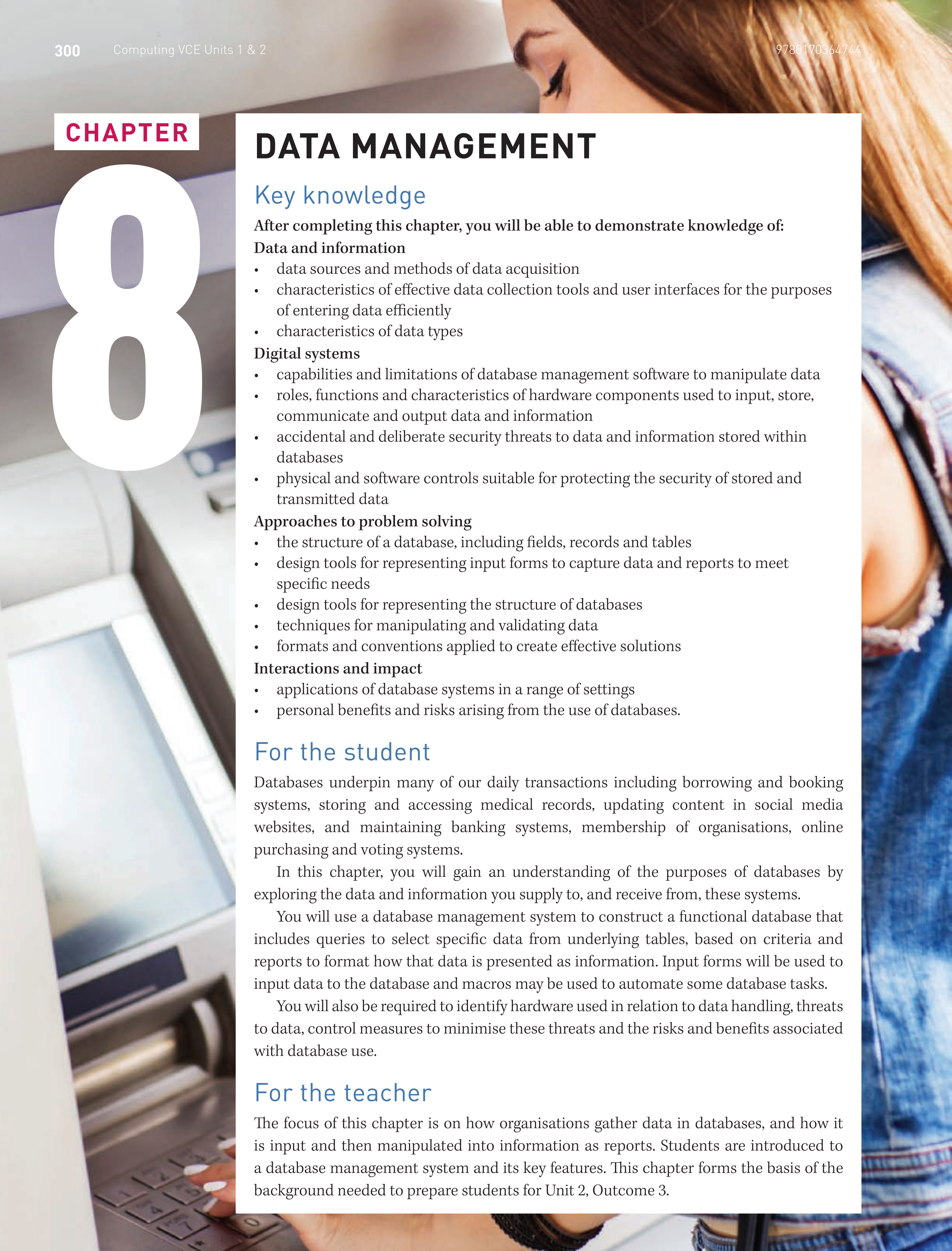


CHAPTER 8

DATA MANAGEMENT

Key knowledge

After completing this chapter, you will be able to demonstrate knowledge of:

Data and information

- data sources and methods of data acquisition
- characteristics of effective data collection tools and user interfaces for the purposes of entering data efficiently
- characteristics of data types

Digital systems

- capabilities and limitations of database management software to manipulate data
- roles, functions and characteristics of hardware components used to input, store, communicate and output data and information
- accidental and deliberate security threats to data and information stored within databases
- physical and software controls suitable for protecting the security of stored and transmitted data

Approaches to problem solving

- the structure of a database, including fields, records and tables
- design tools for representing input forms to capture data and reports to meet specific needs
- design tools for representing the structure of databases
- techniques for manipulating and validating data
- formats and conventions applied to create effective solutions

Interactions and impact

- applications of database systems in a range of settings
- personal benefits and risks arising from the use of databases.

For the student

Databases underpin many of our daily transactions including borrowing and booking systems, storing and accessing medical records, updating content in social media websites, and maintaining banking systems, membership of organisations, online purchasing and voting systems.

In this chapter, you will gain an understanding of the purposes of databases by exploring the data and information you supply to, and receive from, these systems.

You will use a database management system to construct a functional database that includes queries to select specific data from underlying tables, based on criteria and reports to format how that data is presented as information. Input forms will be used to input data to the database and macros may be used to automate some database tasks.

You will also be required to identify hardware used in relation to data handling, threats to data, control measures to minimise these threats and the risks and benefits associated with database use.

For the teacher

The focus of this chapter is on how organisations gather data in databases, and how it is input and then manipulated into information as reports. Students are introduced to a database management system and its key features. This chapter forms the basis of the background needed to prepare students for Unit 2, Outcome 3.

Applications of database systems

In the last few years, an unprecedented quantity of data has been generated as people flock to a range of online or networked systems. Apart from business uses, social media websites such as Facebook, Twitter and Tumblr, and even Google searches, accumulate vast amounts of data. When you borrow a book or DVD from a library, or book concert tickets online, data is generated and stored. When you visit your local doctor, your health records are updated so that your complete medical history can be viewed by other doctors in the future.

This data can be used for a variety of useful purposes but in order for that to happen, it needs to be stored somewhere. A **database management system (DBMS)** is software designed to store data and allow users to search and select from it in order to extract the information they need in order to make decisions.

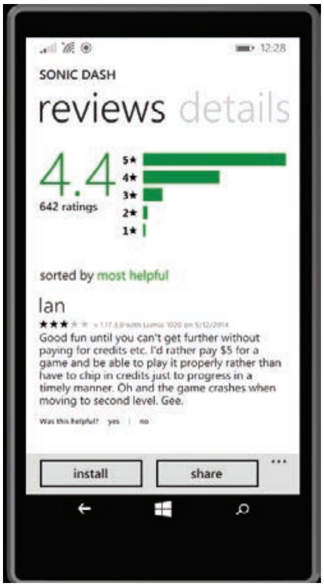
You will find DBMSs used in a variety of contexts. Businesses use databases to record transactions such as bill payments or shopping purchases. Banks are particularly heavy users of databases because of the volume of financial transactions they must be able to process and store every second. Bank customers now also perform much of their own banking online, which is recorded in their bank's DBMS. Libraries catalogue books and store details of borrowers in their databases. Scientists and museums use DBMSs to catalogue their findings and specimens. When you join a club or other group, you will be asked to supply a number of personal details. These details will be collected and stored in a database. Many people purchase goods online through websites such as Amazon and eBay. To register for these types of transactions, you must be able to supply at least your email address, delivery address and payment details. In some cases, you may even be involved in online voting, which also requires a database to store and tally the votes for each option and report on the most favoured popular choices. Rating sellers on eBay, videos on YouTube or 'apps' for mobile devices are common types of voting systems you may encounter (Figure 8.1). Google and Wikipedia use powerful DBMSs to provide a list of relevant websites or topics when you type in a search term, and every item in your Facebook profile is also stored in a massive database. A list of names and phone numbers of friends to invite to your party would also be considered a database.

Well-known non-digital databases include Ancient Egyptian papyrus scrolls with lists of quantities of crops harvested for the Pharaoh and the Domesday book of William the Conqueror, which listed the property and value of every landholder in England, and was used to calculate taxes.

Well-known DBMSs include Microsoft Access, FileMaker, MySQL, PostgreSQL, Microsoft SQL Server, Oracle, SAP and IBM DB2.

THINK ABOUT COMPUTING 8.1

Think of some other examples of databases you may regularly encounter.



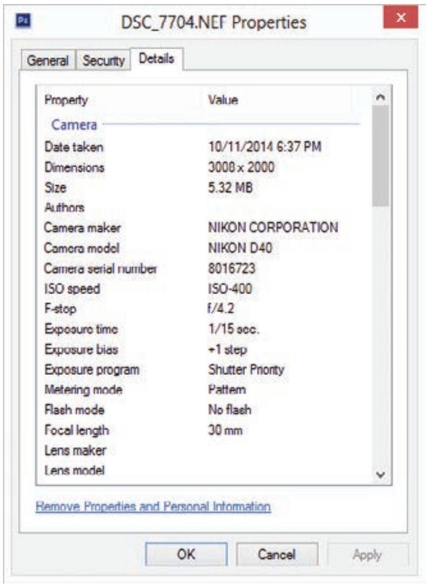
Ian

★★★★★ v 1.17.3.0 with Lumia 1020 on 5/12/2014

Good fun until you can't get further without paying for credits etc. I'd rather pay $5 for a game and be able to play it properly rather than have to chip in credits just to progress in a timely manner. Oh and the game crashes when moving to second level. Gee.

Was this helpful? yes | no

FIGURE 8.1 A rating from online voting for a popular game in the Windows Store



Property	Value
Camera	
Date taken	10/11/2014 6:37 PM
Dimensions	3008 x 2000
Size	5.32 MB
Authors	
Camera maker	NIKON CORPORATION
Camera model	NIKON D40
Camera serial number	8016723
ISO speed	ISO-400
F-stop	f/4.2
Exposure time	1/15 sec.
Exposure bias	+1 step
Exposure program	Shutter Priority
Metering mode	Pattern
Flash mode	No flash
Focal length	30 mm
Lens maker	
Lens model	

FIGURE 8.2 An example of metadata saved in a photo taken on a digital camera. Many images also save GPS coordinates to allow for geotagging.

THINK ABOUT COMPUTING 8.2

List the data that a bank might need in order to process a supermarket EFTPOS transaction.

In addition, all digital files now contain metadata, the primary purpose of which is to allow databases to locate files that match certain search criteria more easily. Figure 8.2 on the previous page provides an example of metadata for a photograph. DBMSs also contain a specific type of metadata that record the details of structural elements in a database file. These details are often referred to as a data dictionary and may include details such as the size and data type of each field and characteristics of tables and other objects. Essentially, it is data about data!

Database management systems (DBMSs)

For data handling, no other software type can compare with a database package. While each package may differ in terms of its look and features, several key capabilities are common to all DBMSs.

Capabilities and features

- Data redundancy is reduced by centralising different data sources into one database file. Data consistency is a flow-on effect of reducing redundancy.
- Sharing of data allows multiple users to access the same data simultaneously. For example, an accounts department might want to send an invoice to a customer, while the marketing department wants to send an invitation to a special product launch. Both departments will send their emails addressed the same way and including the same personal details because they are both taking the data from the one company database.
- Data integrity can be maintained through the use of validation (see below).
- Data security measures mean only authorised users can access certain data. Data is often **read only** to most users in order to restrict the number of people who can edit it.
- Multiple data elements can be updated in different tables when a transaction takes place.
- A common structured query language (SQL) is used in all packages to select data.
- Input forms can be created to make data entry easier for users.
- Reports can be created to display information extracted from database files. They can be saved and modified for later reuse. Reports can include graphic representations such as charts.
- Concurrency of data is maintained when two users try to edit the same record simultaneously.
- Data is independent of the database application, so properties of fields can be modified without altering the application or the database file itself.

Limitations

DBMSs also have a number of limitations that need to be considered if you are planning to implement a database.

- Equipment (software and hardware) can be costly, especially if regular upgrades are needed.
- Conversion from a computer file used for data storage, such as a spreadsheet, to a DBMS can be costly and time consuming. Fields and tables must be carefully constructed to ensure that no data is lost.
- Staff need training on how to use the new system. This can be costly.
- Specialised database administrators need to be employed.
- If a database is corrupted, (for example, as a result of a technical failure), it has a huge impact across the entire organisation.

Benefits and risks

A DBMS can be a very useful tool for a variety of reasons. By storing all of your relevant data in one place, the time and effort spent searching for information is greatly reduced, especially when compared with a manual search. This is especially useful for medical histories and banking details. Being able to find information more quickly could also save you money, especially in the form of wages for data-entry personnel.

When you need to ask a particular question of your data, such as, 'How many widget sales have been recorded in Dandenong during the previous three months?' or 'What is the distribution of Pobblebonk Frogs across Victoria?' you can answer the question in a timely manner with both relevance (only the information you require) and completeness (all the information you require). Using validation and calculation formulas can guarantee the accuracy of data and information, respectively.

An easy-to-use interface and selection tools can make your data more accessible and usable. Selected data can be formatted into well-presented and attractive reports that make the information clearer and more readable to a wider audience on a variety of different platforms. These all contribute to delivering the message that the information needs to convey to the end user.

Benefits of a database solution are classified in terms of efficiency (time, cost, effort) and effectiveness (completeness, readability, attractiveness, clarity, accuracy, accessibility, timeliness, communication of message, relevance and useability).

There are also several risks that need to be considered before and while using a DBMS.

As per the **garbage-in equals garbage-out (GIGO)** principle, the input must be validated to avoid inaccurate data being entered. You may also have difficulties with version control of data.

Privacy must be protected. How will you prevent unauthorised users accessing confidential data? Security must also be protected. How will you prevent hackers and other unauthorised users from stealing or damaging data?

Database systems can be complex, difficult and time-consuming to design. Training is required for all programmers and users, or data can be damaged. In addition to the costs of training and running the system, the costs of purchasing suitable hardware and software can be high.

Database structure

A DBMS is composed of several key parts. All the objects defined in the following sections can be contained within a single database **file**.

Fields, records and tables

This information is perfect as a sample entry for a database:

> Mrs Maria Johnson is 26 years old. She lives at 23 Example Street, Fairfield, 3078. Her mobile phone number is 04 [redacted] 123 456.

Please note that some personal information is redacted for privacy reasons.

The individual data elements highlighted below could be described as database **fields**:

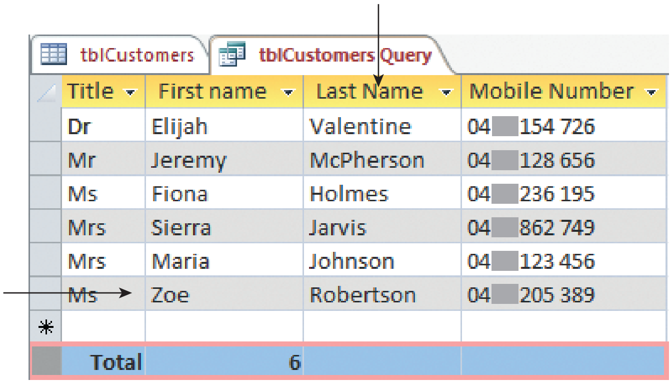
> Mrs Maria Johnson is 26 years old. She lives at 23 Example Street, Fairfield, 3078. Her mobile phone number is 04 [redacted] 123 456. Her customer ID number at the local newsagent is 4820M.

A field is designed to hold a single type of raw data only. It cannot contain a formula. For example, a field may hold a name, a postcode or a telephone number. Specific data types used by fields are discussed below.

All the data that relates to one 'entity', in this case Maria Johnson, is called a **record**. Records are distinguished from one another by creating a field in which the data it contains has a unique value for each record. This is called a primary key. In the example above, the customer ID number would be the primary key to uniquely identify Maria's record.

A set of records is stored in a database **table**. When viewed, each column of the table represents a field while each row represents a record. See Figure 8.3.

'Last name' is a field.



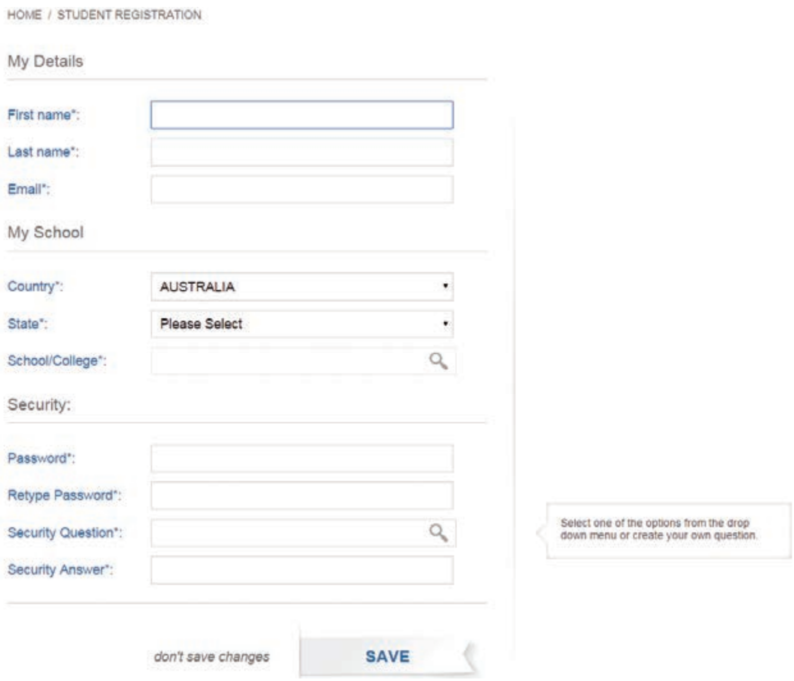
	Title	First name	Last Name	Mobile Number
	Dr	Elijah	Valentine	04 [redacted] 154 726
	Mr	Jeremy	McPherson	04 [redacted] 128 656
	Ms	Fiona	Holmes	04 [redacted] 236 195
	Mrs	Sierra	Jarvis	04 [redacted] 862 749
	Mrs	Maria	Johnson	04 [redacted] 123 456
	Ms	Zoe	Robertson	04 [redacted] 205 389
*				
	Total	6		

This row is a record.

FIGURE 8.3 An example of DBMS table with a field (`Last Name`) and a record (Zoe Robertson) indicated.

Input forms

While a table holds the DBMS data, it is not necessarily a user-friendly object. To make acquisition of data easier and more accurate, input forms are used. Their structure guides users to understand what has to be entered and in what format. Forms use labels to explain the purpose of fields, hiding the complex field name from the user. Data entered via a form goes directly into the fields of the underlying table.



HOME / STUDENT REGISTRATION

My Details

First name*:

Last name*:

Email*:

My School

Country*: AUSTRALIA

State*: Please Select

School/College*:

Security:

Password*:

Retype Password*:

Security Question*:

Security Answer*:

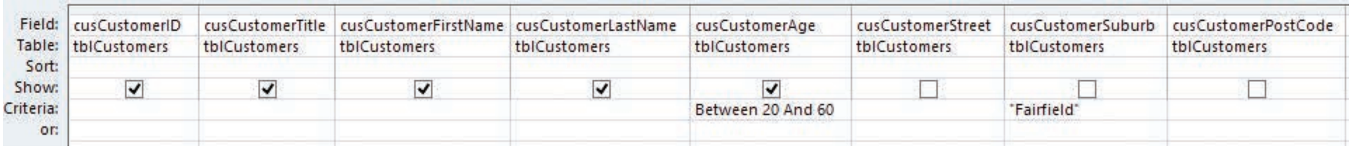
Select one of the options from the drop down menu or create your own question.

FIGURE 8.4 An example of an online input form

Queries

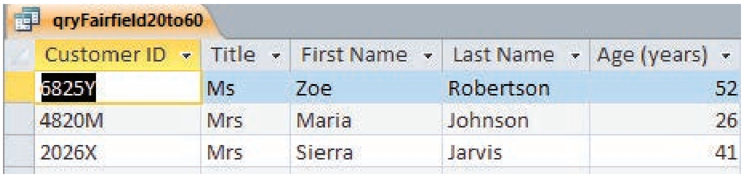
An important feature of a database is the ability to select a set of specific data based on a series of criteria. The database object that defines this selection is called a **query**. The query criteria are the result of a question we might have of the data. For example, 'How many of our customers in Fairfield are between the ages of 20 and 60?' (Figure 8.5) or 'Can I have a list of names and numbers for all our customers in Fairfield or Northcote?' (Figure 8.7).

Modern DBMSs use a structured query language or SQL as a way of standardising how data is managed in databases. Even DBMSs with GUI-based query creation, such as Microsoft Access, use SQL in the background (Figure 8.9).



Field:	cusCustomerID	cusCustomerTitle	cusCustomerFirstName	cusCustomerLastName	cusCustomerAge	cusCustomerStreet	cusCustomerSuburb	cusCustomerPostCode
Table:	tblCustomers	tblCustomers	tblCustomers	tblCustomers	tblCustomers	tblCustomers	tblCustomers	tblCustomers
Sort:								
Show:	☑	☑	☑	☑	☑	□	□	□
Criteria:					Between 20 And 60		"Fairfield"	
or:								

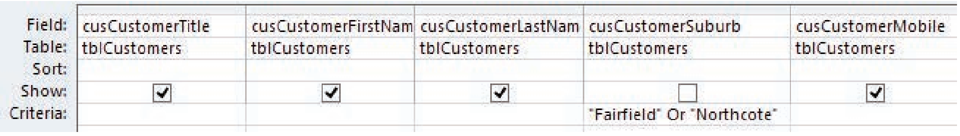
FIGURE 8.5 The design of a query that selects all customers living in Fairfield between the ages of 60 and 60



qryFairfield20to60

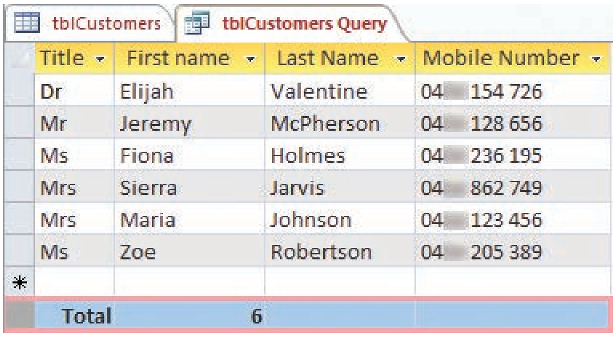
Customer ID	Title	First Name	Last Name	Age (years)
6825Y	Ms	Zoe	Robertson	52
4820M	Mrs	Maria	Johnson	26
2026X	Mrs	Sierra	Jarvis	41

FIGURE 8.6 The results of a query of all customers living in Fairfield between the ages of 20 and 60



Field:	cusCustomerTitle	cusCustomerFirstNam	cusCustomerLastNam	cusCustomerSuburb	cusCustomerMobile
Table:	tblCustomers	tblCustomers	tblCustomers	tblCustomers	tblCustomers
Sort:					
Show:	☑	☑	☑	□	☑
Criteria:				"Fairfield" Or "Northcote"	

FIGURE 8.7 The design of a query to list the names and mobile phone numbers of all customers living in Fairfield or Northcote

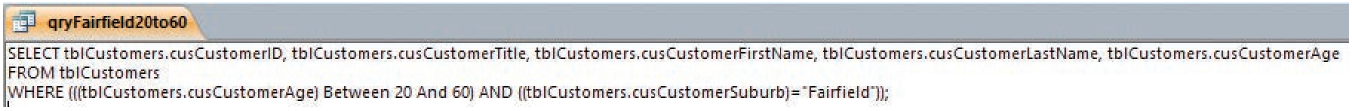


tblCustomers | tblCustomers Query

Title	First name	Last Name	Mobile Number
Dr	Elijah	Valentine	04 [redacted] 154 726
Mr	Jeremy	McPherson	04 [redacted] 128 656
Ms	Fiona	Holmes	04 [redacted] 236 195
Mrs	Sierra	Jarvis	04 [redacted] 862 749
Mrs	Maria	Johnson	04 [redacted] 123 456
Ms	Zoe	Robertson	04 [redacted] 205 389
*			
Total	6		

FIGURE 8.8 The results of a query to list the names and mobile phone numbers of all customers living in Fairfield or Northcote. The suburbs are not shown in the final query because they are not relevant to the needs of the user.

In Figures 8.5 and 8.7, each query has selected only the data we requested. A properly structured query helps to ensure any information we create is both relevant and complete.



qryFairfield20to60

```
SELECT tblCustomers.cusCustomerID, tblCustomers.cusCustomerTitle, tblCustomers.cusCustomerFirstName, tblCustomers.cusCustomerLastName, tblCustomers.cusCustomerAge
FROM tblCustomers
WHERE (((tblCustomers.cusCustomerAge) Between 20 And 60) AND ((tblCustomers.cusCustomerSuburb)="Fairfield"));
```

FIGURE 8.9 The SQL for the query in Figure 8.5

You query a database of websites when you type words in the search box on the Google main page.

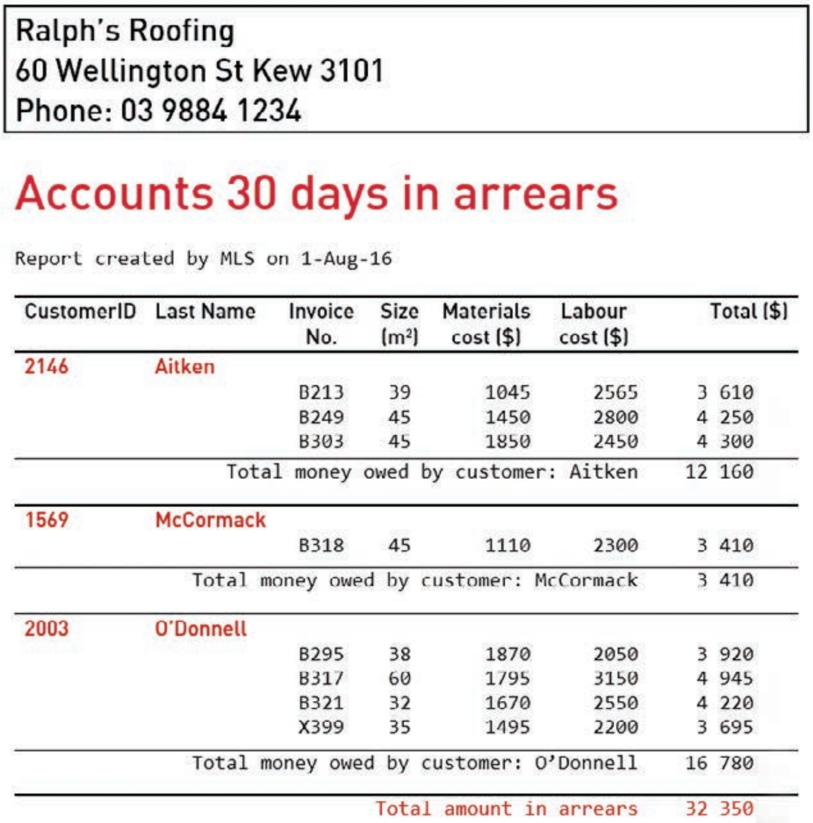
THINK ABOUT COMPUTING 8.3

Experiment with the advanced search tools in Google to see how to select a smaller set of results.

Elements of effectiveness related to solutions like reports include completeness, readability, attractiveness, clarity, accuracy, accessibility, timeliness, communication of message, relevance and useability.

Reports

As useful as a query is, it is merely a simple list of selected records. To convert this list into real information, you must manipulate its formatting and layout, and add summary totals. A report is the information produced at the end of this process. This is what you communicate to the person for whom the information is created. It will be laid out as effectively as possible to make the information stand out and be easy to read and interpret.



Ralph's Roofing
60 Wellington St Kew 3101
Phone: 03 9884 1234

Accounts 30 days in arrears

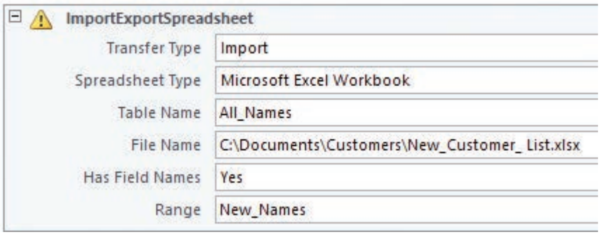
Report created by MLS on 1-Aug-16

CustomerID	Last Name	Invoice No.	Size (m^2)	Materials cost ($)	Labour cost ($)	Total ($)
2146	Aitken					
		B213	39	1045	2565	3 610
		B249	45	1450	2800	4 250
		B303	45	1850	2450	4 300
					Total money owed by customer: Aitken	12 160
1569	McCormack					
		B318	45	1110	2300	3 410
					Total money owed by customer: McCormack	3 410
2003	O'Donnell					
		B295	38	1870	2050	3 920
		B317	60	1795	3150	4 945
		B321	32	1670	2550	4 220
		X399	35	1495	2200	3 695
					Total money owed by customer: O'Donnell	16 780
					Total amount in arrears	32 350

FIGURE 8.10 An example of a database report. The information has been grouped and sorted, and summary totals calculated to make the report easier to understand.

Macros

In many cases, we may have to run a series of tasks whenever a database is opened or while it is in operation. For example, a data set may regularly need to be imported from another data source. The series of steps required to complete this process can be automated by a macro. The benefit of a macro is that it carries out the exact steps we ask of it when a single button or key combination is selected. This ensures the steps are followed accurately every time, and saves time and effort.



ImportExportSpreadsheet

Transfer Type	Import
Spreadsheet Type	Microsoft Excel Workbook
Table Name	All_Names
File Name	C:\Documents\Customers\New_Customer_ List.xlsx
Has Field Names	Yes
Range	New_Names

FIGURE 8.11 An example of the coding of a macro. This macro imports data from a spreadsheet file and appends it to a table called `All_Names`.

Characteristics of data types

The data held by a DBMS must be in a format suitable for storage. Typically, each field will be created to hold a specific type of data. This ensures data consistency in each field and assists with validation, because a field will refuse to accept data of the wrong type. For the purposes of this study, you must be familiar with text (string), numeric (integer and floating point), date, character and Boolean data types. Each of these is discussed in the section to follow Table 8.1.

Data types are also discussed in Chapters 6 and 7.

TABLE 8.1 The most common data types used in databases

Data type	Characteristics or uses	Examples
Text	• Alphanumeric up to 255 characters, searchable	• Name, address, postcode and phone number
Numeric	• Numbers only (see below for different forms)	• Any number to be used in a calculation
Currency	• Numbers, but in dollar amounts formatted with $ symbol and includes .00 at the end as a default	• Any number used to represent a financial value, usually only applied to a total, rather than values in a list
Date/Time	• A variation of numbers formatted to represent a date and/or time	• Any date can be used in calculations
Boolean	• Represents one of two states, usually True/False can be used in logical tests	• Also represented as Yes/No and On/Off

Text (string)

The majority of fields can be set as a text data type. Text may also be referred to as a string, particularly in connection with programming. This type of field holds a mix of characters (letters, numbers, special characters), also referred to as alphanumeric, to a limit of 255. Names and addresses would be considered to be text data. Also, postcodes and telephone numbers are normally formatted as text because they may contain spaces and are not intended to be used in the same way as a numeric value. It is also more efficient to store the values as text rather than as a large numeric value.

Numeric – integer, floating point

Another key type of data format used in databases is numeric. These fields will only allow numbers to be entered. They are often used when the value is to be used in a calculation of some kind. For example, the quantity of an item purchased might need to be multiplied by its price in order to calculate a total amount payable. This sort of calculation cannot be performed on a field formatted for text. Numeric fields can also be categorised into different variations. Integer is used for whole numbers, including negative numbers. Where decimal numbers are required, such as when dealing with financial transactions or percentages, then the floating point data type is used. Depending on the database package you use, the specific names of these numeric data types may vary, but their function will be the same.

Date

Strictly speaking, a date format is another variation of a numeric data type. The value is normally based on the number of days since the 'zero' day built into the operating system or DBMS. For example, the day '1' might be displayed as 01 January 1900, while '42673' would be displayed as 30 October 2016. Calculations can be performed on dates, which can be handy when comparing the difference between them. Dates can be formatted to show a combination of years, months, days, hours, minutes and seconds, depending on the needs of the user. In terms of time, they can also display 12- and 24-hour clocks.

Character

This is a text field that will only accept a single alphanumeric character. It is used where there are multiple options for a value, but they can be represented with a single character to make data entry easier and to save storage space. For example, a wooden box that comes in small, medium and large sizes might be entered as S, M and L, respectively.

Boolean

Boolean data type is used when the data to be entered falls into one of two categories. This is usually True/False, but it may also represent Yes/No or On/Off. It is sometimes shown as a checkbox or radio button on forms.

DATA TYPES

List two examples of each of the data types below. Note: You may also find it useful to refer back to Chapters 6 and 7.

1 Text (string)

2 **a** Numeric – integer

b Numeric – floating point

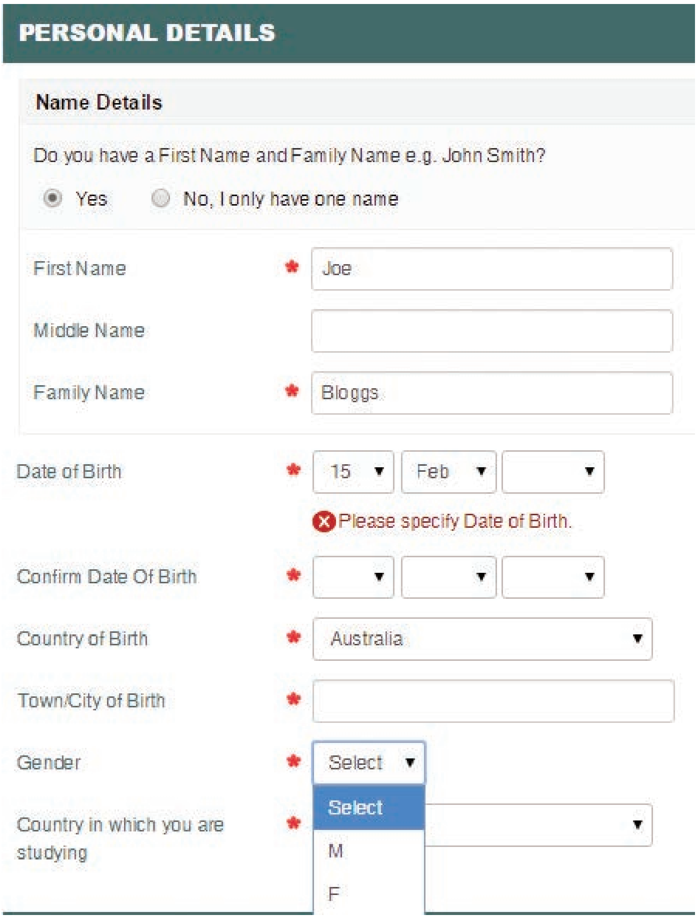
3 Date/Time

4 Boolean

Data sources and methods of data acquisition

In this Outcome, students are expected to acquire their own data via a data collection tool that they design. The data collection tool does not have to be online, instead it will most likely be a form constructed from within their DBMS.

In Chapter 1, you covered various data sources and methods by which data can be acquired. These included various types of surveys and questionnaires involving a variety of open and closed questions. As has been discussed earlier in this chapter, to make use of any online system, such as banking, to join a group, to purchase products or to vote, you will be asked to register. As a minimum, you will need to supply basic contact details. Authentication may only require a valid email address, but banks would also ask for formal identification such as a driver's licence or birth certificate. To supply the requested data, you would complete an input form, similar to Figure 8.12 (characteristics of these forms are covered in more detail on page 315). When an online form is completed and submitted, the data is checked or validated, then written to a database. If a hardcopy form is used, then the details are either manually transcribed or scanned electronically into the database.



PERSONAL DETAILS

Name Details

Do you have a First Name and Family Name e.g. John Smith?

● Yes ○ No, I only have one name

First Name	*	Joe
Middle Name		
Family Name	*	Bloggs
Date of Birth	*	15 ▾ Feb ▾ ▾
		☒ Please specify Date of Birth.
Confirm Date Of Birth	*	▾ ▾ ▾
Country of Birth	*	Australia ▾
Town/City of Birth	*	
Gender	*	Select ▾ Select M F
Country in which you are studying	*	▾

Department of Education and Training Creative Commons Attribution 3.0 Australia Licence

FIGURE 8.12 An example of an online application form. Note the mandatory fields marked with a red * and the use of dropdown boxes to assist with data validation.

Requirements for safeguarding personal information are discussed on page 332.

THINK ABOUT COMPUTING 8.4

List the personal details you need to supply in order to join a social media site like Facebook. Why might they need each of those specific elements of data? What items of data are optional?

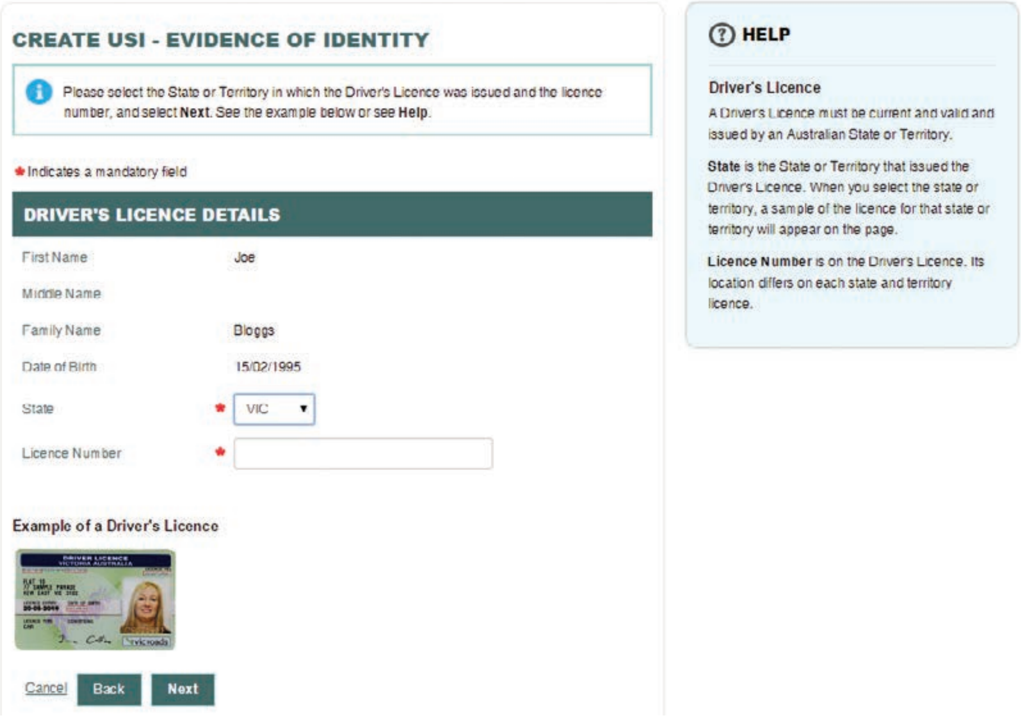
Collection tools and user interfaces for data entry

To ensure that data gathered via input forms is as accurate as possible, there are several techniques that make it easier for users to enter data. The form must guide users as to what types of data must be entered in different fields and what the accepted parameters or boundaries are for that data. The form itself needs to be arranged logically, usually to allow data entry starting from the top, and ensure that a user cannot complete the form without entering key data elements, like an email address and password.

Forms will involve several types of data validation to ensure the data entered is as accurate as possible. You will recall from previous chapters that validation involves checking that data exists in a field, is the correct type of data for that field and that it fits within the field's accepted value boundaries. **Existence checking** can be achieved by the use of mandatory entry fields. These are often marked with an asterisk (*) or might be in a different, eye-catching colour to the other fields. **Data type** validation can be achieved if you include labels showing examples of the expected values beside each field, and provide appropriate data-entry tools, such as numeric keypads and calendars, to encourage the correct data type to be entered. Labels can also indicate **correct ranges** of values that can be entered or by using dropdown lists with prefilled options. The form might generate a suggested list of correct spellings if it detects the user has misspelled something important, such as part of an email address. These tools can also reduce errors caused by inconsistency.

An effective input form will also incorporate a number of other important design features to make the experience better for the user. It is important that forms reduce the amount of onscreen clutter by only displaying fields relevant to the user. In this way, forms can be adaptive; for example, when a user selects a certain option (such as delivery method for goods purchased), only fields relevant to that option (the type of delivery) are shown. Alternatively, as a user types in a credit card number, the type of card is determined and the options available change based on the card.

Forms must also display an appealing and attractive design overall.



CREATE USI - EVIDENCE OF IDENTITY

Please select the State or Territory in which the Driver's Licence was issued and the licence number, and select **Next**. See the example below or see **Help**.

*Indicates a mandatory field

DRIVER'S LICENCE DETAILS

First Name		Joe
Middle Name		
Family Name		Bloggs
Date of Birth		15/02/1995
State	*	VIC ▾
Licence Number	*	

Example of a Driver's Licence

HELP

Driver's Licence

A Driver's Licence must be current and valid and issued by an Australian State or Territory.

State is the State or Territory that issued the Driver's Licence. When you select the state or territory, a sample of the licence for that state or territory will appear on the page.

Licence Number is on the Driver's Licence. Its location differs on each state and territory licence.

FIGURE 8.13 Before continuing with a registration, a user must be validated by providing details of an accepted type of personal identification. Notice how the form tries to assist the user by providing an example of the form of identification to use and other helpful advice.

Department of Education and Training Creative Commons Attribution 3.0 Australia Licence

The role of a user experience (UX) designer is to create an online environment that encourages the user to interact with the system with ease by providing an interface that is both usable and accessible.

Checklist of common effective form elements

- Explain form fields clearly with adjacent labels
- Use indicators to show strength of passwords
- Use inline validation of fields
- Automatically suggest corrections for misspelled email addresses
- Provide a simple interface with large buttons for mobile phone users
- Adapt forms depending on options chosen (for example, delivery method, type of credit card)
- Convert text to a consistent format
- Vertically align labels and fields
- Provide context-sensitive help or information
- Provide clear steps if a user must navigate several input screens
- Use input tools that match the type of data to be entered (such as numeric keypads, sliders and tick boxes)
- Reduce form clutter so only what is relevant is on the screen
- Use an appealing overall design

The case study below is broken into several parts throughout this chapter. Its purpose is to show you a worked-through example of a database solution. Put yourself in the role of Saro. You, as the student, will be expected to come up with your own design brief and acquire your own data via a data-collection tool that you design. In this example, an online data-collection tool is used, but generally you are more likely to use a form you create within your DBMS.

CASE STUDY KALUGANGA JUNIOR SWIMMING CLUB

The Kaluganga Junior Swimming Club is located in Doncaster. It provides swimming training and its members compete in regular local swimming meets. As this is a junior swimming club, members are 20 years old or younger, except for committee members, who are adults. The club records address, age and contact details for each member as well as their best times for several 50 m events. At present, each swimmer's record is stored in a separate word processing document that uses their name as the filename. After training sessions or after competition meets, the new times are compared with the existing times and updated if necessary on each swimmer's record. The club coach and other committee members often need to use the members' details to determine who the best swimmers are for certain events and who is up to date with membership fees.

There have been some problems with the current system. Sometimes the lists of swimmers are inaccurate because names are mistakenly put into the wrong age group or left off the list totally! The club's finanaces are not in order, as it has not accurately identified and followed up members who have not paid their annual fees. This has led to embarrassment for club officials when payment notices have been sent to the wrong club members.

As a result, Felix, the swimming club president, has asked his friend Saro to look at how to solve the problems with a digital-systems-based solution.

Saro has conducted an analysis of the problems and suggested a database solution that includes the ability to add and edit swimmer details and generate lists for the coaches and club treasurer. She has established the following requirements.

Functional requirements

- New swimmers must be able to be added.
- Existing swimmers can have their details edited.
- A list of all swimmers, grouped by age level and event can be generated.
- A list of all swimmers who owe club fees can be generated.
- The list for coaches must include all relevant performance details for each swimmer, including their best times.
- A macro will be used to open and print reports.

Non-functional requirements

- Interface for entering data must be simple to use.
- Results from reports must be accurate.

Constraints

The club wants to use its existing PC running a recent edition of an operating system and a popular office package that includes a DBMS.

Felix and the other committee members are confident with PC use, but unfamiliar with database software.

The solution needs to be in use before the next swim meet in a month's time.

Case study continues ▶

◀ *Case study continued*

Scope of solution

The solution will allow the club to:

- add, remove and edit records of swimmers
- generate lists of swimmers by age level and event within a certain swimming time and grouped by gender
- generate lists of swimmers with overdue payment of fees.

The solution will not:

- allow swimmers to edit their own details
- support online payments of fees.

This solution will benefit the club because:

- it will create accurate lists of swimmers and age groups for the coaches, allowing them to choose from the right pool of members for each competition event
- it will create accurate and up-to-date lists of members who are yet to pay their annual fees, which will assist in improving club revenue and reducing financial difficulties.

Database design tools

Database structure and appearance can be planned using a variety of common planning tools. Data dictionaries, data-structure designs, flowcharts, input–process–output (IPO) charts and query diagrams are useful for representing structure. **Layout diagrams** and mock-ups can also be used to plan the appearance of input forms and reports.

Several of these design tools will be further explained in the context of a DBMS in the following section.

Naming conventions

Every field and object within a database must be named properly so that it can easily be identified by a developer or a user. For example, you might prefix tables with `tbl` (such as `tblCustomer`, `tblProducts`) and fields within the Customer table with `cus` (such as `cusCustomerID`, `cusAddress`). Queries might be prefixed with `qry`, forms with `frm` and reports with `rpt`. Because users will interact with forms and reports often, ensure that the rest of the name is descriptive; for example, `frmInputCustomerDetails` or `rptOverdueInvoices` (Figure 8.14). From these examples, note that spaces and underscores are avoided in the object names. To aid understanding there is, however, a mix of upper- and lower-case characters in each name. In most databases a user would interact with a menu system with properly labelled buttons, rather than the individual objects in the main database screen.

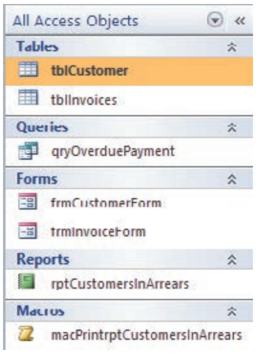


FIGURE 8.14 Examples of named objects within a Microsoft Access database file

Design tools to represent the structure of databases

The most common tool used to represent the basic structure of a database table is a **data dictionary**. This tells a database developer exactly how to setup the properties of each field in the tables. Remember that tables do not hold formulas, but can include validation. When planning fields, think about how the data might be used and how it might be best broken down. For example, the name 'Dr Andrea Smith' might be broken into three fields of `Title` (Dr), `FirstName` (Andrea) and `FamilyName` (Smith). Also look carefully at the type of data that the field will hold to determine the most appropriate data type to select.

TABLE 8.2 Example of a data dictionary to store invoices for a roofing contractor

Field	Data type	Field size	Input mask	Caption	Description	Validation rule	Validation text
`invInvoiceNum`	Text	4		Quote number	Quote number (primary key)		
`invCustomerID`	Text	4		Customer ID	Customer's identification code		
`invSize`	Number	Integer			Size of roof	Between 0 and 500	Normal building size should be between 0 and 500
`invCostMaterials`	Number	Integer		Cost of materials ($)	Cost of roofing materials used	Between 0 and 100 000	Cost of materials must be between 0 and 100 000
`invCostLabour`	Number	Integer		Cost of labour ($)	Cost of labour used	Between 0 and 20 000	Cost of labour must be between 0 and 20 000
`invColour`	Text	15		Roof colour	Colour of roof		
`invJobDate`	Date/Time	dd-MM-YYYY		Date of job	Date the job was completed	Between #01/01/16# and #31/12/17#	Job completion must be in 2016 or 2017
`invAccountPaid`	Yes/No			Account paid?	Indicates whether or not the account has been paid		

When relational databases are designed, a data-structure diagram is used to show the relationships between each of the tables.

CASE STUDY

KALUGANGA JUNIOR SWIMMING CLUB – PART 2

Saro has taken the existing list of swimming club members and their details to determine how they will be represented in the database table. She has also added fields and included validation on the gender, postcode, date-of-birth and members-fees fields.

TABLE 8.3 Data dictionary for the Kaluganga Junior Swimming Club swimmers database `tblSwimmersDetails` table

Field	Data type	Field size	Input mask	Caption	Description	Validation rule	Validation text
`swmID`	Text	5		Membership number	Membership number		
`swmFirstName`	Text	20		First name			
`swmFamilyName`	Text	20		Family name			
`swmGender`	Text	6		Gender (M/F)	Male or Female	'Male' or 'Female'	Gender must be Male or Female
`swmAddress`	Text	100		Address			
`swmSuburb`	Text	20		Suburb			
`swmPostcode`	Text	4		Postcode		Between 3000 and 3999	Postcode must be between 3000 and 3999 inclusive

Case study continues ▶

◀ Case study continued

Field	Data type	Field size	Input mask	Caption	Description	Validation rule	Validation text
swmMobNumber	Text	14		Mobile number			
swmParent	Boolean	N/A		Parent contact?	Is the contact number that of a parent (True) or the swimmer (False)?		
swmDOB	Date	N/A	dd-MM-yyyy	Date of birth		<=Date()	DOB must be today or earlier
swmMemberFees	Numeric	3		Annual fee ($)	How much a member needs to pay each year	90	Annual fees are $90
swmFeesPaid	Numeric	3		Fees paid ($)	How much a member has paid this year		
swmFreestyleTime	Text	12		Freestyle time	Best recorded time for 50 m freestyle		
swmBreastStrokeTime	Text	12		Breaststroke time	Best recorded time for 50 m breaststroke		
swmBackstrokeTime	Text	12		Backstroke time	Best recorded time for 50 m backstroke		
swmButterflyTime	Text	12		Butterfly time	Best recorded time for 50 m butterfly		

Layout diagrams are also discussed in Chapter 5.

The other elements that need to be designed for the database include the input forms, queries to select data, reports to format the query results and macros to make database elements more efficient.

The key design tool used for input forms and reports is the layout diagram. It is essentially a sketch of what an input form or the reports of the solution will look like. It shows an interface developer the location of elements such as headings, labels and fields. This layout is based on appropriate use of formats and conventions. The layout diagram is annotated to show formatting details for all elements, including font type, size and style and options for selection lists. Any formulas to be added to a form or report will also be shown on a layout diagram. Like other design elements, layout diagrams are usually done by hand (see Figure 8.15), although pre-printed 'templates' can be used for some common elements or structures.

Balsamiq is a free handy online tool for creating layout diagrams. Your school can create a free account.

Design tools for representing input forms to capture data

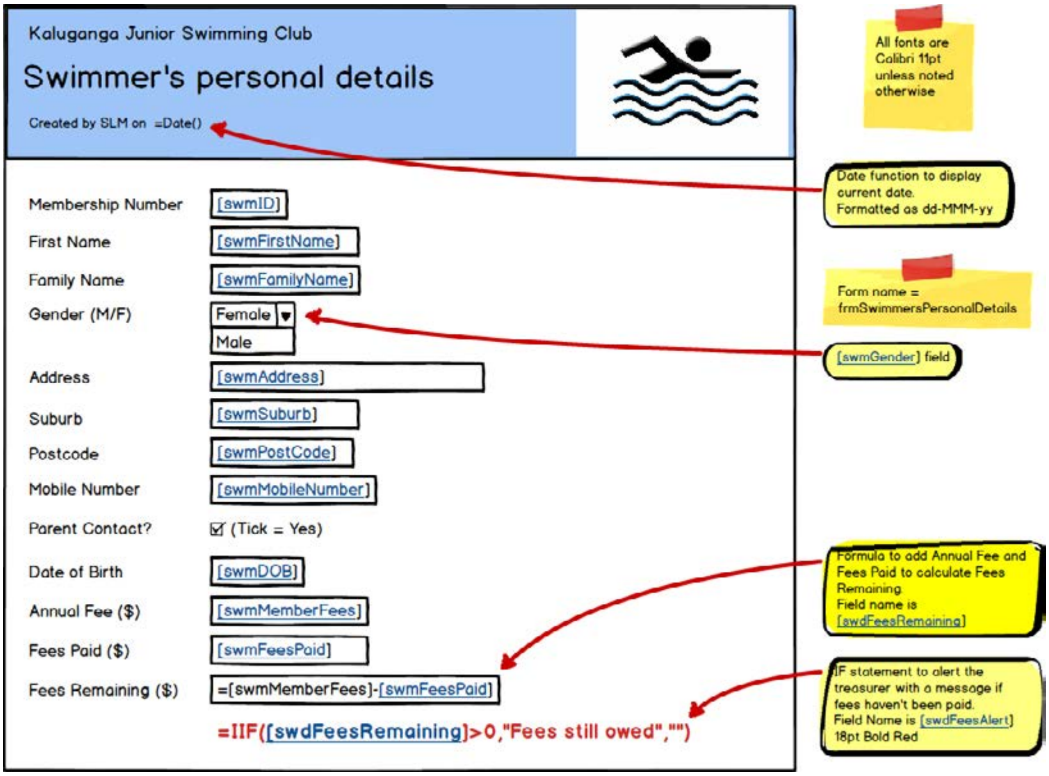
When planning a layout diagram for a form, you need to ensure that it contains all the fields a user needs for their purpose. Elements of effective forms were explained earlier in this chapter. The fields should be laid out in a logical order, usually from top to bottom and in columns if there are too many fields for a single column. Headings can be used to arrange relevant fields into logical groups. Ensure that they have clear labels and make use of dropdown lists to assist with data entry. All formatting must also be clearly indicated. See Figures 8.15 and 8.16 for examples of annotated layout diagrams for input forms.

CASE STUDY KALUGANGA JUNIOR SWIMMING CLUB – PART 3

Saro has designed a simple input form that Felix and the treasurer can use to manage (add, edit and delete) swimmers. She has incorporated a dropdown list for the gender field and a tick box for the field that notes if the mobile phone number field is a parent's number.

She has also designed a second form for the coaches that only shows key personal details and swimming event information.

Both forms use the same underlying table, but they only show data relevant to the needs of each user.



Kaluganga Junior Swimming Club

Swimmer's personal details

Created by SLM on =Date()

Membership Number	[swmID]
First Name	[swmFirstName]
Family Name	[swmFamilyName]
Gender (M/F)	Female ▾ Male
Address	[swmAddress]
Suburb	[swmSuburb]
Postcode	[swmPostCode]
Mobile Number	[swmMobileNumber]
Parent Contact?	☑ (Tick = Yes)
Date of Birth	[swmDOB]
Annual Fee ($)	[swmMemberFees]
Fees Paid ($)	[swmFeesPaid]
Fees Remaining ($)	=[swmMemberFees]-[swmFeesPaid]

=IIF([swdFeesRemaining]>0,"Fees still owed","")

All fonts are Calibri 11pt unless noted otherwise

Date function to display current date. Formatted as dd-MMM-yy

Form name = frmSwimmersPersonalDetails

[swmGender] field

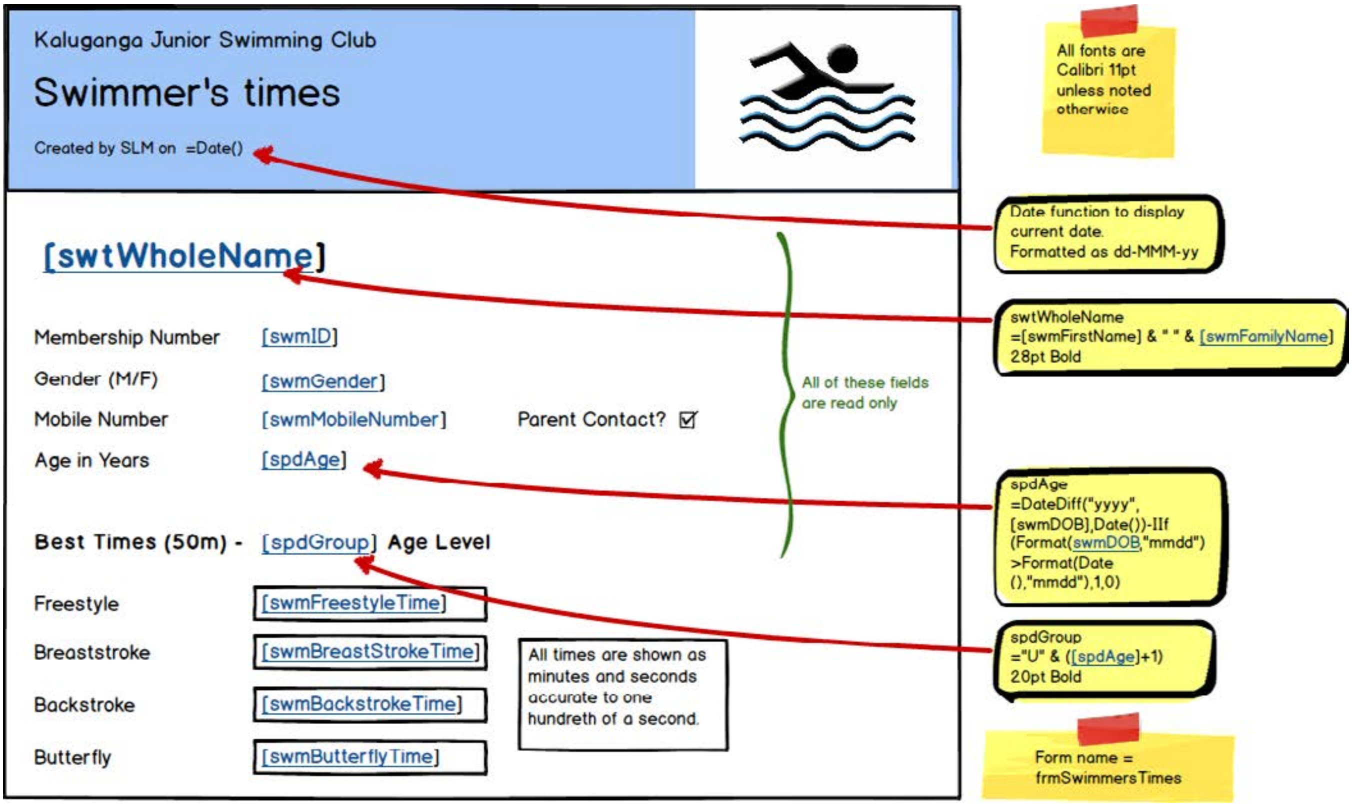
Formula to add Annual Fee and Fees Paid to calculate Fees Remaining Field name is [swdFeesRemaining]

IF statement to alert the treasurer with a message if fees haven't been paid. Field Name is [swdFeesAlert] 18pt Bold Red

FIGURE 8.15 Saro's layout diagram for the main input form for the Kaluganga database `frmSwimmersPersonalDetails`. She has listed all the key personal details for each swimmer plus their membership fee payment details. The form has a heading section with the title, club logo and the current date. There are no swimming event times on this form, as it is used by Felix and the club treasurer.

Case study continues ▶

◀ Case study continued



Kaluganga Junior Swimming Club

Swimmer's times

Created by SLM on =Date()

[swtWholeName]

Membership Number [swmID]

Gender (M/F) [swmGender]

Mobile Number [swmMobileNumber] Parent Contact? ☑

Age in Years [spdAge]

Best Times (50m) - [spdGroup] Age Level

Freestyle [swmFreestyleTime]

Breaststroke [swmBreastStrokeTime]

Backstroke [swmBackstrokeTime]

Butterfly [swmButterflyTime]

All times are shown as minutes and seconds accurate to one hundreth of a second.

All of these fields are read only

All fonts are Calibri 11pt unless noted otherwise

Date function to display current date. Formatted as dd-MMM-yy

swtWholeName =[swmFirstName] & " " & [swmFamilyName] 28pt Bold

spdAge =DateDiff("yyyy",[swmDOB],Date())-IIf(Format(swmDOB,"mmdd")>Format(Date(),"mmdd"),1,0)

spdGroup ="U" & ([spdAge]+1) 20pt Bold

Form name = frmSwimmersTimes

FIGURE 8.16 Saro's layout diagram for the input form coaches can use to update swimmers' times for events (`frmSwimmersTimes`). This form only has the personal details relevant to what the coaches need. They cannot edit any details except the times. This form also includes a calculation to show each swimmer's current age, in years, and their age group. It also contains a header with the same information as the other form.

Designing queries

Queries are used in databases to select only those records that meet the query criteria. They can also be used as a control structure to hold formulas used on forms and on other queries. The result of a query is a list of selected records organised by specified fields. This list is then used as the basis for a printed or electronic report.

The design of a query should specify the fields to be included and the tables to which they belong. The design will also include the criteria for the query and how the resulting records are to be ordered. The query design is usually hand drawn, although you may set up a basic table as a template. In your solution, you should use at least two criteria and both a primary and a secondary sort (sorting is explained in 'Setting up the queries' later in this chapter). Also, be aware that not all fields used for selecting data need to be shown in the final list.

If formulas are created in the query, you should ensure that they have the correct data type (for example, text or numeric) and a proper name applied.

TABLE 8.4 A sample query design for calling up records of overdue accounts. The records will be ordered by the last name of the customer and then the invoice number, since the leftmost sort designation (primary sort key) is assigned to `LastName`.

Field	cusCustomer ID	cusLast Name	invInvoice Num	invSize	invCost Materials	invCost Labour	TotalCost:=[invCost Materials]+[invCostLabour]	invJob Date	invAccount Paid
Sort		Ascending	Ascending						
Show	✓	✓	✓	✓	✓	✓		✓	
Criteria								<#01/08/16#-30	False

Query criteria

The query criteria can use <, >, =, <=, >=, and <> symbols when dealing with numbers or text. Alternatively, plain language such as 'between 0 and 20 000' or 'less than 500' can be used. The '*' and '?' symbols are also valid wildcards. Text used in query criteria needs to be placed inside single inverted commas. Other expressions such as 'is null' or 'is not null' can be used to show records where a field is either empty or contains some data. A query can designate a date by placing '#' marks at each end of the date. If more than one field has a query criterion assigned to it, the criteria are applied in an 'and' relationship. That is, both the criteria must be met before data is returned by the query.

CASE STUDY KALUGANGA JUNIOR SWIMMING CLUB – PART 4

Saro has designed two queries for the Kaluganga Junior Swimming Club database. One of the queries will select all swimmers who have overdue fees. Because Felix wants to call the parents of these swimmers to discuss payment, the query will also only show those swimmers whose parent's mobile phone number is listed. The other query will select all under-17 swimmers with swimming times below a certain threshold in backstroke (40 seconds) to enable the coaches to select the fastest swimmers for that race.

TABLE 8.5 The design of the query to select swimmers with overdue fees (`qryOverdueFeesParentContact`). A formula called `swdFeesRemaining` has been created to calculate the value of unpaid fees. If it is greater than 0 (0 means all the fees have been paid) then the record is selected.

Field	swmFamily Name	swmFirst Name	swmID	swmMobile Number	swmMember Fees	swmFees Paid	swdFeesRemaining: [swmMemberFees]-[swmFeesPaid]	swm Parent
Sort	Ascending	Ascending						
Show	✓	✓	✓	✓	✓	✓	✓	
Criteria							>0	True

Case study continues ▶

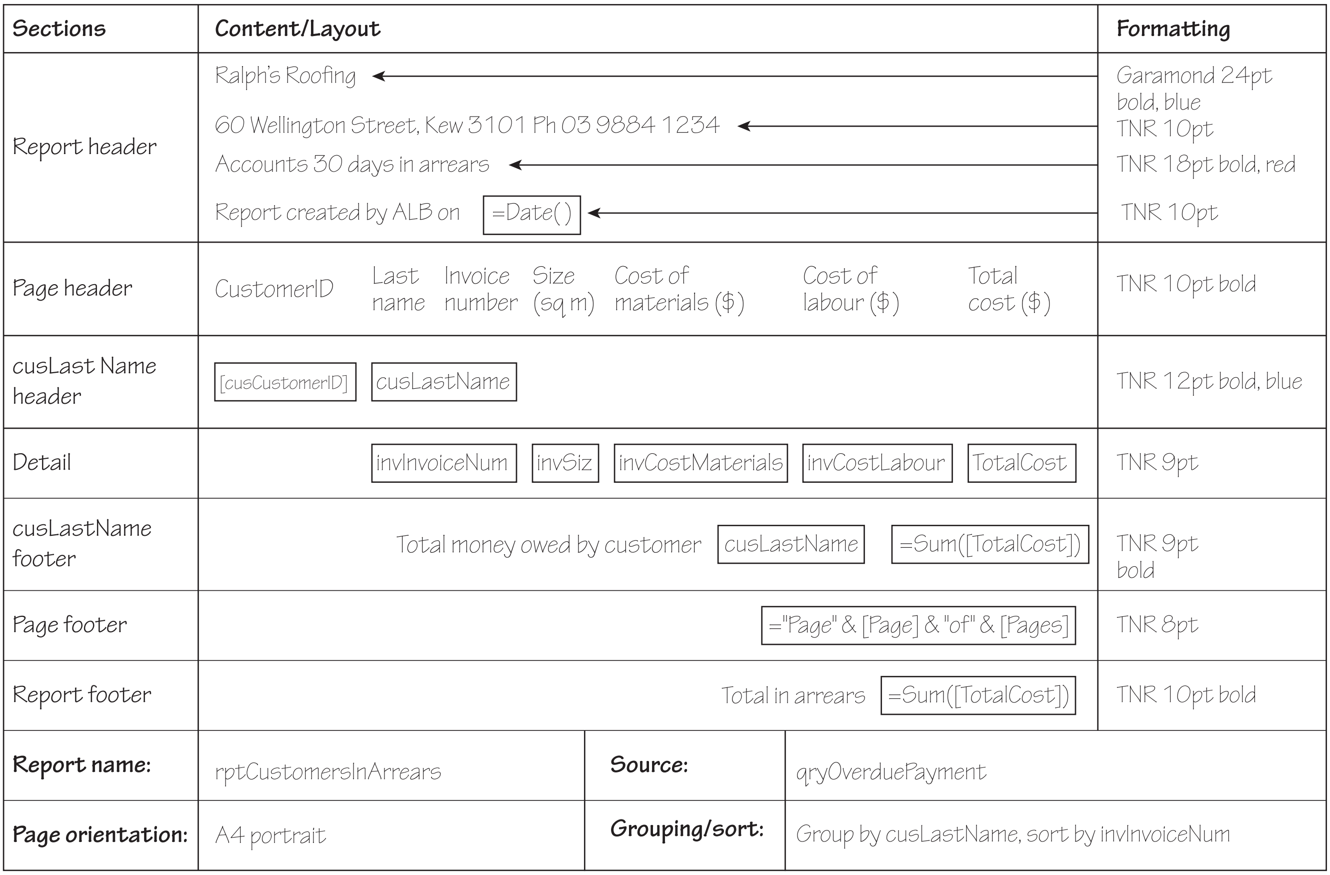
◀ *Case study continued*

TABLE 8.6 The design of the query to list the fastest U17 backstroke swimmers (`qryU17BackstrokeBestTimes`). It selects swimmers who have a backstroke time of less than 40 seconds and who are also under 17. The long formula is intended to calculate the correct age in years as accurately as possible. All swimmers are under 17, so that field does not need to be shown when the query runs.

Field	swmGender	swmBackStrokeTime	swmFamilyName	swmFirstName	"U" & DateDiff("yyyy",[swmDOB],Date())-IIf(Format([swmDOB],"mmdd")>Format(Date(),"mmdd"),1,0)+1
Sort	Ascending	Ascending	Ascending		
Show	✓	✓	✓	✓	
Criteria		<"00:40.00"			"U17"

Design tools to help reports meet specific needs

As discussed earlier in this chapter, the result of a query is used to generate a report. A layout diagram is used to represent the structure of a report. Reports have common sets of formats and conventions. Other essential elements specific to reports include the organisation's details and date in the report header, grouping of information and summary statistics. The records that are returned from the query are placed in the detail section of the report.



Sections	Content/Layout	Formatting
Report header	Ralph's Roofing 60 Wellington Street, Kew 3101 Ph 03 9884 1234 Accounts 30 days in arrears Report created by ALB on =Date()	Garamond 24pt bold, blue TNR 10pt TNR 18pt bold, red TNR 10pt
Page header	CustomerID · Last name · Invoice number · Size (sq m) · Cost of materials ($) · Cost of labour ($) · Total cost ($)	TNR 10pt bold
cusLast Name header	[cusCustomerID] · cusLastName	TNR 12pt bold, blue
Detail	invInvoiceNum · invSiz · invCostMaterials · invCostLabour · TotalCost	TNR 9pt
cusLastName footer	Total money owed by customer · cusLastName · =Sum([TotalCost])	TNR 9pt bold
Page footer	="Page" & [Page] & "of" & [Pages]	TNR 8pt
Report footer	Total in arrears · =Sum([TotalCost])	TNR 10pt bold

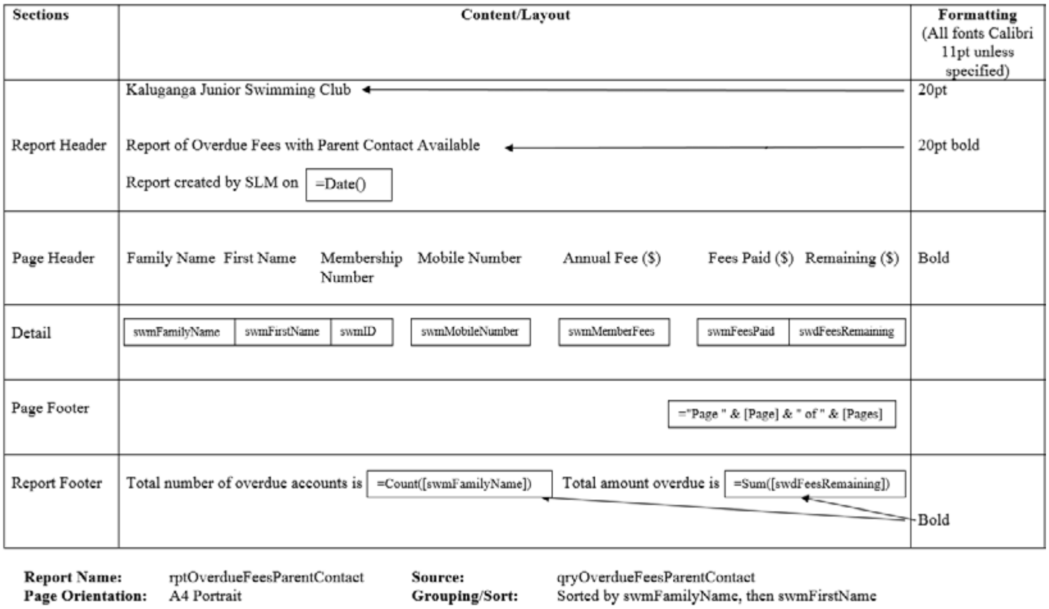
Report name:	rptCustomersInArrears	**Source:**	qryOverduePayment
Page orientation:	A4 portrait	**Grouping/sort:**	Group by cusLastName, sort by invInvoiceNum

FIGURE 8.17 An example of a grouped report design. This is the report shown in Figure 8.10.

Note the report elements down the left-hand side. These correspond with the section headings used in the Design View of the report in Microsoft Access.

CASE STUDY KALUGANGA JUNIOR SWIMMING CLUB – PART 5

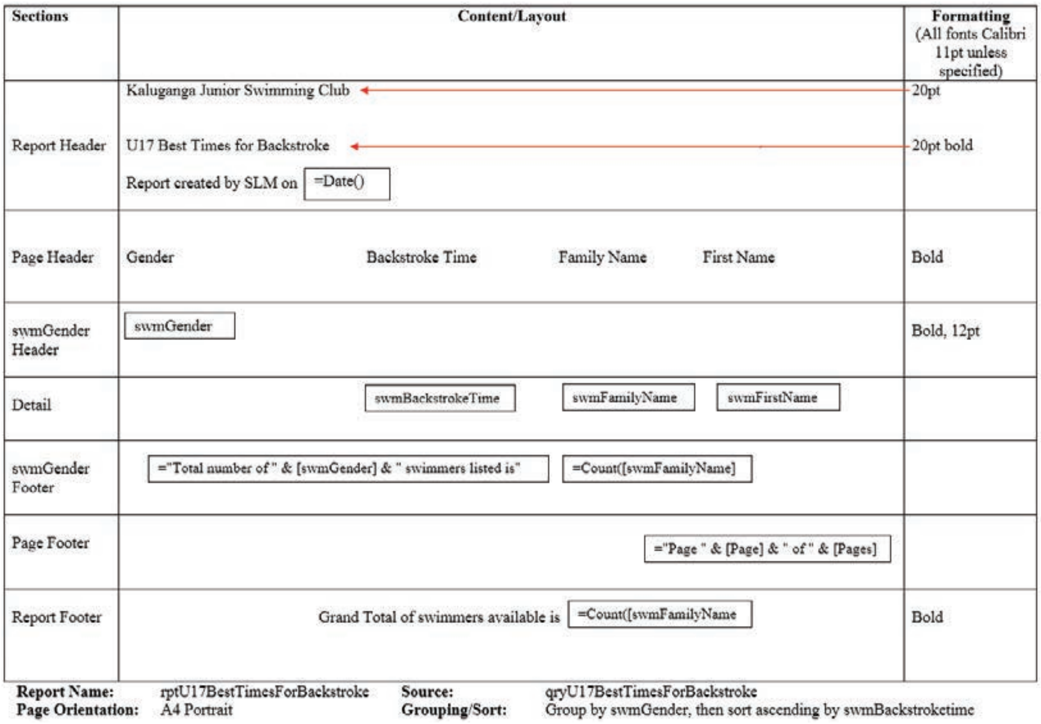
Saro has designed two reports for the swimming club. Each report is based on one of the queries designed above. Saro shows the designs to Felix and the treasurer, who make suggestions for minor changes that will make the reports clearer and more readable.



Sections	Content/Layout	Formatting (All fonts Calibri 11pt unless specified)
Report Header	Kaluganga Junior Swimming Club Report of Overdue Fees with Parent Contact Available Report created by SLM on =Date()	20pt 20pt bold
Page Header	Family Name First Name Membership Number Mobile Number Annual Fee ($) Fees Paid ($) Remaining ($)	Bold
Detail	swmFamilyName swmFirstName swmID swmMobileNumber swmMemberFees swmFeesPaid swdFeesRemaining	
Page Footer	="Page " & [Page] & " of " & [Pages]	
Report Footer	Total number of overdue accounts is =Count([swmFamilyName]) Total amount overdue is =Sum([swdFeesRemaining])	Bold

Report Name: rptOverdueFeesParentContact **Source:** qryOverdueFeesParentContact
Page Orientation: A4 Portrait **Grouping/Sort:** Sorted by swmFamilyName, then swmFirstName

FIGURE 8.18 This is the design for Felix's 'Report for overdue fee with parent contact available' report. The report is to be sorted on the 'FamilyName' field, then the 'FirstName' field.



Sections	Content/Layout	Formatting (All fonts Calibri 11pt unless specified)
Report Header	Kaluganga Junior Swimming Club U17 Best Times for Backstroke Report created by SLM on =Date()	20pt 20pt bold
Page Header	Gender Backstroke Time Family Name First Name	Bold
swmGender Header	swmGender	Bold, 12pt
Detail	swmBackstrokeTime swmFamilyName swmFirstName	
swmGender Footer	="Total number of " & [swmGender] & " swimmers listed is" =Count([swmFamilyName]	
Page Footer	="Page " & [Page] & " of " & [Pages]	
Report Footer	Grand Total of swimmers available is =Count([swmFamilyName	Bold

Report Name: rptU17BestTimesForBackstroke **Source:** qryU17BestTimesForBackstroke
Page Orientation: A4 Portrait **Grouping/Sort:** Group by swmGender, then sort ascending by swmBackstroketime

FIGURE 8.19 This layout diagram shows the design of the report for the coaches, which selects the best under-17 backstroke swimmers. This report is grouped by gender, then sorted in ascending order by time.

Designing macros

Macros can be planned using pseudocode or a flowchart. When using pseudocode to design a macro, be sure to use indenting and a logical order of steps, so that it is easy to see what it is intended to do and to make testing easier.

Pseudocode, also known as structured English, is discussed in Chapter 6.

CASE STUDY KALUGANGA JUNIOR SWIMMING CLUB – PART 6

Saro has decided to create two simple macros that print the two reports. The tasks that she wants to perform include opening the report, printing it to the default printer and then closing it. Her pseudocode design for one of these macros is shown in Figure 8.20. Saro will also create a macro that opens the swimmer's times form for editing. This macro may be used in conjunction with a button on a 'Dashboard'. The pseudocode for this macro is shown in Figure 8.22.

```
Begin
    Open Report 'rptOverdueFeesParentContact'
    Print report to default printer
Close report
End
```

FIGURE 8.20 Saro's pseudocode design for one of the macros that opens a report and prints it, then closes the report

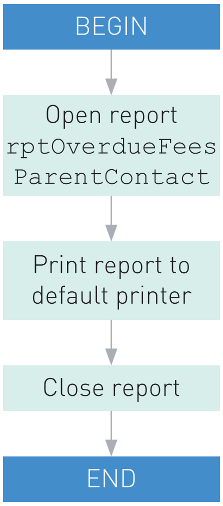

BEGIN
Open report rptOverdueFees ParentContact
Print report to default printer
Close report
END

FIGURE 8.21 This flowchart represents the same steps as the macro design in Figure 8.20.

```
Begin
    Open Form 'frmSwimmersTimes'
    Show on screen
End
```

FIGURE 8.22 Saro's pseudocode design for the macro that opens the swimmer's times form

Development of the DBMS

Once the designs have been finalised and agreed to by the client, the development of the database can take place. The DBMS file is first created and stored; then, each of the key objects inside the database is created, starting with the tables.

Creating tables

It is crucial that the developer constructs the tables properly as all other parts of the database depend on this. They must ensure that the correct names and data types, as well as validation rules, are assigned to each field.

Creating user input forms

Now that tables have been constructed, input forms can be created. Remember that an input form is designed to enable easy data entry. It can also be used to calculate formulas, provide selection lists and other data input control tools. Layout and various formats and conventions to follow were discussed earlier in this chapter.

Setting up the queries

The queries that will select the required subsets of data are constructed once the tables and reports have been finalised. Each database might involve multiple queries with different criteria.

Creating reports

Reports can be generated once the queries have been set up. Each report uses the data selected by its corresponding query. The report turns the data into information by presenting it in a more useful format for the end user.

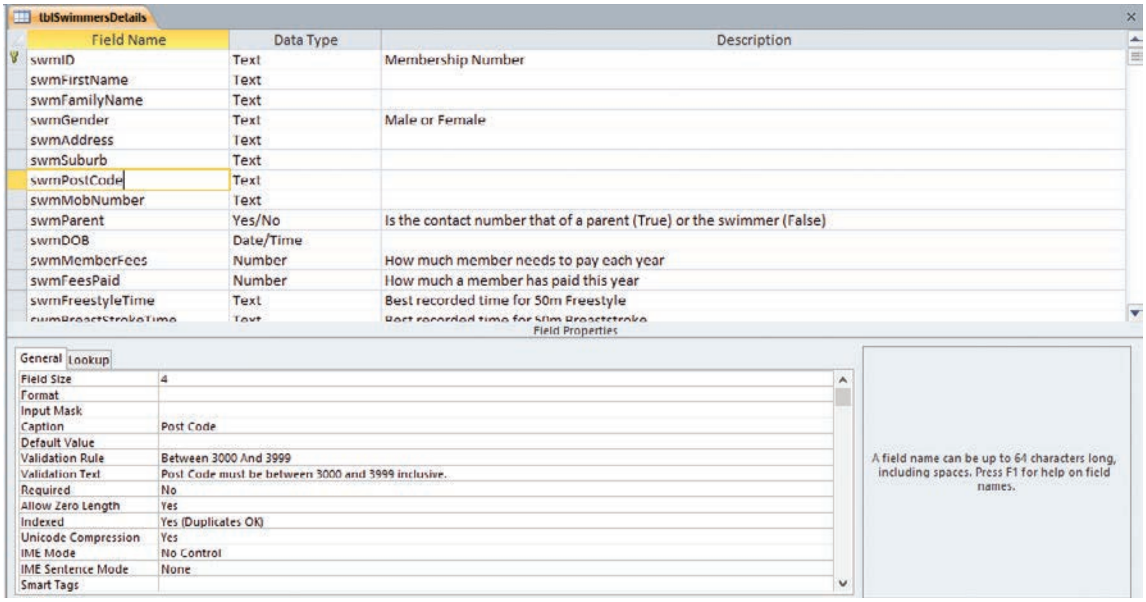
Creating macros

Once the key objects in the database have been constructed, macros can be recorded to automate common tasks. Macros can even be placed on forms to create an interactive menu.

CASE STUDY

KALUGANGA JUNIOR SWIMMING CLUB – PART 7

Saro creates a database file called 'Kaluganga Swimmers Details'. She then constructs the 'tblSwimmersDetails' table, which will hold all of the swimming club data relevant to the swimmers (Figure 8.23). Once happy with the table, Saro constructs the two forms – for individual swimmers' personal details (see Figures 8.25 and 8.26) and for individual swimmers' best times (see Figures 8.30 and 8.31). These are followed by the two queries – regarding parent contact for swimmers with overdue fees (Figures 8.32 and 8.33) and for swimmers' best times for backstroke (see Figures 8.35 and 8.36) – and then the two reports – the overdue fees parent contact report (see Figures 8.39 and 8.40) and the best times for backstroke report (see Figures 8.41 and 8.42). Finally, Saro creates the macros to make it easier and faster to generate the reports (see Figures 8.43 and 8.44). At this stage, Saro is satisfied that the software side of the solution is complete.



Field Name	Data Type	Description
swmID	Text	Membership Number
swmFirstName	Text	
swmFamilyName	Text	
swmGender	Text	Male or Female
swmAddress	Text	
swmSuburb	Text	
swmPostCode	Text	
swmMobNumber	Text	
swmParent	Yes/No	Is the contact number that of a parent (True) or the swimmer (False)
swmDOB	Date/Time	
swmMemberFees	Number	How much member needs to pay each year
swmFeesPaid	Number	How much a member has paid this year
swmFreestyleTime	Text	Best recorded time for 50m Freestyle
swmBreastStrokeTime	Text	Best recorded time for 50m Breaststroke

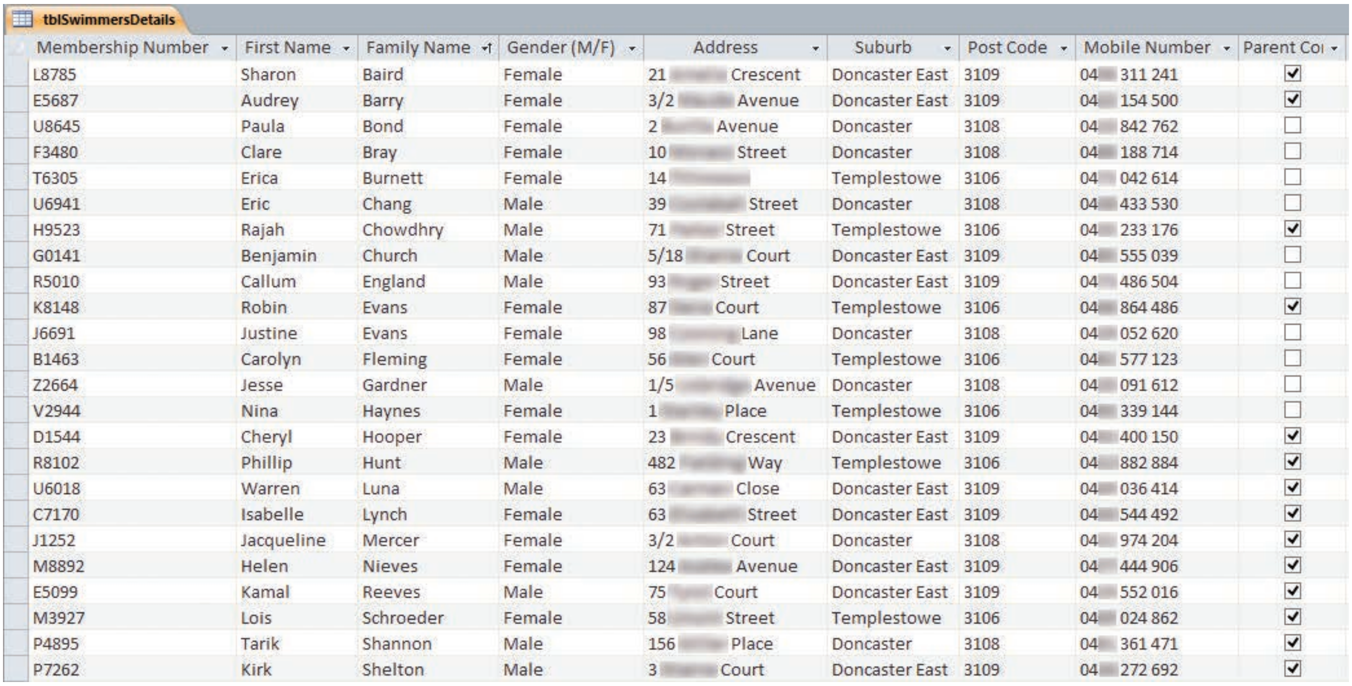
Field Size	4
Format	
Input Mask	
Caption	Post Code
Default Value	
Validation Rule	Between 3000 And 3999
Validation Text	Post Code must be between 3000 and 3999 inclusive.
Required	No
Allow Zero Length	Yes
Indexed	Yes (Duplicates OK)
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

FIGURE 8.23 The `tblSwimmersDetails` table in Design View in Microsoft Access. Note how validation has been set on the postcode field.

Case study continues ▶

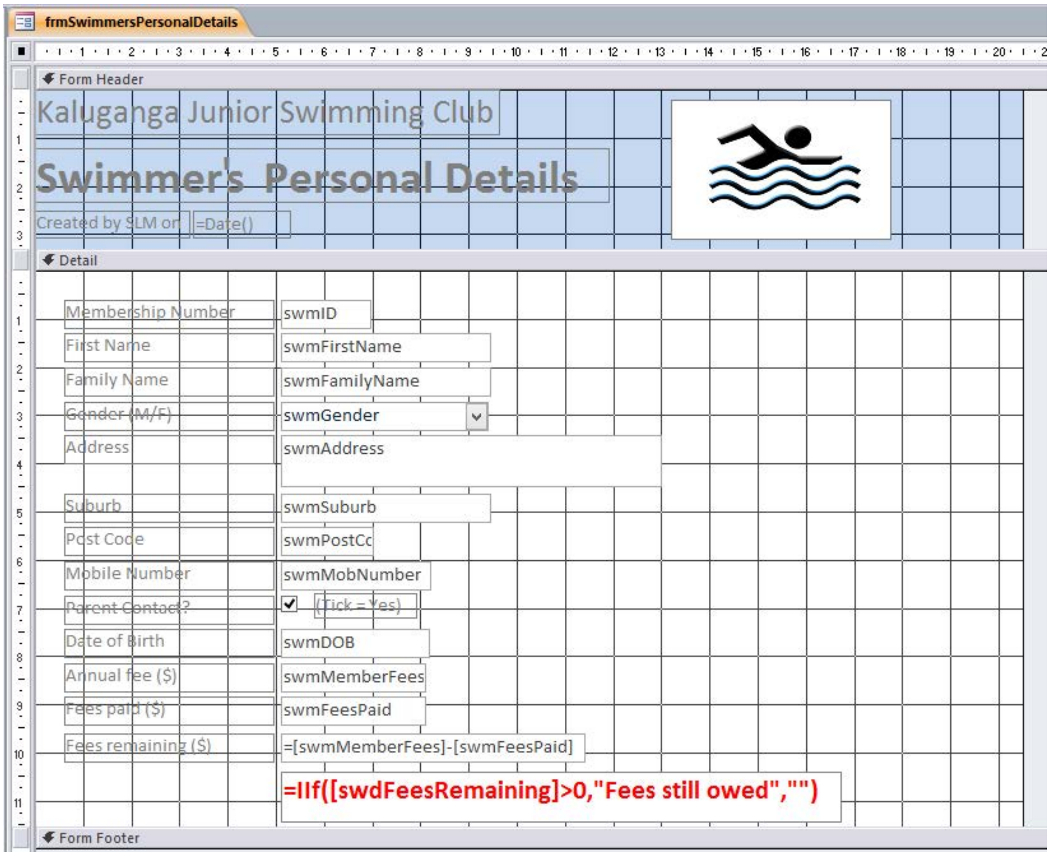
◀ Case study continued



tblSwimmersDetails

Membership Number	First Name	Family Name	Gender (M/F)	Address	Suburb	Post Code	Mobile Number	Parent Co
L8785	Sharon	Baird	Female	21 [redacted] Crescent	Doncaster East	3109	04[redacted] 311 241	☑
E5687	Audrey	Barry	Female	3/2 [redacted] Avenue	Doncaster East	3109	04[redacted] 154 500	☑
U8645	Paula	Bond	Female	2 [redacted] Avenue	Doncaster	3108	04[redacted] 842 762	□
F3480	Clare	Bray	Female	10 [redacted] Street	Doncaster	3108	04[redacted] 188 714	□
T6305	Erica	Burnett	Female	14 [redacted]	Templestowe	3106	04[redacted] 042 614	□
U6941	Eric	Chang	Male	39 [redacted] Street	Doncaster	3108	04[redacted] 433 530	□
H9523	Rajah	Chowdhry	Male	71 [redacted] Street	Templestowe	3106	04[redacted] 233 176	☑
G0141	Benjamin	Church	Male	5/18 [redacted] Court	Doncaster East	3109	04[redacted] 555 039	□
R5010	Callum	England	Male	93 [redacted] Street	Doncaster East	3109	04[redacted] 486 504	□
K8148	Robin	Evans	Female	87 [redacted] Court	Templestowe	3106	04[redacted] 864 486	☑
J6691	Justine	Evans	Female	98 [redacted] Lane	Doncaster	3108	04[redacted] 052 620	□
B1463	Carolyn	Fleming	Female	56 [redacted] Court	Templestowe	3106	04[redacted] 577 123	□
Z2664	Jesse	Gardner	Male	1/5 [redacted] Avenue	Doncaster	3108	04[redacted] 091 612	□
V2944	Nina	Haynes	Female	1 [redacted] Place	Templestowe	3106	04[redacted] 339 144	□
D1544	Cheryl	Hooper	Female	23 [redacted] Crescent	Doncaster East	3109	04[redacted] 400 150	☑
R8102	Phillip	Hunt	Male	482 [redacted] Way	Templestowe	3106	04[redacted] 882 884	☑
U6018	Warren	Luna	Male	63 [redacted] Close	Doncaster East	3109	04[redacted] 036 414	☑
C7170	Isabelle	Lynch	Female	63 [redacted] Street	Doncaster East	3109	04[redacted] 544 492	☑
J1252	Jacqueline	Mercer	Female	3/2 [redacted] Court	Doncaster	3108	04[redacted] 974 204	☑
M8892	Helen	Nieves	Female	124 [redacted] Avenue	Doncaster East	3109	04[redacted] 444 906	☑
E5099	Kamal	Reeves	Male	75 [redacted] Court	Doncaster East	3109	04[redacted] 552 016	☑
M3927	Lois	Schroeder	Female	58 [redacted] Street	Templestowe	3106	04[redacted] 024 862	☑
P4895	Tarik	Shannon	Male	156 [redacted] Place	Doncaster	3108	04[redacted] 361 471	☑
P7262	Kirk	Shelton	Male	3 [redacted] Court	Doncaster East	3109	04[redacted] 272 692	☑

FIGURE 8.24 Part of the swimmers' details table with data entered.



frmSwimmersPersonalDetails

Form Header

Kaluganga Junior Swimming Club

Swimmer's Personal Details

Created by SLM on =Date()

Detail

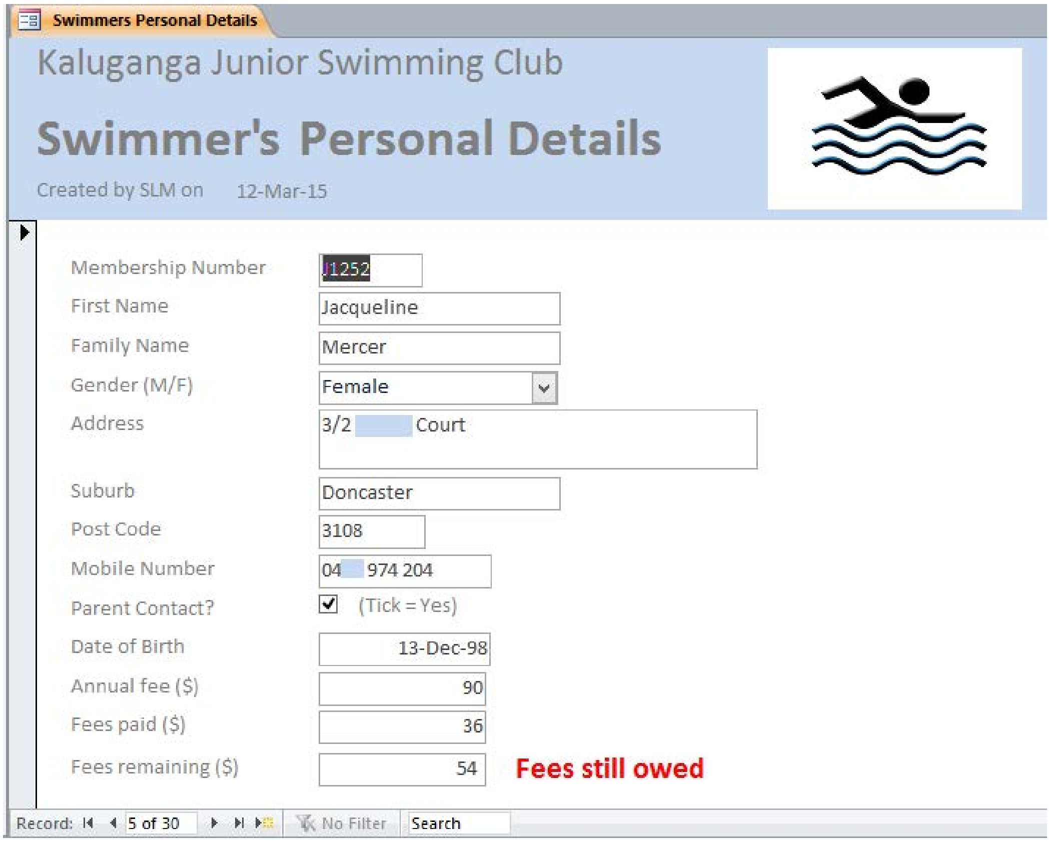
Membership Number	swmID
First Name	swmFirstName
Family Name	swmFamilyName
Gender (M/F)	swmGender
Address	swmAddress
Suburb	swmSuburb
Post Code	swmPostCc
Mobile Number	swmMobNumber
Parent Contact?	☑ (Tick = Yes)
Date of Birth	swmDOB
Annual fee ($)	swmMemberFees
Fees paid ($)	swmFeesPaid
Fees remaining ($)	=[swmMemberFees]-[swmFeesPaid]
	=IIf([swdFeesRemaining]>0,"Fees still owed","")

Form Footer

FIGURE 8.25 The `frmSwimmersPersonalDetails` form in Design View in Access.

Case study continues ▶

◀ *Case study continued*



Kaluganga Junior Swimming Club

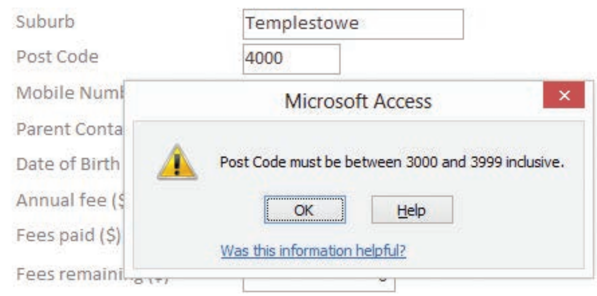
Swimmer's Personal Details

Created by SLM on 12-Mar-15

Membership Number	J1252
First Name	Jacqueline
Family Name	Mercer
Gender (M/F)	Female
Address	3/2 [redacted] Court
Suburb	Doncaster
Post Code	3108
Mobile Number	04 [redacted] 974 204
Parent Contact?	☑ (Tick = Yes)
Date of Birth	13-Dec-98
Annual fee ($)	90
Fees paid ($)	36
Fees remaining ($)	54

Fees still owed

FIGURE 8.26 The swimmer's personal details form as it appears for data entry or editing. Note that because the 'fees remaining' field is greater than 0, the 'Fees still owed' message is displayed to alert the club treasurer.

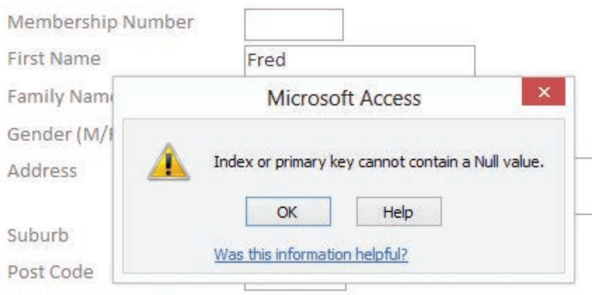


Suburb	Templestowe
Post Code	4000
Mobile Numb	
Parent Conta	
Date of Birth	
Annual fee ($	
Fees paid ($)	
Fees remaini	

Microsoft Access

Post Code must be between 3000 and 3999 inclusive.

FIGURE 8.27 This is the message that appears when a range checking validation rule is activated caused by an incorrect value being entered in the postcode field on the form.

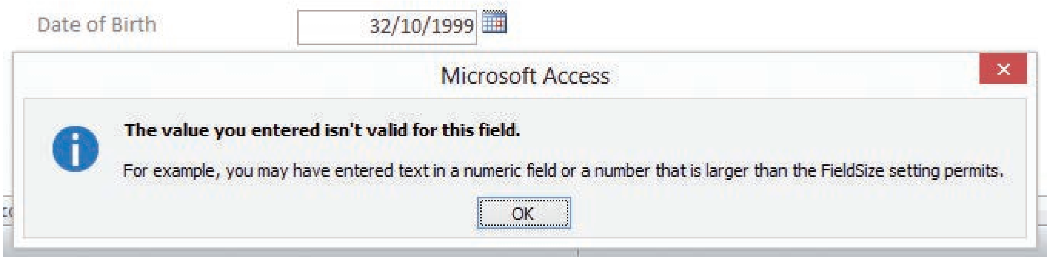


Membership Number	
First Name	Fred
Family Nam	
Gender (M/I	
Address	
Suburb	
Post Code	

Microsoft Access

Index or primary key cannot contain a Null value.

FIGURE 8.28 This is an alert for an existence check on the membership number field. The membership number is the primary key so a value must exist for each record.



Date of Birth 32/10/1999

Microsoft Access

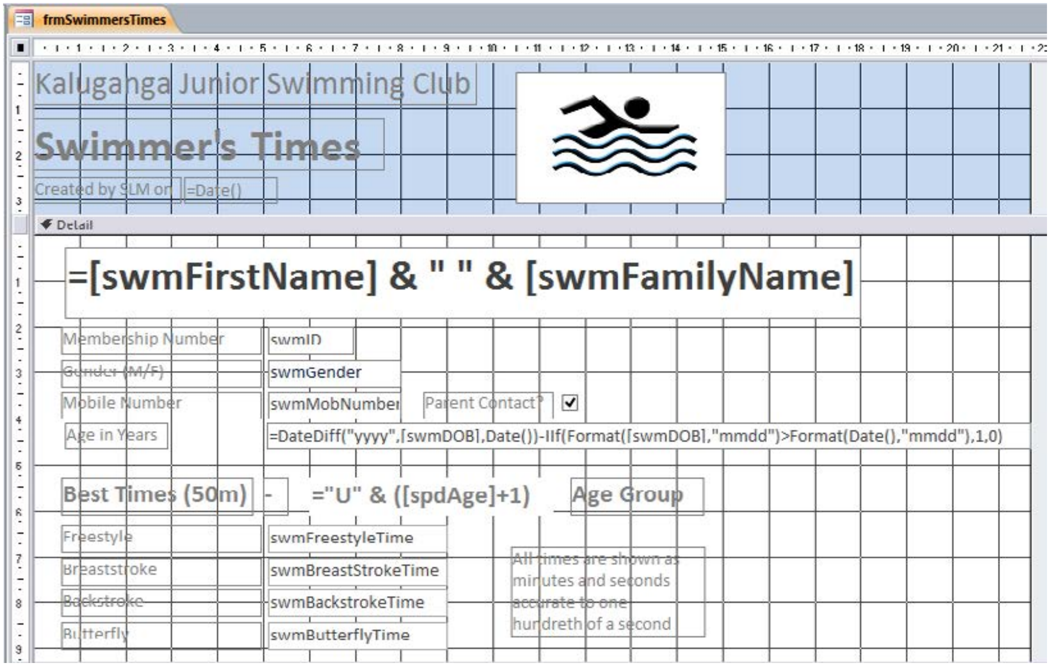
The value you entered isn't valid for this field.

For example, you may have entered text in a numeric field or a number that is larger than the FieldSize setting permits.

FIGURE 8.29 A data-type check alert occurs when an invalid date is entered in the date-of-birth field. A similar message will also appear when a non-numeric value is entered into the fees-paid field.

Case study continues ▶

◀ Case study continued



frmSwimmersTimes

Kaluganga Junior Swimming Club

Swimmer's Times

Created by SLM on =Date()

Detail

=[swmFirstName] & " " & [swmFamilyName]

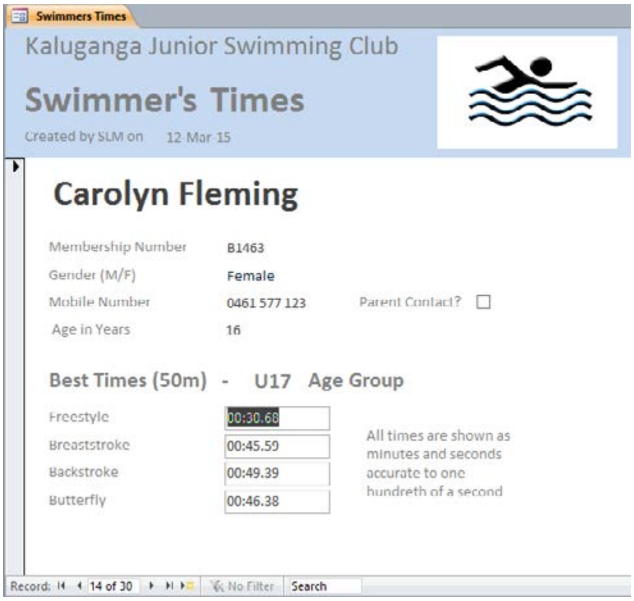
Membership Number	swmID		
Gender (M/F)	swmGender		
Mobile Number	swmMobNumber	Parent Contact?	☑
Age in Years	=DateDiff("yyyy",[swmDOB],Date())-IIf(Format([swmDOB],"mmdd")>Format(Date(),"mmdd"),1,0)		

Best Times (50m) - ="U" & ([spdAge]+1) Age Group

Freestyle	swmFreestyleTime
Breaststroke	swmBreastStrokeTime
Backstroke	swmBackstrokeTime
Butterfly	swmButterflyTime

All times are shown as minutes and seconds accurate to one hundreth of a second

FIGURE 8.30 The `frmSwimmersTimes` form in Design View in Access. The fields have been expanded so the various formulas can be seen.



Swimmers Times

Kaluganga Junior Swimming Club

Swimmer's Times

Created by SLM on 12-Mar-15

Carolyn Fleming

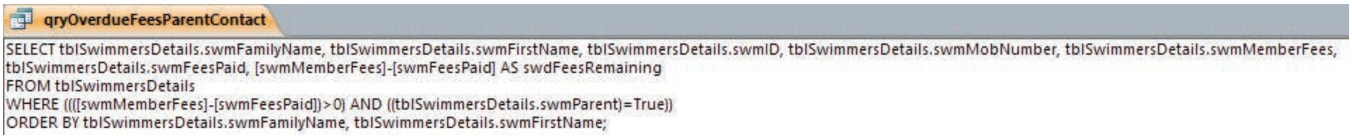
Membership Number	B1463		
Gender (M/F)	Female		
Mobile Number	0461 577 123	Parent Contact?	☐
Age in Years	16		

Best Times (50m) - U17 Age Group

Freestyle	00:30.68
Breaststroke	00:45.59
Backstroke	00:49.39
Butterfly	00:46.38

All times are shown as minutes and seconds accurate to one hundreth of a second

FIGURE 8.31 The swimmer's times form to be used by the coaches for updating details for best times in each swimming event. Personal details cannot be changed on this form, only the times.



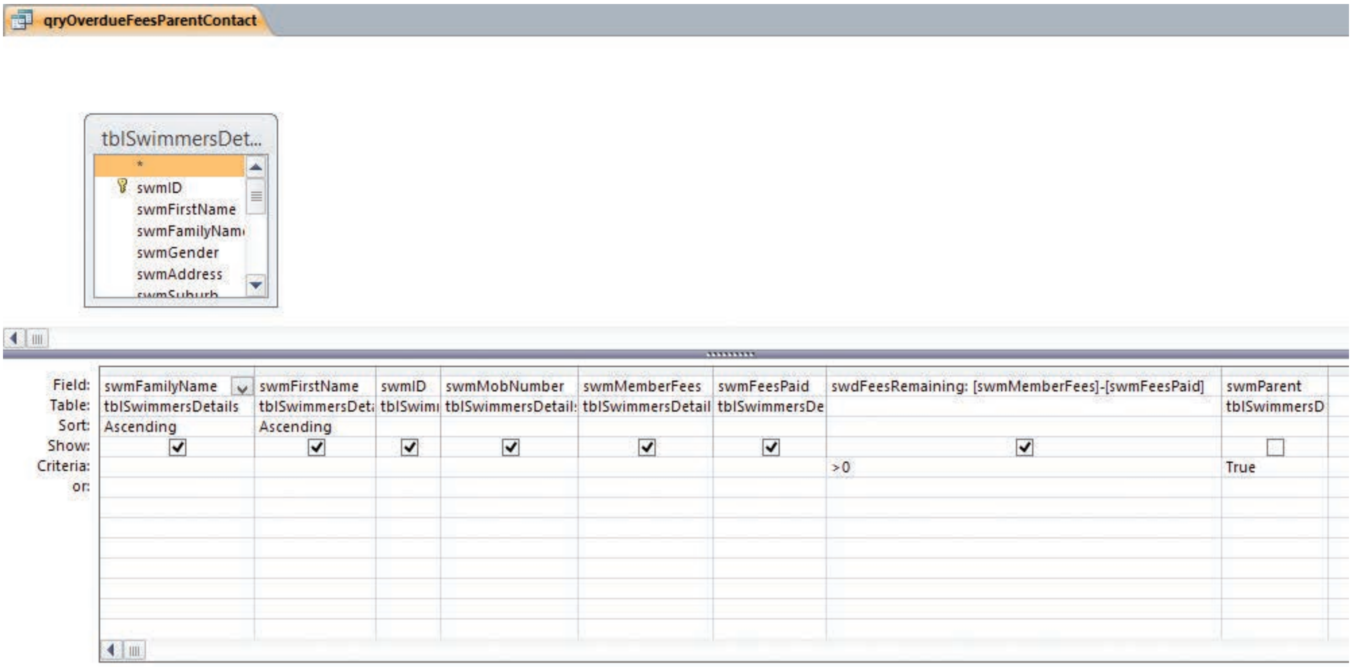
qryOverdueFeesParentContact

```
SELECT tblSwimmersDetails.swmFamilyName, tblSwimmersDetails.swmFirstName, tblSwimmersDetails.swmID, tblSwimmersDetails.swmMobNumber, tblSwimmersDetails.swmMemberFees,
tblSwimmersDetails.swmFeesPaid, [swmMemberFees]-[swmFeesPaid] AS swdFeesRemaining
FROM tblSwimmersDetails
WHERE ((([swmMemberFees]-[swmFeesPaid])>0) AND ((tblSwimmersDetails.swmParent)=True))
ORDER BY tblSwimmersDetails.swmFamilyName, tblSwimmersDetails.swmFirstName;
```

FIGURE 8.32 The `qryOverudueFeesParentContact` query, displayed as an SQL statement

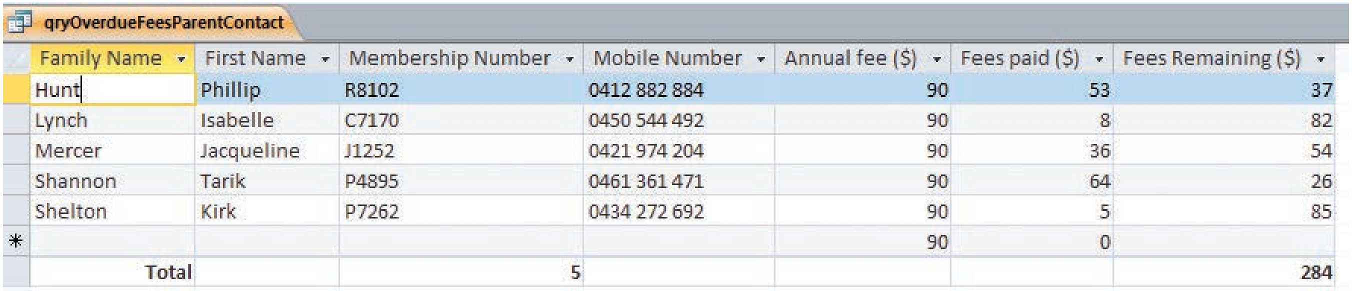
Case study continues ▶

Case study continued



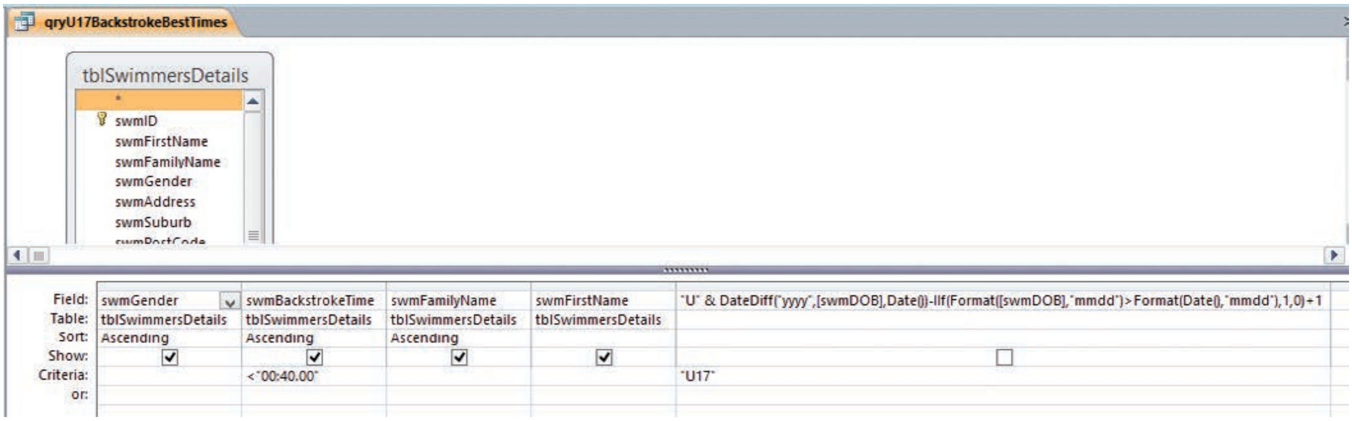
Field:	swmFamilyName	swmFirstName	swmID	swmMobNumber	swmMemberFees	swmFeesPaid	swdFeesRemaining: [swmMemberFees]-[swmFeesPaid]	swmParent
Table:	tblSwimmersDetails	tblSwimmersDet	tblSwimi	tblSwimmersDetail:	tblSwimmersDetail	tblSwimmersDe		tblSwimmersD
Sort:	Ascending	Ascending						
Show:	☑	☑	☑	☑	☑	☑	☑	□
Criteria:							>0	True
or:								

FIGURE 8.33 The Design View in Access for the `qryOverdueFeesParentContact` query. The widths have been changed on some of the fields so that the formula calculating the fees remaining to be paid can be seen.



	Family Name	First Name	Membership Number	Mobile Number	Annual fee ($)	Fees paid ($)	Fees Remaining ($)
	Hunt	Phillip	R8102	0412 882 884	90	53	37
	Lynch	Isabelle	C7170	0450 544 492	90	8	82
	Mercer	Jacqueline	J1252	0421 974 204	90	36	54
	Shannon	Tarik	P4895	0461 361 471	90	64	26
	Shelton	Kirk	P7262	0434 272 692	90	5	85
*					90	0	
	Total		5				284

FIGURE 8.34 When run, the overdue fees query selects only those swimmers with outstanding membership fees and whose contact number belongs to their parents. Only fields relevant to the treasurer's needs are used so that only the required data is displayed.

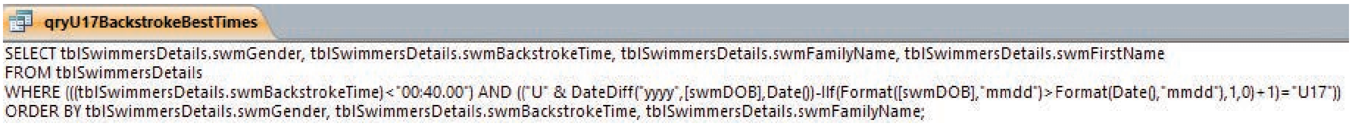


Field:	swmGender	swmBackstrokeTime	swmFamilyName	swmFirstName	"U" & DateDiff("yyyy",[swmDOB],Date())-IIf(Format([swmDOB],"mmdd")>Format(Date(),"mmdd"),1,0)+1
Table:	tblSwimmersDetails	tblSwimmersDetails	tblSwimmersDetails	tblSwimmersDetails	
Sort:	Ascending	Ascending	Ascending		
Show:	☑	☑	☑	☑	□
Criteria:		<"00:40.00"			"U17"
or:					

FIGURE 8.35 The Access Design View of the `qryU17BackstrokeBestTimes` query. It selects swimmers with a backstroke time of less than 40 seconds and whose age group is under 17. The age-group field does not need to be shown when the query is run because all records listed will have the same age group, which makes this data redundant.

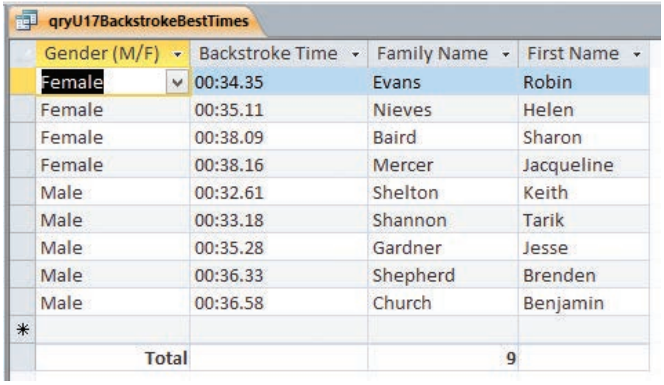
Case study continues

◀ *Case study continued*



```
SELECT tblSwimmersDetails.swmGender, tblSwimmersDetails.swmBackstrokeTime, tblSwimmersDetails.swmFamilyName, tblSwimmersDetails.swmFirstName
FROM tblSwimmersDetails
WHERE (((tblSwimmersDetails.swmBackstrokeTime)<"00:40.00") AND (("U" & DateDiff("yyyy",[swmDOB],Date())-IIf(Format([swmDOB],"mmdd")>Format(Date(),"mmdd"),1,0)+1)="U17"))
ORDER BY tblSwimmersDetails.swmGender, tblSwimmersDetails.swmBackstrokeTime, tblSwimmersDetails.swmFamilyName;
```

FIGURE 8.36 The under-17 backstroke swimmers query in SQL view.



Gender (M/F)	Backstroke Time	Family Name	First Name
Female	00:34.35	Evans	Robin
Female	00:35.11	Nieves	Helen
Female	00:38.09	Baird	Sharon
Female	00:38.16	Mercer	Jacqueline
Male	00:32.61	Shelton	Keith
Male	00:33.18	Shannon	Tarik
Male	00:35.28	Gardner	Jesse
Male	00:36.33	Shepherd	Brenden
Male	00:36.58	Church	Benjamin
*			
Total		9	

FIGURE 8.37 The query selects the best under-17 backstroke swimmers and is sorted by gender, then by time with the fastest at the top (ascending). It only selects the data the coaches need.

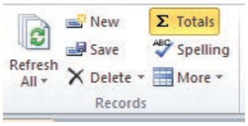


FIGURE 8.38 In Access, simple totals like sums and counts can be shown in queries by choosing the totals option.

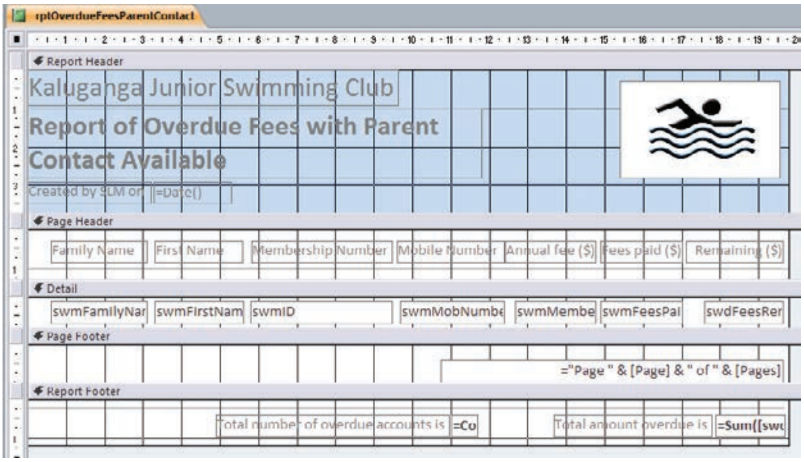
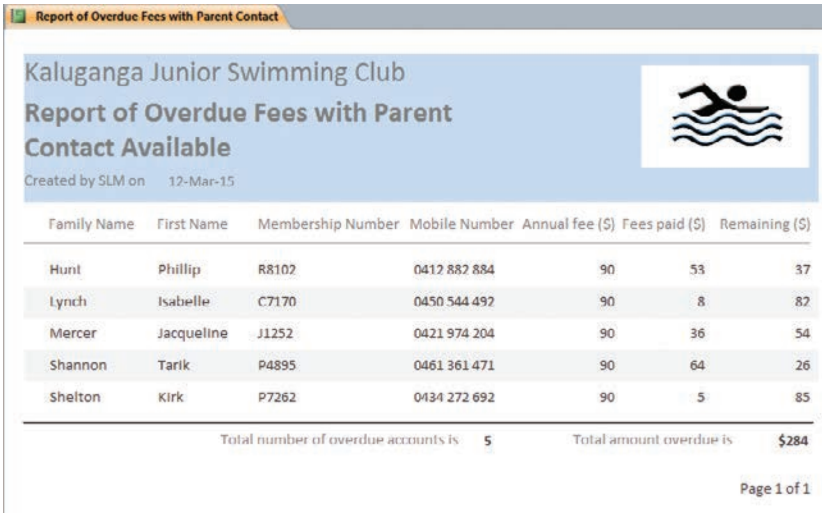


FIGURE 8.39 This is the design of the `rptOverdueFeesParentContact` report in Access. A Count and a Sum function are both used at the end of the report to provide summary statistics. These increase readability and communication of the message.



Kaluganga Junior Swimming Club

Report of Overdue Fees with Parent Contact Available

Created by SLM on 12-Mar-15

Family Name	First Name	Membership Number	Mobile Number	Annual fee ($)	Fees paid ($)	Remaining ($)
Hunt	Phillip	R8102	0412 882 884	90	53	37
Lynch	Isabelle	C7170	0450 544 492	90	8	82
Mercer	Jacqueline	J1252	0421 974 204	90	36	54
Shannon	Tarik	P4895	0461 361 471	90	64	26
Shelton	Kirk	P7262	0434 272 692	90	5	85

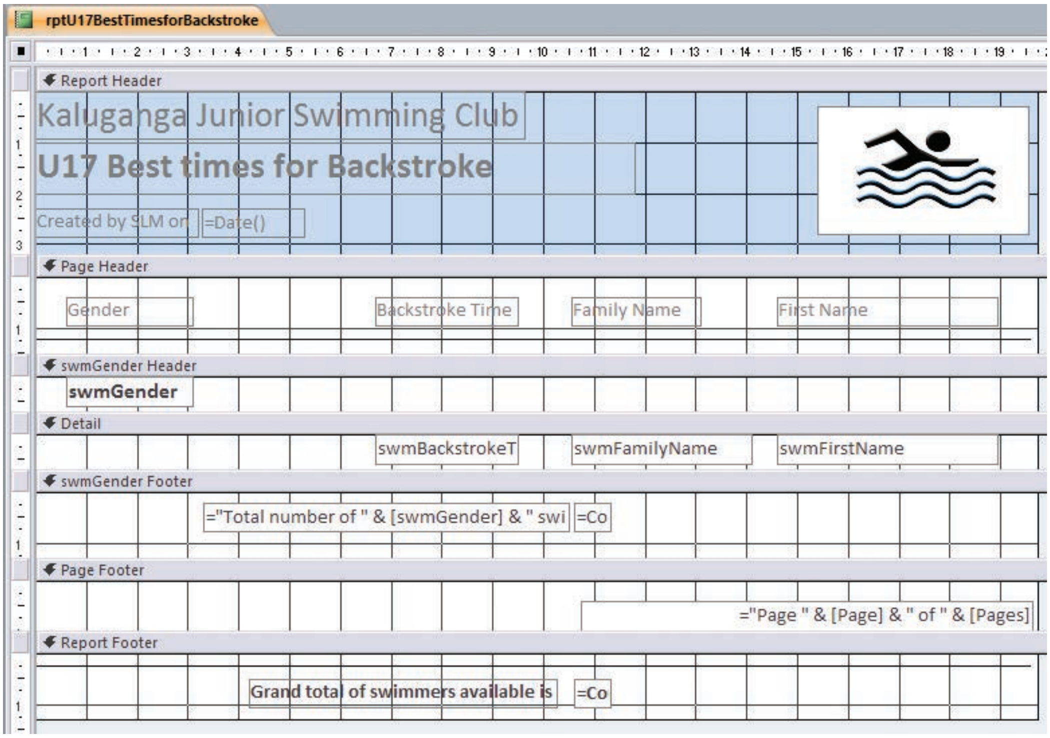
Total number of overdue accounts is 5 Total amount overdue is $284

Page 1 of 1

FIGURE 8.40 The final version of the overdue fees report for the club treasurer. It includes a summary total of the number of records listed as well as the total amount owed to the club. It is designed to be printed on an A4-size portrait page.

Case study continues ▶

◀ *Case study continued*



rptU17BestTimesforBackstroke

Report Header

Kaluganga Junior Swimming Club

U17 Best times for Backstroke

Created by SLM on =Date()

Page Header

Gender Backstroke Time Family Name First Name

swmGender Header

swmGender

Detail

swmBackstrokeT swmFamilyName swmFirstName

swmGender Footer

="Total number of " & [swmGender] & " swi =Co

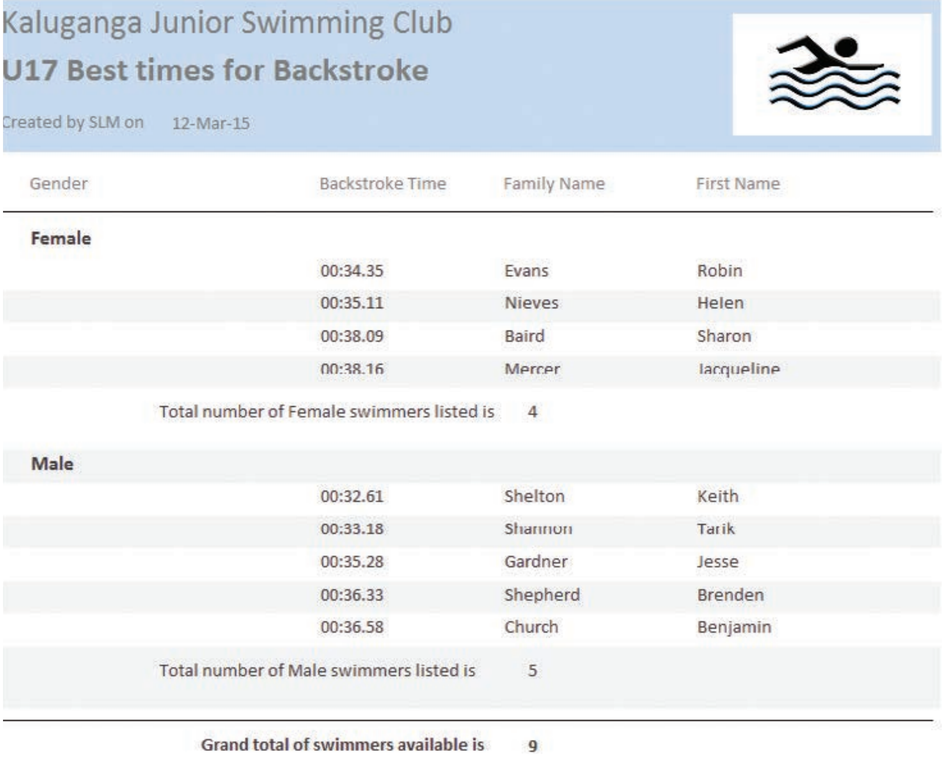
Page Footer

="Page " & [Page] & " of " & [Pages]

Report Footer

Grand total of swimmers available is =Co

FIGURE 8.41 The `rptU17BestTimesforBackstroke` report in Access Design View. Note the header and footer sections for the gender field because it is being used to group data.



Kaluganga Junior Swimming Club

U17 Best times for Backstroke

Created by SLM on 12-Mar-15

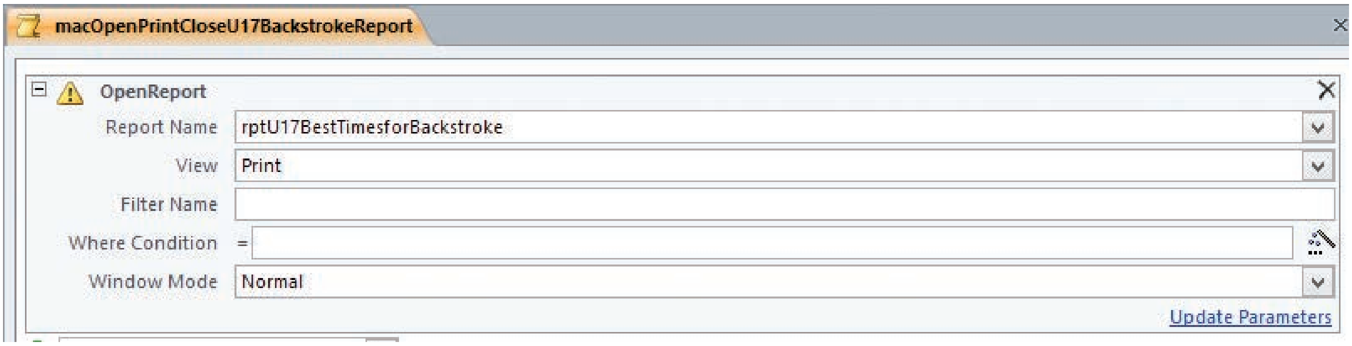
Gender	Backstroke Time	Family Name	First Name
Female			
	00:34.35	Evans	Robin
	00:35.11	Nieves	Helen
	00:38.09	Baird	Sharon
	00:38.16	Mercer	Jacqueline
Total number of Female swimmers listed is	4		
Male			
	00:32.61	Shelton	Keith
	00:33.18	Shannon	Tarik
	00:35.28	Gardner	Jesse
	00:36.33	Shepherd	Brenden
	00:36.58	Church	Benjamin
Total number of Male swimmers listed is	5		

Grand total of swimmers available is 9

FIGURE 8.42 This is the grouped list of the best under-17 backstroke swimmers sorted by time for the coaches. It includes a subtotal for each gender group as well as an overall total of swimmers who fit the selection criteria. This report is designed to be printed on an A4 portrait page.

Case study continues ▶

◀ *Case study continued*

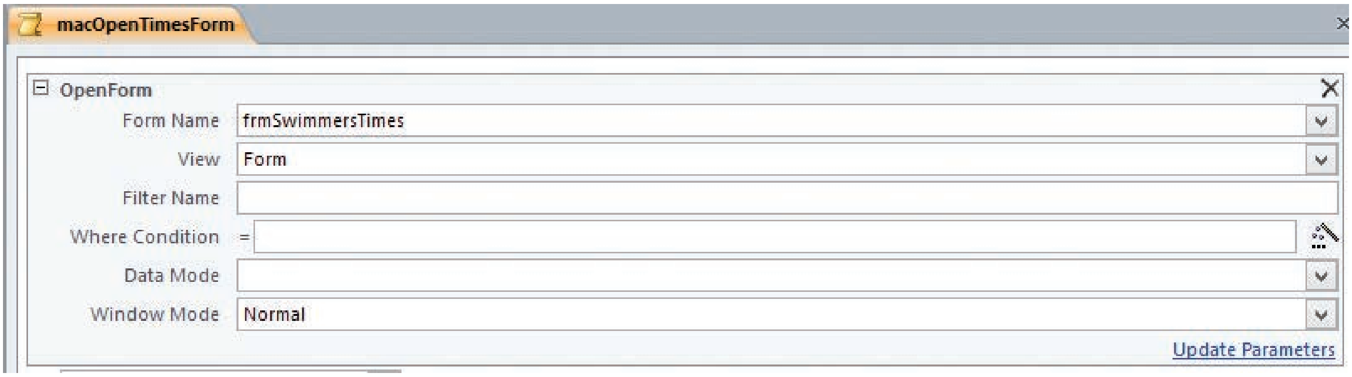


macOpenPrintCloseU17BacksstrokeReport

OpenReport

Report Name	rptU17BestTimesforBackstroke
View	Print
Filter Name	
Where Condition	=
Window Mode	Normal

FIGURE 8.43 The Design View for the macro that opens the under-17 backstroke swimmers report, prints it, then closes it.

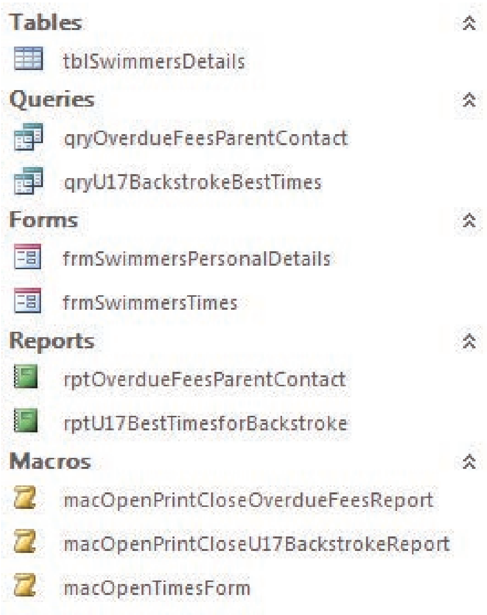


macOpenTimesForm

OpenForm

Form Name	frmSwimmersTimes
View	Form
Filter Name	
Where Condition	=
Data Mode	
Window Mode	Normal

FIGURE 8.44 This is the Design View for the macro that opens the Swimmer's Details form when run.



Tables
- tblSwimmersDetails

Queries
- qryOverdueFeesParentContact
- qryU17BackstrokeBestTimes

Forms
- frmSwimmersPersonalDetails
- frmSwimmersTimes

Reports
- rptOverdueFeesParentContact
- rptU17BestTimesforBackstroke

Macros
- macOpenPrintCloseOverdueFeesReport
- macOpenPrintCloseU17BackstrokeReport
- macOpenTimesForm

FIGURE 8.45 This is a listing of all the database objects Saro has created and their names.

Roles, functions and characteristics of hardware components

In Chapter 5, you were introduced to hardware used for input, storage, communication and output. In this section, we explore common hardware devices that perform these actions in the context of databases. You may recall that each piece of equipment needs to be identified by its role, functions and its characteristics. It is important that you are able to advise a client, or confirm for yourself if you are creating the database for your own needs, of suitable hardware that they may need to operate the database solution you create for them.

Input devices

The keyboard is a frequently used hardware device for typing data directly into fields. This may be users filling in a form on a webpage or staff transcribing data from a printed form completed by hand. A mouse is also used to interact with dropdown lists or as a convenient way to activate tick boxes or similar controls. But these devices are inefficient when a database requires a high volume of data to be entered rapidly from a non-electronic source. For example, imagine how much longer shopping would take if a person at the checkout had to manually enter the barcode of every product purchased!

To improve the speed of data entry, a range of scanners and reading devices exist. These include optical scanners and readers, barcode readers, RFID readers, magnetic stripe card readers, MICR readers and various electronic sensors. Often these input devices are grouped into handheld data-collection devices so that a user can input data at the point where the item or person is located. They will often communicate wirelessly with the database. Many waiters may use such a device when they take your order in a restaurant. You may also see them in use in large stores and supermarkets when managers walk along beside the shelves noting which items need to be restocked. In some cases, you may even have been asked to sign for a parcel on such a device.

TABLE 8.7 Input devices

Device	Role	Function	Characteristics
Keyboard	Manual input of data	Users press keys to input data	From 101–5 keys Assignable special function keys Can handle rapid input of typed data
Mouse		Users press buttons to select data to input from premade lists	2–3 buttons Slow data speed
Optical scanner	Automatic data acquisition and input via optical reader or radio signal.	Source documents or devices either pass in front of, or nearby, the reader.	Can handle high volume of data input Source documents are brought to reading device Immediate input High level of accuracy Often wireless
Optical reader			
Barcode reader			
RFID reader			
Magnetic stripe card reader			
MICR reader			
Sensor			
Data-collection device	Manual or electronic input of data	Device can go to the location of data being input and use one of several input devices to send the data to the main database wirelessly.	Combination of different acquisition and input devices Immediate input Wireless Portable (hand-held)

FIGURE 8.46 A barcode reader

Shutterstock.com/Ruslan Kuzmenkov

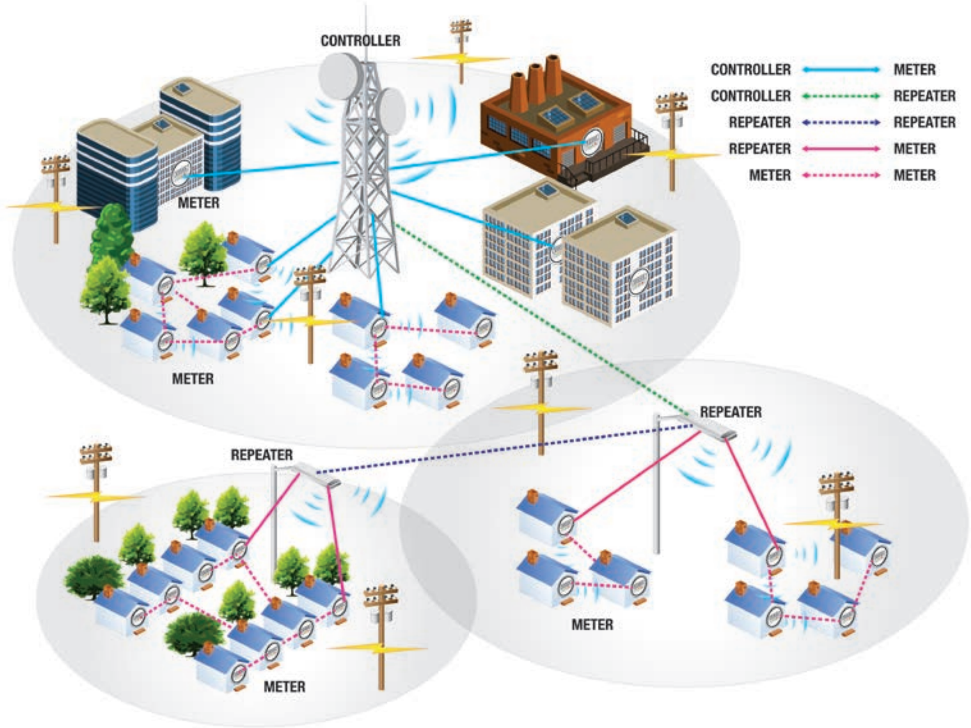


FIGURE 8.48 This diagram represents a mesh network used to collect electricity usage data via Smart Meters. AMI stands for Advanced Metering Infrastructure.

courtesy of EDX Wireless

FIGURE 8.47 A data collection device that reads RFIDs and can communicate via wi-fi, Bluetooth and mobile phone network.

Shutterstock.com/Voznikevich Konstantin

Storage devices

You will remember that storage is the permanent saving of data for later retrieval. Storage hardware that is suitable for databases is the same as for any electronic files. These include internal or external hard disk and solid state drives, flash memory drives such as SD cards and USB flash drives. A more permanent backup copy of a database may be saved onto optical media like DVD or magnetic tape. Some databases may be stored in the cloud, but this is usually a massive data store with many hard drives connected using RAID (redundant array of inexpensive disks) to minimise data loss if there is some kind of threat, whether accidental or deliberate.

FIGURE 8.49 Inside a typical cloud-storage data centre

Getty Images/Erik Isakson

Communication devices

Databases can be transferred across and accessed over almost any networking device. You will recall from Chapter 2 that common networking hardware consists of network cards, wireless access points, routers, switches and cables. Data to be input might be transferred across any number of these devices if it is being entered remotely across the internet or an organisation's own network.

An important consideration is to ensure that any data is protected from interception and viewing by unauthorised users by using encryption.

Encryption is further discussed on page 334.

Output devices

Output is data that has been manipulated into a useful and meaningful form. It is also referred to as information. Information generated from a database is in the form of a report. The two most relevant output devices for information contained in a database would be via a screen (softcopy) or a printer (hardcopy). Information displayed on a screen is referred to as a softcopy because it is temporary and only exists electronically. The layout of the information is designed to suit the particular display device. For example, the format for Google search results will look different on a mobile device when compared with a large desktop screen.

In contrast, printed material is more permanent and exists physically, which is why it is called hardcopy. We might commonly also refer to hardcopy as a printout. The orientation of the hardcopy depends on the media being used and the layout of the information. Where possible, reports should be printed in a portrait orientation as this is generally easier to read.

Device	Role	Function	Characteristics
Screen (softcopy)	Displays information so it can be read and interpreted	Information is displayed on an electronic device	Temporary Only electronic
Print (hardcopy)		Information is printed onto physical media like paper.	More permanent Exists physically Can use a range of media

CASE STUDY

KALUGANGA JUNIOR SWIMMING CLUB – PART 8

Saro has investigated the hardware components that she thinks will be required to successfully allow input, output, communication and storage of the swim club's database.

For input she suggests that a mouse and keyboard will be adequate for entering swimmers' details and for updating financial information. The coaches will use a tablet with data collection facility and touchscreen so that times for events can be entered immediately. The tablets will communicate wirelessly with the database. The database itself will be stored on a hard drive on the club's server, but will also have copies saved in real time to a cloud-based storage service. Felix and the treasurer will be able to view swimmer details onscreen, although they can also print the report of overdue fees if required. The coaches can also view the list of swimmers by event and time on the tablets, but they are also able to receive a printed version of the report for when it is impractical for them to use the tablets.

Accidental and deliberate security threats

Once our database is constructed and in use, it is important that it be safeguarded. If the data is altered or damaged by someone who is not authorised to do so, it can easily be compromised. This will lead to inaccurate information being produced for users. Most of the key types of threats to data were explained in Chapter 2. The table below summarises these threats.

TABLE 8.8 Types of threats

Category of threat	Types of threat	Explanation
Accidental	Data altered by accident	An inexperienced or careless user may change or delete data. This creates inaccuracies.
	Stored files are lost	Database objects or data are stored on a portable device, like USB flash memory, and this device is misplaced.
	Not deleting data from an old storage device	A storage device that may have stored database records may be replaced by a newer device, but if the files are not removed then they could be recovered and accessed by whoever the storage device is passed on to.
Deliberate	Hackers	Some people may target a database in order to alter or delete its contents. A common method of attacking a database is by using SQL Injection.
	Malware	These types of malicious software may be designed to automatically attack a specific database, like your email contact list, or may damage database files as a by-product of targeting any files on storage devices
Event based	Power related	In these event-based threats, the results can be the same. Data may be lost or corrupted or the entire file might be destroyed.
	Crash or failure of a storage device	
	Remote connection fails	If a transaction, say from a web-based form, is interrupted, it may result in an incomplete record being saved.

SQL injection is a way of attacking databases via their input forms. Malicious SQL statements are inserted in data-entry fields in order to target and damage underlying data tables or steal their contents. Proper validation on any input box can prevent this form of attack on data.

CASE STUDY

KALUGANGA JUNIOR SWIMMING CLUB – PART 9

Saro has considered the likely threats the swim club database may face. She has assessed that the overall threat level from hackers is fairly low, but there is a larger possibility of data loss occurring by accident, as Felix is less familiar with how to operate a database package. She also thinks it is important to consider the possibility of malware infecting the server. The other key threats are the possibility of a blackout or the failure of the storage device, which could lead to lost data.

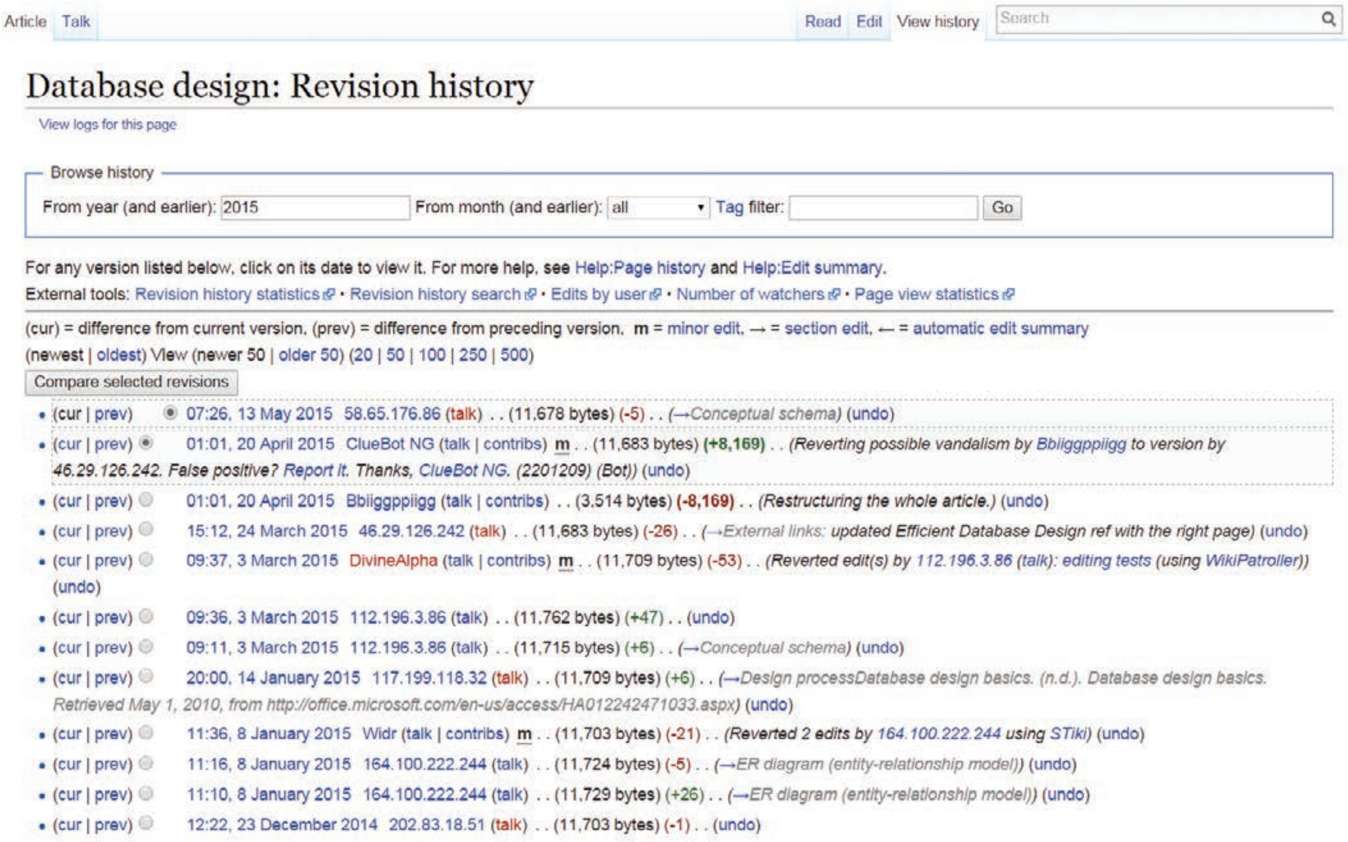
Physical and software controls for protecting security

All data in a database is valuable – we need to protect the integrity of the data so it can produce the most accurate information. A database may hold confidential data that must only be accessed by authorised users. Most government databases fall into this category, but this is crucial for businesses too as they store confidential information such as business contacts or customer details.

In Chapter 1, you were introduced to several physical and software-based controls that can protect data both when stored and transmitted. These measures help to reduce or eliminate the possible damage caused by the threats discussed above. These measures include backing up, electrical protection, usernames and passwords, using system security software and encryption.

Backing up

Databases are generally 'live', but they need the ability to have data 'rolled back' to an earlier time in case of data corruption or malicious activity. Full copies of databases can be made onto any of the storage devices listed above. This means the entire database can be restored to a previous point in time if necessary. Alternatively, many databases support version control, meaning they actually store changes to each data element in any record so that, if necessary, only that particular field is restored. A wiki site like Wikipedia is a good example of an implementation of this type of database 'protection'.



Database design: Revision history

View logs for this page

Browse history

From year (and earlier): 2015 From month (and earlier): all Tag filter: Go

For any version listed below, click on its date to view it. For more help, see Help:Page history and Help:Edit summary.

External tools: Revision history statistics • Revision history search • Edits by user • Number of watchers • Page view statistics

(cur) = difference from current version, (prev) = difference from preceding version, **m** = minor edit, → = section edit, ← = automatic edit summary

(newest | oldest) View (newer 50 | older 50) (20 | 50 | 100 | 250 | 500)

Compare selected revisions

- (cur | prev) ● 07:26, 13 May 2015 58.65.176.86 (talk) . . (11,678 bytes) (-5) . . *(→Conceptual schema)* (undo)
- (cur | prev) ● 01:01, 20 April 2015 ClueBot NG (talk | contribs) **m** . . (11,683 bytes) **(+8,169)** . . *(Reverting possible vandalism by Bbiiggppiigg to version by 46.29.126.242. False positive? Report it. Thanks, ClueBot NG. (2201209) (Bot))* (undo)
- (cur | prev) ○ 01:01, 20 April 2015 Bbiiggppiigg (talk | contribs) . . (3,514 bytes) **(-8,169)** . . *(Restructuring the whole article.)* (undo)
- (cur | prev) ○ 15:12, 24 March 2015 46.29.126.242 (talk) . . (11,683 bytes) (-26) . . *(→External links: updated Efficient Database Design ref with the right page)* (undo)
- (cur | prev) ○ 09:37, 3 March 2015 DivineAlpha (talk | contribs) **m** . . (11,709 bytes) (-53) . . *(Reverted edit(s) by 112.196.3.86 (talk): editing tests (using WikiPatroller))* (undo)
- (cur | prev) ○ 09:36, 3 March 2015 112.196.3.86 (talk) . . (11,762 bytes) (+47) . . (undo)
- (cur | prev) ○ 09:11, 3 March 2015 112.196.3.86 (talk) . . (11,715 bytes) (+6) . . *(→Conceptual schema)* (undo)
- (cur | prev) ○ 20:00, 14 January 2015 117.199.118.32 (talk) . . (11,709 bytes) (+6) . . *(→Design processDatabase design basics. (n.d.). Database design basics. Retrieved May 1, 2010, from http://office.microsoft.com/en-us/access/HA012242471033.aspx)* (undo)
- (cur | prev) ○ 11:36, 8 January 2015 Widr (talk | contribs) **m** . . (11,703 bytes) (-21) . . *(Reverted 2 edits by 164.100.222.244 using STiki)* (undo)
- (cur | prev) ○ 11:16, 8 January 2015 164.100.222.244 (talk) . . (11,724 bytes) (-5) . . *(→ER diagram (entity-relationship model))* (undo)
- (cur | prev) ○ 11:10, 8 January 2015 164.100.222.244 (talk) . . (11,729 bytes) (+26) . . *(→ER diagram (entity-relationship model))* (undo)
- (cur | prev) ○ 12:22, 23 December 2014 202.83.18.51 (talk) . . (11,703 bytes) (-1) . . (undo)

https://en.wikipedia.org/w/index.php?title=Database_design&action=history

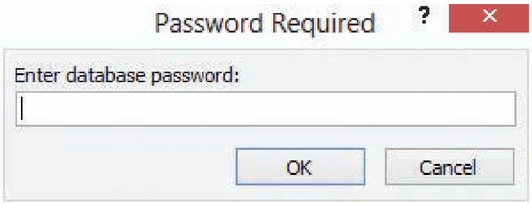
FIGURE 8.50 Wikipedia stores backup copies of articles and changes made to them. This allows editorial staff to roll back to a previous version of an article without having to rewrite it.

Electrical protection

A common way of protecting against power events is to use a well regulated power supply that can intercept power surges or spikes. But a database may need to remain accessible even under adverse power conditions. For this reason, backup power supplies and batteries are often used. An Uninterruptable Power Supply or UPS can regulate electricity coming into a computer and supply backup electricity for a short time if a blackout occurs.

Usernames and passwords

Usernames and passwords can be used to restrict access to the entire database, so that only authorised users can open it. Alternatively, usernames and passwords allow certain users to edit records, while others can only read, but not alter them. These types of databases will also make a record of who changes what data as an auditing feature.



Enter database password:

FIGURE 8.51 A password can be used in Access to prevent unauthorised users from opening a database file.

Systems security software

Malware is discussed in Chapter 3.

It is important to use a variety of systems security software to prevent malware or other malicious coding from running on your system and either damaging or destroying the contents of a database.

Encryption (storage and transmitted)

Encryption is used to prevent interception and theft. It involves coding data into a form that only authorised users can read and decrypt. The encryption and decryption processes both require a 'key'.

If data is entered via a website, then it will use SSL or TLS over an HTTPS connection. When SSL or TLS protocols are used over HTTPS (on port 443), an encrypted and secure 'pipeline' is created between the user and the database so that if the data being transmitted is intercepted, it cannot be read without the key. You will often notice that online application forms, registration forms and order forms use this type of encryption, as do internet transactions involving credit card details.

When the plain text is encrypted, it is called 'ciphertext'.

When stored, databases can also be encrypted. The contents can be encrypted inside fields. This is known as '**hashing**' and is often used to protect passwords. Hashing passwords is a way of taking a variable-length password and creating a cryptic, fixed-length password from it. You do this by generating and using a **salt value**. A salt value is a random value that you use to generate the hashed password. If the password field is viewed by a user, they cannot decrypt the hashed password.

The entire database file can also be encrypted so that if stolen, its contents cannot be opened and used by another person.

Encryption involves coding data into a form that only authorised users can read and decrypt. The encryption and decryption process both require a 'key'. When the plain text is encrypted, it is called 'ciphertext'. When SSL or TLS protocols are used over HTTPS (on port 443), an encrypted and secure 'pipeline' is created between the user and the database so that if the data being transmitted is intercepted, it cannot be read without the key. This system is commonly used for internet transactions involving registration or order forms, and for credit card details.

CASE STUDY

KALUGANGA JUNIOR SWIMMING CLUB – PART 10

For the Kaluganga database, Saro has limited access to personal and financial details to authorised users by placing a password and username on the files. The coaches can view basic details about each swimmer but can only make changes to their times for events. Any changes to data are logged. Anti-malware software will be installed on the tablets, PC and server while an encrypted connection will be used to transmit data between the server and tablets. The cloud storage service also acts as a backup and it keeps multiple versions of the database. A portable external hard drive is also used as a secondary backup of the database each night.

ESSENTIAL TERMS

data dictionary a set of information used to design the structure of a database table; it includes field names, data types and size as well as validation rules

data redundancy where the same piece of data, a customer's address for example, is stored in two or more places in a database. This can waste storage space if the data is merely the same data stored elsewhere, but can also be useful if it is deliberately saved as a form of backup

database software designed to store data and allow users to search and select from it in order to extract information

database management system (DBMS) software designed to store data and allow users to search and select from it in order to extract the information they need

hashing a way of taking a variable-length password and creating a cryptic, fixed-length password from it

input form a form that allows user to enter data into a database via a more user friendly interface than a table

layout diagram a hand-drawn sketch that shows the elements to be included on an input form; The diagram would indicate the placement of fields and labels, the fonts to be used, and any graphics or other elements to be included

metadata data about data; in a DBMS context it consists of details about the size and type of each field within a database file

primary key a field attached to each record of a relational database; this is a unique identifier for each record, such as a customer ID number or phone number

query to select specific data based on a series of criteria in order to answer questions and make links between data; the criteria are the results of questions about the data; for example, 'How many of our customers are female?' or 'Can I have a list of names and mobile numbers for all customers aged 20 to 40, male, who live in Fairfield or Northcote?'

RAID (redundant array of inexpensive disks) banks of hard drives that are configured in such a way that if one or two fail the data they contain isn't lost. Instead the remaining disks can rebuild the missing data in their own storage areas.

salt value a random value that you use to generate the hashed password

SQL (structured query language) a language used by many database packages to provide a consistent way of selecting required data from a table

IMPORTANT FACTS

1. The problem-solving methodology (PSM) involves four stages: Analysis, design, development and evaluation.
2. DBMSs are used in a range of contexts including banking, membership, online purchasing and in voting systems.
3. Reasons for using a DBMS for data handling include reduction of data redundancy, sharing of data, maintenance of data integrity, increased data security, easy updating of data, the ability to select data (with a query), use of forms for data input, creation of structured reports, concurrency of data and data independence.
4. Limitations of a DBMS include expensive equipment, potentially costly conversion from other file types, staff training, the need for specialist database administrators and the impact across an organisation if data is corrupted.
5. Benefits of using a DBMS include improvements in efficiency and effectiveness.
6. Risks associated with using a DBMS include inaccurate data leading to inaccurate information being produced, unauthorised users accessing data (privacy and security); they are time consuming to design properly, the possibility of accidental damage to data by untrained users, cost of purchasing suitable equipment.
7. A field contains the same type of data for a series of records. The same field for a series of records will contain the same type of data.
8. A database record is a set of data about one entity; for example, a person, event or object.
9. A database table stores sets of records.
10. A primary key uniquely identifies each record in a database table.
11. Records that meet specified criteria can be selected from the total number of records by performing a query. Each criterion can be a number, a piece of text or an expression.
12. Reports format and present the data selected by a query as usable information.

13 Common data types used in a DBMS include text, numeric, date, character and Boolean.

14 To gather data effectively, key characteristics of forms include validation, a logical arrangement of fields, uncluttered layout, and adaptive to what users select.

15 Use a consistent naming convention for DBMS objects to make it clear what they are (for tables and queries) and what they are part of (for fields). Examples are tblCustomers for a table, or qryAllProductsSoldToday. Fields should begin with letters that indicate the object they come from, such as cusFamilyName.

16 Common design tools used with databases include data dictionaries, data structure diagrams, data structure charts, flowcharts, input–process–output charts and layout diagrams.

17 A query design includes fields and criteria in order to select only the data required.

18 Reports are designed with a more structured style of layout diagram.

19 Macros can be designed with either pseudocode (structured English) or a flowchart.

20 Input–process–output (IPO) charts are used to identify inputs, outputs, and the processing steps required to transform the inputs into the outputs.

21 A list tool can be used to direct the entry of data where there is a limited number of allowed values. A list box on a form will display a predetermined set of values from which the user must select.

22 A range of devices is used to input, store, communicate and output data and information.

23 Input devices include a keyboard, mouse, optical scanner, optical reader, barcode reader, RFID reader, magnetic stripe card reader, MICR reader, sensor and data collection device.

24 Common storage devices include hard and solid state drives, flash memory devices, optical media and magnetic media. Data can also be stored in the cloud and for added security RAID can be used.

25 Communication devices include network interface cards, wireless access points, routers, switches and cables.

26 Types of output include screen (softcopy) and print (hardcopy).

27 Threats to databases can be classified as accidental, deliberate or event based.

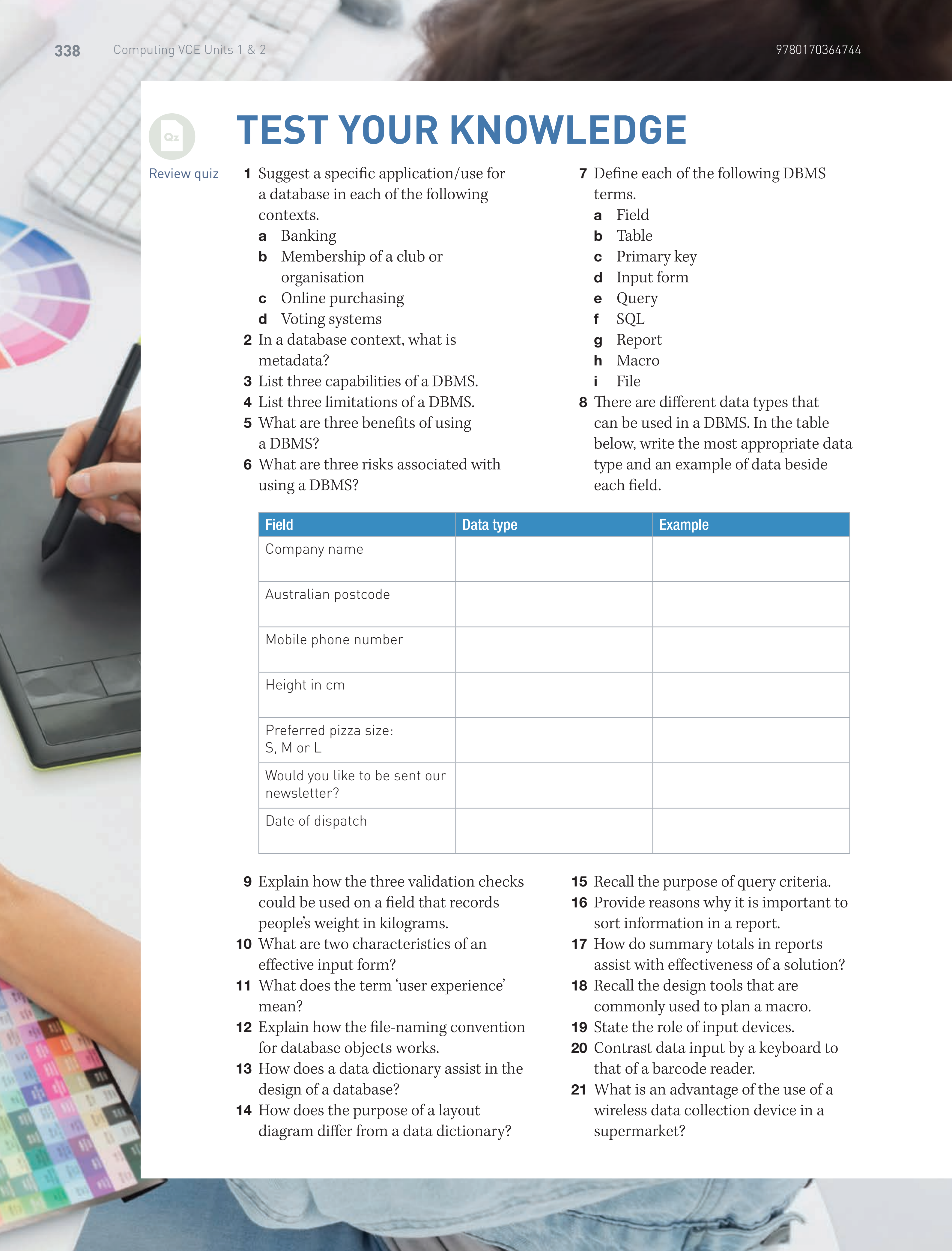
28 Accidental threats include data that is altered by accident, stored files that are lost, and data not being deleted from old storage devices.

29 Deliberate threats to data include hackers and malware.

30 Hackers may wish to damage or steal data in a database.

31 Event-based threats to databases include the crash or failure of a storage device, failure of remote connection and power loss or surge.

32 Organisations use a range of software and physical controls to protect their data. These controls include backing up, providing electrical protection, using usernames and passwords, installing system security software and using encryption.



TEST YOUR KNOWLEDGE

Review quiz

1 Suggest a specific application/use for a database in each of the following contexts.
   - a Banking
   - b Membership of a club or organisation
   - c Online purchasing
   - d Voting systems

2 In a database context, what is metadata?

3 List three capabilities of a DBMS.

4 List three limitations of a DBMS.

5 What are three benefits of using a DBMS?

6 What are three risks associated with using a DBMS?

7 Define each of the following DBMS terms.
   - a Field
   - b Table
   - c Primary key
   - d Input form
   - e Query
   - f SQL
   - g Report
   - h Macro
   - i File

8 There are different data types that can be used in a DBMS. In the table below, write the most appropriate data type and an example of data beside each field.

Field	Data type	Example
Company name		
Australian postcode		
Mobile phone number		
Height in cm		
Preferred pizza size: S, M or L		
Would you like to be sent our newsletter?		
Date of dispatch		

9 Explain how the three validation checks could be used on a field that records people's weight in kilograms.

10 What are two characteristics of an effective input form?

11 What does the term 'user experience' mean?

12 Explain how the file-naming convention for database objects works.

13 How does a data dictionary assist in the design of a database?

14 How does the purpose of a layout diagram differ from a data dictionary?

15 Recall the purpose of query criteria.

16 Provide reasons why it is important to sort information in a report.

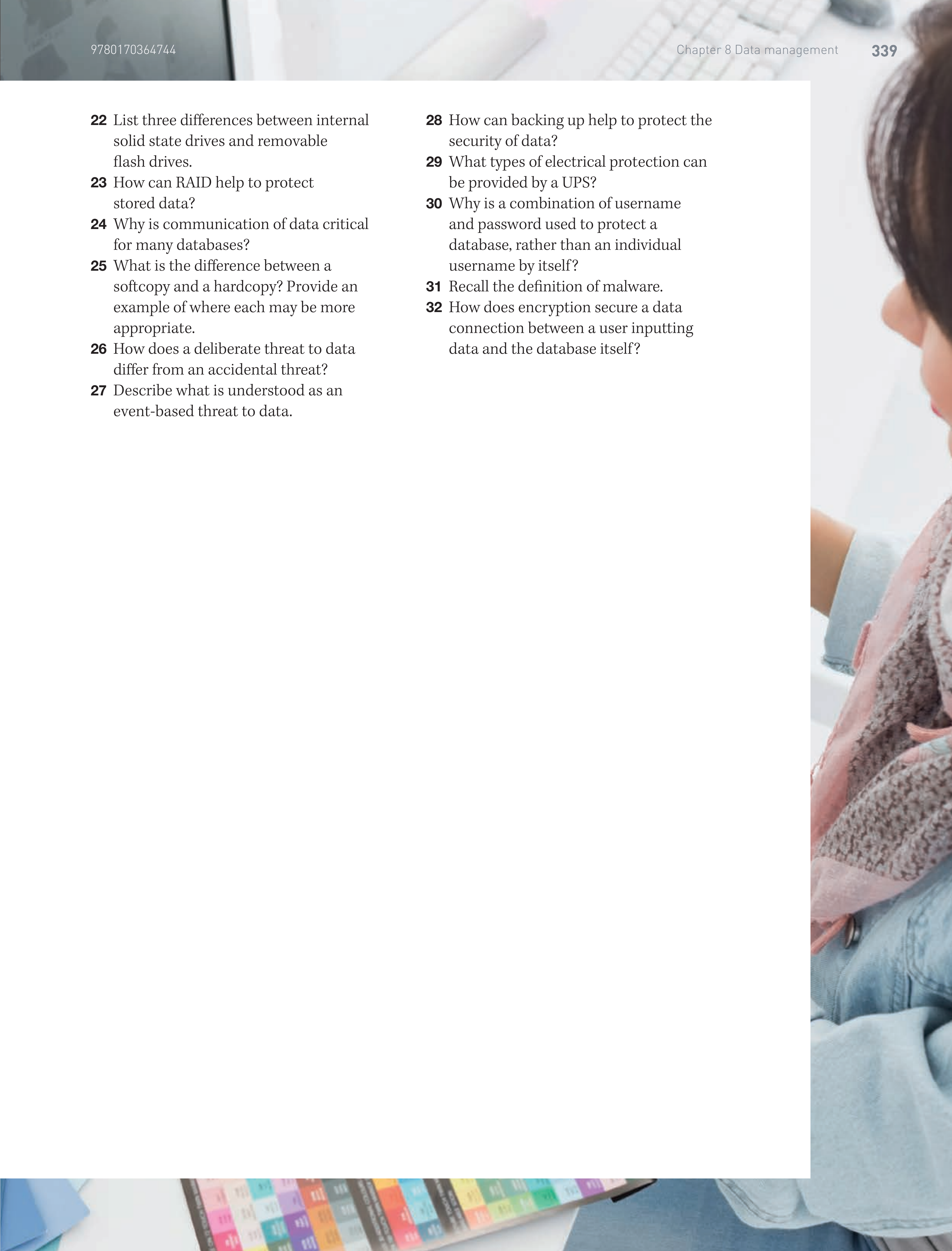
17 How do summary totals in reports assist with effectiveness of a solution?

18 Recall the design tools that are commonly used to plan a macro.

19 State the role of input devices.

20 Contrast data input by a keyboard to that of a barcode reader.

21 What is an advantage of the use of a wireless data collection device in a supermarket?



22 List three differences between internal solid state drives and removable flash drives.

23 How can RAID help to protect stored data?

24 Why is communication of data critical for many databases?

25 What is the difference between a softcopy and a hardcopy? Provide an example of where each may be more appropriate.

26 How does a deliberate threat to data differ from an accidental threat?

27 Describe what is understood as an event-based threat to data.

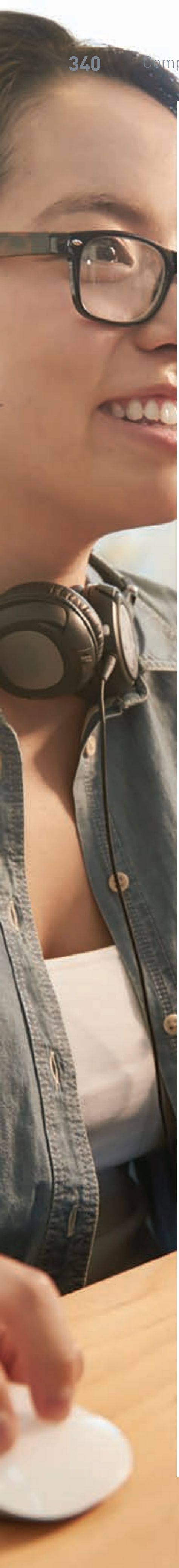
28 How can backing up help to protect the security of data?

29 What types of electrical protection can be provided by a UPS?

30 Why is a combination of username and password used to protect a database, rather than an individual username by itself?

31 Recall the definition of malware.

32 How does encryption secure a data connection between a user inputting data and the database itself?



APPLY YOUR KNOWLEDGE

DATABASE ACTIVITY

Catherine works as a junior programmer for a small software company. She currently earns $2500 a month and pays $300 in rent. She has many other monthly expenses. Sometimes she has overspent and run out of money before her next pay period. This has led to embarrassment when she has been unable to buy food for herself or her cat at the local shop. In order to manage her budget better, Catherine has asked you to create a database in which she can store her bank transactions. At the end of each month she will use this database to analyse her spending and determine if she is in credit or debit. For the purposes of this task, she is working with the month of November 2016.

The two reports Catherine would like are as follows.

a A report that shows all the debits from her cheque account for the month of November 2016. It should be grouped by transaction category and sorted by transaction date within each category. Be sure to include summary statistics that show the percentage and subtotals for each group as well as an overall total so Catherine can see what money remained at the end of the month.

b A report that lists all expenses from all of Catherine's accounts for the 'Car/Travel' category for the month of November 2016. The report should be sorted by transaction date (from earliest to most recent) and then by amount (smallest to largest amount).

1 Table 8.9 shows the data set that Catherine has provided.

2 Looking carefully at the data, create a data dictionary to plan the database table. You may rearrange the data into different fields as you see fit. Also remember to include validation and to choose appropriate data types and formats.

3 Design a suitable data input form that will enable Catherine to add, review and edit her bank records. Be sure to make use of labels and dropdown lists to assist in more accurate data entry.

4 Create designs for the two queries that will be used to create the reports outlined above.

5 Design the two reports outlined above.

6 Design macros that will open, print and close each of the reports when they are run.

7 Using a DBMS package, create Catherine's bank transactions database. This will involve constructing the table, the input form, the two queries, two reports and the two macros you designed above. Remember to include the data validation and appropriate formulas and functions.

8 Create annotated screenshots of the various database objects you have created to show their key features and how you have met Catherine's needs.

9 In a paragraph, list the roles, functions and characteristics of suitable hardware components that Catherine could use to input, store, communicate and output the data and information in her database.

10 Identify one each of an accidental, deliberate and an event-based security threat to the data and information stored in Catherine's database.

11 Identify and explain suitable physical and software controls that will protect the data when it is stored in Catherine's database and when it is transmitted (both physically and electronically).

12 Evaluate the potential success of the DBMS you have created for Catherine by comparing relevant benefits and risks from its use.

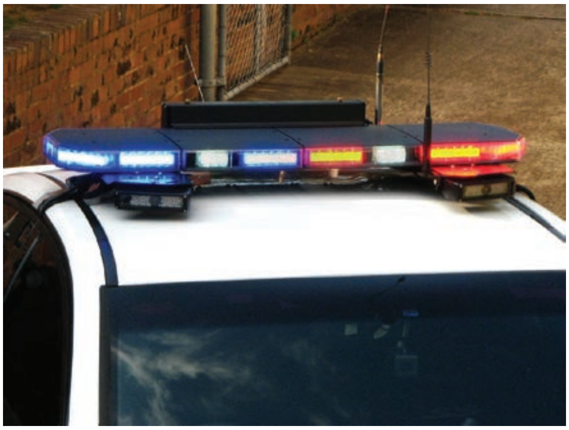
TABLE 8.9 The table of data Catherine has supplied you with for her bank transaction database.

Transaction ID	Account (cheque or savings)	Date	Details	Category	Transaction type (debit or credit?)	Amount ($)
21	Cheque	01-Oct-16	Monthly salary	Salary	C	2500.00
30	Cheque	15-Oct-16	Rent	Miscellaneous	D	300.00
5	Cheque	15-Oct-16	Food shopping	Food/groceries	D	15.20
10	Cheque	01-Nov-16	Petrol for car	Car/Travel	D	34.56
14	Cheque	01-Nov-16	Movie at cinema	Entertainment	D	25.00
15	Cheque	01-Nov-16	Popcorn and ice-cream at cinema	Entertainment	D	15.00
6	Cheque	01-Nov-16	Food shopping	Food/Groceries	D	8.54
24	Cheque	01-Nov-16	Monthly salary	Salary	C	2500.00
22	Cheque	03-Nov-16	Mobile recharge	Miscellaneous	D	20.00
9	Cheque	05-Nov-16	Dinner with Nathan	Entertainment	D	125.87
7	Cheque	07-Nov-16	Food shopping	Food/groceries	D	185.21
34	Savings	08-Nov-16	Refund from tax dept	Salary	C	350.28
32	Savings	12-Nov-16	Birthday present and card for Mum	Miscellaneous	D	125.99
31	Cheque	15-Nov-16	Rent	Miscellaneous	D	300.00
11	Cheque	15-Nov-16	Petrol for car	Car/travel	D	56.28
36	Savings	15-Nov-16	Puncture repair	Car/travel	D	35.00
33	Cheque	15-Nov-16	MSO concert	Entertainment	D	85.00
2	Cheque	15-Nov-16	Car insurance	Car/travel	D	551.67
4	Cheque	17-Nov-16	Red shirt	Clothing	D	65.00
17	Cheque	17-Nov-16	Black pants	Clothing	D	35.50
1	Cheque	17-Nov-16	New shoes	Clothing	D	125.00
25	Cheque	18-Nov-16	Car service	Car/Travel	D	245.89
23	Cheque	18-Nov-16	Cat food	Miscellaneous	D	35.99
8	Cheque	20-Nov-16	Food shopping	Food/Groceries	D	57.66
16	Cheque	21-Nov-16	Placemats for table	Miscellaneous	D	30.00
12	Cheque	22-Nov-16	Petrol for car	Car/travel	D	72.45
29	Savings	22-Nov-16	Repair bike	Miscellaneous	D	65.80
28	Cheque	23-Nov-16	Movie and pizza with Bill	Entertainment	D	85.00
19	Cheque	25-Nov-16	Sleeping bag	Miscellaneous	D	156.00
20	Cheque	29-Nov-16	Sunscreen 50+	Miscellaneous	D	15.99
18	Cheque	29-Nov-16	Sun hat	Clothing	D	24.99
13	Cheque	01-Dec-16	Petrol for car	Car/travel	D	22.60
35	Cheque	01-Dec-16	Monthly salary	Salary	C	2500.00
3	Cheque	03-Dec-16	Food shopping	Food/groceries	D	34.55

AUTOMATIC NUMBERPLATE RECOGNITION

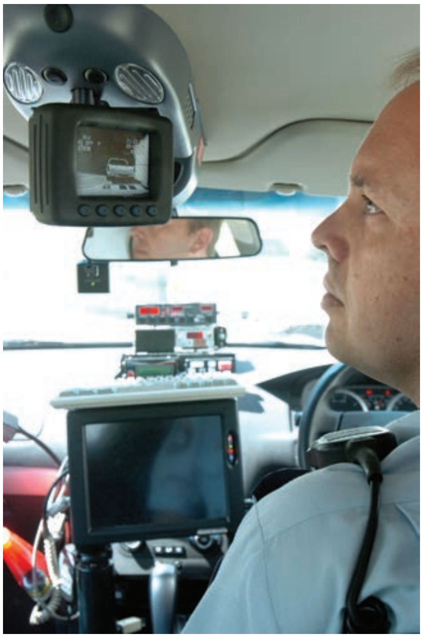
New South Wales launched an automatic numberplate recognition (ANPR) system in 2009. It can scan and check up to six numberplates a second from passing and oncoming cars. The system is typically mounted on police cars (see Figure 8.52). There is a terminal inside the car that displays information about cars that are scanned, in real time (see Figure 8.53) and alerts police if that car ought to be pulled over for further investigation. The numberplate details and the date, time and location of the scan for every car the ANPR system detects are transmitted immediately and stored in a police controlled database. This data is stored for a minimum of 5 years. Privacy advocates have raised concerns about the quantity of data being collected and the purposes for which it may be used, other than to help with highway patrols.

1 Identify one each of an input, storage, communication and output device that might be used in the ANPR system.
2 Suggest two reasons as to why privacy might be a concern with this system.
3 List some ways the data could be misused.
4 Suggest two threats that the ANPR database might face.
5 Explain a suitable control measure to protect the ANPR data once it is stored in the database and while it is transmitted to and from a police car.



Eastwood Highway Patrol, https://creativecommons.org/licenses/by/2.0/

FIGURE 8.52 Roof mounted lightbar and ANPR cameras on a NSW patrol car



Newspix/Tracee Lea

FIGURE 8.53 The ANPR terminal inside a police car

PREPARING FOR UNIT 2 OUTCOME 3

Apply the problem-solving methodology to create a solution using database management software, and explain the personal benefits and risks of interacting with a database

Devise a need or opportunity for a database solution, collect relevant data and create that solution using database management software (DBMS). The solution must make use of a table, input forms, queries and reports. Where appropriate, apply mathematical calculations to the data and create macros to automate repetitive tasks. Identify and explain suitable hardware that can be used with the database, as well as risks to data and information, and ways to minimise these risks.

You will apply all four stages of the PSM: Analysis, design, development and evaluation.

OUTCOME MILESTONES

1. Identify a need for a data-handling task that requires the use of a DBMS as its solution.
2. Prepare an analysis of the need, which includes an outline of the organisation, a list of functional and non-functional requirements of the solution, a list of constraints, and the scope of the solution.
3. Identify input data and information required to fulfil the need.
4. Design the solution.
5. Use relevant functions of DBMS software to develop the solution.
6. Use relevant software functions to demonstrate features of the software.
7. Identify relevant hardware components to input, store, communicate and output data and information.
8. Identify threats to data and information stored within databases.
9. Identify physical and software controls to protect data when stored and transmitted.
10. Identify risks and benefits associated with using database solutions.

STEPS TO FOLLOW

The analysis, design, development and evaluation stages of the problem-solving methodology will be used to create the solution, although you will not be required to create any documentation nor conduct any testing.

1. You must identify a user or organisation where a need for a DBMS exists. The user can be yourself and the database need may be as simple as recording your music collection or sports club statistics.
2. Analyse and define the need. In the analysis clearly identify:
   - a the solution requirements (needs of the users and problem to be solved, functional and non-functional requirements)
   - b the solution constraints
   - c the scope of the solution (solution boundaries and what it will and will not do).

The problem to be solved and the needs of the users are best written as simple statements; for example, 'Sometimes the lists of swimmers are inaccurate because they are mistakenly put into the wrong age group or left off the list totally! In addition, some financial problems have occurred for the club as they have not accurately followed up which members have not paid their annual fees. This also leads to embarrassment for club officials when payment notices are sent to the wrong club members'.

3 Design the solution. The following are required.
   - a Use of a data dictionary to represent the table plan
   - b Layout diagrams to plan the forms
   - c Query plans to design the queries
   - d Layout diagrams for the reports to be generated
   - e Pseudocode or flowcharts for any macros used

4 Develop the solution. For a DBMS-based solution, ensure that you have created all the objects required and named them correctly.

5 Validate the data and input it.

6 Generate the required information from the DBMS solution (the reports).

7 Evaluate the solution by commenting on the following.
   - a Appropriate hardware for input, storage, communication and output
   - b Accidental and deliberate threats relevant to the database
   - c Relevant physical and software controls to protect the stored or transmitted data
   - d Identification of benefits or risks for the user arising from the use of this database

DOCUMENTS REQUIRED FOR ASSESSMENT

1 Documentation produced during the analysis stage, including the solution requirements, constraints and scope of the solution

2 A representation of the design of each part of the solution

3 Printouts of the solution and information

4 Annotated copies of the solution and information indicating features you have used and conventions you have followed and how it meets the needs of the users

5 Documentation to evaluate the hardware, threats, security controls, risks and benefits associated with the database

ASSESSMENT

A set of assessment criteria will be prepared and distributed by your teacher before you start the task. Details on the actual criteria for this Outcome and the breakdown of how each criterion will be marked are available at http://computing1and2.nelsonnet.com.au.