3. Compare the HCI of your improved form with the customer form as it was before you made any changes to it (see page 100).
   a In what ways is this form easier and more pleasant to use?
   b Has the re-arrangement of the fields made the connections between the data simpler to comprehend?
   c What further improvements could be made to the form to improve the HCI of this form?
4. Before leaving this activity it is important to realise that despite the changes we have made to this form, the basic data has not altered.

   Click on the datasheet view button and you will see the data is still as it was, and the order of the fields is still the same. The only changes we have made have been to the appearance of the form.

Activity 6.3 – Inserting, deleting, updating and sorting

In this activity we will look at maintaining and sorting the data in an existing database.

1. Start *Access* and open your copy of the *Vet* database.

   Switch to either form view or datasheet view.

   Note: do *not* make the following changes in design view.
2. There are two new customers who visited Dr Harry today.

   Click on the new record button  (one of the navigation buttons at the bottom of the form) and add the following data:
   - *Allen Collins* of *13 Hardy St, Newtown* (ph *4354678*) has a *3* year old pet *galah* called *Golly*
   - *Leanne Bradford* of *34 Ford Av, Harlaxton* (ph *4394567*) has a *12* year old collie called *Lassie*

   Both owe *$35.50* for today's visit.
3. *Peter Gillam* of Drayton has paid the money he owes and has moved away. His data can now be removed from the database.

   To do this click on his record and from the top menu choose *Edit* > *Delete Record*.

   Click *Yes* to delete the record.

   Note: this delete action *cannot* be undone. Once the record is deleted it cannot be brought back.
4. Two customers have notified us that their details have altered. Make the following changes:
   - Nick Scott has moved around the corner to *4/15 Jenkins St*; he has kept the same phone number
   - Janet Quinn's old cockie has died; she has bought another which she is calling *George II*; it is *one* year old.

5. The data in the database is not very well organised at the moment. If we wish we can re-sort it into a different order.

   Sorting can be done in either form or datasheet view. In either, simply click on the field you want in order and then click on one of the two sort buttons (at right).

   a Click on the *surname* field. Sort the records into alphabetical order of *surnames*. Who is the last person alphabetically?

   b Sort the records into alphabetical order of *pet name*. Which is the first pet?

   c Sort the records in order of *amount owing*. Who owes the least money?

   d Sort the records of *amount owing* in descending order  . Who owes the most money?

   e Sort the data into *suburbs*. How many records are there for the Harlaxton area?

   f Sort the data by *pet type*. How many guinea pigs are there?

Note: It is not necessary to save an Access database before closing. Any changes you make are automatically recorded as you go. Data will however be saved in the order in which it was last sorted.

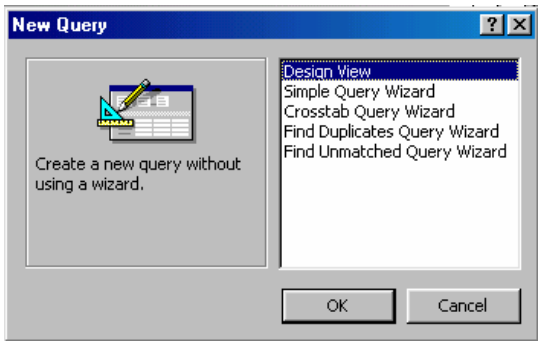
Activity 6.4 – Simple queries

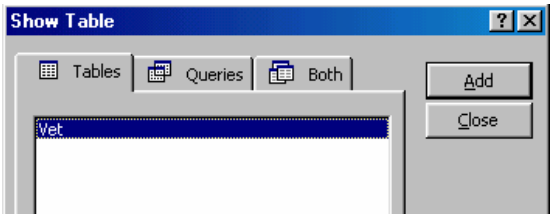
In the last activity we used the sort function to find information. In this activity we will use a more powerful data collection method called a *query* or *filter*.

1. In the *Vet* database close any open form  and switch to the *Queries* object:

2. Say we wish to just view the records of people who live in Newtown. To do this we will need to create a filter to block out all other records.

   a Click on the *New* button and choose *Design View*. (Or click *Create query in Design view*.)



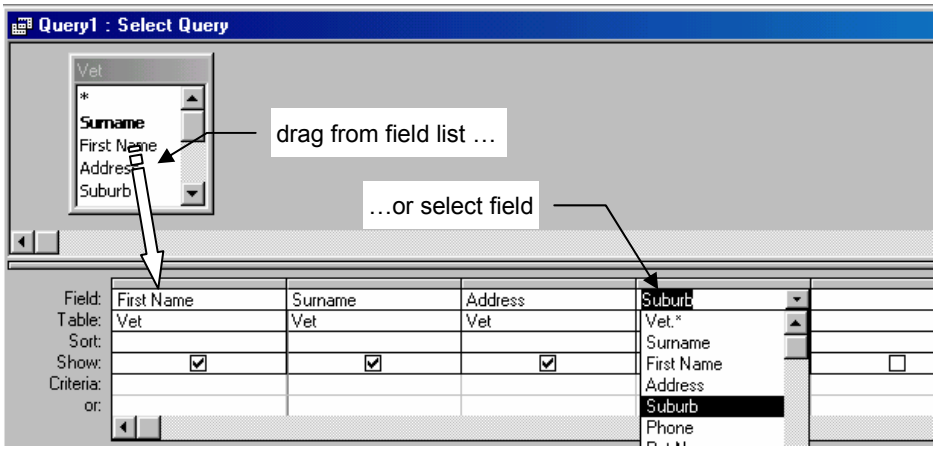


   In the dialogue that appears click on *Add* and then *Close*. This will give us the Vet table to work on.

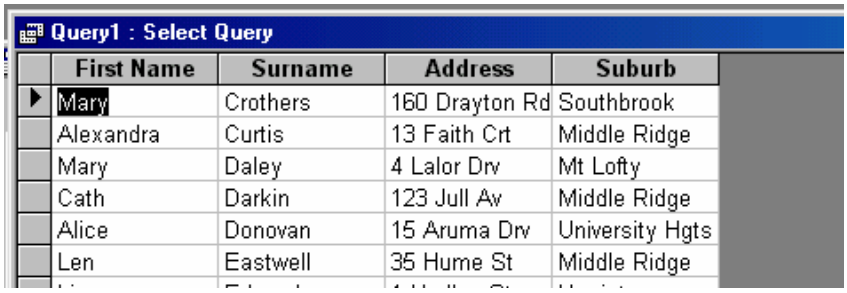
In more complex databases we can build queries based on several different tables of information, each of which would have to be added to the query grid.

b We now need to decide which fields of information we want to display. Say we just want to show *first name, surname, address* and *suburb* only.

To get this put these four fields in the first four columns. This can be done either by clicking and dragging from the field list, or by using the drop down arrows in each column:



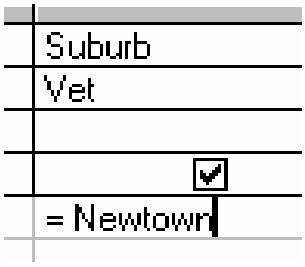
c To see what we have achieved so far click on the *Run* button  on the toolbar at top.



Just the four selected columns are displayed. (While these are displayed they can also be sorted if desired.)

d Now we need to filter out the suburbs that are not Newtown.

Return to design view  and under *Suburb* in the *Criteria:* row write = Newtown:



When you run  the query only data for Suburb = Newtown is displayed.

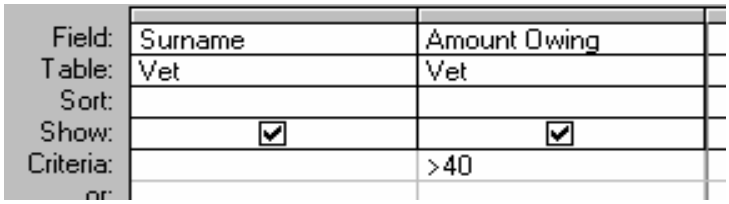
(Note that when you return to design view of this query Newtown now has " " quotes around it. *Access* adds these because *suburb* is a text field.)

e Close this query  and when prompted save it as *Newtown*:

3. Our next query will find people who owe more than $40.00 .

   a In the *Queries* tab again click on *New,* then *Design View* and *Add* the *Vet* table.

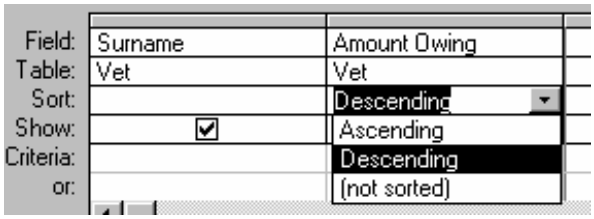
   b This time we will only work on surname and amount owing:



   In the criteria row put >40 (i.e. show people who owe more than $40.00) and run the query.

   c Once the query has found these people we might want to find who owes the most. We can either do a sort in the datasheet view or we can make the sorting part of the query.

   To do this simply select *Descending* from the drop down list in the sort row under *Amount Owing*:



   We now have the people who owe more than $40, with those owing the most first.

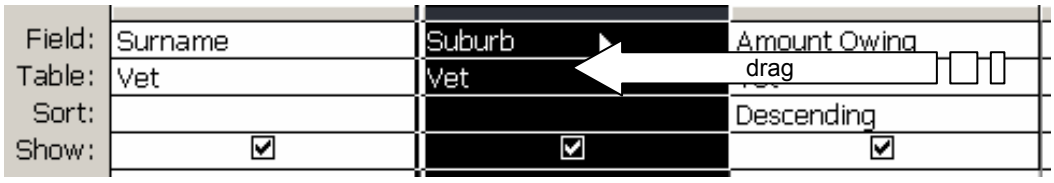
   d Save this query as *Debtors*.

4. The advantage of making the sort part of a query is that we can sort on more than one field.

   Say Dr Harry wants to send a debt collector around to collect from those who owe him more than $40. He needs a list of customers by suburb, with the customer owing the most in each suburb listed first.

   a Open the *Debtors* query in design view. (In the *Query* object click once on *Debtors* then on *Design*, or double click on *Debtors* and then switch to design view.)

   b Add the *Suburb* field and click and drag it so that it is ahead of the *Amount Owing* field.



c Sort the suburbs into alphabetical order:

Field:	Surname	Suburb	Amount Owing
Table:	Vet	Vet	Vet
Sort:		Ascending	Descending
Show:	☑	☑	☑
Criteria:			>40

d Run the query.

Why was it necessary to put suburb before amount owing? (Hint: *Access* works out queries from left to right.)

5 Make queries to determine the following:

a What is the maximum amount owing in Harristown?

b Find the name of the cat Dr Harry has treated most recently.

c What is the name of the six year old dog that has the most amount owing on it?

6. Queries can work on more than one condition.

a Say we wish to find out how many owners live in the Harristown area *AND* have pets under 5 years old.

In this case *both* Suburb = Harristown *and* Pet Age < 5 must be true for the records displayed.

Field:	Surname	Suburb	Pet Age
Table:	Vet	Vet	Vet
Sort:			
Show:	☑	☑	☑
Criteria:		="Harristown"	<5

Develop and test the above query.

b On the other hand we might want to find which owners live in the Wilsonton *OR* the Middle Ridge area.

In this case *either* Suburb = Wilsonton *or* Suburb = Middle Ridge. To show this we place both conditions in the *Suburb* column:

Field:	Surname	Suburb
Table:	Vet	Vet
Sort:		
Show:	☑	☑
Criteria:		="Wilsonton"
or:		="Middle Ridge"

Develop and test this query.

Remember:

AND means *both* conditions have to be true. To *AND* conditions place them in *separate* columns.

OR means *either* condition can be true. To *OR* conditions place them in the *same* column.

7. Create queries to find out answers to the following:
   a How many owners have dogs?
   b What is the number of owners who live in the Mt Lofty area *and* who owe more than $40?
   c How many owners have pets aged between 3 *and* 10?
   d How many owners made their last visit between 1st April *and* 31st Oct last year?
   e How many pets are rodents (rat, mouse, hamster *or* guinea pig)?
   f How many owners live in the Middle Ridge area *and* have a dog *or* a cat?
   g How many cats has Dr Harry seen this year?
8. This form of developing queries where fields are dragged onto a grid and sample data entered into criteria rows is called query by example (QBE).
   a Explain why you think this form might be called query *by example*.
   b QBE is used to set up and use a query. How difficult is QBE to understand, simple, moderate, difficult or complex?
   c What features of QBE do you think support this assessment.
   d Identify any metaphors or affordances used in the QBE system.
   e Suggest another possible way of querying a database other than QBE.

Activity 6.5 – Video database

Having seen some of the different things we can do with a database it is now time to create one of our own. The database we will develop will be for a video store and will have tables for videos to hire, members who borrow them, and a list of who has which video out on hire.

1. Click on the *New* button to create a new blank database called *Video Hire*. Save it to a suitable folder.
2. a Choose the *Tables* object and double click on *Create table in Design view*:
   b Enter the field names at right.

   Access will automatically select *Text* as the data type for each field but this is not always suitable.

   Click on the data type for member#, and then on the drop down arrow that appears.

   Choose *Number* from the list offered.

   member : Table

Field Name	Data Type
member #	Number
name	Text
address	Text
phone	Text
DoB	Date/Time

   c Set the indicated data type for the other fields.

   (Why is *phone* set as text?)

   d In turn click on the data type for each field and set suitable properties at the bottom.

   e.g. for the text data types set an appropriate field length i.e. how many letters long *name* or *address* might be; make the number property a fixed integer with no decimal places; choose your preferred date format.
3. In organising and querying data it is necessary to have a way of identifying each record. In this database the best way of telling records apart will be the member number.

Unit 1 – Information Systems

Overview

In this unit we will begin our investigation into information and intelligent systems by looking at:

- what information is and how by itself information does not provide knowledge or wisdom
- the different ways information systems can be viewed
- some of the different forms of information systems
- basic terms and concepts of relational databases
- information systems and privacy.

Introduction

We live in the Age of Information. Information is a valuable commodity that can be bought and sold; whoever has information has power.

Over the last 250 years society has moved from the Agricultural Age, to the Industrial Age, to the Information Age. In western countries information workers now outnumber all other workers combined. As members of this age we need to understand and be able to use information technology to keep up with the changes that are occurring in our society.

Information is the basis of the effectiveness and efficiency of the organisations you will be part of when you leave school. There is an increasing need for more information and more (human) information processors. Those with the skills to use computerised information systems will have a distinct advantage over those who do not.

Through the study of information and intelligent systems that you are now beginning you will give yourself the basis of knowledge and skills needed to be an effective member of the information age. In addition you will develop skills in analysis, synthesis, evaluation, problem solving and communication that you can use in other fields.

Data, information, knowledge and wisdom

Where is the wisdom we have lost in knowledge?
Where is the knowledge we have lost in information?
T S Eliot

The pessimistic words of the poet T S Eliot are very apt in the world of the Internet. Anyone who has run a search on a simple query that has returned thousands of hits will agree with this. Eliot warns us we can have too much information to be able to derive any knowledge from it. Even if we do develop knowledge, there may be little chance of wisdom emerging.

Information is not the same as knowledge, and knowledge is not the same as wisdom. To understand the distinctions we must go back one further level to data, the basic building block of comprehension.

Data is the plural of datum. A *datum* is a simple recognisable fact. Examples of data include *red, 25,* or *true*. *Red* might be the colour of a car or a person's nickname; *25* might be a temperature or a score; and *true* could be the result of an experiment or an answer in a test.

Each datum is simple (singular) and states a fact. *25* is not *15* and is not *35*. It is something that we recognise stands for the value of two tens and five units. We do not concern ourselves if the datum is correct or not. Even though we call it a fact, truth does not come into it. We accept data as given or, in an information system, as entered.

From the building blocks of data, information may be structured. From information, knowledge may be constructed and, if we are very lucky, wisdom may arise. To understand this we will look at an everyday example.

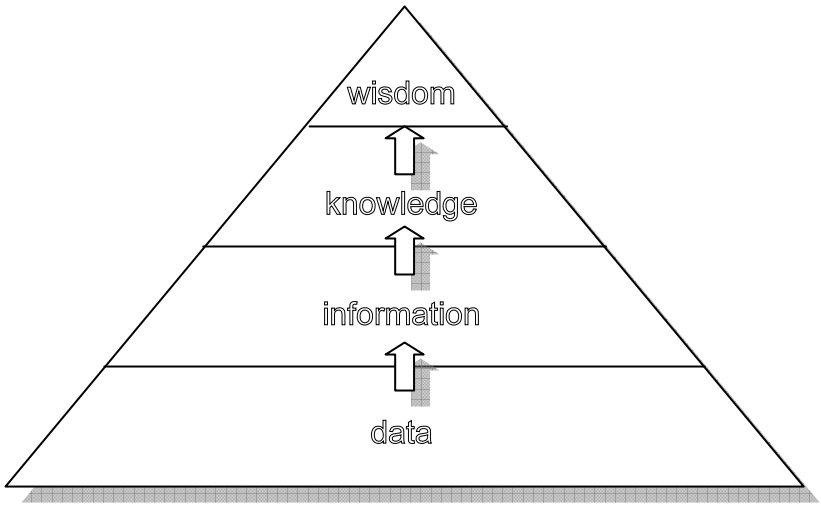
As I listen to the weather on the radio I might hear that the temperature outside is 25°C. Somewhere, someone looked at a thermometer and saw the number 25. This 25 is *data* (actually a datum).

The value 25 was reported to the weather bureau where they gave the datum form. They structured it by adding °C. This implies that the 25 exists on a scale of values that represents air temperature. The datum has now become *information*. It has been organised using the Celsius scale so that it has meaning. The meaning would be very different if the weather bureau was in America and had added °F. This is from a different structure, the Fahrenheit scale. 25°F is below freezing.

When I hear from the radio the information that the outside temperature is 25°C I can construct some meaning from it; I know it is not too cold and not too hot. By decoding that 25°C is a pleasant temperature, I have drawn inferences from the information and I now have *knowledge* of the situation. The information has been arranged within an overriding format so that it is useful. This overriding format is sometimes called a *metacontext*. In this case the metacontext is formed from my experience of weather conditions in the past.

Once I know that it is an agreeable day outside I can use the knowledge in ways that are compatible with my needs and with what is acceptable in society. I can make the decision not to wear a thick overcoat, or not to go outside in a swimming costume. If I have *wisdom* I use the knowledge effectively in a given situation. (Or not, as the case may be. Knowledge that has been constructed from information, but then not used wisely, has been described as being *inert knowledge*.)

To summarise: data are basic recognisable facts; information is data that has been organised so that it has meaning; knowledge is information arranged into a metacontext so that it is useable; and wisdom is knowledge being used effectively in a given situation.



From data to wisdom

Together these form a hierarchy. Wisdom comes from knowledge, which comes from information, which comes from data. At the bottom level a great deal of data is required to develop the needed information. As we move up there is only a limited amount of useable knowledge derived from this information. At the top there is usually very little wisdom displayed in the use of this knowledge.

The point of all this is that knowledge and wisdom do not come from an information system. The system if used effectively will combine data into information, but whether this information is developed into knowledge that is applied with wisdom depends upon the person operating the system.

Activity 1.1 – From data to wisdom

1. a What is meant by the term *Information Age*?
   b In what ways has the availability of huge volumes of information on the Internet and through other sources put us in the position of not being an age of knowledge?
2. a Explain in your own words what is meant by each of the terms *data, information, knowledge,* and *wisdom*.
   b Give a simple, single example of each.
   c The philosopher Immanuel Kant once said "*Science is organised knowledge, wisdom is organised life*".
   Explain what you think he meant by this.
3. The section above uses the analogy of today's temperature to illustrate the difference between data, information, knowledge, and wisdom.
   Create your own analogy. You may use your own situation or, if you wish, consider the scenario of a runner in a 100m race.
4. Give an example of a piece of information that can have two different meanings depending on the context it is in.
5. Give an example of a situation in which knowledge might be inert.
6. Whether information is developed into knowledge that is applied with wisdom depends upon the user of the system.
   Give an example where an information system may not be used with wisdom.
   Compare your answers with those of others in your class.
7. Hugh McKay has described today's generation as being "answer rich and question poor".
   a What do you think he meant by this? In your answer you might like to refer to the ability to reason logically and powers of discernment.
   b In what ways has the advent of search engines such as Google led to the reduction in the need to consider and contemplate before seeking answers.
   c The artist Pablo Picasso once said "*Computers are useless, they can only give you answers*". Was he correct?

Systems

Think of systems you know about. You have a digestive system, you may use a stereo system, you belong to the education system, and the Earth is part of the Solar System.

So what is a system?

A *system* is defined as a group of interrelated parts that work together for a common purpose or goal. For example your digestive system consists of your teeth and tongue, your oesophagus, stomach, small and large intestines, and a variety of glands and other organs. These work together with the purpose of converting food into simple substances that can be transferred to the blood – where they become part of the circulatory system.