Each of the applications which makes use of this database, such as billing, receiving, stock control, etc. could be located on the various departmental servers located throughout the organisation. One of these servers might even be a web server that encodes all data in such a way that it can be made available via the internet to customers who wish to check stock availability. Finally, each user will interact from his personal computer with the application residing on his departmental server.

7.6 South African Perspective

The Automated Fingerprint Identification Systems (AFIS) is a mobile system connected to the police database in Pretoria, that can track down a person's criminal record in a matter of seconds. The system is being used by police on the East Rand to identify suspected criminals, as well as to assist in recovering stolen cars. Assistant commissioner Gert de Lange commented in a newspaper interview that by enabling police to check 100 cars simultaneously, crime can be targeted with a minimum of inconvenience for motorists.

7.7 Beyond the Basics

Amazon.com has over 20 million customers and has revolutionized the bookselling business, becoming a household name almost overnight. Amazon.com succeeded because of the way it presents information to customers, not just because it was one of the first companies to sell books via the Web. In fact, it would be difficult to label Amazon.com a bookseller or even a retailer. It has no brick-and-mortar stores, no warehouses, and little inventory of its own. What Amazon.com does have is information (lots of it) about customers and what they've purchased, driven by a foundation of very, very large databases. Amazon.com uses this information to ensure that customers have a valuable online shopping experience. For example, Amazon.com proactively recommends books to customers based on previous books they've purchased, a profile they've filled out, and buying trends among customers who have similar profiles. The same information capacity is used to support sales of products such as CDs and clothes.

Amazon.com is an example of a new breed of company whose core business is information and whose chief competitive advantage is the sophistication of its information systems. Companies like Amazon.com are dramatically changing the business landscape because they know how to integrate databases, operational systems, and the Web to cement relationships with their customers, suppliers, and partners.

7.8 Exercises

7.8.1 Database management system

Pete Anderson owns a small chemist's shop in Wynberg, and recently replaced his cash tills with a PC-based sales system from Mikrotek Systems that makes use of a product price file, containing product codes, descriptions and prices, which is updated by his bookkeeper on a weekly basis.

Pete is so impressed by the accuracy and speed of this system, that he has been looking at other computer systems that might be suitable for his business. For a reasonable price he can buy a separate customer accounting system from Aardvark Computing, which will keep track of his customers' account details and balances, and print monthly statements. As part of the same deal, Aardvark will also provide him with a stock control system, to be used for ordering stock and keeping track of what is still available in his storeroom (i.e. has not been moved into the shop for display or sales).

As he explains it to you, stock orders would be entered on the **stock system**. When new stock is received, the quantity available in the store would be increased, and when stock is moved onto the shop floor, the quantity available in the store would be decreased. Below a certain stock level, new orders would be automatically generated.

The **sales system** accumulates totals of sales per day for each product, so that Pete will be able to identify fast-moving lines and move stock out of the storeroom as needed.

When a sale is made on account, the cashier will simply write the customer's account number on a copy of the sales docket, which the bookkeeper will then enter into the **customer accounting system** the following day.

Pete's son is studying IS, and says that it would be better to install a single integrated database system which combines all of his business data. Pete is not sure whether his son really knows what he is talking about, and has asked you to explain why using a database system would be even more efficient than the solution outlined above, for managing his stock, sales and accounts.

- Describe three examples of how business activities and record keeping could be enhanced by the use of an integrated database
- Outline two "queries" or reports that could be useful in managing the business, which combine data from two or more of the subsystems (stock, sales, customers).
- Design a set of database tables to be used in a new integrated database system that includes stock, sales and customers. Base your solution on the empty tables provided below.

<u>Note</u>: Pete needs to keep a record of account holders' names, addresses and telephone numbers. In addition, account sales to customers must be recorded in the database. Each sale

