Chapter 51 - Entity Relationship diagrams

51.1 E-R diagrams (Entity-Relationship diagrams) and normalisation

There are two approaches that database designers commonly use to come up with a design for a relational database. They are used together, even though at first you may think they are completely different approaches that produce different designs for the same system!!

1) **E-R diagrams.** The first approach is to produce an E-R diagram of the proposed system. The designer will ask themselves what obvious 'entities' exist in a system and how are they related. Entities are identifiable objects in a database about which you would store information. We have already seen lots of examples of entities such as Member, Dog, Breed, Undergraduate and Degree. Each of these entities requires a table to store real-life examples of that entity in (known as 'records'). Each actual record is stored in a row in the appropriate table. Each record is made up of 'fields'. A field is a piece of information you keep about an entity. In the Member's table in the previous section, fields included, Initial, Surname, Title, Sex and Postcode, for example. Database designers often also refer to the 'attributes' of a particular record rather than 'fields' but they mean the same thing. Notice that fields are the columns in tables. Once the designer has come up with an E-R diagram of logically linked entities, they can then go ahead and build the database.
2) **Normalisation.** A second technique the database designer can use is known as 'normalisation'. This has its roots in mathematical analysis and can produce a very efficient design. It involves identifying all of the possible attributes in a database and then applying a set of rules to them in turn. Each stage in the process of normalisation can result in a 'better' design.

51.2 E-R diagrams and normalisation together

Normalisation will produce database designs that can be shown mathematically to be the 'best' design. By this, we mean a design that minimises the amount of data redundancy. However, it may not necessarily produce the best design in terms of ease of understanding for humans! In practice, the designer will use both techniques together!

51.2.1 The first approach

1) The designer may well start a new design by producing an E-R diagram of the proposed system.
2) They might then produce a Data Dictionary that details what attributes make up each entity.
3) Finally, they might take each table in turn and 'normalise' it, to check that there are no data redundancy problems and that problems associated with adding and deleting records and amending data are removed. Normalisation in this case is used to **validate** the E-R diagram the designer has come up with.

51.2.2 The second approach

1) The designer may well start a new design by producing an E-R diagram of the proposed system.
2) They might then produce a Data Dictionary that details what attributes make up each entity.
3) They might then list all the attributes they have identified in the Data Dictionary and normalise them together. This will produce a set of related tables.
4) They then compare the design of the database using the E-R diagram they produced with the design produced by normalising the attributes.
5) They will decide which design they want to go with (if they are different). It may mean that they decide to go with the E-R diagram because it is an easier design to follow, or they may go for the normalised design because it is the best for eliminating data redundancy, or they may use a hybrid of both designs, based on the designer's prior experience. Whichever design is chosen, however, it is up to the database designer to fully justify it!!

51.3 E-R diagrams and their use in designing relational databases

An E-R diagram is a diagram that database designers use to show the relationships between groups of data (each group being known as an 'entity'). It gives a simple yet effective overview of the entities in a system and how they relate to each other.

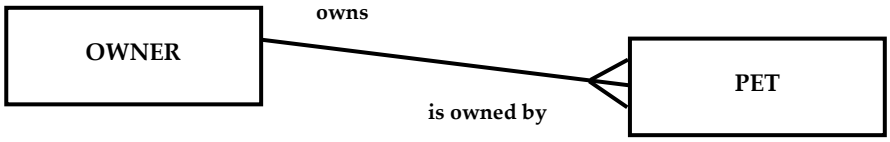
- It is a useful way of summarising what entities (tables) are needed in a relational database.
- It is a concise way of representing the relationships between records in tables. This can then be used both as a reference for the designer when setting up foreign keys and as a means of identifying 'many-to-many

relationships', which can then be 'resolved'. (See later in this section for further details about many-to-many relationships and why you need to resolve them).

- It can be used as a check to normalisation and a discussion aid. The results of both methods could be used together in design meetings. The design team can then discuss these how to go forward.
- It should be produced because it is a record of the design of the database and needs to be part of the technical manual. In the future, some maintenance of the database may be necessary. The person doing it (who may not be the original designer) needs to know the overall design and will expect to find the E-R diagram in the technical manual!

We will now look at the building blocks of E-R diagrams.

51.4 One-to-many relationships



In this database, there are two tables. One is called OWNER and the other is called PET. OWNER and PET are 'entities'. They are things about which we keep information, or 'attributes'. For example, the attributes we might store in the entity OWNER include ID, Surname, Contact phone number, Gender and so on.

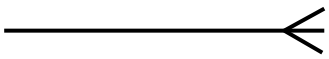
A relationship between 2 entities must be read in two directions. In other words, to fully describe the relationship between two entities, you need **two sentences, not one**. The process of reading E-R diagrams is very mechanical.

1) Always begin with the word 'Each'.
2) Add the entity name.
3) Add the verb or phrase closest to that entity.
4) Follow the relationship line to the other entity.
5) Write down whether it is a 'one' or a 'many'. (If it has three prongs then it is the 'many' side of the relationship. If it has one prong then it is the 'one' side.)
6) Finally, write down the name of the other Entity.

In the above example, the two sentences that describe the relationship between OWNER and PET are:

- each OWNER owns many PETs
- each PET is owned by one OWNER.

The line used to show a one-to-many relationship is this:



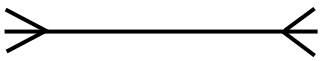
Note that the 'crow's foot' is the many side of the relationship.

An E-R diagram is constructed in the following way:

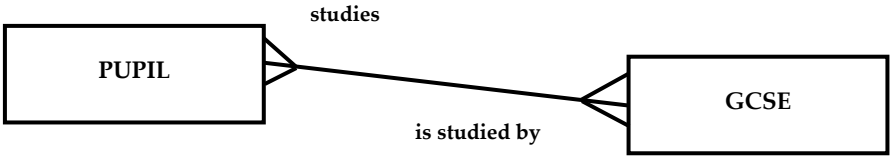
1) Each entity is shown in a box.
2) Entity names are always singular (PET not PETS, OWNER not OWNERS, for example).
3) The correct type of relationship line is drawn, the right way round! The 'crows foot' is the many side.
4) Verbs or phrases are added to aid understanding.

51.5 Many-to-many relationships

The symbol used to represent a many-to-many relationship is:



Here is an example of a many-to-many relationship:



Using the same method as before, we need to describe the relationship with two sentences:

- Each pupil studies many GCSEs.
- Each GCSE is studied by many pupils.

How would you implement this database? To help us understand the design, we will put some records in the tables, as before.

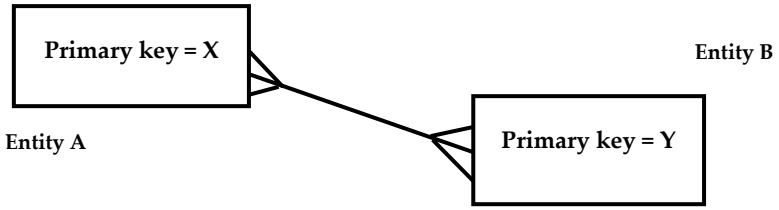
Pup_ID	Name	GCSE1	GCSE2	GCSE3	etc
1	Smith	1	4	5	
2	Jones	1	2	3	
3	Ali	2	5	6	
4	Potts	2	5	6	
5	Kanu	2	3	5	
6	Stay	4	5	6	
7					

GCSE_ID	Code	Pupil1	Pupil2	Pupil3	etc
1	Maths	1	2	11	
2	PE	2	3	4	
3	RE	2	5	15	
4	ICT	1	6	10	
5	D&T	1	3	4	
6	Eng	3	4	6	
7					

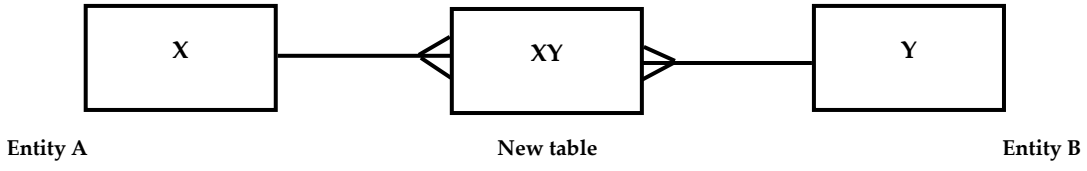
This is a far from ideal solution! For a start, how many attributes will you need in the GCSE table? You could have many indeed if there were lots of pupils taking a particular GCSE. More importantly, however, is that you (or rather Access or whichever software you are using) are going to have a problem recombining the two tables back into one! For example, a record in the GCSE table doesn't match to just one record in the pupil table. One GCSE will match back to many different records in the pupil table. This is not good!

51.6 Resolving many-to-many relationships

Fortunately, the solution is straightforward. It is known as 'resolving a many-to-many relationship'. If you have a many-to-many relationship, you can turn it into two one-to-many relationships by creating an extra linking table. In other words, this:



is converted into this:



If the primary key of an entity A is attribute X and the primary key of entity B is attribute Y, then the primary key of the extra table is a 'compound primary key' made up of X and Y.

- A **simple** primary key is a primary key that is made up of only one attribute.
- A **compound** primary key is a primary key that is made up of more than one attribute.

51.7 An example of 'resolving a many-to-many relationship'

Looking at some sample records for the Pupil-GCSE many-to-many relationship problem, we can see the following:

Pupil

Pup_ID	Name
1	Smith
2	Jones
3	Ali
4	Potts
5	Kanu
6	Stay
7	

Pupil_GCSE

Pup_ID	GCSE_ID
1	1
1	4
1	5
2	2
2	3
2	4

GCSE

GCSE_ID	Code
1	Maths
2	PE
3	RE
4	ICT
5	D&T
6	Eng
7	

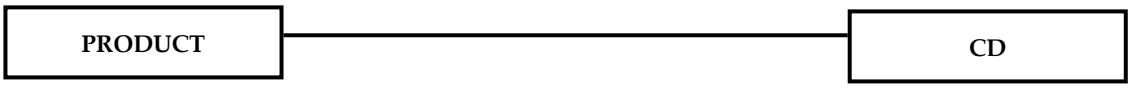
Reading the relationships, we have:

- Each record in the Pupil table (i.e. each actual pupil) has many entries in the Pupil_GCSE table. Have a look at pupil number one. She has three entries in the Pupil_GCSE table because she does three GCSEs.
- Each record in the Pupil_GCSE table (each row) relates to only one record (one pupil) in the Pupil table. Look, for example, at the last record in the Pupil_GCSE table (Pup_ID = 2 and GCSE_ID = 4). This record is related to only one record in the Pupil table. This particular record relates only to pupil number 2.
- Each GCSE appears in many different records (rows) in the Pupil_GCSE table. Look, for example, at the ICT GCSE. This GCSE appears twice in the Pupil_GCSE table. It appears in the second record and in the last one.
- Each record in the Pupil_GCSE table relates to just one GCSE. Look, for example, at the third record in the Pupil_GCSE table. (Pup_ID = 1 and GCSE_ID = 5). This record is related to only one record (one GCSE) in the GCSE table. It is related to GCSE number five, D&T.

We have resolved our many-to-many relationship. We store each pupil's details only once. This is good! We store the details about each GCSE only once. This is also good! We have a third table that matches up each pupil to every GCSE they are taking. This table has a compound primary key made up of the primary key from the pupil table and the primary key from the GCSE table. Each compound primary key combination is unique! (Each row is unique - you cannot have two rows the same). At any time, we can recombine the data from any related records from the three tables into one record!

51.8 One-to-one relationships

This type of relationship is shown with a simple straight line. An example of this kind of relationship is shown below:

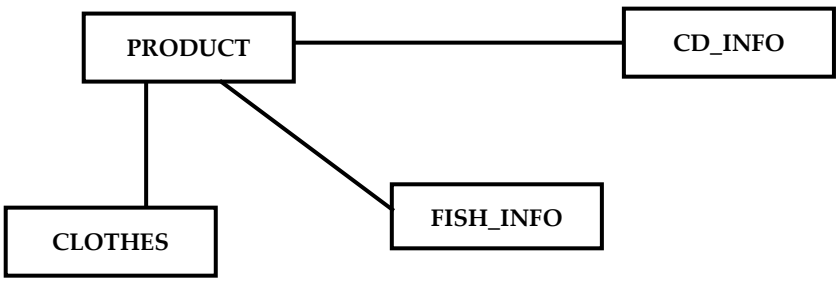


This kind of relationship is used to keep the database design compact and clear. For example, suppose you had a PRODUCT table for all products sold by a supermarket. Some of those products are CDs and some are, for example, fish! If you are keeping information (attributes) about the products, one possible set of attributes for the PRODUCT table might be the following:

PROD_ID	Name	Price	Country	Fresh / frozen	Gutted	CD_length	No_of_tracks

Of course, the number of attributes needed could be very large. This is because 'products' in a supermarket aren't restricted just to CDs and fish. They will include items such as clothes, insurance, cars, fresh vegetables and so on. Each of these products will need their own specialist attributes. For example, you might need an attribute to hold the size of clothes. Clearly, as you fill out this table with records of each actual product, some of these attributes will be left blank. For example, if you entered the details for ABBA's greatest hits, the 'Fresh / frozen' attribute will need to be left blank! So not only will you have a table with a great

many different attributes, many of them for a particular product will be left blank. This is not the clearest of designs! It would be better to split up the PRODUCT table into logical entities. One entity would be purely for CDs. One would be purely for fish. Another would be for clothes, and so on.



Now you can keep the *common* product information in the PRODUCT table (like ID, Name, Price etc). However, any specific information relating to a particular product type would go in that product's table. For example, if you have an ABBA CD for sale, the ID code, Name and Price would go in the Product table. There would be one entry in the CD_INFO table, where you would put the CD length, year of recording and artist. The result of using one-to-one relationships in a database is a clearer design.

51.9 Producing E-R diagrams from the beginning

Now you know how to interpret E-R diagrams, you can practice putting them together from the start. You would begin by investigating whatever system you have been asked to design a database for. You could do this (amongst other methods) by interviewing people individually, carrying out group interviews, collecting example documents, observing people using the system or sending out questionnaires by email, fax or post. The next step would then be to write down how the system works.

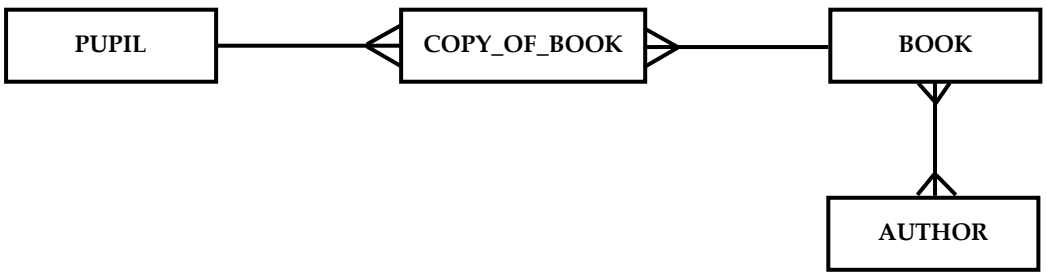
51.10 A library database example

A library contains books! Pupils take out up to six books from the library. There may be more than one copy of a particular book. Books are usually written by only one author but sometimes they are written jointly, by more than one author.

This is a typical description of part of a library database. The entities in this particular system, the 'objects' about which we will keep details might be identified in a first attempt as BOOK, COPY_OF_BOOK, PUPIL and AUTHOR. We know that:

- each pupil can take out many different copies of books
- each copy of each book can only be taken out by one pupil at a time
- each book may have many copies
- each copy of each book can only be about one particular book
- each book can be written by many authors (more than one author can contribute to a single book)
- each author can write many books.

Drawing a first attempt for the E-R diagram gives us:



51.11 A few more points to note …

It is important to recognise a number of things:

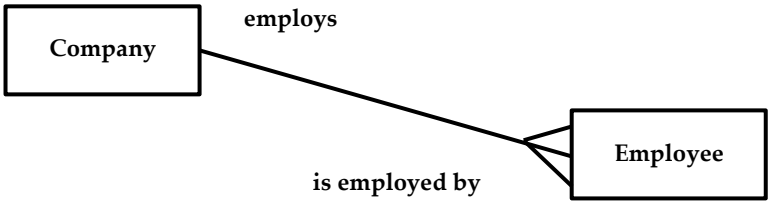
1) There is always more than one possible E-R diagram you can design that will do the job. The key is to be able to justify any particular design.
2) E-R diagrams 'evolve'. The process involves investigating a system and then designing a draft E-R diagram. This diagram needs to be thought about, discussed with other database designers or team members and reviewed. It then needs to be redrawn. The review process is repeated until everyone is happy! This is not a quick job! It takes time and experience to be able to design and review E-R diagrams effectively.
3) The E-R diagram only shows the designer what tables they need in the database and how those tables are related. It doesn't show the designer what fields will be in each table.
4) So long as tables are related, any data held anywhere within the whole database can be retrieved.

Q1. What does an E-R diagram show you?
Q2. What is the diagram to use for a 1-1 relationship?
Q3. What is the diagram to use for a 1-M relationship?
Q4. What is the diagram to use for a M-M relationship?
Q5. Why are M-M relationships not allowed in a relational database design?
Q6. How do you resolve a M-M relationship?
Q7. What is a simple primary key?
Q8. What is a compound primary key?
Q9. What is a foreign key?
Q10. After analysing the system and writing down a description of it, the following relationships were identified:

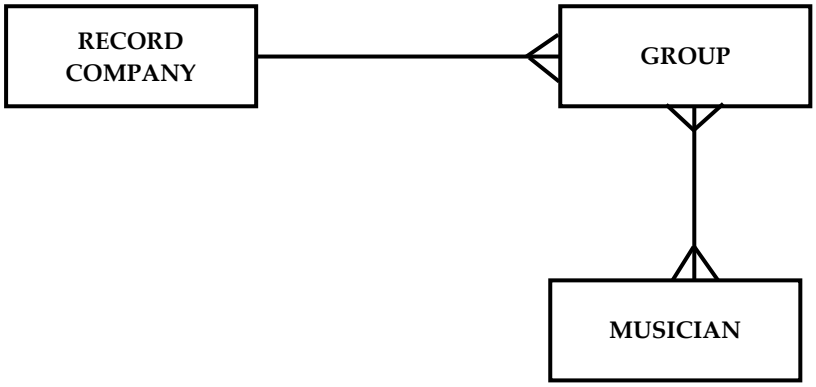
- Orders are places with a shop.
- An order can be for many different products.
- A product can appear on many different orders.
- Products are divided into two main categories, phones and accessories.

Draw the E-R diagram for the above analysis.

Q11. Write down the two sentences that describe this one-to-many relationship:



Q12. Look at the following diagram. Add some verbs/phrases in the correct places to help someone understand the diagram.



Chapter 52 - An introduction to Entity Life Histories (ELH)

52.1 Introduction to Entity Life History diagrams

Entity life histories are used to describe what happens to an entity over time. They describe the events that can happen to an entity. For example, if you had an entity PUPIL in which you stored all the details of pupils at a school, the kinds of the things that could happen to the table of records include:

- a record could be added. (when a pupil joins the school)
- a record could be deleted. (when a pupil leaves the school)
- a record could be amended (when a pupil changes address, for example).

52.2 Why use ELH diagrams at all?

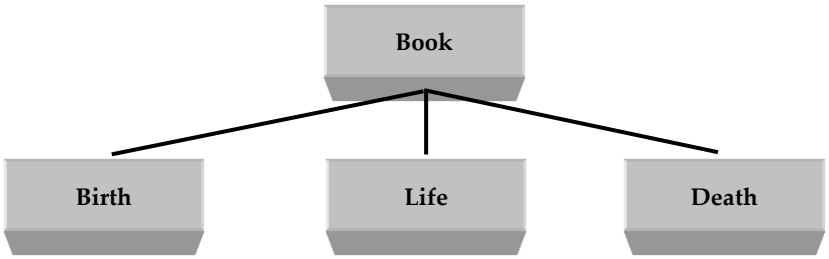
Entity life histories are important because they help you design into the database those forms and actions that are needed to service each event. For example, to cater for when a new pupil record is added, you will need to design an input form. If you have to output a report listing all pupils then you will need a query to do it and so on. ELH diagrams help a designer because they provide a checklist of functions that must be catered for. A function often overlooked by pupils when doing coursework is that of deleting data from a database. Many build systems that forever allow data to be entered but never removed! Had an ELH been drawn for each entity, they would have identified those entities where data 'dies'. They could then have designed functions for the user so that they could carry out data removal tasks at appropriate times.

An entity life history diagram should be prepared for each entity in a system. The diagrams used are based on the Jackson Structured Programming diagrams and generally need to show three main stages in the life of the entity:

- The Birth
- The Life
- The Death

52.3 An example of producing an Entity Life History diagram

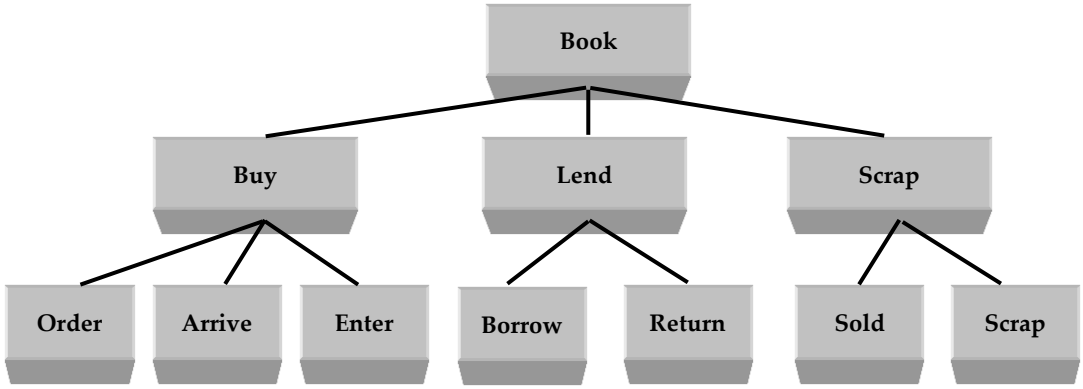
Consider a library. If you had an entity 'BOOK', you might start with the following diagram:



You might decide that the events that could happen to that entity are:

- it is ordered
- it arrives
- it is entered into the catalogue system ready for borrowing
- it gets borrowed
- it gets returned
- it gets removed from the library, either sold or thrown away.

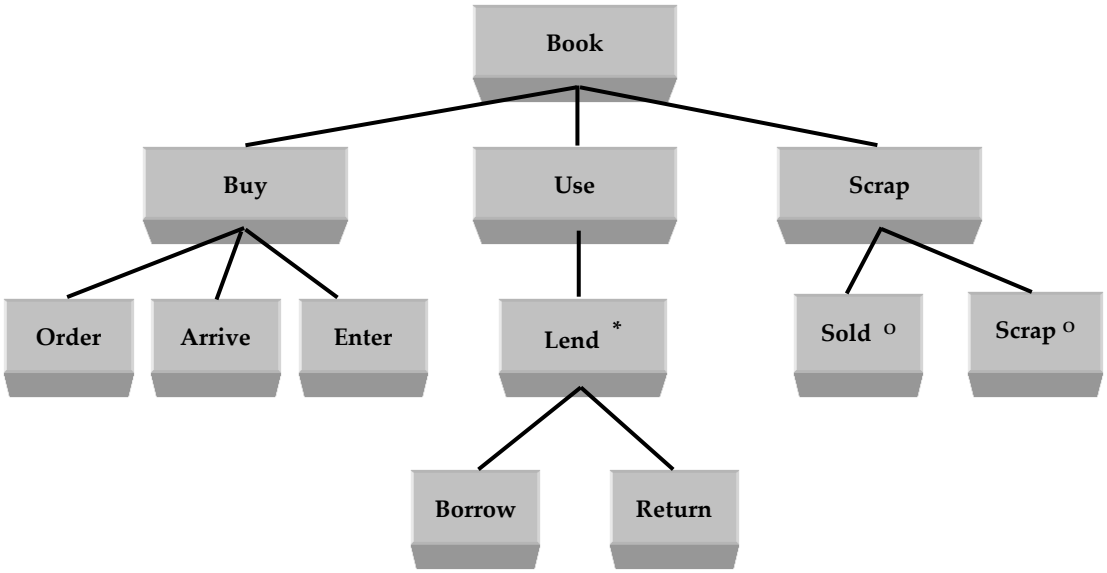
We will change the names of Birth, Life and Death to something more appropriate to libraries and books. A first attempt at the ELH diagram might be:



ELH diagrams use JSP notation. If you remember the work done previously on JSP diagrams, you will know that there are three important constructions that need to be shown on the diagram. These are **sequence**, **selection** and **iteration**.

- **Sequence.** The diagram needs to be read from left to right. Time passes from left to right so you can see each event happening over time.
- **Selection.** When a book is going to be removed from the library, one of two things can happen to it. It can be sold OR scrapped. Selection (this event OR that event) is shown by putting a small zero in the top right hand corner of the possibilities.
- **Iteration.** A book can be lent out and returned many times. This is iteration. It is shown by adding an extra box with a star in it and a comment that signals how the iteration ends.

Your diagram should now look like this.



Well, nearly! A common mistake when representing iteration is to miss something out. Look at the diagram above and see if you can spot what was missed out.

Q1. What does ELH stand for?
Q2. Explain why ELH diagrams are useful to a designer.
Q3. Draw how you show 'sequence' in an ELH diagram.
Q4. Draw how you show 'selection' in an ELH diagram.
Q5. Draw how you show 'iteration' in an ELH diagram.

Chapter 53 - More about databases

53.1 Different types of database

There are 4 classifications of databases that you need to know about. These are flat files, relational databases, hierarchical databases and network databases. We have already discussed flat files and relational databases in some detail but we will recap the essential points again and then briefly discuss the other two types of databases you should be aware of.

53.2 Flat files

When your database is kept in a single table, it is known as a 'flat file'. Each row in the table corresponds to one record. Each column in the table corresponds to a field. Flat files are excellent if you have only a very small, very simple database. They are quick and easy to set up and require little in the way of design skills. However, as the database grows in size and complexity problems develop. These include, amongst other things:

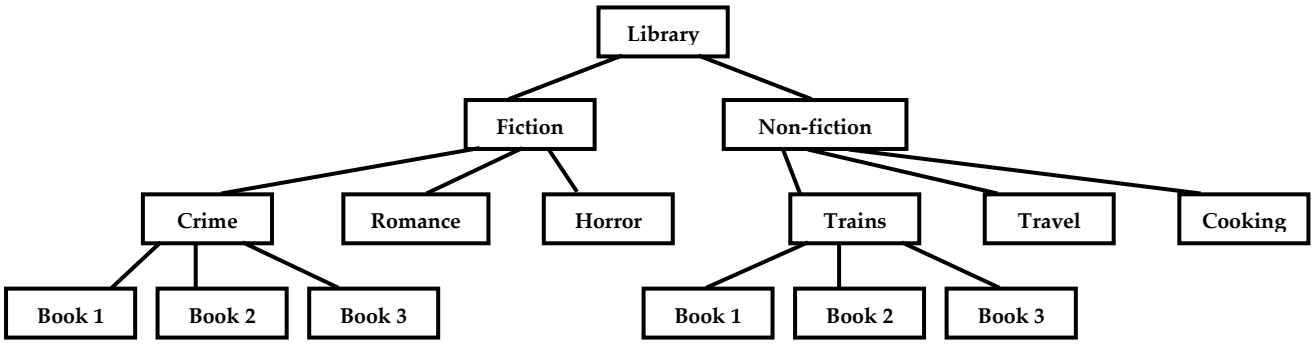
1) problems when you have to update records or data
2) problems when you need to delete records or data
3) problems of data duplication, known as 'data redundancy'.

53.3 Relational databases

When a database is kept as a number of related tables, it is known as a 'relational database'. Each table corresponds to a collection of records about one type of item, thing, or entity. For example, you may have the entity CUSTOMER that holds all the records about customers and another called ORDERS, which hold details about each order a customer has placed. These are then linked together using a common attribute known as a 'foreign key'. A foreign key in one table is a primary key in another table. There are quite a few reasons why you would want to organise your database as a set of related tables, not least of which is because it gets over those problems associated with flat files.

53.4 Hierarchical databases

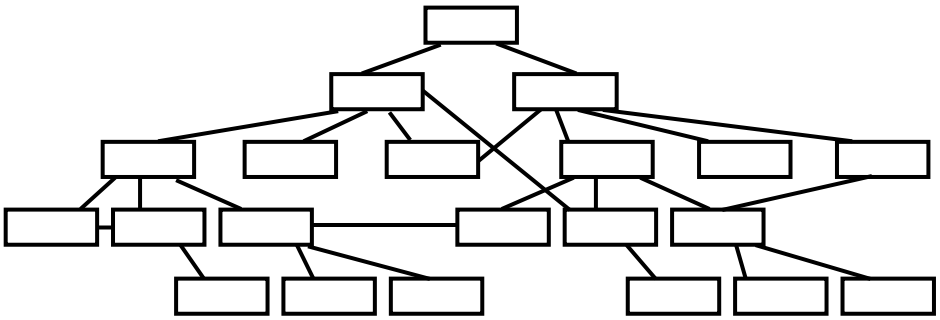
Records held in a flat file are held in a sequential order. In other words, the records come one after another. In a relational database, the records are also ordered sequentially within each table. A hierarchical database organises records differently, using a tree structure. Each record has attached to it a number of pointers. These pointers link one record to the record above and below it in the tree. As you go to a lower level in the tree, you can retrieve ever more detailed information. Here is part of a diagram of a hierarchical database as an example:



53.5 Network databases

These are similar to Hierarchical database. However, data can be provided with many pointers indeed, not just above and below a data item and this can allow very fast accessing to data because data can then be accessed up the tree structure, down the structure but also along the tree structure, too!

A generic diagrammatical representation of a network database is as follows:



Records are cross-linked as well as linked above and below in a network database.

53.6 Primary keys, foreign keys and secondary keys

Databases use primary keys, foreign keys and secondary keys. These are explained in detail below.

53.6.1 Primary Keys

- Every entity must have one attribute that is also known as the primary key. If an entity of records does not have a primary key attribute, then you must add one!
- A primary key is used to either find one record, or is used to sort a file in primary key order.
- The primary key, usually an ID number of some kind, is unique, for each record. You cannot have two records with the same primary key value.
- Many primary keys are made up of just one attribute. There are lots of circumstances, however, when the primary key is made up of 2, 3 or even more attributes. It is then known as a compound primary key. If you have normalised a database to 3NF, you will probably have seen an entity with a compound primary key.

53.6.2 Foreign Keys

- Foreign keys are used to link entities.
- A foreign key in one table is a primary key in another table.
- Although primary keys cannot have duplicate values in a table, foreign keys most definitely can. You have a situation where the same attribute cannot be duplicated in one table but can be duplicated in another!
- When you have a one-to-many relationship between two entities, you will need to link them using a foreign key. To do this, always copy the primary key from the entity on the 'one' side of the relationship and put it in the table on the 'many' side. In the table on the 'one' side, it is known as a ‘primary key’. In the table on the 'many' side, it is known as a ‘foreign key’.

53.6.3 Secondary keys

Very often, you may want to access a database by an attribute other than the primary key. Consider a textbook. Most of the time, a reader will want to access a particular topic in a book, so there exists a main contents page to allow them to do this. Sometimes, however, they want to look up specific things, so an index at the back of the book is provided. And sometimes, however, the reader may be more interested in looking at all the pictures! To enable this to happen, another index is provided - an index of pictures and where to find them! In this example, if a reader wanted to look at all the pictures, they could still go to the contents page and use that if they wanted to but it would take a long time, especially if it was a thick book. Now they have a secondary index file they can get back just the pictures whenever they want - and quickly. Secondary keys, then, enable a user to access data in an order other than the primary key order and they allow a user to filter what data to display. New secondary indexed files are created because a different set of information is required (possibly in a different order) from the main set of records frequently enough to justify doing it. If a particular set of information were needed once in ten years, it wouldn't be worth the overheads of setting one up.

53.6.4 Another example of the use of secondary keys

For example, imagine a bus company that stores bus details. It might have in a typical record the bus code, departure and arrival points, times of arrival and departure, number of seats in the bus and drop-off points. You could easily get a list of all the buses, listed by the primary key (the bus code) from the main file. But when someone goes to the bus company to buy a ticket, they just want to know only when the buses leave from their town to London. Now this is a question that the bus company might get asked one hundred times a day! They could run a query. They would need to set one up and then check the record of

every bus. If they had 20000 buses, checking each of them would be time consuming, especially if they had to do this so many times every single day! So instead, they set up a secondary index file, which only has buses from the hometown to London and the time of departure. It will be ordered according to a secondary key that has been set up. The result of using a secondary index file with a secondary key is faster access to the information the customer wants!

53.7 Access Rights, Views and the role of the DBMS.

Consider a relational database in a company. It is one database of related tables that serves the entire business. However, you do not want everyone having access to all the data in the system at all times from all computers. It would be a breach of the Data Protection Act for one thing! You would not want your personal details available to anyone who felt like looking them up for another. In addition, Information in companies, especially financial information, is valuable and therefore needs to be restricted. You want to be able to limit who can access what, at what times and from where. For example:

- The receptionist should be able to see who works for the company and his or her telephone extension but not personal details of each employee.
- The personnel manager should be able to view, add, delete or amend an employee's details, but not be able to read the sales figures for a company.
- A salesman should be able to view all orders, create new ones and view his own sales figures, but not view other people's sales figures, or the total sales figures.
- The senior management should have the right to view any salesman's sales figures.

53.8 The role of the DBMS

Restricting access to data is done through the DBMS (See DDL in the next section). This is a piece of software that, amongst other things, provides an interface for the database administrator to allocate 'views' and 'rights' to individuals or groups of individuals. There are a number of things an administrator could do to a user or group of users using the DBMS.

1) A user or group could only be allowed to 'view' certain records but not others.
2) A user or group could be allowed to both view certain records and modify them but not others.
3) A user could be allowed to delete certain kinds of records but not others.
4) A user could be allowed to execute certain programs but not others.

53.8.1 How does the database know who is at the computer and what rights they have?

Each user is given an account. An account consists of a user ID and a password. When a user correctly logs on using the User ID and Password, their rights and views are accessed by the DBMS. They will only be able to get to the views of data that they have been allocated with and will only be able to use the rights they have been given. It is possible for individual files to have extra security, for example by being password-protected. The user will then have to correctly enter a password to access those files. This is useful to prevent casual access to a computer that has been left unattended but not correctly logged out. As an additional security method, users can be logged out automatically after a period of inactivity.

53.9 Locking

Databases occasionally need to be 'locked'. This is important when multiple users potentially have access to the same file at the same time. For example, if a file is being updated by User A, you don't want User B accessing the file until User A has finished with it. Reserving a file is known as 'locking' and ensures that the integrity of the file contents is not compromised. An example of a file needing to be locked is in a real-time booking system. You wouldn't want two different booking agents who book cinema tickets booking the same seat at the same time! This will lead to overbooking. One of them must have the right to modify the file and to book tickets. The other agent should have the right to look at the current seat availability, but not the right to book seats, until the file has been updated and released as a result of any bookings taken by the first booking agent.

53.10 More on the Database Management System (DBMS)

If you have used Access, or any other main relational database software, then you have used a DBMS. In Access, the kinds of things you will have done are varied. You may have used the special window to design tables, or you may have designed them using SQL. You might have entered data directly into the tables in 'data sheet view'. You could have spent some time designing forms and entered the data in that way. You may have used QBE to design a search for information or again used SQL. Whichever way you used, you probably presented the findings using a report. All of these functions are provided by the DBMS.

In general, a DBMS is a piece of software that lets you create, manipulate and maintain a database. It lets you:

- set up tables of data about data (also called 'meta-data' and the 'data dictionary'.)
- create forms to enter data
- create accounts and rights for users and groups of users
- do queries on data
- present data in various report formats
- have access to a programming language so that you can do extra-clever things
- set up different 'views' of data.

Above all, the DBMS provides users with a 'conceptual view of data'. In other words, it provides the user with a user interface that they can understand and relate to, for example, a picture of tables of data related to each other (an E-R diagram) or a table of data showing the name of each piece of data, their data type, the validation rules and so on. Data is not stored like this on disk! In fact, the data might be stored in indexed files, linked lists or sequential files, for example. This is the 'physical view of the data', which humans cannot understand. To make use of the data on disk they need it to be interpreted and shown in a form they can understand. This is what the DBMS software does. It provides a user interface!

Broadly speaking, there are two main functioning parts to a DBMS. These are the DDL and the DML.

53.11 Data Definition Language (DDL)

The Data Definition Language (DDL) is one part of the DBMS. It is used by the designer of a database rather than a user. The DDL is used to set up the structure of a database as well as for maintaining the database. For example, using the DDL, they can set up tables of data about data (names of attributes, data types, validation rules etc). The data about data is often called '**meta-data**'. The DDL is also used to set up the security systems in the database. This is done using the DDL by describing the whole database (the 'schema') and then describing data in *parts* of the database (the 'sub-schema'). Once this has been done, the designer can control access to each of the sub-schema. For example, if a receptionist needs to know each employee's department, extension and job title, then a sub-schema will be defined with these data items in it. The receptionist is then given access to that sub-schema (she is allowed to view that data on her screen) but not other sub-schema. That means she cannot, for example, access how much an employee earns, or the contents of their personnel file. Using the DDL, then, different users of a database can be allowed to access certain sub-schema but not others. Users can be given different '**views**' of the same data in a database.

53.12 Data Manipulation Language (DML)

The DML is another important part of the DBMS. It is commonly used by the users of a database to access and update data. This language allows users to actually work on the data in your database. They could, for example, write programs that add, delete, modify or retrieve data! The DML is a very high level language. This ensures that minimal programming skills are necessary to actually use it. You will see and get some experience of a DML when you use SQL in the next chapter.

Q1. Summarise the benefits of a relational database compared to a flat file.
Q2. What is meant by 'data redundancy'?
Q3. Describe how a hierarchical database organises data.
Q4. Why are secondary keys used in databases?
Q5. What does DBMS stand for?
Q6. What a DBMS allow you to do?
Q7. What is a 'view' of a database?
Q8. What does the DDL in a DBMS do?
Q9. What does the DML in a DBMS do?
Q10. What is SQL?

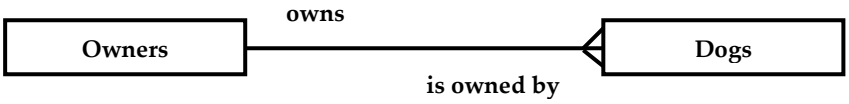
Chapter 54 - Querying a database

54.1 Introduction

SQL, or Structured Query Language, is a language that is used for gaining access to and manipulating tables of data. It is a 'standard' language in that it has been defined by the American National Standards Institute (**ANSI**) and so will work with many different applications such as Access, Informix and Oracle, for example. Although there are a number of different SQL versions in existence, they all support the basic ANSI standard and so all can carry out certain functions in the same way, such as SELECT and DELETE. We will see examples of these later. You should note that the instructions that make up SQL are part of a database management system's **DML**, or Database Manipulation Language.

54.2 A dog club's database

Here is a design for a dog club.



An E-R diagram for a dog club database.

We must describe any **one** E-R relationship with **two** sentences. In this case, we have:

- each owner can own many dogs
- each dog can be owned by only one owner.

Here are the two tables with some record in.

tblOwners

ID	Title	Surname	Phone No.	Registration date
1	Mr	Smith	091 123456	12/10/1996
2	Mr	Jones	081 654321	01/03/2000
3	Miss	Patel	081 222222	14/08/1999
4	Mrs	Tung	030 666666	21/08/1999
5	Mr	Fredricks	030 888888	29/11/2002

tblDogs

ID	Name	Date of Birth	Type	Competition wins	Owners ID
1	Red	21/09/1998	Poodle	4	1
2	Capper	01/10/2000	Greyhound	0	1
3	Alisister	20/02/1999	Spaniel	1	3
4	Bluey	15/05/1999	Spaniel		1
5	Patch	15/05/2002	Poodle	0	4
6	Whitefoot	24/03/2001	Alsatian	1	4
7	Twinkletoes	01/02/1995	Bulldog	7	5
8	Fluffy	18/03/1996	Spaniel		5
9	Carpin	19/05/1997	Bloodhound	12	1
10	Mischief	22/06/1999	Spaniel	1	5
11	Lappy	02/08/1999	Spaniel	6	3
12	Smoz	12/06/1998	Alsatian	2	3
13	Bluenose	26/01/1996	Poodle		3
14	Flipflop	11/05/1999	Poodle	3	1

54.3 Using Access to do the examples

You must get some actual experience using SQL to really understand it. Use Access or any other relational database package. To try out the following examples in Access 2000, do the following.

1) In the database window (The database window is the screen you get when you first open your database, where you can see a window with a list on the left that says tables, queries, forms reports, pages etc.) click on **Queries**, **New**, **OK** and then don't **ADD** any tables. Just **CLOSE** the pop-up box.
2) Now click the **design view icon**. It is probably situated under the **FILE** menu.
3) An SQL box should pop-up and you are ready to type in your queries. If you get stuck getting started, you will need to ask your peers or your teacher for help or refer to the HELP menu in Access or use the Internet.

IMPORTANT: When you do each query, you must be sure that you get the correct results. Just because you get some results does not mean they are correct. Whilst you are learning how to construct queries, you should always:

1) **Predict** the results of running a query before you actually run it.
2) **Run** the query.
3) **Compare** your prediction to the actual results obtained.

54.4 Using SELECT - FROM

Type into the SQL box the following command:

```
SELECT Name
FROM tblDogs;
```

Run the query (in Access, by pressing the exclamation mark) and you should see a list of the names of the dogs. To go back to SQL view, simply right-click on the results of the query and then select SQL view. The **SELECT** command is used to select columns to display in a table. The **FROM** command is used to pick the table(s) you want to get the columns from. You can select more than one column from a table and more than one table. Try running this:

```
SELECT Name, Type
FROM tblDogs;
```

You should see a list of names of dogs and their types. Try this:

```
SELECT Name, Type, [Date of Birth]
FROM tblDogs;
```

You should get a list of names of dogs, their type and their data of birth. Note the use of square brackets, used because the field name has spaces in it. You can display all of the fields in any particular table easily by using the **wildcard** symbol, *. Try this:

```
SELECT *
FROM tblDogs;
```

You will see all of the fields in all of the records in the dog's table displayed. You can also display fields from more than one table in the same query. Try this:

```
SELECT tblDogs.Name, tblDogs.Type, tblOwners.Title, tblOwners.Surname
FROM tblOwners, tblDogs;
```

Notice the syntax in the SELECT statement. For example **tblOwners.Surname** means 'go to the table called **tblOwners** and get the column called **Surname'**.

You can experiment in Access easily by designing queries in QBE (Query By Example). QBE is the graphical way of designing queries in Access. You can check that they work and then switch to SQL view to the SQL code (use the 'design view' icon under the FILE menu). You can also write queries in SQL and then see the equivalent in QBE. For example, if you design a query in QBE to select the names of dogs and their types and who owns them. When you convert this to SQL, you will get the following:

```
SELECT tblDogs.Name, tblDogs.Type, tblOwners.Title, tblOwners.Surname
FROM tblOwners INNER JOIN tblDogs ON tblOwners.[Owner ID] = tblDogs.[Owner ID];
```


This SQL code is different from the code we successfully used in the previous example. The relationship has been defined in the SQL code. It is important to get into the habit of defining the links between tables. **Take note of this example and use it in any future SQL queries where you need data from more than one table.** If you don't, you *may* get strange results!

54.5 Using SELECT - FROM - WHERE

You can search your tables according to some criteria specified in the **WHERE** statement. There are lots of criteria you can use.

Search criteria	Meaning	Example
=	Equals.	Type="Spaniel" [Competition wins]=1
<>	Not equal to.	Type<>"Poodle" [Competition wins]<>1
>	Greater than.	[Competition wins]>0
<	Less than.	[Date of Birth]<#01/01/2000#
>=	Greater than or equal to.	[Date of Birth]>=#01/01/2000#
<=	Less than or equal to.	[Registration date]<=#01/01/2000#
BETWEEN AND	Between two values.	[Competition wins] BETWEEN 2 AND 4
LIKE	Similar to	Name LIKE "C*"
IS NULL	No data	[Competition wins] IS NULL
AND	Satisfy both criteria.	Type="Poodle" AND [Date of Birth]>#01/01/2000#
OR	Satisfy either criteria.	Type="Poodle" OR Type="Spaniel"
NOT IN	Do not include.	Type NOT IN ("Spaniel")

Note that dates in Access must be surround by hash symbols. **NULL** does **not** mean **zero**. It means 'no data'. There is a big difference! There are other criteria! Try out the following SQL queries. Remember to **predict** the results *before* you run the query, then **run** the query and then **compare** the prediction to the results.

```
SELECT [Name], [Type], [Date of Birth]
FROM tblDogs
WHERE Type="Poodle" OR Type="Spaniel"

SELECT Name, Type, [Date of Birth]
FROM tblDogs
WHERE [Competition wins] IS NULL;

SELECT tblDogs.Name, tblDogs.Type, tblOwners.Title, tblOwners.Surname
FROM tblOwners INNER JOIN tblDogs ON tblOwners.[Owner ID] = tblDogs.[Owner ID]
WHERE Type="Poodle" OR Type="Spaniel";
```

Don't forget that if you are getting data from more than one table then you should link them properly in the **FROM** statement.

54.6 Using SELECT - FROM - WHERE - ORDER BY

You can sort the results easily, either in ascending or descending order by any field. For example:

```
SELECT tblDogs.Name, tblDogs.Type, tblOwners.Title, tblOwners.Surname
FROM tblOwners INNER JOIN tblDogs ON tblOwners.[Owner ID] = tblDogs.[Owner ID]
WHERE Type="Poodle" OR Type="Spaniel"
ORDER BY tblOwners.Surname;
```

This will order the results in ascending order by the owner's surname. The reverse order can be obtained like this:

```
ORDER BY tblOwners.Surname DESC;
```

54.7 Using SELECT - SUM - AS - FROM - GROUP BY

This command can be used to find the totals for each unique entry in columns in tables. For example, suppose you wanted to know how many wins each unique owner in the dog club had in total. You would do it like this:

```
SELECT tblDogs.[Owner ID], Sum(tblDogs.[Competition wins]) AS [Total number of wins]
FROM tblDogs
GROUP BY tblDogs.[Owner ID];
```

This SQL command adds up the competition wins for each owner in the tblDogs table. It then displays the total number of wins for each owner under the heading 'Total number of wins'.

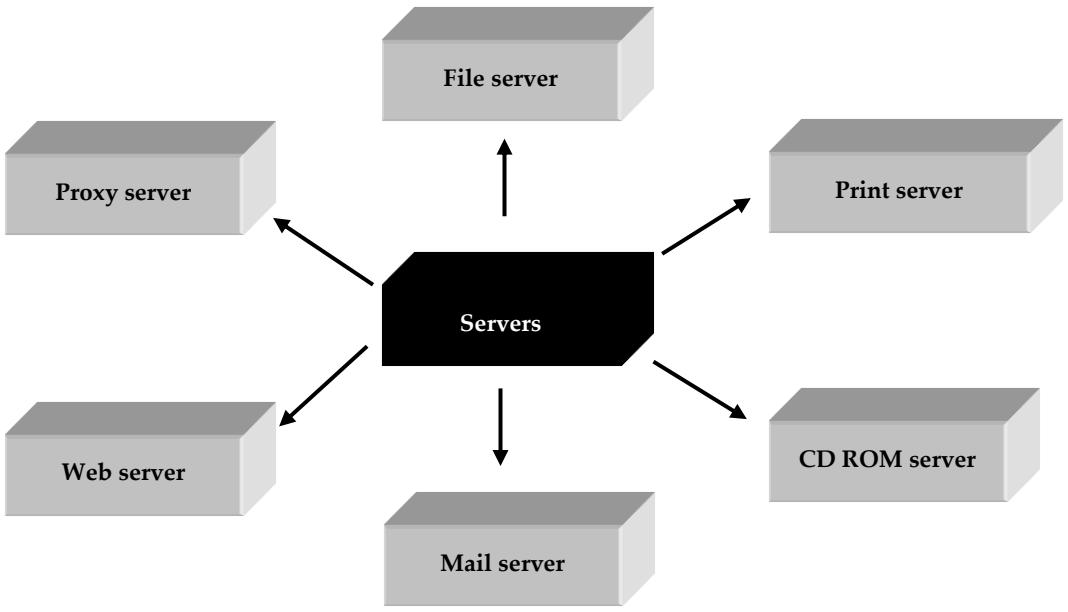
Q1. What does SQL stand for?
Q2. Which part of the DBMS does SQL form part of?
Q3. What is a relational operator?
Q4. List six maths relational operators.
Q5. What is a logical operator?
Q6. List three logical operators.
Q7. What is meant by 'Null'?
Q8. What SQL command picks out the tables to use in an SQL query?
Q9. What SQL command picks out the fields to display in an SQL query?
Q10. What SQL command puts the results of an SQL query into descending order?

Chapter 55 - Network systems

55.1 Network hardware - servers

A client-server network is a way of organising computers (clients) so that they can make use of the resources of one or more servers. A server is simply a computer that has resources that can be used by clients. There are different types of server.

- **A file server.** This is a computer that stores all of the users' data files. The NOS (Network Operating System) allows accounts to be set up. Users have to enter in a user ID and password to gain access to the network and to their own files. This system also means that users cannot access other people's files.
- **A print server.** This is a computer that allows clients to have access to a shared printer. Print servers usually come with spooler software. Spoolers collect any jobs that have been sent to the printer, queues them and then sends them to the printer. If you sent a file to be printed straight to the computer rather than the spooler program, your computer would slow down because the printer works much slower than the computer. The spooler program frees up an application so you can carry on working.
- **A CD-ROM server.** This is a computer that has many CD drives (or one drive that can change CD-ROMs automatically). Clients can access the information from any of the CD-ROMs currently in this server.
- **A mail server.** This manages mail into and out of a network.
- **A web server.** If a business wants to have an Intranet or wants to host its own web site then it will need a web server. This is responsible for storing web pages and distributing them in response to requests.
- **A proxy server.** The proxy server acts as a security buffer between the main file server and the users on a computer system. A user requests data held in a file server. The proxy server intercepts this request. The proxy server checks that the user is a valid user and is making a request for data that he/she is entitled to. If the user passes these checks, the proxy server gets the requested data from the file server and passes it to the user. The user cannot access the file server directly. They must go via the proxy server. Proxy servers are mainly used as a security buffer between a computer network and the Internet, to stop threats from the Internet entering the computer system.



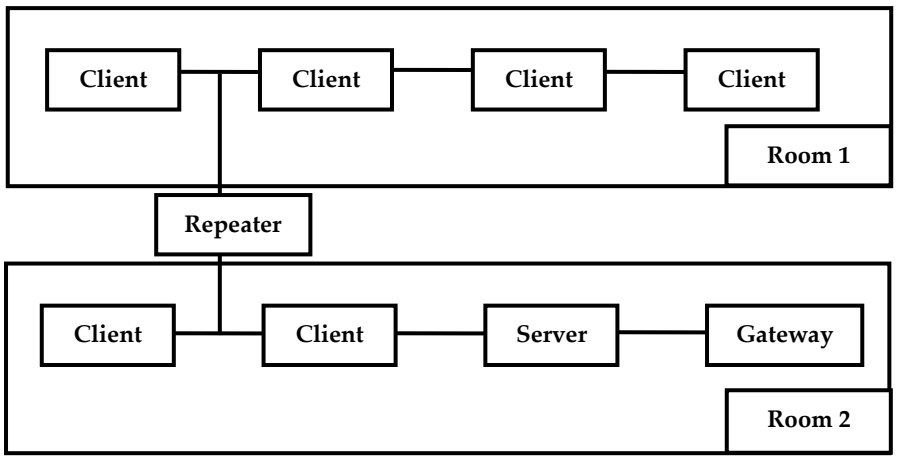
Different kinds of servers.

55.2 Network hardware - additional equipment

In addition to clients and various kinds of servers, a network will also need additional equipment. You should read widely to understand fully what the components that make up a network actually do because their definitions and functions are blurred and are constantly changing! If in doubt as to what any particular piece of equipment does then stick to the definitions given by the British Computer Society!

55.2.1 Repeater

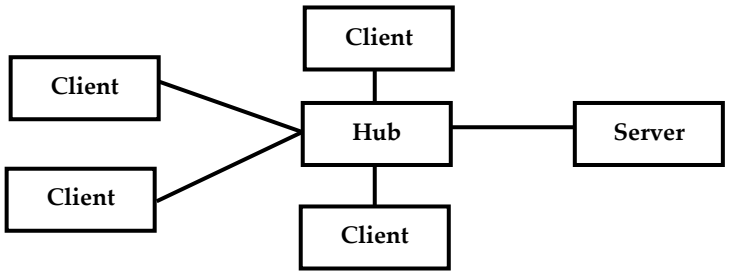
Repeaters can be used when you want to extend a network but where there may be problems with the strength of data signals. This would be because the cable lengths between different parts of the new network have become very long; data strength (i.e. voltage) drops as the distance increases.



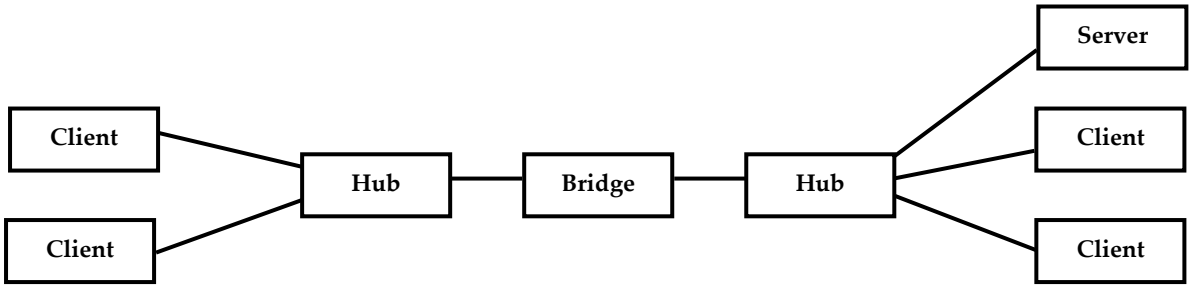
Two rooms are to be connected up. They are a little far from each other. The signals going between the two rooms are weak and so a repeater has been used to boost the signals.

55.2.2 Hub

A hub is used to connect many computers to one place. For example, in a star network, all the cables from each individual computer go back and connect to a hub. The hub then connects to the server. Hubs can also boost signal strength if needs be.



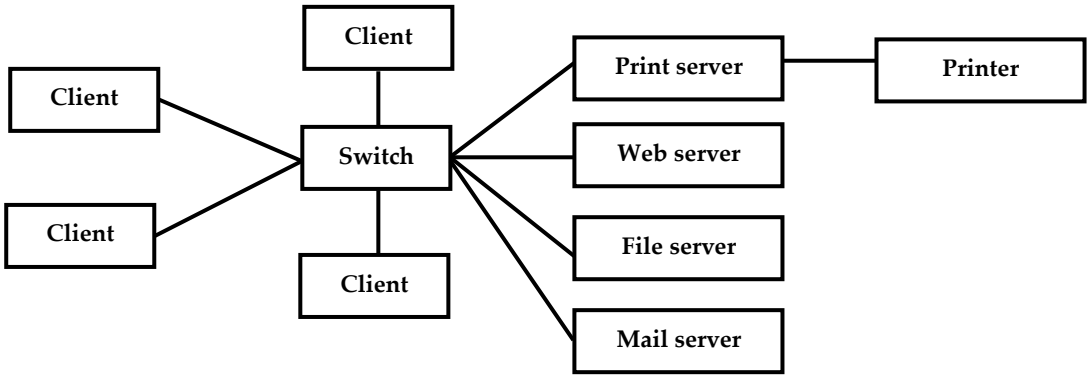
Four computers are connected up in a star formation. A hub is used to provide a physical connection for each computer to the network. The hub is then connected up to the server.



Four computers are connected up using hubs. This network has been 'segmented' using a bridge. (See the Bridge section later).

55.2.3 Switch (commonly used in 'switched Ethernet' networks)

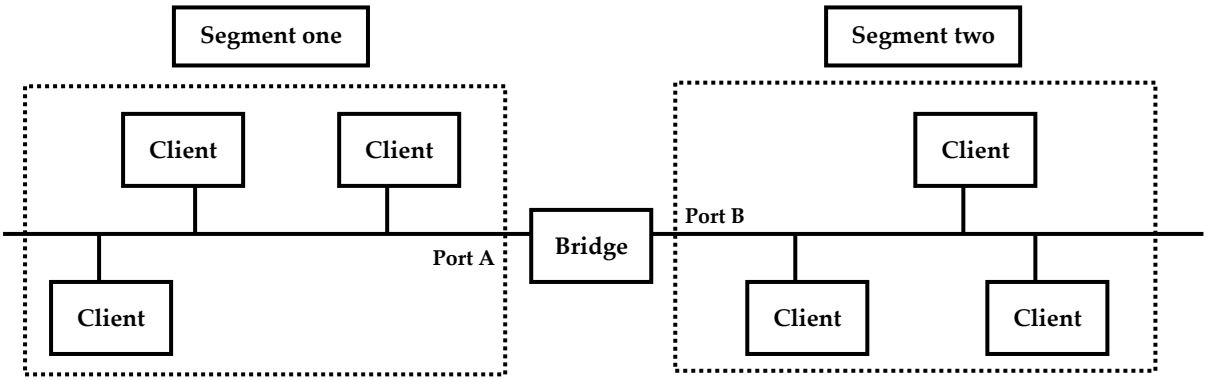
A switch is a more 'intelligent' hub. It can set up communication paths between different clients and different servers at the same time! If you have large files to transmit, or a large volume of data, then switches would be more appropriate than hubs. See later in this chapter the section on Ethernet LANs.



4 web designers in a small agency share both small and massive data files. They send and receive emails (sometimes with attached video files), download animations, upload large web pages, save and retrieve large files and print a range of files. The star topology has been selected because it is both fast and secure. A switch rather than a hub has been chosen because a switch will efficiently direct files to different servers, making multiple connections at the same time. If all 4 designers are sending and receiving a range of files at the same time, the switch will work out the best way, the most efficient way, to make multiple, concurrent connections to the different servers.

55.2.4 Bridge

A bridge connects two **similar** LANs together. Users think it is logically one LAN even though it is physically two! Bridges enable the users of one network to use the resources of the other. They are also used to split up one large network into 'segments'. Consider part of a bus network design:



A segmented bus network.

Segments are simply parts of the whole network. Networks are often split into segments because there is too much traffic on the whole network. By splitting the network up, you are keeping communications (the packets of information that move around the network), into the areas where they are commonly used. For example, segment one might be one department whilst segment two might be a separate department that generally never communicates with the first one. Packets are prevented from accessing the computers where communications rarely take place, unless the packets are actually addressed for the other segments. This will reduce the number of packets moving around the whole network and therefore speed up the network. It will also reduce the number of 'collisions' that occur. The cable on the bus network can only carry one packet of data at a time. If two different PCs see that the cable is free and both try to put a packet on the cable at an identical time then the packets 'collide'. Luckily, the computers can detect when this happens and they simply resend the packets.

A bridge has some 'ports'. These are where connections to different segments are made. The bridge works out and remembers which PCs are connected to which port. When a packet of data is sent from a PC in segment one to another PC in that segment, the bridge blocks it from crossing over into the other segment.

55.2.5 Gateway

These are 'more intelligent' bridges. If you have two separate networks, either LANs or WANs, that are each set up in **different** ways using different protocols then you cannot simply connect them together and pass information between them using a bridge. They will need to be plugged into a gateway first. Gateways convert data from one network's format into the other, so that each network can send and receive information to the other, usually using the public telephone network. LANs wanting an Internet connection (i.e. to a WAN) typically use gateways.

55.2.6 Network card (or Network Interface Card – NIC)

Each computer that is going to be part of a network needs to be actually connected up to the network. To do this, you will need to buy and fit a network card into an expansion slot in each computer. The network card inside a computer provides:

- a means of connecting a computer to a network
- a way of splitting up data that you want to send across a network and then getting that data actually onto the network successfully
- a way of collecting 'packets' of data addressed to that particular computer from the network. Each network card has a unique identifier called a 'MAC address' that enables this to happen.

55.2.7 Router

A router takes a 'packet' of data, looks at its IP address and then redirects it, or routes it, to the correct destination on a network. The network might be a LAN or WAN, such as the Internet.

55.3 IP addresses

When you send a message on the Internet, it is split into packets. Each packet is then given some extra information, such as how to reassemble all the packets that make up a message, the sender's address and the destination address, known as IP or Internet Protocol addresses. A typical address looks like this: 23.101.10.114. The packets are then sent on the Internet, each one going by the best route available at the time, being constantly re-routed towards the destination by the routers on the networks that make up the Internet. In fact, each packet in a message might arrive at the destination by completely different routes. This is why the Internet is known as a 'packet-switched network'. But how does each packet 'know' which route to take? This is where the routers on the network come in.

55.3.1 Routers on the Internet

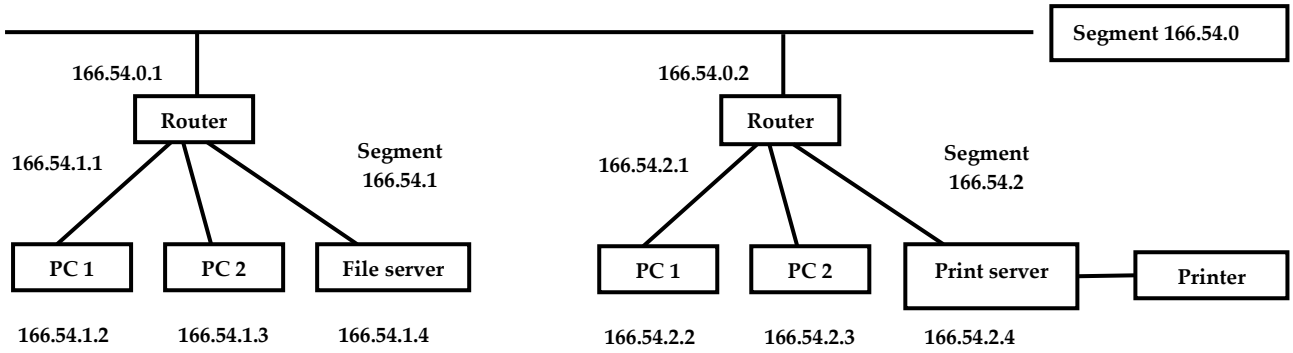
Each LAN connected to the Internet has a router. Consider your school's network. When you send a message or request a web page, you are sending/receiving packets of data. The router on the school network exchanges information with other routers on other networks on the Internet. They tell each other how busy a communication connection is between the routers, how quickly packets are being moved along a section of the line and whether there are any problems on it, for example. The router can then look at and evaluate the information it has received from the other routers and make decisions about which way to send a packet! Routers are key to the workings of the Internet. Without routers, the Internet could not work! You can see for yourself the route that a packet takes using the 'tracert' DOS command! Go to the DOS prompt on your computer and at the C:\WINDOWS prompt type in: **tracert www.theteacher.info** and press Enter. You will then see a list of:

- all the routers used in transferring a packet between the site and yourself
- the times a packet takes to travel between your computer and the router
- the name of the network router
- the IP address of the network router.

55.3.2 More examples of uses of routers

Imagine a situation where you have computers on a LAN and all but three admin computers are sending large music files to the other computers on the same LAN. Each file is split into packets. Each packet is put on the network cable and every computer examines every packet's network address to see if the packet is for itself. This means that a lot of packets are being put on the network and a lot of network activity is taking place. The consequence of this is that all computers are slowed down, even the admin ones. One solution is to split the network into two networks called segments (one for the computers that need to transfer large files and one for those that only ever need to transfer small files, the admin computers). They are then joined with a router. Now when large music files are sent, they are read only by the network they are on: the router examines every packet put on the network, sees that they aren't addressed to any of the admin computers and stops the admin computers from reading them. Their performance isn't now affected by the large file transfer activity.

Now consider the next example:

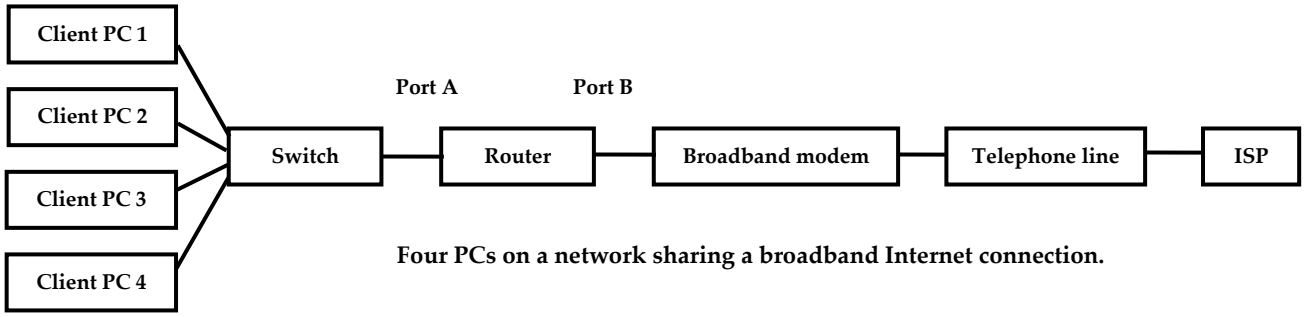


How does PC 1 in segment 166.54.1 send a document to be printed, if the printer is in segment 166.54.2?

- The document is split up into packets.
- The destination address of the print server is attached to each packet.
- The destination IP address of every packet doesn't match the segment IP address of PC 1.
- The packet is therefore sent to the current segment's router.
- The router looks at the destination address of each packet.
- The router then forwards the packet to the correct segment's router.
- The correct segment's router then forwards the packets to the print server.
- The printer server de-spools the print job to the printer.

55.4 More about ports, routers, IP addresses and the Internet

Consider this network:



Four PCs on a network sharing a broadband Internet connection.

If you had a phone and wanted to receive calls, you would need to publish your phone number (or at least tell your friends what it is). Similarly, if a company wants to have an Internet presence then it must have a way of accepting only those packets of data that are destined for its network. It must publish an address on the Internet. The router here shows two ports. Port B is connected to the broadband modem and this in turn is connected to the Internet via the telephone line. Port B will have an IP address and it is this address that is published. It is a unique address (no one else on the Internet has it) and it is visible to the entire Internet. Whenever packets of data are received by the router, it will accept those packets with the right IP address into the company's network and reject those ones which are not addressed to it. Port B might have an address 23.65.10.252 for example. This address will have been registered for you by your ISP. For the PCs in the company's network to use the Internet, they will need to be set up so that any communications they send can pass through the router. They will all need to be told what the address of Port A is.

55.4.1 Sockets

Imagine PC 3 on the company network sends a request to a web server with an IP address 231.33.321.98. We know from earlier in the chapter that a web server is a special type of server that dishes out web pages. Companies (and individuals) may have their own web servers, may rent space on someone else's web server (known as a 'virtual' web server) or may be granted space on their ISP's web server. Most of you with an Internet connection at home will have some web space by your ISP.

Each web server has many 'ports'. These can be thought of as holes that lead into the web server. Each of these holes has an address, such as Port 80 or Port 25. Given that a web server can actually carry out various jobs (one of which is dishing out web

pages), it is not unreasonable to allocate one port to one type of job. The web server spends all day and night 'listening' for requests for web pages on port 80 (the port used for http requests). When it receives a request, it fetches the web pages and sends them off.

Sometimes, we talk about the 'socket address' of a particular service on a server such as web page distribution. The socket address is made up of the IP address of the web server plus the port address. In our example, the socket address of the web page server is **231.33.321.98:80**

Web servers are constantly receiving requests from computers for web pages that it holds. Before a PC can request a web page from a web server, it must first make a connection to the web server. The web server sets up a new socket to deal with each connection request. If it receives multiple concurrent requests, it simply sets up a new socket for each of them. If it didn't do this, there would be no way of distinguishing which computer wanted what web pages. Once a socket has been allocated for a communication session, the web server is ready to service requests from the client. It will be able to send and receive packets of information between itself and a remote computer for the duration of that session. Sockets are simply used by a web server as a convenient way of setting up individual connections when multiple requests are being made at the same time by lots of different computers.

55.4.2 Ports

The TCP / IP protocol (Transmission Control Protocol / Internet Protocol) is a set of protocols that allow users on the Internet to communicate with each other, regardless of the equipment they are using. It is a set of rules that determine how data is sent and uses other protocols as part of its make-up. When you communicate across the Internet, you will be using the TCP / IP protocol. Generally speaking, you don't need to know what port number is required for any particular type of communication. However, there are times when you have to set up a communication session manually so it is worth knowing what ports are used for the most common types of Internet communication.

Port number	Internet application or protocol
21	**Used for ftp, or File Transfer Protocol.**
23	**Used for Telnet.**
25	**Used for SMTP, or Simple Mail Transfer Protocol.**
80	**Used for HTTP, or Hyper Text Transfer Protocol (requesting web pages).**
194	**Used for IRC, or Internet Relay Chat.**

Table showing what ports are used for the most common types of Internet connection.

55.5 Interconnections

In addition to clients, servers and other hardware, some thought needs to be given to the type of interconnections. The final choice will depend upon a number of factors including the predicted bandwidth, the environment that the interconnections will exist in, the distance between clients, the level of security needed and the cost. Computers on a network can be connected up using different media. These include twisted pair cables, coaxial cables, fibre optic cables and wireless connections.

55.5.1 Unshielded Twisted Pair cable (UTP)

Many networks use UTP cable. It is very light, flexible and cheap and has been around a long time - installation engineers are familiar with it. It is used extensively in the home to connect up the telephone system! It consists of pairs of conductors covered in insulation material and then twisted together. Within one cable, you might have 4 pairs but there are different designs, each with their own characteristics.

Twisting wires together reduces the effects of electrical interference at minimal cost. Twisting is helpful but you can provide even more shielding from interference by using a silver foil wrapped around the cables. Although you can certainly buy STP (Shielded Twisted Pair) cable, UTP is perfectly sufficient for many networks. You can read a debate on the merits of UTP compared to STP on this web site: **http://www.anixter.com/techlib/vendor/cabling/d0503p02.htm** Importantly, the bandwidth for UTP isn't as high as for coaxial or fibre optic and you need repeaters more often with UTP than coaxial or fibre optic cable. For many 'standard' networks in a building, however, it is the cable of choice because of its cheapness, adequate-for-the-job bandwidth, sufficient resistance to electrical interference and easy installation.

55.5.2 Coaxial cable

It is made up of a central conducting core covered with some protective insulation. Wrapped around the insulation is a thin metal sheath that provides the electrical interference protection. Finally, the cable has an outer covering. Coaxial cable is the cable used to connect your television aerial to the television.

It is heavier and less easy to manipulate than UTP. It is also more expensive. You can have longer cable runs than for UTP but still not as long as for fibre optic. This means that you may need fewer repeaters compared to UTP cable, but more than for a fibre optic cable. It is mechanically strong and resistant to interference.

55.5.3 Fibre optic cable

This consists of a glass core wrapped in protection within a cable. Data is sent down the fibre optic cable as light, not electrical signals. The signals therefore don't suffer from electrical interference. Fibre optic cables are also resistant to the effects of moisture because they are non-metallic, unlike conventional cables, which are metal-based. The cables themselves are very brittle when compared to conventional metal-based cables and need to be well-protected.

Fibre-optic communications is high bandwidth compared to e.g. UTP. For this reason, it is selected for networks where video-conferencing will be needed. The price of fibre optic cable has dropped rapidly in recent years so that it is comparable to metal-based cables. However, installing it and any modifications or repairs do need specialist engineers and this makes it relatively expensive to install and modify.

Fibre optics connections are often used for the 'backbone' of LAN networks. This means that they are used to connect the servers to the main switches, which are located in different areas where there are clusters of computers. From the switches, normal coaxial or twisted pair connections are made. Data transmission speeds of 1Gbit/s are easily reached using fibre optics connections.

55.5.4 A wireless LAN

You can connect up networks without using cables! For example, you can use lasers to 'fire' data signals between different parts of a network or use radio signals to send information across a LAN. You could use lasers to link two buildings onto one network, for example. Of course, you must have a clear view between the sending and receiving equipment in each building or the laser signal will be blocked.

Consider a school. The school has a couple of computer rooms linked using UTP. What happens when one department wants to use a computer room and it is booked? What happens when a teacher wants to use the computers in their own area rather than the computer room? One answer is to use wireless technology to provide a 'roaming computer room' like this:

- Buy laptop computers.
- Put radio network cards in them.
- Set up Access Points around the school in all departments. (These hardware devices are 'black boxes' fixed to the wall. They relay signals from the server to each laptop's radio network card).
- Use the laptops as if they were part of the main network!

The beauty of this system, also known as **Wi-Fi**, is that you can move the laptops around the school to different departments (so long as you have remembered to install a transceiver there). You also don't have to worry about cable installation. One drawback is that the bandwidth of these systems may not perform as the main part of the network does. For example, you should be able to run generic applications such as a word processor or a spreadsheet but multimedia or bandwidth-intensive Internet applications may be slower than networks that use physical connections. They also suffer from electrical interference problems far more than cable-based networks and they are potentially very insecure! Signals can be easily intercepted if the network has not been set up with security in mind. If this is a problem for an organisation than it should consider other technologies or think about encryption, perhaps PGP.

If you need to send information over longer distances then satellite, cellular phone technology and microwave technology should be looked at.

55.5.5 Using satellites

Satellites are used widely for communication in the media, navigation, video conferencing, data transfer and for military purposes, for example. A transmitter on the ground sends signals to a satellite using microwaves or some other form of transmission. Using a different frequency to avoid interference, the signals are then redirected back to the planet. If it doesn't reach the intended recipient, it is then bounced to other satellites and on to its destination.

55.5.6 Pros and cons of using satellites for communication

If you transfer data using satellite technology, the cost of data transfer is not dependent on distance. A short distance transfer costs the same as a long distance one. Overall, the cost of transfer is higher than non-satellite methods. Satellites can handle a very high number of simultaneous communication lines (they can handle a large bandwidth). Satellite communications can be

used in the remotest parts of the world, from the arctic to mountainous areas to deserts although the actual equipment is very expensive. One use of this technology has been to provide up-to-date educational materials for schools in remote parts of Africa. Because of the distance between the surface of the planet and the satellites, there is a short delay in any communication. Weather conditions can also affect the quality of transmission.

55.6 Ethernet

Ethernet is a widely-used design for baseband LANs, working until recently to speeds of up to 10 Mbps. Now Fast Ethernet (also called 100Base-T) can achieve 100 Mbps and Gigabit Ethernet is faster still. Ethernet networks connect stations using coaxial cable and can spread over about 100 meters although they can be extended with the right hardware.

We mentioned in a previous chapter about collision avoidance and collision detection, to try to prevent two workstations putting packets of data on a network at the same time. Ethernet makes use of collision avoidance and detection strategies. As traffic increases on an Ethernet network, the number and frequency of collisions increase. This is because every station is broadcasting to every other one and the more workstations there are broadcasting, the more the communications on the network increase.

Ethernet will try its best to prevent data collisions. When it does detect a collision, however, it will have the packets put on the network again but this time with a delay between the events.

55.6.1 Reducing data collisions on an Ethernet LAN

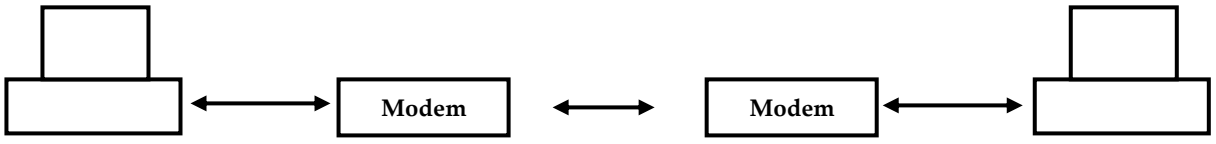
There are strategies for reducing the number of collisions on a network. We saw at the beginning of this chapter the use of a **switch**. This is a piece of hardware in common use in Ethernet LANs and when one is used it is known as a '**switched Ethernet LAN**'. A switch can be used to split up the network into areas that improve the efficiency of communication. It can be used to ensure each station is given a 'time-slice' in which to send data. Two stations can also be given a temporary communication link between them using a switch, to ensure successful communication.

55.7 The Internet and the PSTN

We need to remind ourselves of some of the work we did earlier in the book, when we talked about making an Internet connection on a network. You can also refer to earlier in this chapter, when we discussed how a company would share a broadband connection amongst the computers on its network. There is a range of choices for getting an Internet connection. These include:

- a dial-up connection
- an ISDN connection
- broadband and baseband
- a Leased line.

55.7.1 Using a dial-up connection



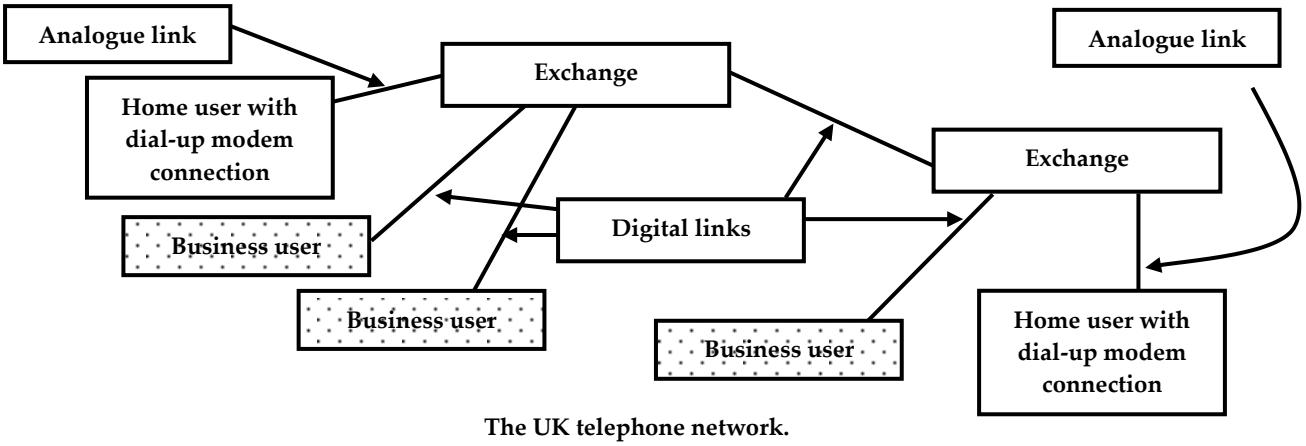
A typical dial-up connection.

If a company is using a modem for its Internet connection then access will be subject to the modem's limitations. Modems typically work at **56 Kbps**. This is fine for a company that only has low bandwidth communication needs such as sending text files and light Internet use, but if higher bandwidths are needed, for example a lot of video, picture and sound files need to be sent and received, then communication using a dial-up connection can become painfully slow.

55.7.2 Using an ISDN connection

Businesses (and increasingly homes) could choose an ISDN line (Integrated Services Digital Network) for faster access to the Internet, reaching download speeds of **128 Kbps**. The computer is connected to the ISDN line (which is usually just an extra normal phone line) with a 'special interface'. You don't need a modem because the connection is digital.

The ISDN line goes back to the digital part of the local PSTN exchange. The PSTN exchanges together form a digital network. Home computers with modems send and receive their information **to start with** using analogue connections to and from the analogue part of the exchange. Then the information is sent along the network using digital connections.



The UK telephone network.

55.7.3 Broadband and baseband

A further development is ADSL (Asymmetric Digital Subscriber Line). It works by 'splitting' the telephone line into two. It then uses one half for phone calls whilst the other half is used for sending and receiving digital data, using many different frequencies **at the same time**. By sending different packets of data at the same time using different frequencies, a high bandwidth can be achieved. WANs like the Internet make good use of broadband connections because WANs are expensive to set up and maintain - they are big networks! You need to ensure that you send and receive as much data as possible and share resources as much as possible to keep costs down and to ensure that users don't clog up the system. Compare broadband to the **baseband** found on a typical LAN, where data is sent using the whole frequency of the connection. LANs are smaller and less complex than WANs and don't need the same amount of transmission power.

Home users (as well as business users) with ADSL can reach **download** speeds of **2 Mbps** whilst **upload** speeds are **256 Kbps**. Higher speeds are possible at a price! Note that download speeds are faster than upload speeds, which is why ADSL is known as an '**asynchronous**' connection. The price of this technology has been dropping considerably recently.

55.7.4 Leased lines

An alternative to a dial-up connection is a leased line connection. In this type of connection, you pay a fixed fee for a permanent Internet connection. You have access to the Internet 24 hours a day. You do not need to dial-up a phone number because you are always connected! Modems used in dial-up connections work at speeds of up to 56 Kbytes. This may be fine for light Internet use. However, leased lines typically give much higher bandwidths and this may be necessary if there is heavy use in the office by a lot of people or use involving large multimedia files.

Q1. Describe the role of the proxy server.
Q2. Clearly explain the difference between a hub and switch.
Q3. What is a gateway used for?
Q4. What does a repeater do?
Q5. What is the purpose of a network card?
Q6. What is a network segment?
Q7. Use the Internet for research. What is a Wi-Fi hotspot, typically found in many cafes?
Q8. What is meant by a client-server network?
Q9. What is meant by satellite communications?
Q10. Compare dial-up Internet connections to broadband Internet connections.