

Databases: Querying and analysing data using Access

How to Use This Course Book

This handbook accompanies the taught session for the course. Each section contains a brief overview of a topic for your reference and then one or more exercises.

The Exercises

Exercises are arranged as follows:

- A title and brief overview of the tasks to be carried out
- A numbered set of tasks, together with a brief description of each
- A numbered set of detailed steps that will achieve each task

Your lecturer will direct you to the location of files that are needed for the exercises. If you have any problems with the text or the exercises, please ask the lecturer or one of the demonstrators for help.

This book includes plenty of exercise activities – more than can usually be completed during the hands-on sessions of the course. You should select some to try during the course, while the teacher and demonstrator(s) are around to guide you. Later, you may attend Course Clinics at the IT Learning Programme, where you can continue work on the exercises, with some support from IT teachers. Other exercises are for you to try on your own, as a reminder or an extension of the work done during the course.

Writing Conventions

A number of conventions are used to help you to be clear about what you need to do in each step of a task.

- In general, the word **press** indicates you need to press a key on the keyboard. **Click**, **choose** or **select** refer to using the mouse and clicking on items on the screen (unless you have your own favourite way of operating screen features).
- Names of keys on the keyboard, for example the Enter (or Return) key, are shown like this ENTER.
- Multiple key names linked by a + (for example, CTRL+Z) indicate that the first key should be held down while the remaining keys are pressed; all keys can then be released together.
- Words and commands typed in by the user are shown like `this`.
- Labels and titles on the screen are shown like **this**.
- A button to be clicked will look like this.
- The names of software packages are identified *like this*, and the names of files to be used like `this`.

Software Used

Access 2013

Files Used

Inventory.accdb
New Dentists.accdb

Revision Information

Version	Date	Author	Changes made
1.0	August 2014	Pamela Stanworth	Created
1.1	January 2015	Pamela Stanworth	Small edits
2.0	July 2015	Pamela Stanworth	Revised for Access 2013

Copyright



Pamela Stanworth makes this document and the accompanying PowerPoint presentation available under a Creative Commons licence: Attribution-NonCommercial-ShareAlike CC BY-NC-SA. pamela.stanworth@it.ox.ac.uk

Screenshots in this document are copyright of Microsoft.

The Oxford University crest and logo and IT Services logo are copyright of Oxford University and may only be used by Oxford University members in accordance with the University's branding guidelines.

Contents

1 Introduction	1
1.1 What You Should Already Know	1
1.2 What You Will Learn.....	1
1.3 What is <i>Access</i> ?	1
1.4 Using <i>Access (Office) 2013</i>	2
1.5 Keyboard Methods for Using <i>Access</i>	2
1.6 Where Can I Get A Copy?.....	3
1.7 App or Desktop Database?	3
1.8 Using the Database Files for ITLP Exercises.....	3
2 Select Queries.....	4
2.1 Using Queries	4
2.2 Creating a New Query Using the Query Wizard.....	5
2.3 Naming Queries and Other Objects in a Database	6
2.4 Query Results.....	7
2.5 Looking at the Query in Design View	7
2.6 Switching Query Views	8
2.7 Re-using a Query.....	8
3 Working on a Query in Design View	9
3.1 Relationship Between Two Tables	9
3.2 Creating a Query With Two Tables.....	9
3.3 Adding Fields to a Query	10
3.4 Running a Query Based on Two Tables.....	11
4 Criteria in a Query	12
4.1 Criteria With Equals.....	12
4.2 Changing Criteria in a Query Design.....	12
4.3 Criteria With Comparison Operators	12
4.4 Criteria Using Null And Not	13
4.5 <i>Access</i> Edits Punctuation in Criteria	14
4.6 Wildcard Symbols in Criteria	15
4.7 Yes/No Criteria	15
4.8 AND and OR in Query Criteria.....	15
5 Editing the Query Design.....	17
5.1 Removing a Field.....	17

5.2 Rearranging Columns	17
5.3 Adjusting Column Width	17
5.4 Hiding/Showing A Field Column	17
5.5 Sorting Query Results	17
5.6 Finishing	18
6 Parameter Queries	19
6.1 Entering Parameter Expressions as Criteria	19
6.2 Parameter Query Problems	20
6.3 Parameter Query Using Wildcards (<i>Optional</i>)	20
7 Some Special Kinds of Query	22
7.1 Top Values (<i>Optional</i>)	22
7.2 Query to Find Duplicate Records	22
7.3 Query to Show Unmatched Values	24
7.4 Crosstab Query - What Is It For?	26
7.5 Creating a Crosstab Query	27
8 Calculations and Functions In A Query	30
8.1 Calculated Columns In a Query	30
8.2 Operators in Calculations	31
8.3 “Totals” in Queries – Grouping Records Together	31
9 Calculated Controls in a Form	33
9.1 A Note on Views of Forms	33
9.2 Bound and Unbound Controls	33
9.3 Creating a Control With A Calculation	33
9.4 More Built-In Functions (<i>Optional</i>)	34
9.5 Using Calculated Form Controls	36
10 Calculated Controls in a Report	37
10.1 A Note on Views of Reports	37
10.2 Bound and Unbound Controls in a report	37
10.3 Creating a Calculated Control	37
10.4 Formatting a Calculated Control	38
10.5 Running Sum Format (<i>Optional</i>)	39
10.6 Summarising “Totals” in Reports	39
10.7 Calculations In a Grouped Report (<i>Optional</i>)	40
11 Conditional Formatting For Controls	43
12 Action Queries	45

12.1 Update Values Query	45
12.2 Delete Records Query.....	46
12.3 Make-Table Query.....	47
12.4 Append New Records Query	48
12.5 Running Action Queries from the Navigation Pane	49
13 What Next?	50
13.1 Downloadable Course Materials and More – the ITLP Portfolio...	50
13.2 Database Courses Which Precede This.....	50
13.3 Word-processing and Spreadsheet Courses	50
13.4 Course Clinics.....	50
13.5 IT Services Help Centre.....	50
Appendix 1: Viruses and Security Levels in Access.....	51
1. Why Worry About Viruses?	51
2. Scanning for Viruses	51
3. Access 2013 and Virus Security	51
4. Further Help with Security and Trust Center Settings	55
Appendix 2: Student Exercises.....	56

1 Introduction

Welcome to the course “Databases: Querying and analysing data using *Access*”.

This booklet accompanies the course delivered by Oxford University’s IT Learning Programme. Although the exercises are clearly explained so that you can work through them yourselves, you will find that it will help if you also attend the taught session where you can get advice from the teachers, demonstrators and even each other!

If at any time you are not clear about any aspect of the course, please make sure you ask your teacher or demonstrator for some help. If you are away from the class, you can get help by email from your teacher or from help@it.ox.ac.uk.

1.1 What You Should Already Know

This session is one of a series that cover the important aspects of building and managing a database, using *Microsoft Access* as an example.

We will assume that you have already attended the course “Databases: Building a database using Access”, “Databases: User-friendly databases using Access” and “Databases: Reporting data using Access” (or equivalent), and that you are familiar with creating tables in *Access* with suitable fields, and creating basic forms or reports using the wizards.

The computer network in our teaching rooms may differ slightly from that which you are used to in your College or Department; if you are confused by the differences, ask for help from the teacher or demonstrators.

1.2 What You Will Learn

In this session we will cover the following topics:

- Devising select queries and setting criteria
- Parameter queries
- More queries – action queries, calculations and totals
- Calculations in a form
- Summary calculations in a report
- Conditional formatting
- Action queries for housekeeping and organising data

Related *Database* sessions, should you be interested, are given in Part 13 below.

1.3 What is *Access*?

Access is database management software. It enables you to build and maintain a database.

Access 2013 is part of the *Microsoft Office 2013* for Windows package. *Access* is not available for the Mac (Mac users may try *FileMaker Pro*).

An *Access* database consists of the data held in a number of tables, plus a number of other objects which are used to manage the data. All these are saved together in a single file. In this course, we will deal with tables, queries, forms, reports, and macros.

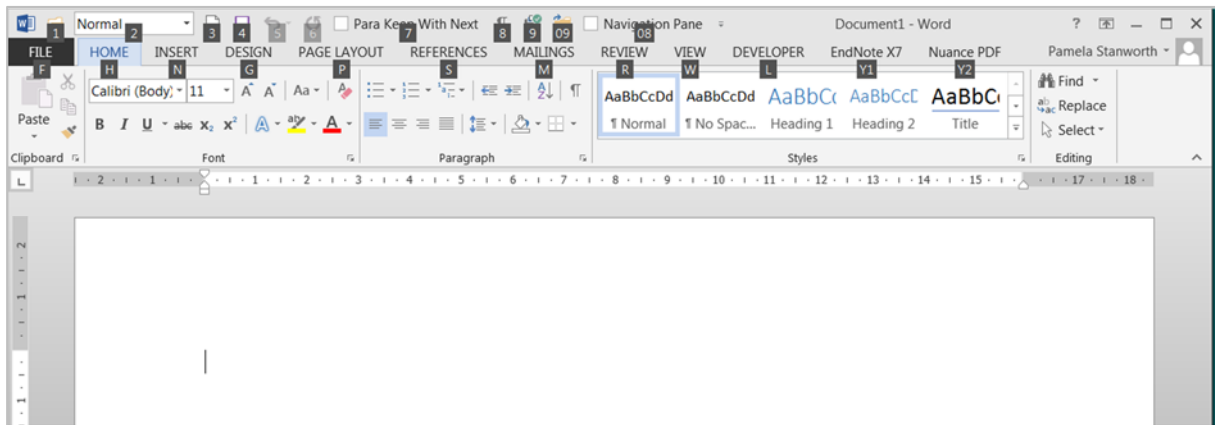
A database *application* may be built – using forms, menus, control buttons and program code that responds to events.

1.4 Using *Access (Office) 2013*

If you have previously used another version of *Office*, you may find *Office 2013* looks rather unfamiliar. “Office 2010: What’s New” is a self-study guide covering the ribbon, Quick Access Toolbar and so on. This can be downloaded from the ITLP Portfolio at <http://portfolio.it.ox.ac.uk>

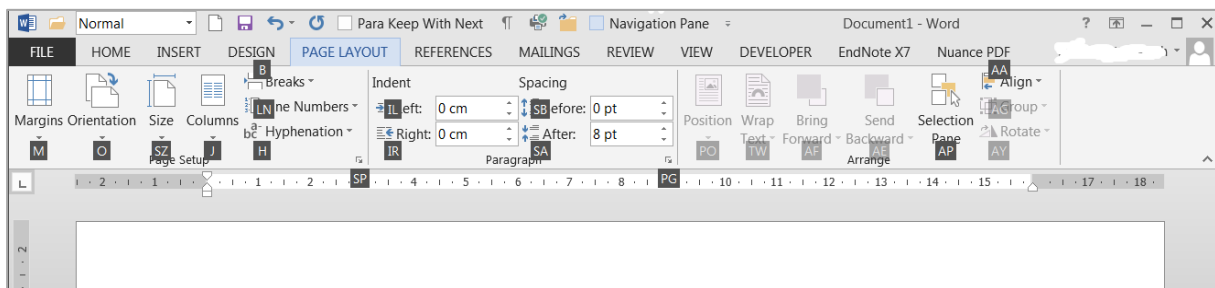
1.5 Keyboard Methods for Using *Access*

For anyone who prefers not to use a mouse to control software, or who finds a keyboard method more convenient, it is possible to control *Office 2013* applications without using a mouse. Pressing ALT once displays a black box with a letter or character next to each visible item on the ribbon and title bar (shown in Figure 1).



**Figure 1 Keystrokes for Controlling Ribbon Tabs and Title Bar
(Press ALT to show these)**

After you have typed one of the letters/characters shown, the relevant ribbon tab or detail appears, with further letters/characters for operating the buttons and controls (shown in Figure 2).



**Figure 2 Further Keystrokes for Controlling Buttons
(press ALT+P to show these on the Page Layout tab of the Ribbon)**

The elements of a dialog can be controlled, as usual with *Windows* applications, by using TAB to navigate between items or typing the underlined character shown beside an item.

1.6 Where Can I Get A Copy?

If you have a copy of *Microsoft Office Professional*, then you already have a copy of *Access*. If you are unable to find it on your computer, it may not have been installed and you should talk to your IT support contact (or the IT Services Help Desk).

If you are a member of staff, you can obtain a copy of *Microsoft Office Professional* from the IT Services Online Shop. Students can obtain a Microsoft Student Licence, but this must be bought through a Microsoft Authorised Education Reseller; the help desk can direct you to a suitable reseller.

1.7 App or Desktop Database?

Access 2013 can also be used to create an **app database**: where users work on the data via a web browser. This would require communication using *Office 365* or *SharePoint 2013*, and is not the subject of this course. We will create a **desktop database**, which is saved locally on your computer or a network drive.

1.8 Using the Database Files for ITLP Exercises

Please note that *Access* only trusts files if they have been saved in a “Trusted Location”. The network drive H:, used for most IT Learning Programme courses, has been designated an Access Trusted Location. If you make copies of the files for these exercises, and save them on your own computer in a location that is not trusted, you may not be able to carry out all the activities described.

Appendix 1 discusses the question of virus protection and trusting locations further.

Exercise 1: Opening a database file

Now look at this exercise (page 56).

2 Select Queries

2.1 Using Queries

Once a quantity of data has been collected in tables, it becomes interesting to analyse it using a query, looking perhaps at only selected data records, or only some fields, or looking at related records from a number of tables.

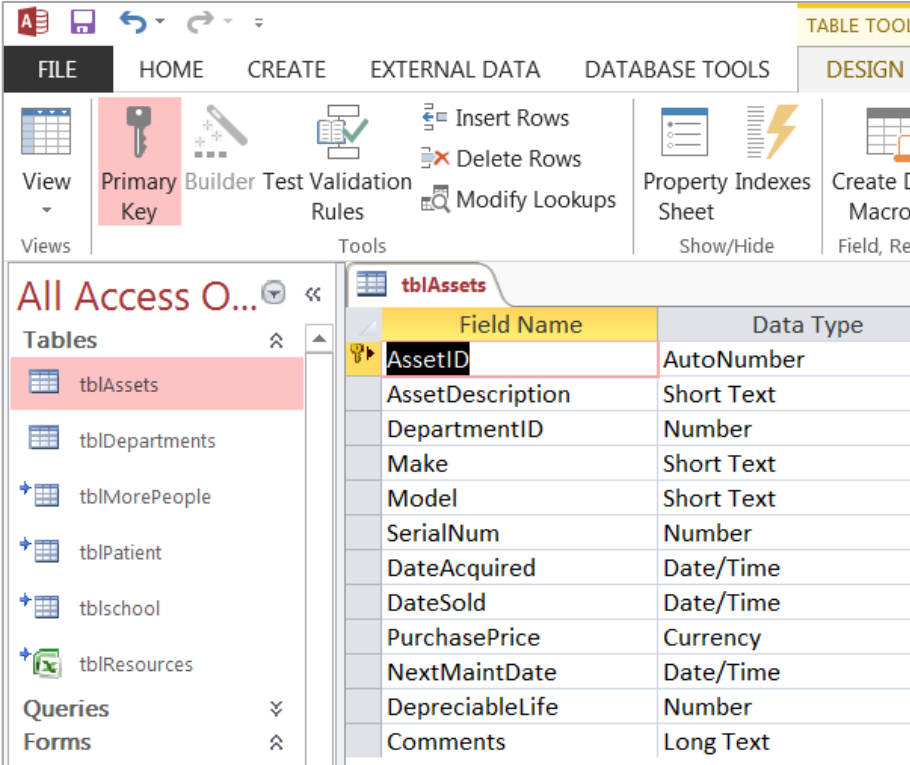
The most common type is the select query, which selects records and/or fields from related tables as requested, then displays the results in a datasheet that looks similar to Table Datasheet View.

A query is used to investigate and manipulate data that has been entered into the tables that make up a database.

A query may be used to:

- Sort records
- Select only some fields
- Select only some records using criteria
- or Collect related data from several tables

Suppose you have a database file open, with a table already created and filled with suitable data.



Field Name	Data Type
AssetID	AutoNumber
AssetDescription	Short Text
DepartmentID	Number
Make	Short Text
Model	Short Text
SerialNum	Number
DateAcquired	Date/Time
DateSold	Date/Time
PurchasePrice	Currency
NextMaintDate	Date/Time
DepreciableLife	Number
Comments	Long Text

Figure 3 Table Design

It may be interesting to present only a few of the fields from the table, to start considering the significance of the data. A query can be created to do this, based on the table.

Queries that have already been created are listed in the Navigation Pane on the left-hand side of the screen. This pane has a menu for controlling the list that it displays: click the white/red bar at the top of the pane (labelled **All Access Objects** in Figure 3 above. If you do not see the queries, select **Object Type** (under **Navigate To Category**), then go back to this menu and make sure **All Access Objects** is selected as well.

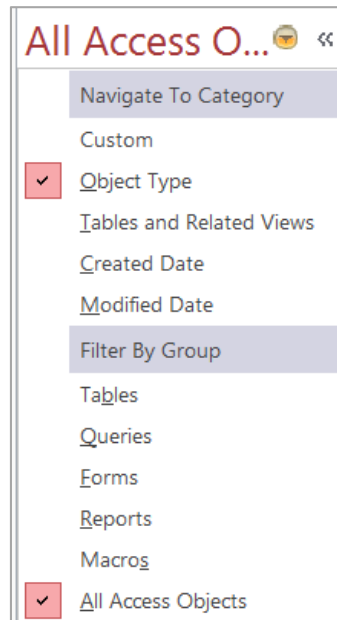



Figure 4 Showing All Access Objects (including Queries, if any)

2.2 Creating a New Query Using the Query Wizard

The Query Wizard is started by selecting  on the **Create** tab of the ribbon, then selecting **Simple Query Wizard** from the list. The wizard takes you step by step through creating a new query. Here you specify the table it is to be based upon, and choose which fields are to be included.

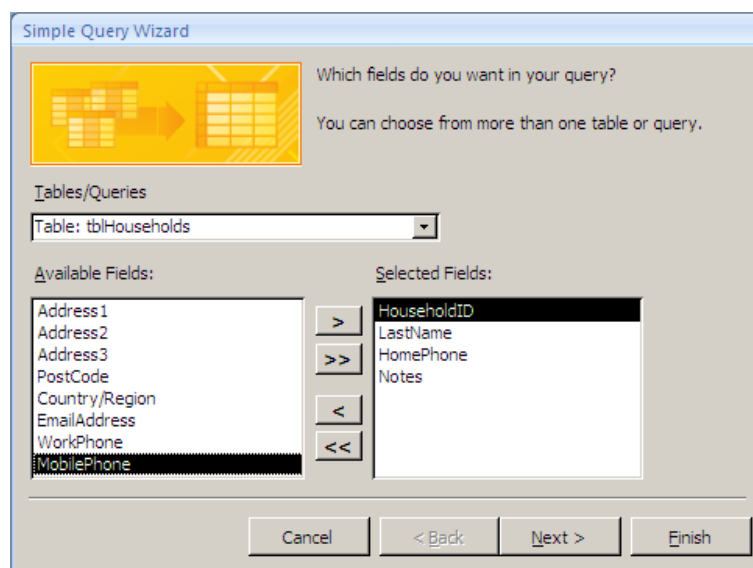


Figure 5 The Query Wizard

2.3 Naming Queries and Other Objects in a Database

In the next part of the wizard, the query is given a name.

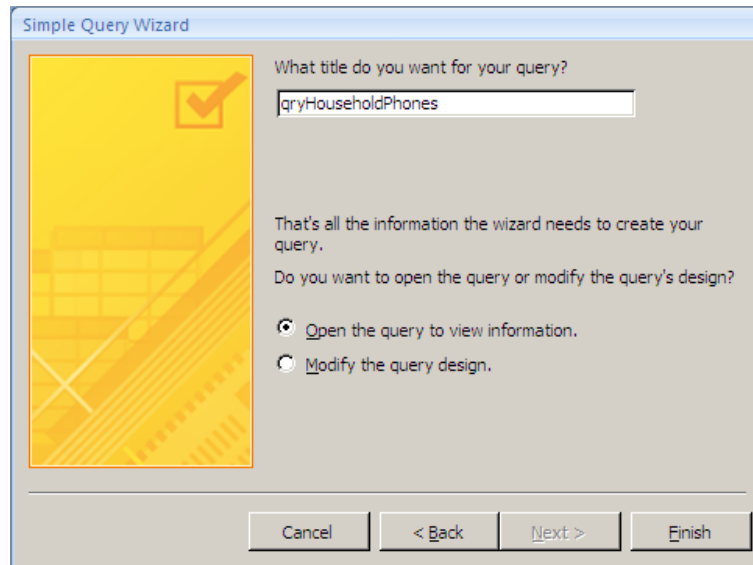


Figure 6 The Query Wizard, continued

By convention, query names always begin with **qry**. Similarly, table names begin with **tbl**, form names begin with **frm** and report names begin with **rpt**. These Reddick naming conventions can be found at <http://mvps.org/access/general/gen0012.htm>.

Although it is possible in *Access* to include spaces in object names, it is good practice to avoid them. This is because when an object name is included in an expression (perhaps in a calculation), then any spaces are difficult for *Access* to parse and you must remember to enclose the object name in [square brackets] every time. If there are no spaces, then the names of queries, forms etc are not ambiguous and expressions are easier for *Access* (and people) to read. This also applies to the names of fields and controls.

Designers may use a mixture of lower case and capital letters, or may use some punctuation symbols to make object names and field name easy to understand.

Examples might be **tblStudentApplications** or **qryLateReplies**, or **frmSimple_Address_List**.

2.4 Query Results

After the wizard finishes, the query results are presented in Datasheet View (by default). This view looks similar to Table Datasheet View, although only the fields requested are shown.

DateAcquired	DateSold	PurchasePrice	NextMaintDate	DepreciableL	Comments	Departn
23 July 1937	07 May 1939	£351.03	27 November 2004	5		
01 December 1997	19 October 1998	£135.03	08 December 2005	2		
17 June 1953	26 December 1953	£42.52	29 May 2005	8		
30 December 1936	25 August 1937	£25.75	13 December 2004	9		
09 September 1948	30 January 1949	£10.73	10 December 2004	10		
31 October 1971	09 April 1973	£321.73	08 October 2005	20		
22 July 1902	14 June 1904	£88.84	02 April 2005	20		
01 October 1973	25 January 1974	£9.71	23 July 2005	9		
12 November 1963	05 January 1965	£227.22	24 September 2004	8		
11 May 1901	13 November 1902	£133.06	13 March 2005	9		
22 November 1996	06 April 1997	£430.23	06 March 2005	9		

Figure 7 Query Results in Datasheet View

Each time the query is run, *Access* re-queries the raw data in the table and presents up-to-date results.

Only those fields chosen in the query wizard are shown. All records are shown (until we set some criteria, as discussed below). The data in this view can be edited – changes made here will affect the underlying data in the table.

2.5 Looking at the Query in Design View



on the **Home** tab of the ribbon will switch between Datasheet View and Design View.


qryAssets				
tblAssets				
* AssetID				
AssetDescription				
DepartmentID				
Make				
Model				
SerialNum				
Field:	[AssetID]	[AssetDescription]	[DepartmentID]	[Make]
Table:	tblAssets	tblAssets	tblAssets	tblAssets
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:				
or:				


Figure 8 Simple Query in Design View

In Design View, all tables that are used in the query are shown at the top. In the design grid below, field names are chosen and set in order.

2.6 Switching Query Views

The **View** button is now visible on the **Home** tab or the **Query Tools Design** tab,

and it has changed appearance to . This will switch back to Design View.

Use  to close the query. You will be prompted to save any changes to the query design (recall that any edits or additions to data are saved without comment as you move between records).

2.7 Re-using a Query

Once a query has been designed, it is saved and assigned a name (conventionally beginning **qry** and avoiding spaces and special punctuation symbols). The query can be run and re-run from the Navigation Pane.

Each time a query is run, *Access* interrogates the underlying data from the table(s), so the results displayed are always up-to-date.

Exercise 2: Creating select queries using the wizard

Now look at this exercise (page 57).

3 Working on a Query in Design View


3.1 Relationship Between Two Tables

If the data in separate tables is related, then relationships must be set up between corresponding fields. This was discussed in a separate course, “Databases: Building a database using Access”, which considers the wider question of defining relationships for a whole database (see Part 13.2 below).

A query that is based on two or more related tables will look for records in one table which correspond to records in the other table. For example, each person in an address book will be associated with the household where they live.

In this way, a one-to-many relationship can be represented, without duplicating data between records. For example many members of a household may live at one address, but each has their own birthday.

3.2 Creating a Query With Two Tables

To create a new query in design view, choose  from the **Create** tab on the ribbon. The empty query design grid is shown, with the Show Table dialog where you can select from the available tables (and queries).

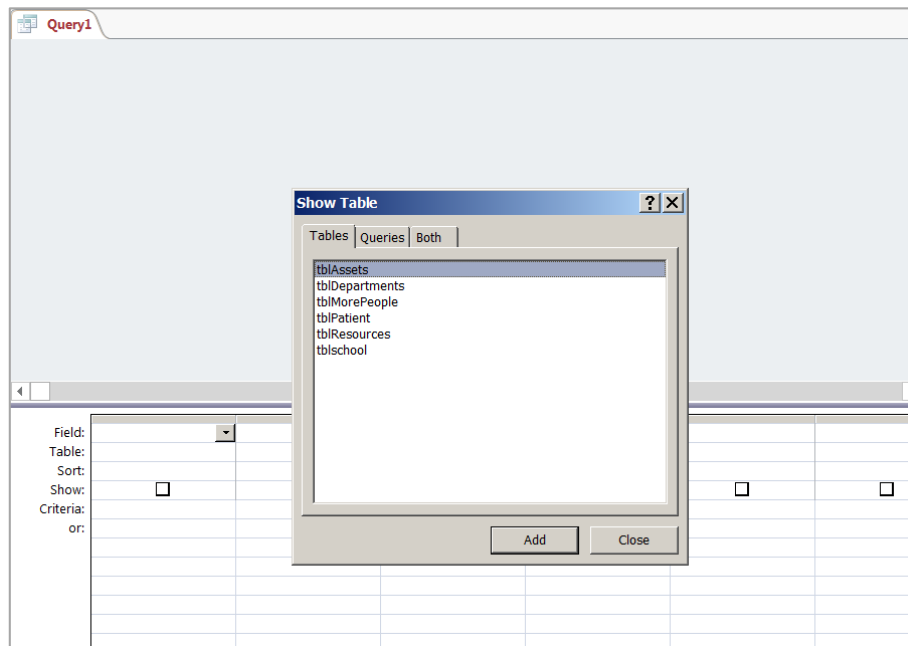


Figure 9 Adding Tables to Query Design

In the Query Design View, lines appear between the tables, representing any joins that have previously been set up. If no such lines appear, you need to work in the Relationships Diagram to join the tables.

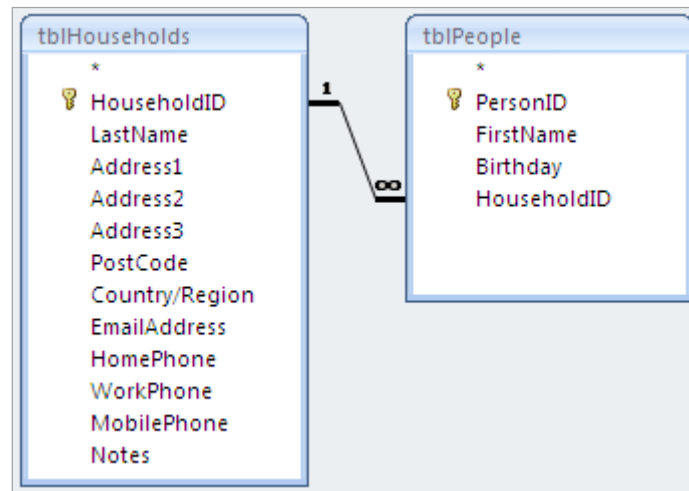


Figure 10 Two Tables With a Join, in Query Design View

3.3 Adding Fields to a Query

The required fields can be added into the empty columns in the design grid, using various methods:

- Drag a field name from a table field list and drop it in an empty column
- Double-click a field name in a table field list – it will appear in the next empty column
- Click at the top of an empty column to display an arrow button; click this to activate a drop-down list, then choose a field name from the list

Add as many fields as are needed, from either or both tables, in any convenient order.

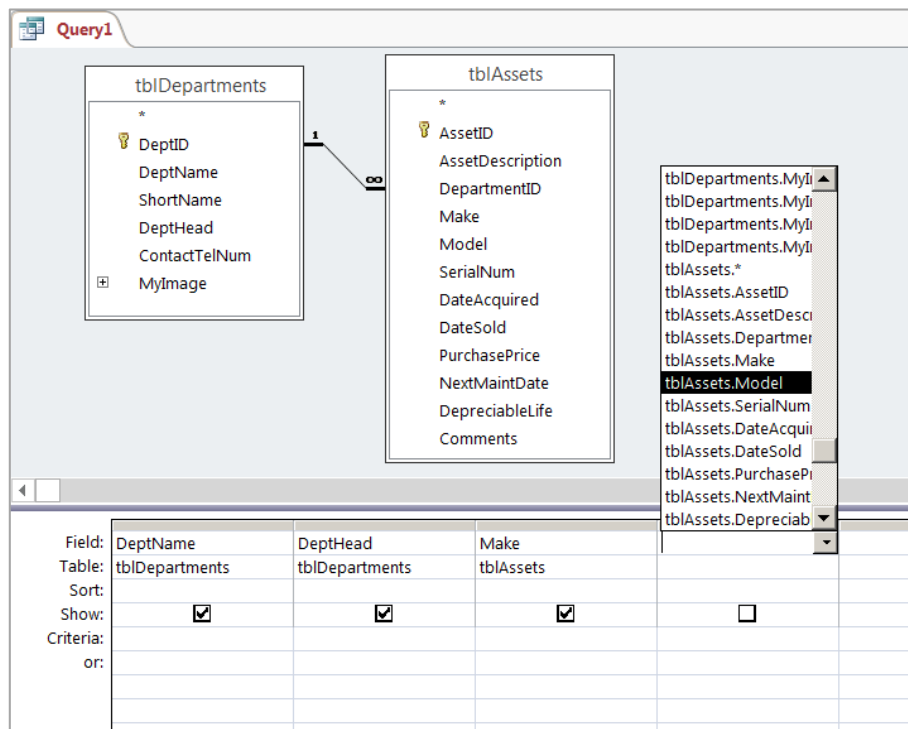





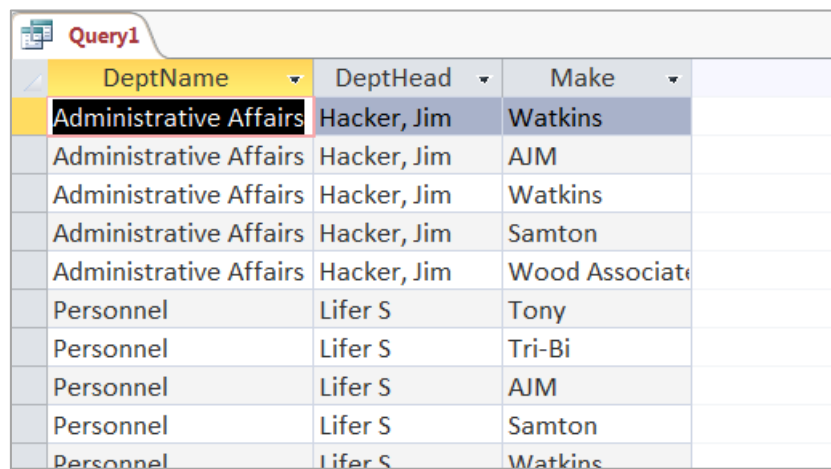
Figure 11 Building a Query With Two Tables

The asterisk * can be used to represent all the fields in a table (this provides a more flexible structure, in case the table design changes in the future).

If necessary later, add a further table by clicking  and choosing the table name in the Show Table dialog.

3.4 Running a Query Based on Two Tables

Click  or  (in the case of select queries, the two buttons are equivalent) to run the query. Access collects current data from the tables and presents them as a datasheet. Access compares the records in both tables, looking at the field in each which is used in the join relationship. Any value of that field which arises in both tables is deemed a match, and that record is included in the results dataset.



DeptName	DeptHead	Make
Administrative Affairs	Hacker, Jim	Watkins
Administrative Affairs	Hacker, Jim	AJM
Administrative Affairs	Hacker, Jim	Watkins
Administrative Affairs	Hacker, Jim	Samton
Administrative Affairs	Hacker, Jim	Wood Associates
Personnel	Lifer S	Tony
Personnel	Lifer S	Tri-Bi
Personnel	Lifer S	AJM
Personnel	Lifer S	Samton
Personnel	Lifer S	Watkins

Figure 12 Results of a Query From Two Tables

Exercise 3: Working on a query in Design View

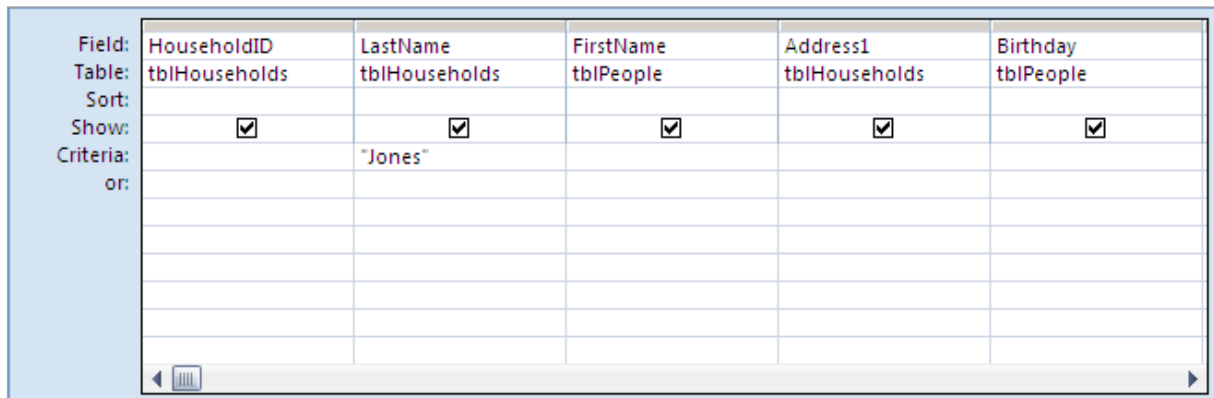
Now look at this exercise (page 59).

4 Criteria in a Query

Criteria can be set in the query design grid, to limit which records are included.

4.1 Criteria With Equals

On the row labelled **Criteria**, text or a number can be entered under the field to be limited. When the query is run, only those records which have this criteria value will be included in the results dataset presented.



The screenshot shows a Microsoft Access Query Design View. On the left, a vertical pane lists the fields: Field, Table, Sort, Show, Criteria, and or. The main area is a grid with five columns corresponding to the fields: HouseholdID, LastName, FirstName, Address1, and Birthday. The rows are: Field, Table, Sort, Show, Criteria, and or. The 'Criteria' row shows the text '"Jones"' entered under the LastName field. The 'or' row is empty. The 'Show' row has checkboxes checked for all five fields. The 'Table' row shows 'tblHouseholds' for HouseholdID, Address1, and Birthday, and 'tblPeople' for LastName and FirstName. The 'Field' row shows the field names. The 'Sort' row is empty. The 'Criteria' row shows '"Jones"' under LastName. The 'or' row is empty.

Field:	HouseholdID	LastName	FirstName	Address1	Birthday
Table:	tblHouseholds	tblHouseholds	tblPeople	tblHouseholds	tblPeople
Sort:					
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		"Jones"			
or:					

Figure 13 A Query with Text Criteria

4.2 Changing Criteria in a Query Design

Add further criteria by typing in the Criteria row, in other field columns.

Remove criteria by deleting in the Criteria row: click in the criteria expression and use DELETE or BACKSPACE.

4.3 Criteria With Comparison Operators

Calculated expressions can be used as criteria. Use <, >, <=, >= and **Between** for comparison expressions.

For example:

>t in a LastName field, to find those people whose surnames begin with t or later in the alphabet

>=18 in an Age field, to find only those who are of voting age

Complex expressions can be built, using (brackets) to maintain clarity.

Field:	Title	Surname	Forename	Address 1	
Table:	tblStudent	tblStudent	tblStudent	tblStudent	
Sort:					
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Criteria:			Between "ja" And "me"		
or:					

qryPeopleMiddleOfAlphabet				
Title	Surname	Forename	Address 1	
Mrs	Trueman	Mary	156 Station Road	
Ms	Jones	Lucy	183 Utttoxeter Road	
Mrs	Bowker	Jenny	10 Derwent Close	
Mr	Smyth	Jake	64 Zoo Lane	
*				

Figure 14 Criteria Using Comparison Operators

4.4 Criteria Using Null And Not

Expressions in query criteria may include the operator **Not**. For example, the query in Figure 15 will show all the data about any driving lessons which are to collect the student from some location other than their home address.

Query1			
<div> <div>tblLesson</div> <div> * Lesson Number Student ID Instructor ID Date Start Time Length of Lesson Collection Point Drop-Off Point LessonTypeID </div> </div>			
Field:	tblLesson.*	Collection Point	
Table:	tblLesson	tblLesson	
Sort:			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:		Not "home address"	
or:			

Figure 15 Criteria Expression Using Not

The **Is Null** operator finds records where no data has been entered in a field. This can be useful for tracing records which have not been completed properly.

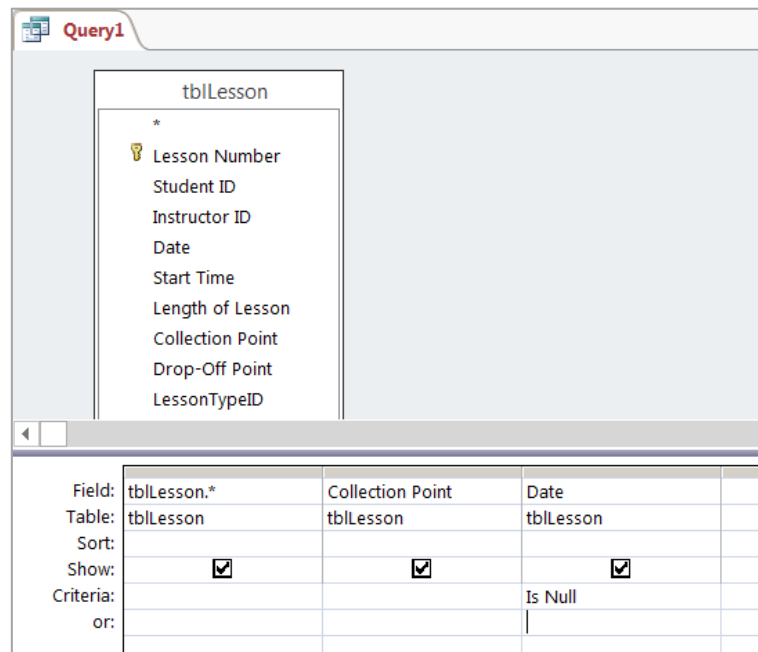


Figure 16 Criteria Using Is Null

4.5 Access Edits Punctuation in Criteria

Access may add punctuation to your criteria, such as “quote marks” around text or #hatch marks# around dates.

In the date example in Figure 17 below, # symbols would appear in the **BirthDay** expression, enclosing the date which Access has recognised. You can type the # symbols to enclose your dates if Access does not recognise them automatically.

Field:	HouseholdID	LastName	FirstName	Address1	BirthDay
Table:	tblHouseholds	tblHouseholds	tblPeople	tblHouseholds	tblPeople
Sort:					
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:					> #01/01/1960#
or:					

Figure 17 Setting Date Criteria

Note that Access will sometimes “correct” the punctuation or syntax of your expression, applying SQL conventions. When this happens, you should read the suggested version of the expression carefully, to make sure that Access has interpreted your intentions correctly.

Similarly, Access replaces some text expressions using the structure **Like “text-pattern”**. An expression **p*** (intended to find all records with a value beginning with p) is replaced with **Like “p*”**.

4.6 Wildcard Symbols in Criteria

Including wildcard symbols in an expression allows *Access* to select all values which partly match a given expression, but allowing some variation:

#	allows <i>Access</i> to match any single numerical digit
?	allows any single character or digit, at the position marked by ?
*	allows any characters or digits, in any quantity, at the position marked by *

For example:

J?lly	will find Jolly, Jelly, Jilly but not Jenelly
am*	will find amicable, American and am but not age nor a

4.7 Yes/No Criteria

In a field which has the Yes/No data type, criteria may be set. Enter **yes** or **no** in the criteria line. Alternatively, enter 0 (equivalent to **no**) but beware that the equivalent to **yes** is -1.

4.8 AND and OR in Query Criteria

Setting several criteria on the same row narrows down the results of a query: you are permitting only those records which satisfy all the criteria (this is a logical AND).

Field:	HouseholdID	LastName	FirstName	Address1	Birthday
Table:	tblHouseholds	tblHouseholds	tblPeople	tblHouseholds	tblPeople
Sort:					
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			Like "j*"		>#1/1/1960#
or:					

Figure 18 Multiple Criteria (AND)

Setting up criteria on separate rows widens the results of a query: you are permitting those records which satisfy any of the criteria (this is a logical OR).

Field:	HouseholdID	LastName	FirstName	Address1	Birthday
Table:	tblHouseholds	tblHouseholds	tblPeople	tblHouseholds	tblPeople
Sort:					
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:					>#1/1/1960#
or:			Like "j*"		

Figure 19 Multiple Criteria (OR)

Exercise 4: Setting criteria in a query

Now look at this exercise (page 61).

5 Editing the Query Design

5.1 Removing a Field

Click the field name at the top of a column, then delete using DELETE or BACKSPACE keys. Note you are only removing the field name from the query design, not deleting data from the underlying table.

5.2 Rearranging Columns

Select a field by clicking on the narrow grey bar at the top of the column. Then drag the column to the desired position and drop.

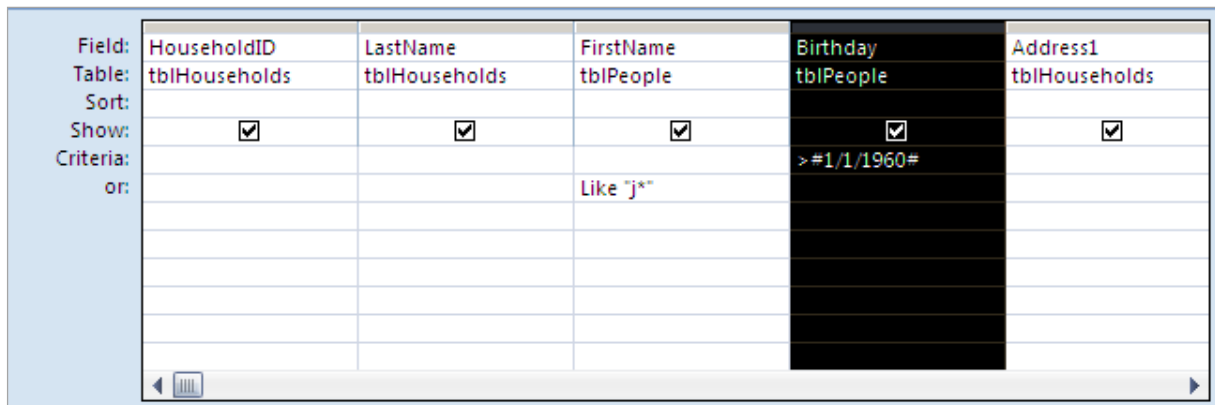


Figure 20 Selecting a Field

5.3 Adjusting Column Width

Drag the divider between two column headings to make one column wider or narrower. Note this changes the column width in Design View. For wider columns in Datasheet View, drag the column dividers there.

5.4 Hiding/Showing A Field Column

In the **Show** row, clear the checkbox under a field to omit that field from the query results. Check the box to show that field again.

5.5 Sorting Query Results

In the **Sort** row, choose **Ascending** or **Descending** in the column of the field you wish to sort by. When the query is run, the records will be presented sorted by that field value.

If more than one column has a Sort, then results will first be sorted by the leftmost marked field, then sorted by the next marked field, and so on working from left to right.

Field:	HouseholdID	LastName	FirstName	Birthday	Address1
Table:	tblHouseholds	tblHouseholds	tblPeople	tblPeople	tblHouseholds
Sort:		Ascending		Ascending	
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:					
or:					

Figure 21 Sorting Families by LastName and then by Birthday (Age)

5.6 Finishing

Close the query by clicking , saving when prompted.

The query name appears on the list in the Navigation Pane. Delete a query if necessary by selecting the query name and pressing DELETE, agreeing to the confirmation dialog.

Exercise 5: Rearranging a query

Now look at this exercise (page 64).

6 Parameter Queries

Criteria can be composed which allow some flexibility. Then when a user runs the query, they can specify to some extent which records are to be displayed. This is useful for a select query which is to be run repeatedly.

6.1 Entering Parameter Expressions as Criteria

An expression is entered in the Criteria row, but instead of specifying exact values in the expression you give a short prompt text in square brackets e.g. **[Tests after which date?]**.

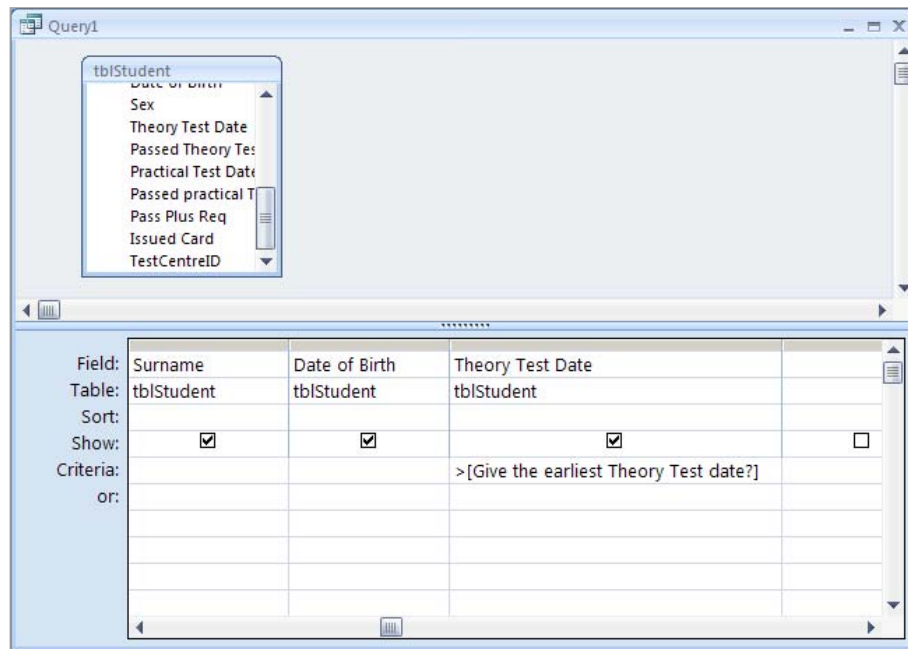


Figure 22 Select Query With Parameter Criteria

When such a query is run, *Access* is unable to calculate the expression in the Criteria row until the user has supplied a value. A dialog appears, showing the prompt text from the square brackets. When the user gives a value, this is used in the Criteria and the query is run.

This provides a flexible query, which the user may use repeatedly to display different subsets of records. The prompt text needs to be carefully phrased so the user understands what response is expected.

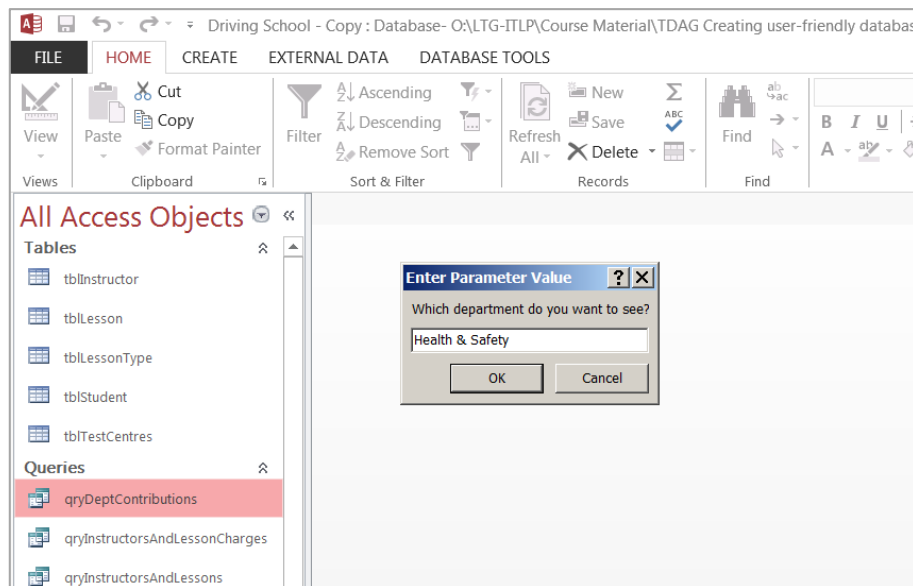


Figure 23 Running a Parameter Query

6.2 Parameter Query Problems

If criteria are entered in the query design grid which cannot be evaluated, for instance if a field name is mis-typed and does not match a real field name, then Access will interpret this as a parameter query and display the **Enter Parameter Value** dialog unexpectedly.

The prompt text shown in the Enter Parameter Value dialog is then the mis-typed field name or expression. This may puzzle the user but it is a clue to which part of the query design has an error.

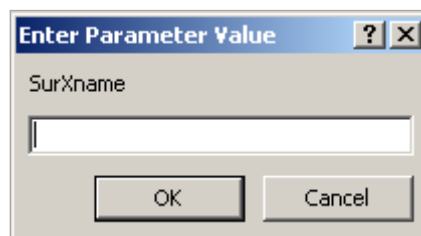


Figure 24 Effect of an Error in Query Design

6.3 Parameter Query Using Wildcards (*Optional*)

If a query is to be used repeatedly, it may be useful to allow users some flexibility in their answer to the parameter question. This can be done using wildcard symbols in parameter criteria. The formal SQL syntax must be used.

For example, the Criteria line might read **Like [Give initial letters]&"**"**. In this case, the word **Like** tells Access to seek all records which match the pattern given; the multiple-character wildcard symbol ***** is enclosed in "double quotes". It is joined, using **&**, to prompt text enclosed in [square brackets]. When the query is run, the prompt dialog appears. The user types one or more initial letters. The query returns all records with field values that start with those initial letters.

If the user types nothing in the prompt dialog, all records are shown in the results.

Exercise 6: Parameters in queries


Now look at this exercise (page 65).

7 Some Special Kinds of Query

7.1 Top Values (*Optional*)

A query usually displays all qualifying records. It can be set up to display only the top few – the top 10, or the bottom 5%, etc.

The data must first be sorted by the desired field, so that the top (or bottom) few records appear at the top of the datasheet. The Return Top Values control

 Return: 25% is then used to set the number of top values: either choose from the drop-down list or type a value or percentage.

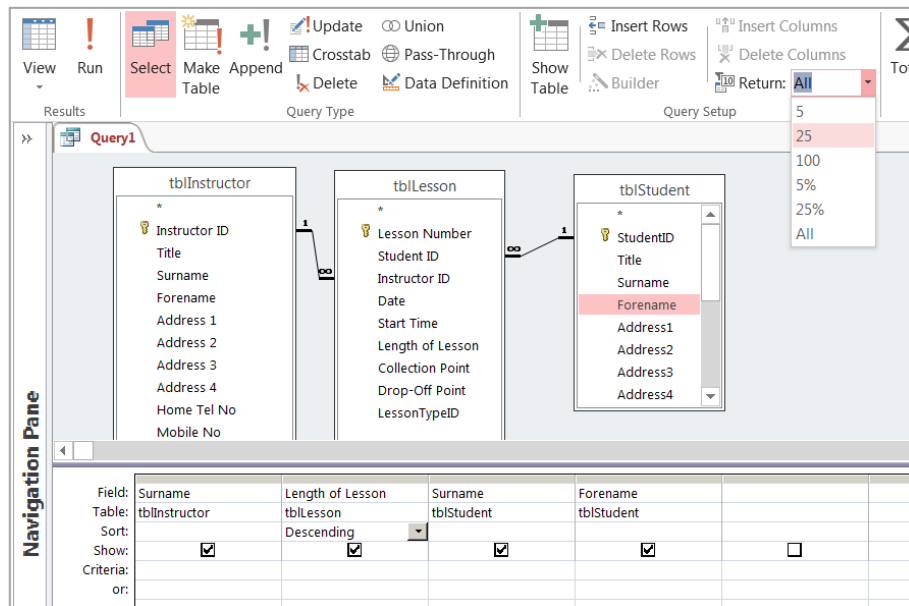


Figure 25 Displaying Only Top 25 of Lesson Lengths

When the query is run, only the top so-many records are displayed.


Alternatively, the bottom so-many records can be shown, by first displaying all records sorted in **Ascending** order, then choosing from the Return Top Values control.

Selecting **All** in the Top Values control will display all records.

7.2 Query to Find Duplicate Records

Access has a wizard for creating a query which finds duplicate or near-duplicate records. This is useful if it is suspected that two or more records have been entered for the same object, for example the same person has been entered twice in a contacts list.



In the **Create** tab, clicking  gives a choice of special query types available, including **Find Duplicates Query Wizard**. The first step on the wizard is to specify the table/query which is to be searched for duplicates. Then you nominate the fields which might contain duplicate information. At this point, you should choose just the minimum number of fields, to avoid including a field which has non-duplicate data.

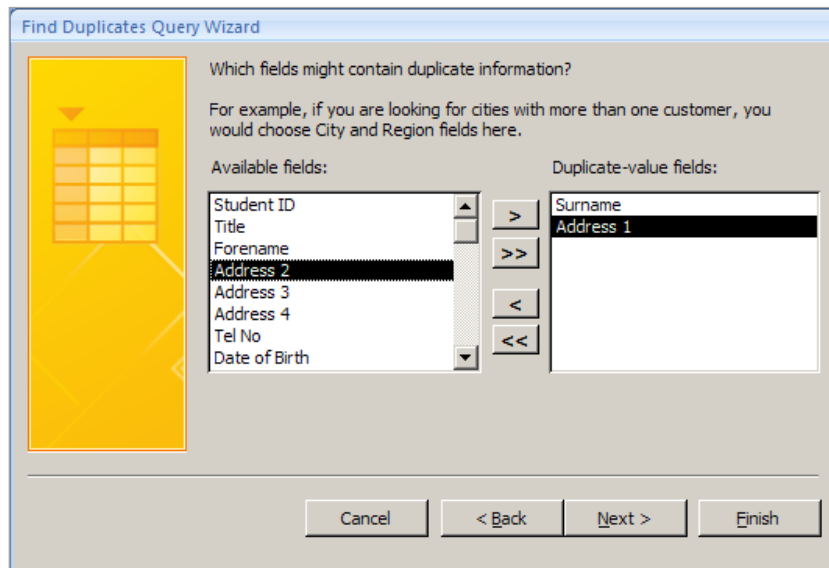


Figure 26 Wizard to Find Duplicate Records

Next you specify some other fields to be displayed. These would usually be fields whose data you expect to use for distinguishing between the “duplicate” records.

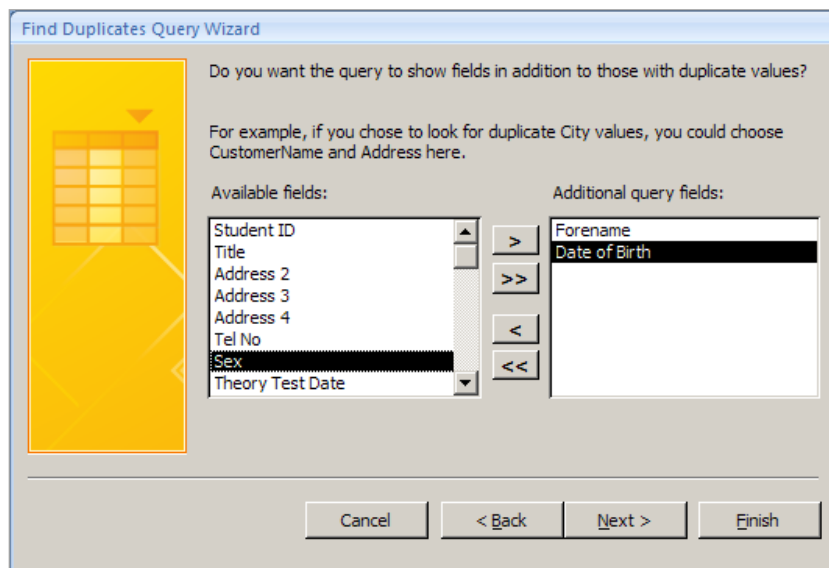


Figure 27 Wizard to Find Duplicate Records

When the query is run, the results show any records which have the same values in each of the chosen fields.

Surname	Address 1	Forename	Date of Birth
Beadwell	76 Maple Road	Steve	01/12/1982
Beadwell	76 Maple Road	Steve	
Jones	183 Uttoxeter Road	Alan	
Jones	183 Uttoxeter Road	Lucy	28/05/1982
Spencer	73 Mayfield Road	Anna	
Spencer	73 Mayfield Road	Emma	
Spencer	73 Mayfield Road	Victoria	20/12/1980

Figure 28 Results of the Find Duplicates Query


In Figure 28, Alan and Lucy Jones, who live at the same address, probably are two different people, but Steve Beadwell seems to have been entered twice. Now you would work through the records which appear to be duplicates, editing values and removing unwanted records.

7.3 Query to Show Unmatched Values

A table may include some records which do not correspond to any records in a related table. For instance, a patient may be included on a dentists' database, in a table showing patients' name and address details, but may not yet have made an appointment to visit. A query based on these two tables would by default show only those records which appear in both tables, and the new patient would be hidden until their first appointment was made.

Access has a wizard for building a query which shows these unmatched values.



In the **Create** tab, clicking  gives a choice of query types available, including **Find Unmatched Query Wizard**. The first step on the wizard is to specify the table or query containing the records you want shown.

The query you create will list records in the table you select below that have no related records in the table you select on the next screen. For example, you can find customers that have no orders.

Which table or query contains records you want in the query results?

Table: tblAssets
Table: tblCostingCategories
Table: tblDepartments
Table: tblResources
Table: tblResourcesLinked

View
☒ Tables ☐ Queries ☐ Both

Cancel < Back Next > Finish

Figure 29 Wizard for a Find Unmatched Query (1)

Next you specify which table contains the related data.

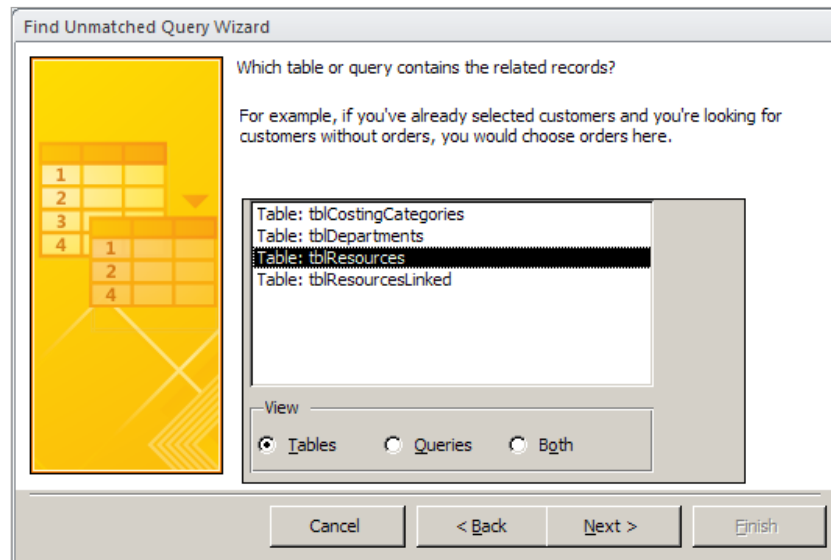


Figure 30 Wizard for a Find Unmatched Query (2)

In the next step, *Access* proposes the fields in the two tables which may form the relationship. If the suggestions are unsuitable, you choose different field names. Note that if the suggestions here are completely unexpected, you should abandon this wizard and review the relationships that are defined for the database.

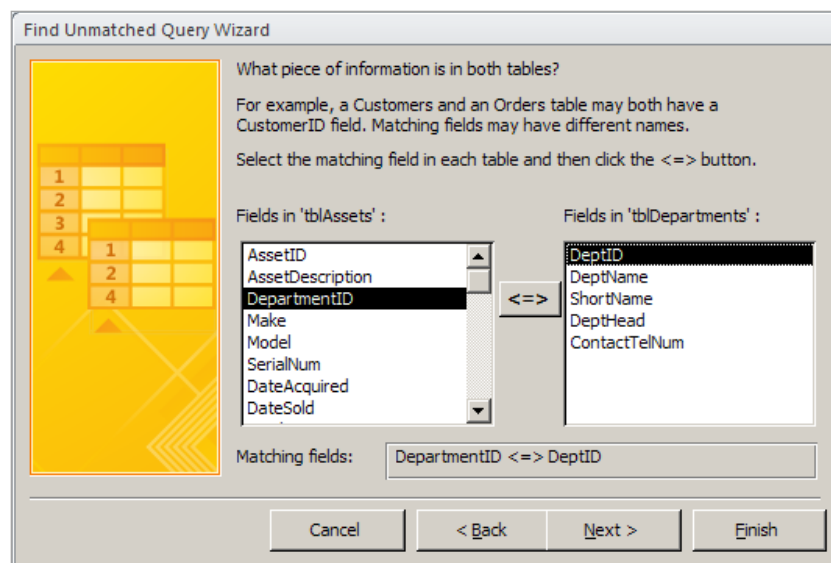


Figure 31 Wizard for a Find Unmatched Query (3)

Next you specify the fields which are to be shown in the results. These should be the fields that you need to recognise the unmatched records.

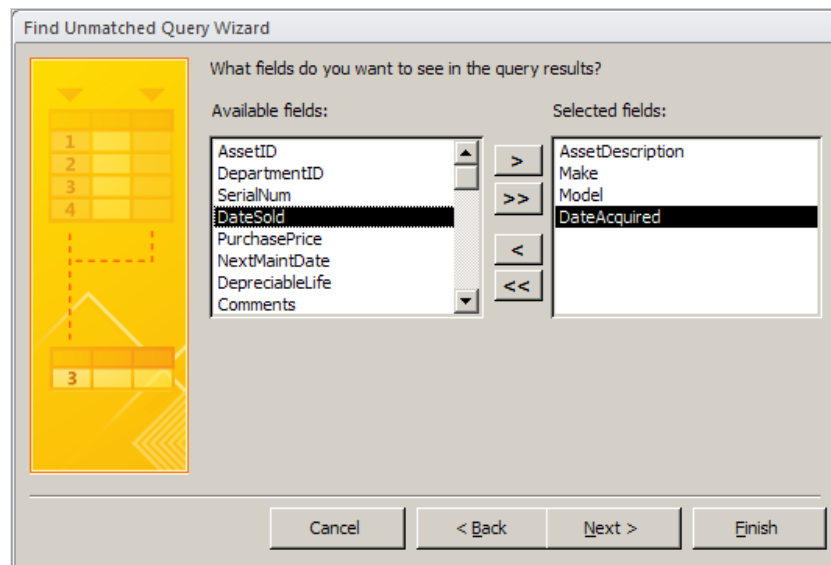
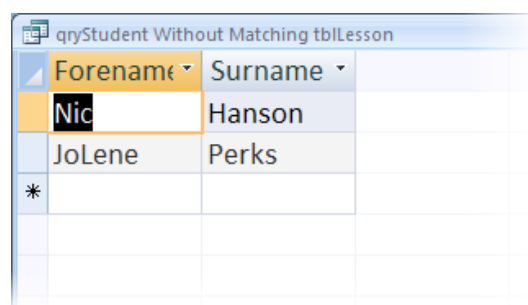


Figure 32 Wizard for a Find Unmatched Query (4)

Finally you name the query, and the results are presented. Records in the first table are listed if they do not find a matching record in the second table, with only the selected fields shown.



Forename	Surname
Nic	Hanson
JoLene	Perks
*	

Figure 33 Results of a Find Unmatched Query

Exercise 7: Special types of query

Now look at this exercise (page 67).

7.4 Crosstab Query - What Is It For?

A crosstab query displays calculated values, broken down in terms of two or more field categories, in a grid layout.

Where numerical data can be grouped by two fields, the values may be grouped and aggregated (as a Sum, Average, Max, Count etc.), then displayed in a two-way table known as a crosstab. For example, Figure 34 shows a simple list of exam results where a number of students sat various end-of-term exams, with the name of each student's tutor.


TutorLastName	SubjectName	Score	student
Hargreaves	maths	82	Robert Brammer
Shelley	maths	60	Michael Beswood
Belmont	maths	91	Charlotte Williams
Belmont	maths	53	Hugh Spencer
Belmont	maths	92	Charlotte Williams
Hargreaves	maths	56	Victoria Spender
Shelley	eng lit	98	Lucy Bowker
Hargreaves	eng lit	89	Greg Watson
Shelley	eng lit	79	Michael Beswood
Shelley	eng lit	86	Michael Beswood
Belmont	eng lit	92	Hugh Spencer
Belmont	eng lit	92	Jenny Bowker
Shelley	eng lit	93	Lucy Bowker
Shelley	eng lit	97	Steven Jenkins
Hargreaves	eng lit	58	Victoria Spender
Shelley	geog	57	Lucy Bowker
Hargreaves	geog	57	David Windsor
Belmont	geog	96	Jenny Bowker
Shelley	geog	98	Steven Jenkins
Belmont	geog	54	Sarah Fowler
Belmont	geog	56	Charlotte Williams

Figure 34 Student Exam Result Data

A crosstab query would make this data easier to understand and analyse.

7.5 Creating a Crosstab Query



Clicking  offers query options, including the **Crosstab Query Wizard**. First you choose a table or query to base the query on. Then choose up to three fields to provide the row headings.

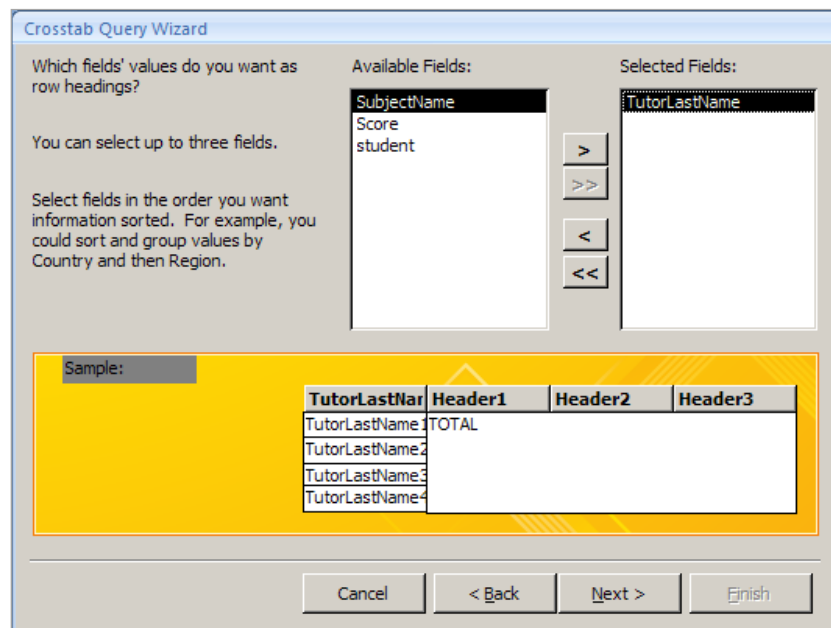


Figure 35 Crosstab Query Wizard (1)

Next you choose the field which is to provide the column headings.

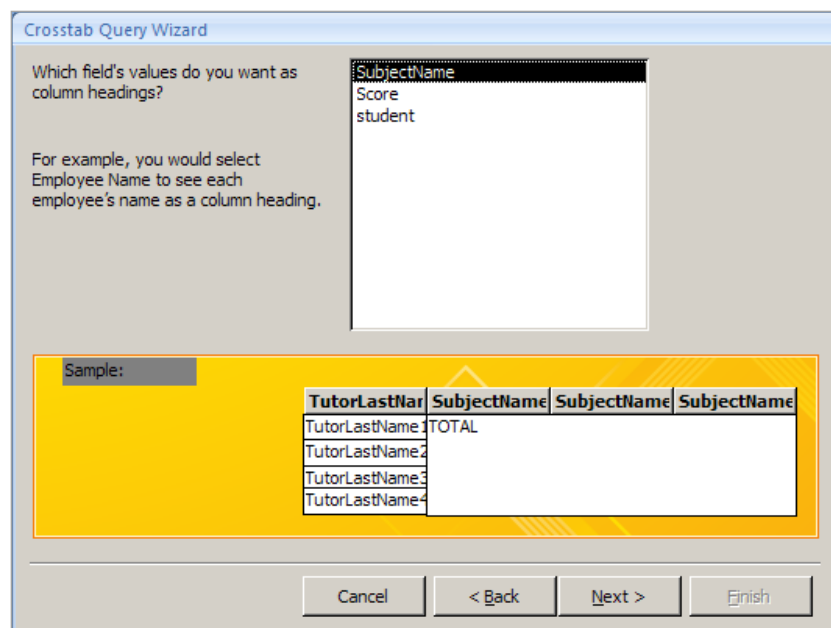


Figure 36 Crosstab Query Wizard (2)

Next you choose which field, usually a numerical field, is to fill the cells of the table. Several functions are offered, for calculating the combined field values.

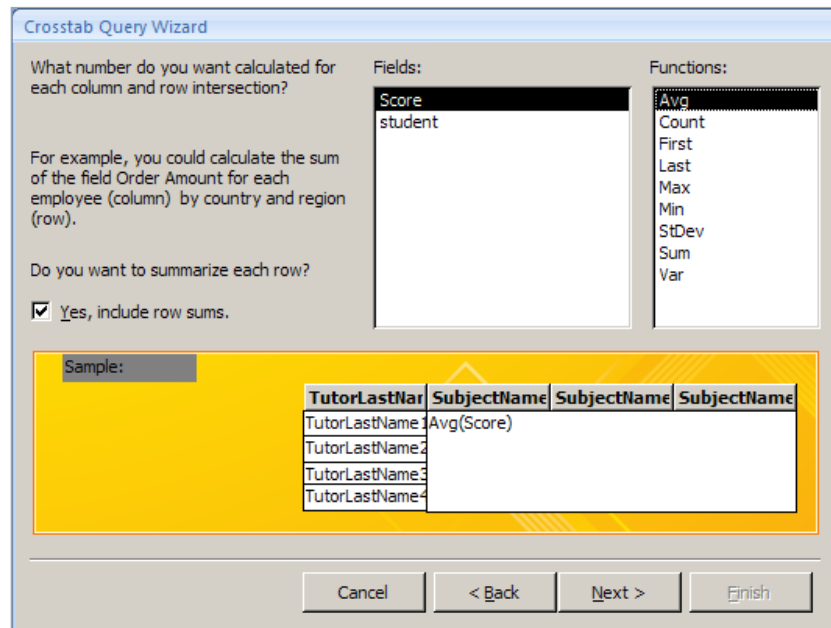


Figure 37 Crosstab Query Wizard (3)

When the finished query is run, the data is collected from the underlying tables/queries and presented in a table.

	TutorLastName	Average Of Sco	biology	chemistry	economics	eng lit	
►	Belmont	76		69	74	92	
	Hargreaves	68	66	73	71	89	
	Shelley	74	72		73	91	

Figure 38 Crosstab Query Results

Exercise 8: Crosstab query (*Optional*)

Now look at this exercise (page 69).

8 Calculations and Functions In A Query

Sophisticated calculation expressions can be built into criteria, using a range of operators and functions. The way these are put together is similar to that used in spreadsheets such as *Excel*.

8.1 Calculated Columns In a Query

A column in a query may contain an expression, rather than a simple field name. A calculation can be typed in the column heading, made up using field names and arithmetic operators. Any field name that has spaces must be enclosed in [square brackets].

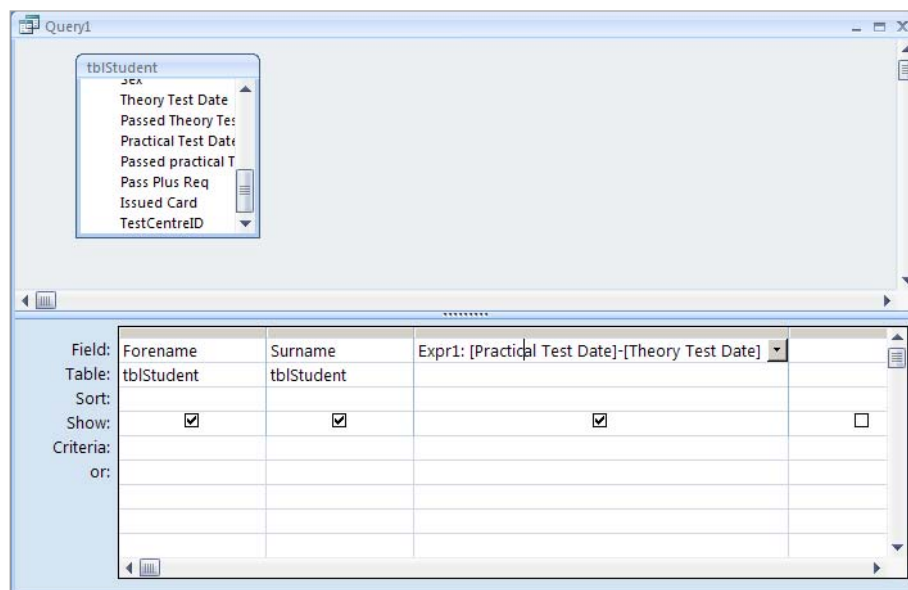


Figure 39 A Calculation in a Query

When the query is run, *Access* evaluates the expression for each record, and shows the results in the dataset.

Access inserts a generic column heading such as **Expr1** before the expression. You may prefer to replace this with a descriptive heading – ensure that the colon: remains as the separator before the expression itself.

When editing an expression, it may be helpful to press **SHIFT+F2** to display the expression in a zoom window.

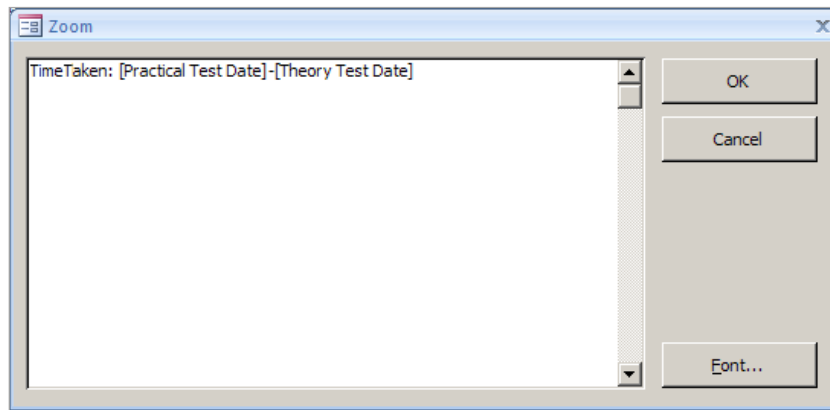


Figure 40 Editing an Expression, Zoomed

8.2 Operators in Calculations


The usual operators used in an expression are + - * / and ^. Brackets clarify how a complex expression is to be evaluated.

Expressions may also use logical operators AND, OR and NOT.

Text can be manipulated, for instance using & to concatenate pieces of text. It may also be useful to insert spaces or punctuation between the field names, each enclosed in "double quote marks". For example
FullName:[First Name]&" "&[Last Name]

8.3 "Totals" in Queries – Grouping Records Together

Records may be grouped together, so that all those sharing the same value in a particular field are combined. Grouped data is then presented using aggregate functions such as SUM, AVG, and MAX etc.

In Query Design view, clicking the Totals button  displays the **Total** row in the grid. One field is chosen for **Grouping By** – in the example shown, students are grouped by tutor. All other fields included in the query *must* now have one of the aggregate functions chosen from the drop-down list on the **Total** row.

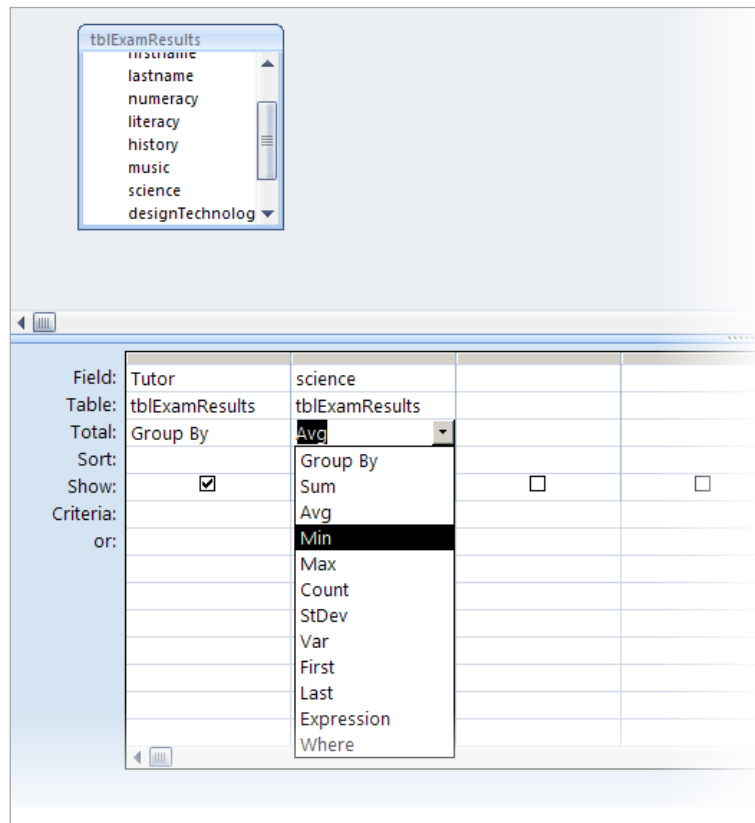


Figure 41 Grouping Students by Tutor, Showing the Average Science Score

When this query is run, *Access* collects all records having the same value in the **Group By** field. It calculates the chosen function, using values from the chosen field. A result is displayed for each available field value.

Tutor	AvgOfscienc	AvgOfhistor	AvgOfmusic
Hargreaves	91	31	56
Salmer	83	77	80
Trueman	82	71	98

Figure 42 Results of a Grouped (Totalled) Query

When building a query with grouping, it is good practice to have just the minimum number of columns showing, as extra fields may confuse the grouping (not to mention confusing the owner).

Exercise 9: Calculations in queries

Now look at this exercise (page 69).

9 Calculated Controls in a Form

9.1 A Note on Views of Forms

Many of the activities discussed in this chapter can be carried out equally in Layout View or Design View. In general, Design View offers more flexibility, so we will use that in most of our illustrations, but you may like to try similar procedures in Layout View.

9.2 Bound and Unbound Controls

Typically, a control in a form is *bound* to a field in the underlying table/query. When the form is opened, the current field value is displayed in a type-in text box, where the user may edit or delete the value. The procedure for creating such a form with controls is discussed in other courses in this series, described in Part 13.1.

An *unbound* control is one which does not read its value directly from table data. This may be useful if the control source is a calculation: an expression is evaluated when the form opens, taking some values from fields or functions and calculating a derived value. The result is displayed in a box in the form and cannot usually be altered by the user.

9.3 Creating a Control With A Calculation

Working in a form in Design View, you can add an unbound control by clicking



then clicking in the form.

An expression can be typed into the control itself, as required. The expression must begin with equals =. Expressions may use operators such as + - * / and brackets, and may be based upon field names from the table or query that underlies the form (any field names should be enclosed in [square brackets]). An expression may also use logic operators such as AND and OR, and text may be concatenated using &.

The label associated with the control may be edited or deleted altogether (it appears at first with unhelpful default label text such as **Text11:**). The control and its associated label can be moved and resized on the form.

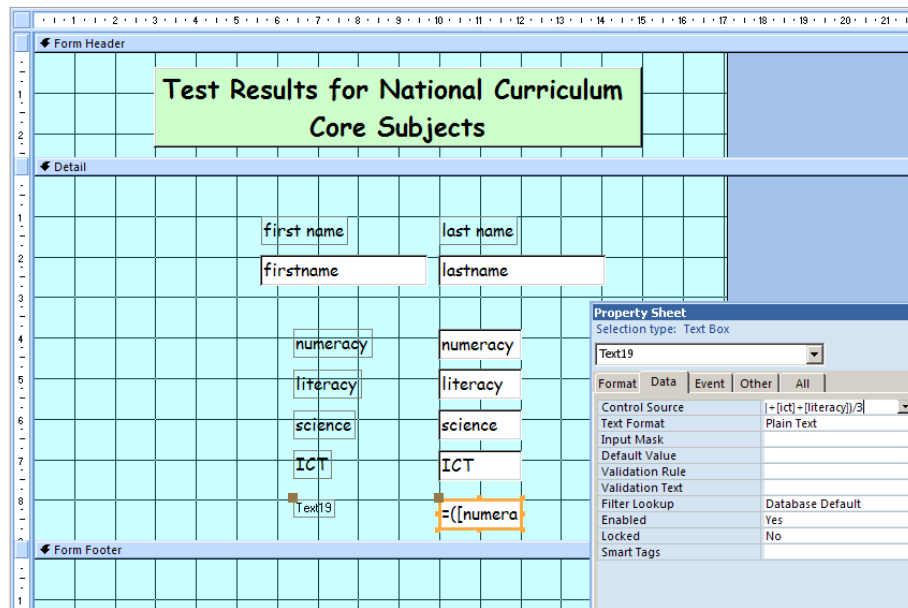



Figure 43 An Unbound Control With a Calculation

9.4 More Built-In Functions *(Optional)*

The Expression Builder offers a library of ready-programmed functions. It is usually available in places where you might type an expression directly.

With an unbound control selected in the form's or report's Design View, the **Control Source** is one of the properties given in the Property Sheet, and this can be created using the Expression Builder. When you click in this property,  appears, which will display the Expression Builder window.

An expression is built up by typing suitable field names and operators. Instead of typing, popular operators including +/÷/* can be found by choosing Operators in the left-hand list of Elements. & is used to concatenate text. Note that field names can also be chosen from the lists of **Categories** found via **Tables** or the current form/report in the left-most list.

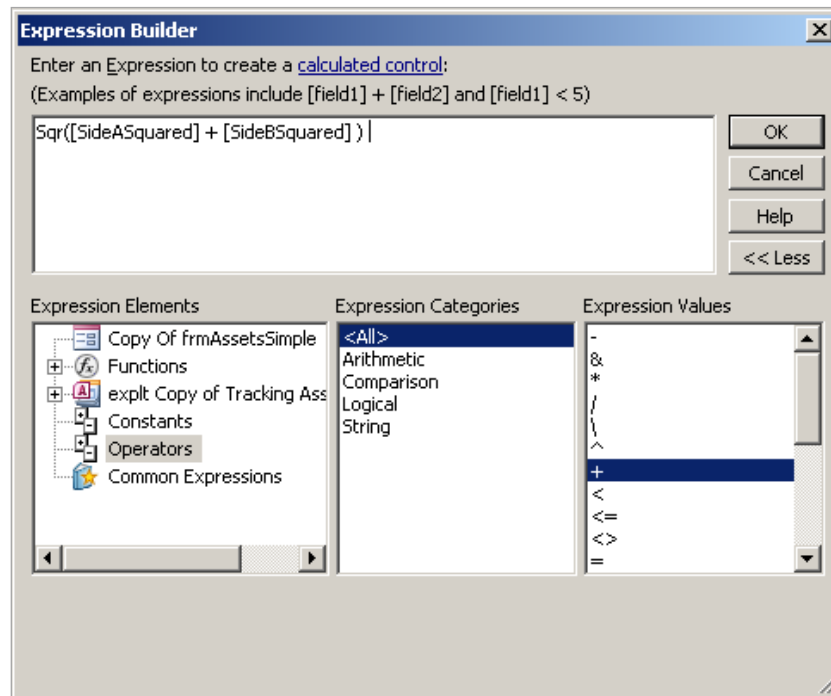


Figure 44 Building an Expression

Choosing **Functions**, then **Built-In Functions**, in the left-most column reveals a choice of categories in the middle column. When a category is chosen, all the relevant functions are listed in the right-most column.

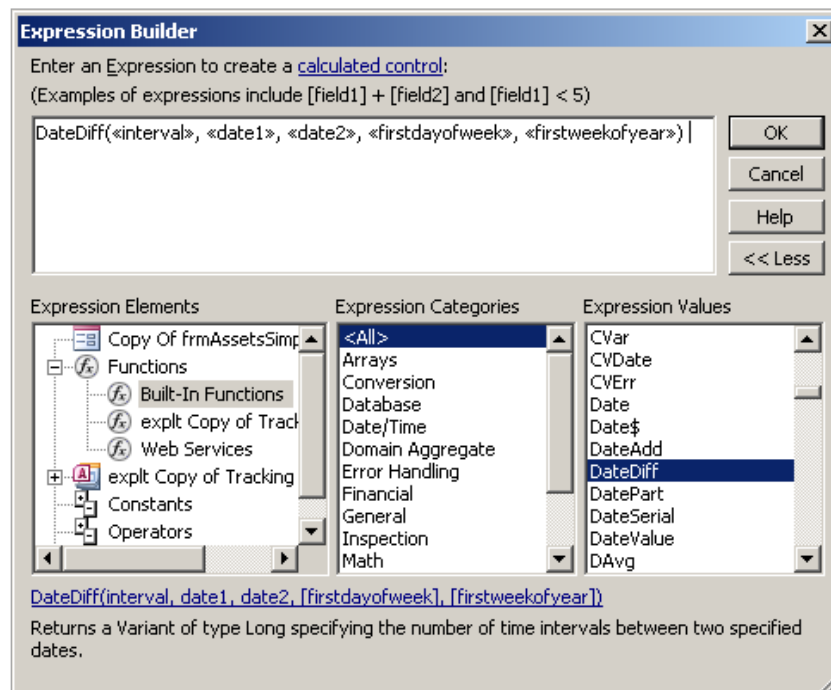



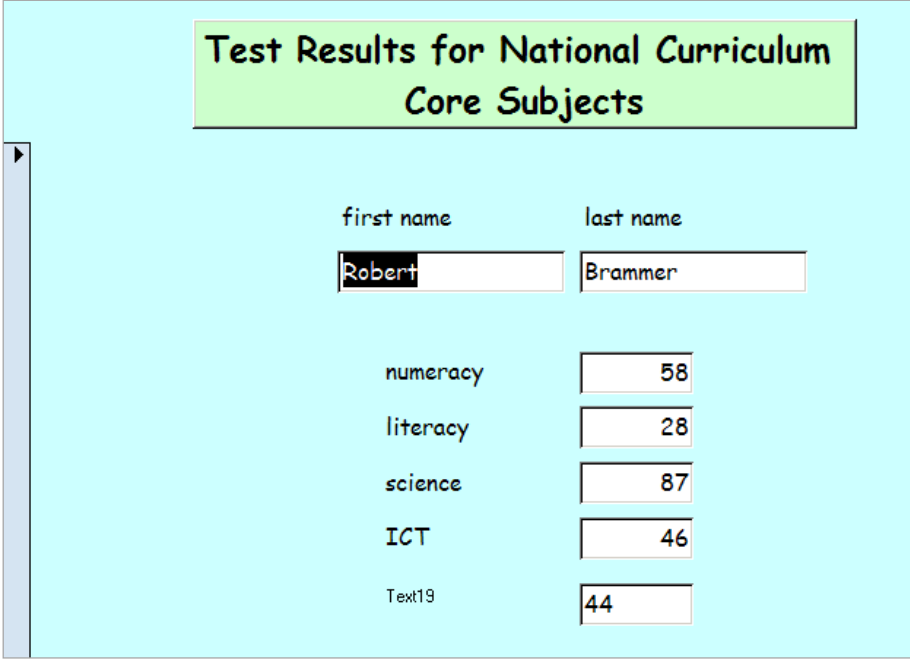
Figure 45 Built-In Functions Listed in the Expression Builder

Double-clicking the desired function name will insert the function in the box at the top of the Expression Builder.

When ready, clicking  will insert the expression from the Expression Builder into a control in the report.

9.5 Using Calculated Form Controls

When the form is opened, *Access* retrieves current data from the underlying table/query, and evaluates the expression for each record. The result for the current record is displayed in the unbound control.



first name	last name
Robert	Brammer
numeracy	58
literacy	28
science	87
ICT	46
Text19	44

Figure 46 Calculated Control

Because the value displayed in the unbound control has been derived from a calculation, the user is usually prevented from editing or over-writing the value.

Exercise 10: Calculations on a form

Now look at this exercise (page 71).

10 Calculated Controls in a Report

10.1 A Note on Views of Reports


Many of the activities discussed in this chapter can be carried out equally in Layout View or Design View. In general, Design View offers more flexibility, so we will use that in most of our illustrations, but you may like to try similar activities in Layout View.

10.2 Bound and Unbound Controls in a report

Typically, a control in a report is **bound** to a field in the underlying table/query. When the report is run, the latest field value is displayed. The procedure for creating such report controls is discussed in other courses in this series, listed in Part 13.1.

An **unbound** control is one which does not read its value directly from table data. This is relevant if the control source is a calculation: an expression is evaluated when the report runs, taking some values from fields or functions and calculating a derived value. The result is displayed in a box in the report.

10.3 Creating a Calculated Control

An unbound text box can be added to a report, in Design View, by clicking  then clicking at a suitable position in the report. An expression is then typed into the text box.

The expression must begin with equals = and may use operators such as + - * / and brackets. It may be based upon field names from the table or query that underlies the report (field names should be enclosed in [square brackets]). An expression may also use logic operators such as AND and OR, and text may be concatenated using &.

For more complex expressions, the Expression Builder may be helpful (described in Part 9.4).

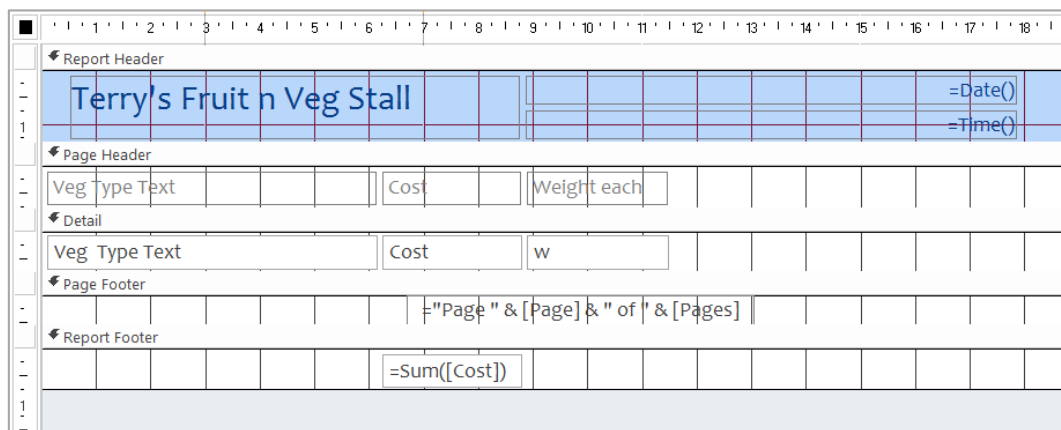


Figure 47 Adding an Unbound Control With A Calculation

The label associated with the control may be edited or deleted altogether (it appears at first with unhelpful default label text such as **Text11:**). The control and its associated label can be moved and resized on the report.

When the report is run, *Access* retrieves the data needed from the underlying table(s) and evaluates the expression, displaying the result for each record.



ItemName	Quantity	Price	
apple	6	£0.30	£1.80
clementine	5	£0.25	£1.25
banana	8	£0.45	£3.60
carrot	10	£0.20	£2.00
cabbage	1	£0.60	£0.60
lettuce	1	£0.75	£0.75
pumpkin	1	£2.50	£2.50
tomato	8	£0.30	£2.40


Page 1 of 1

Figure 48 Report With a Calculated Control

10.4 Formatting a Calculated Control

The **format** of an individual control determines the way the data is presented, such as the number of decimal places for a number. This formatting change does not affect the value stored in the underlying table. Standard formats available include percentage, euro currency and various kinds of date and time.

In Design View, when a control is selected (click it once so that an orange border

appears), its properties can be displayed using . In the **Format** tab of the Property Sheet, a **Format** can be chosen from the drop-down, which will affect the selected object.

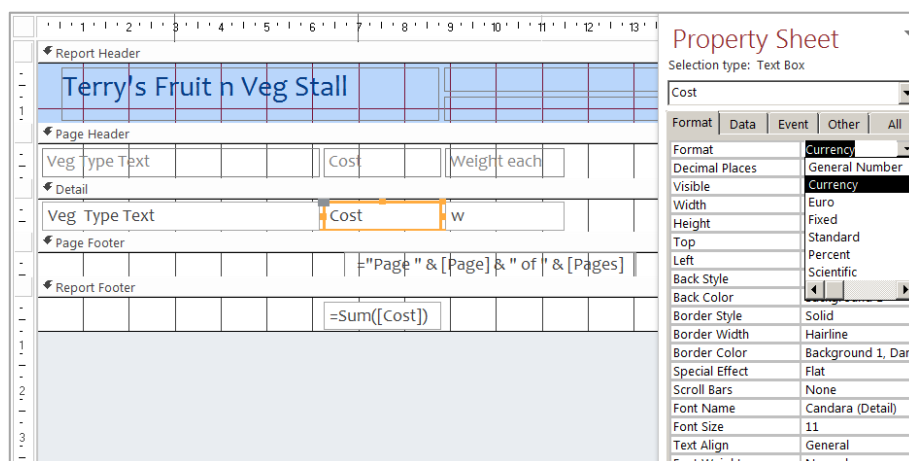


Figure 49 Setting the Format of a Control

10.5 Running Sum Format (*Optional*)

An interesting format which may be relevant to calculated controls (and indeed to any numerical field control) is **Running Sum**, found on the **Data** tab of the Property Sheet. If this is set to **Over All**, then the control will display the total of values in the specified field, summed over all records up to and including the current record.

10.6 Summarising “Totals” in Reports

A control may calculate summary values, evaluated using data from all records, for example an overall total or count. This is typically an unbound control placed in a report footer/header.

Summarising calculations may have been set up when the report was created, if the wizard was used.

On an existing report, some popular summary calculations can be added quickly. With the field control selected that is to be summarised, clicking **Σ Totals** offers relevant functions, such as AVG, SUM, MAX for numerical data and COUNT for any data including text. A calculated control appears in the report footer, and you may wish to create a label to explain it. A footer appears, if necessary.

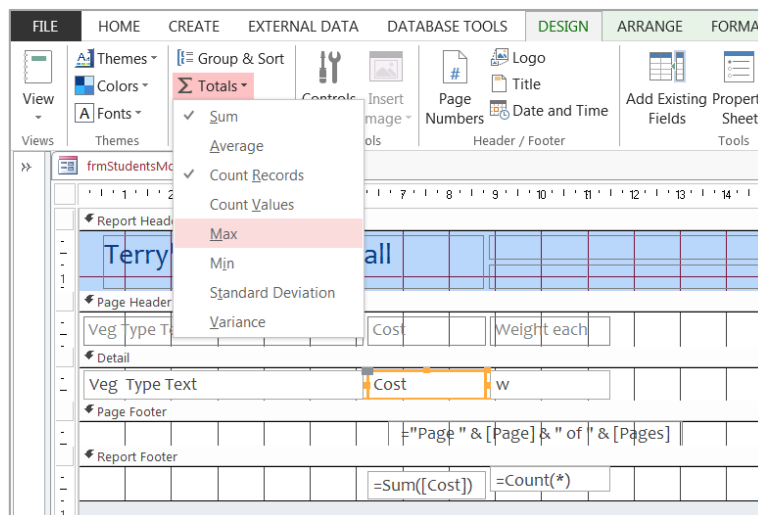


Figure 50 Adding a Summarising Calculation Using the Totals Button

For more variety of expressions, add an unbound control to the Report Footer and type or build an expression in its **Control Source**, as described in Part 10.3. A wide range of pre-programmed functions are available; these can be found through the Expression Builder.

When the report is run, Access evaluates the expression using data values from all the records, and displays the result in a footer at the bottom of the report (or at the end of the group).

Terrys Fruit and Veg Stall			
ItemName	Quantity	Price	
apple	6	£0.30	£1.80
clementine	5	£0.25	£1.25
banana	8	£0.45	£3.60
carrot	10	£0.20	£2.00
cabbage	1	£0.60	£0.60
lettuce	1	£0.75	£0.75
pumpkin	1	£2.50	£2.50
tomato	8	£0.30	£2.40
Number of items bought	40		

Figure 51 Report With a Summary Calculation

Exercise 11: Calculations in a report


Now look at this exercise (page 72).

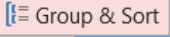
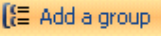
10.7 Calculations In a Grouped Report (*Optional*)

The position where a summarising control is placed is important for the result that will be calculated. For example, if a SUM control is placed in the Report Footer, then the result will be the total including all the records in the whole report. If the data has been grouped, then a SUM control in the Group Footer will calculate a separate total for each group.

Report Header									
Lessons And Students									
Page Header									
Lesson Length	Surname	Forename	Date		Start Time	Collection Point	Drop-Off Point	Lesson Type	
Length of Lesson Header									
h of L	hr								
Detail									
	Surname	Forename	Date		Start T	Collection Poin	Drop-Off Point	Lesson Typ	
Length of Lesson Footer									
Number of lessons	=Count([lesson number])								
Page Footer									
=Now()					="Page " & [Page] & " of " & [Pages]				
Report Footer									

Figure 52 Summary Calculation in a Group Footer

A quick way to achieve this is to use  **Totals** as described above. However, more options are available by using the Group, Sort and Total Pane (this was discussed in a previous IT Learning Programme course, see Part 13.1).

 **Group & Sort** displays the Group, Sort and Total pane at the bottom of the Design View.  **Add a group** will group the data according to a chosen field, inserting an additional header or footer for the boundary between values of that grouping field.

You will probably need to move the field control for the grouping field from the Detail section into the new Group Header (or perhaps Footer), as shown in Figure 53. If the control cannot be dragged then use **Cut** and **Paste**.

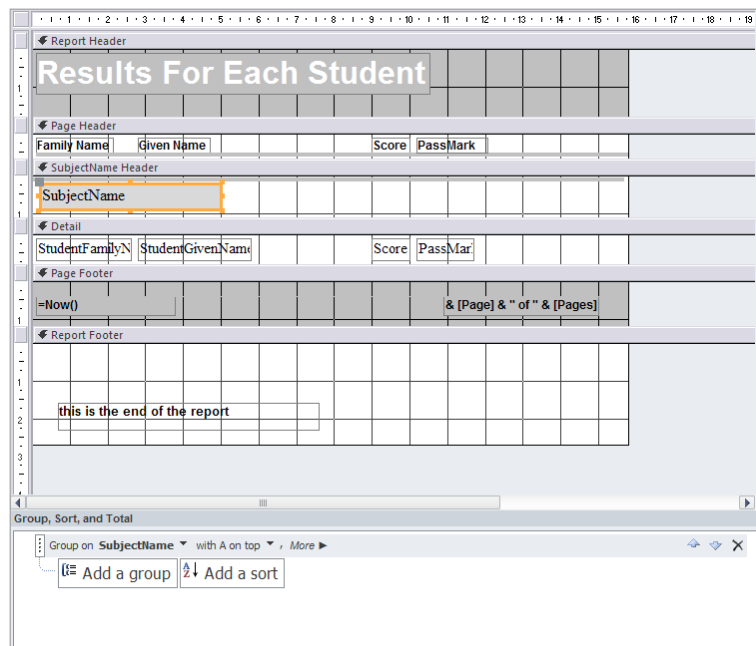


Figure 53 A Grouped Report


 **More**, on a Group bar, displays further details and settings for that group. These include **Totals**. The **Type** control offers functions which are applicable to the field chosen in **Total On**, such as Count for text fields and Sum, Average, Minimum for numerical fields.


Figure 54 Setting a Total on a Group

A calculated control is added to the group Header or Footer, where you can format and label it.

Exercise 12: Summary calculations on a grouped report (*Optional*)

Now look at this exercise (page 73).

11 Conditional Formatting For Controls

Conditional formatting  may be useful in forms and reports, especially to draw attention to individual data values that are exceptional.

With conditional formatting, a control appears with different formatting record by record, depending on the value of the control contents. For instance, the text colour might be set to be green whenever the value in the field represents a “pass” (score exceeds, say, 60%) and to be red if the value represents a “fail”. Formatting the controls like this could help important data stand out on the form.

Enabling: there is a checkbox amongst the conditional formatting to make a control Enabled or Not, which controls whether a user can change the value. However, if you want a control to become hidden or **Not Visible** you need a macro that sets a property (read about creating macros in another course, see Part 13.2 below).

A control may have several conditional formatting rules set up on it. Access will work through the rules in the order shown, applying each format as appropriate.

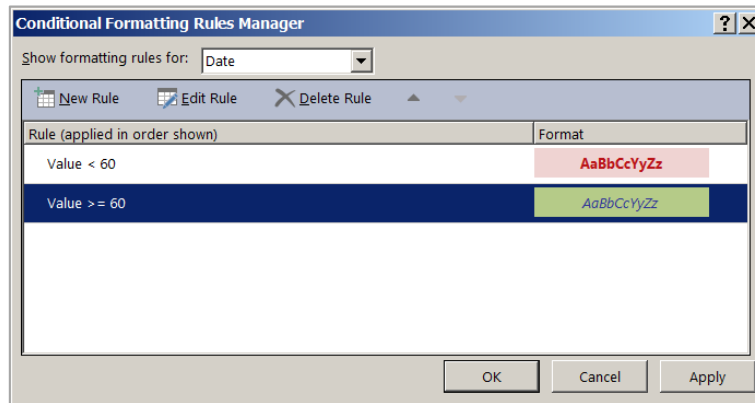


Figure 55 Conditional Formatting for a Control

An alternative kind of conditional formatting presents the current value using a shaded bar, indicating the current value compared with the set of values for all the records. When creating a new formatting rule, select **Compare to other records**. Now you indicate values for the shortest bar and the longest bar (usually let Access determine the lowest and highest values in the data set).

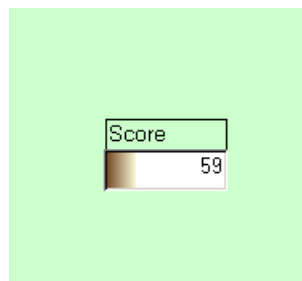


Figure 56 A data control with conditional formatting data bar (this score of 59 is one of the lowest in the dataset)

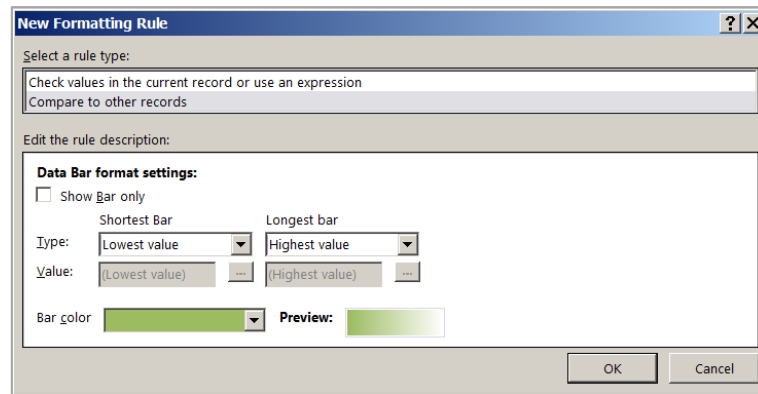


Figure 57 Setting a rule that compares the current value with other records

Exercise 13: Conditional formatting for controls in a form

Now look at this exercise (page 75).

Note that this exercise works with controls on a form, however similar effects can be done on a report.

12 Action Queries

This sort of query is used to make changes to data in tables. Data may be deleted or edited, new values or new records may be appended, following a specification saved in the query design.

Typically a select query is designed first, to display only the required fields and only selected records using criteria, from one or more tables. A precaution would be to run the select query now, to confirm that the intended data has indeed been selected. Once the correct collection of data has been specified, the query is converted to an action query which makes changes to the data values.

Once the query has been designed, it is usually saved (with a conventional query name). Then it can be run repeatedly as needed. This may form part of a procedure for maintaining a database, copying, removing or appending data from time to time.

12.1 Update Values Query

This type of query updates values in records, replacing old values with new values.

Starting from a select query (already made up, specifying only some fields from one or more tables, and specifying criteria to select only some records), the Design tab has buttons for other query types.



Clicking **Update** displays an extra row **Update To** on the query design grid. Here a replacement value (or expression) can be entered in one or more field columns.

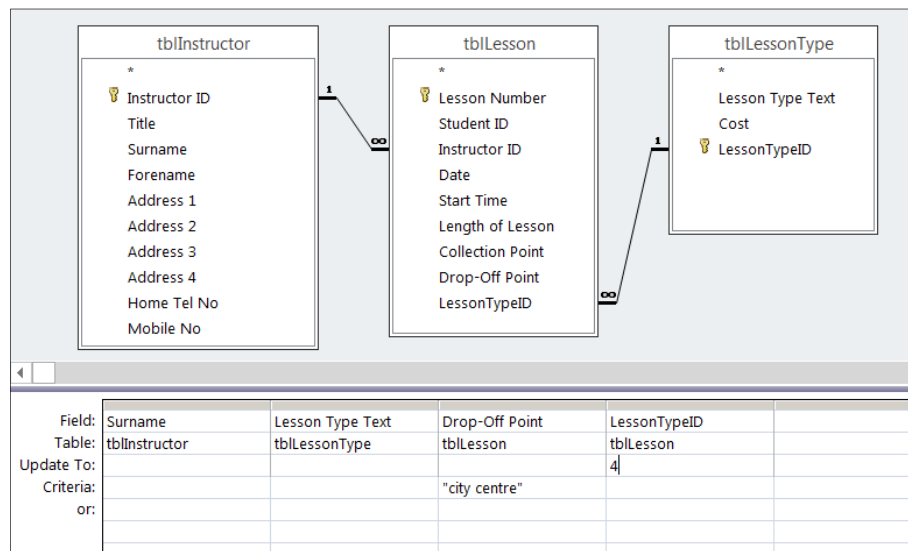


Figure 58 Setting up Criteria and Update Values in a Query (any lessons that drop off in city are designated lesson type 4)



Clicking **Run** runs the update query: Access looks in the table for the fields shown and the records which satisfy the criteria. The new values given in **Update To** are entered in the table, replacing any previous values. A confirmation dialog appears, where you can cancel the operation if necessary before it completes.

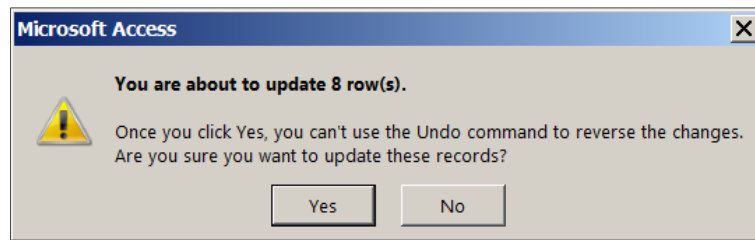


Figure 59 Confirmation Dialog For Update Query

The **Update To** value may be a simple number or text (as in the example of Figure 58) or it may be an expression which depends on the previous value. For example, in a Salary field the **Update To** expression might be **Salary*1.1** (this would give each selected employee a 10% pay rise).

When an update query is saved, it appears with a special icon in the Navigation Pane.

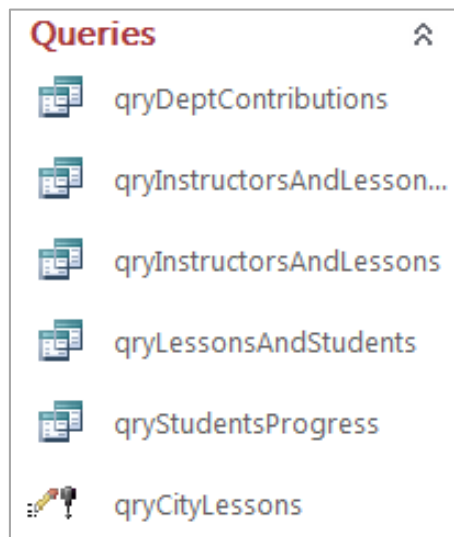



Figure 60 Action Query Icon

Alternatively, a query can be run by selecting the query name, then pressing ENTER. Running an update query several times from the Navigation Pane *will update the values repeatedly*, for instance increasing an employee's salary several times. So update queries should be treated with caution.

12.2 Delete Records Query

A select query which selects only some records may be converted into a Delete

Query by choosing  from the **Design** tab of the ribbon.

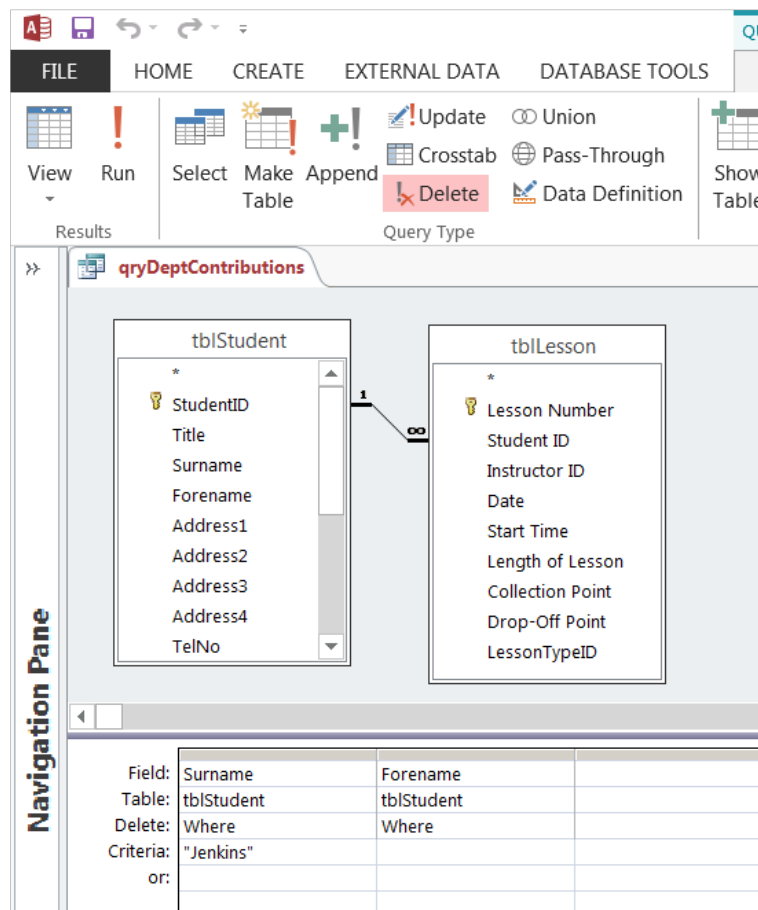


Figure 61 Delete Query

This sort of query should be used with caution, as running a Delete Query *does delete records* from the underlying tables.

12.3 Make-Table Query

Part of a table may be removed to form a new separate table, perhaps as a data archive. This would usually be data selected using criteria.



When you choose **Make Table**, a dialog allows you to give a name for the new table. The new table may be in the current database or in another database file.



Clicking **Run** runs the Make-Table query: a new table is built, with the fields and records specified. The new table then appears in the list of tables in the Navigation Pane.

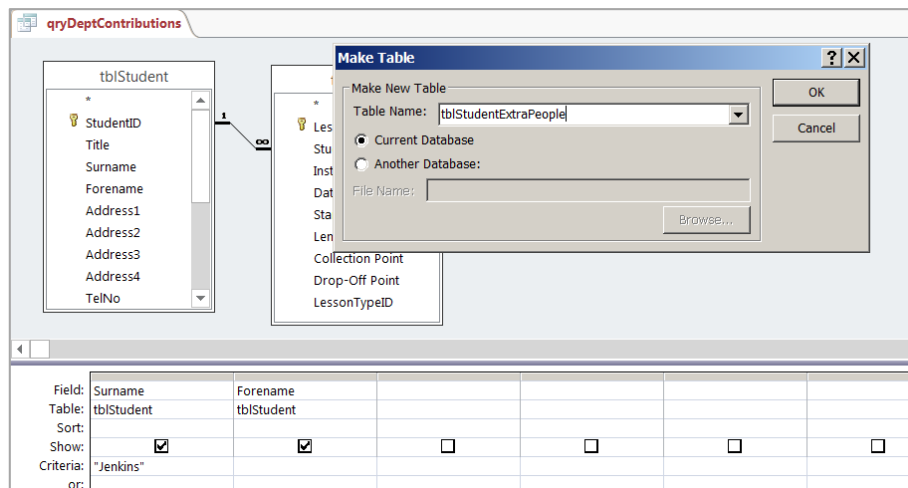


Figure 62 Running a Make-Table Query

12.4 Append New Records Query

This type of query copies selected data from the select query shown, adding the data as new records to another existing table.



Clicking **Append** displays the **Append** dialog. Here you specify the destination table (either in the same database or in some other database file) where the selected data is to be appended.

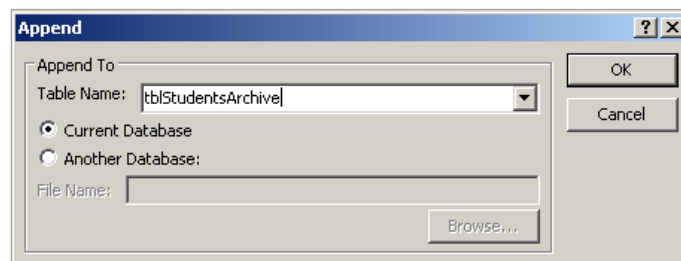


Figure 63 Nominating a Table for Appending Records

For each field shown, you now specify the field in the nominated other table where the selected data is to be appended. If field names in the two tables match, then Access can readily assign which ones correspond. If the field names are different, you must use the drop-down list to specify which field is to be used.

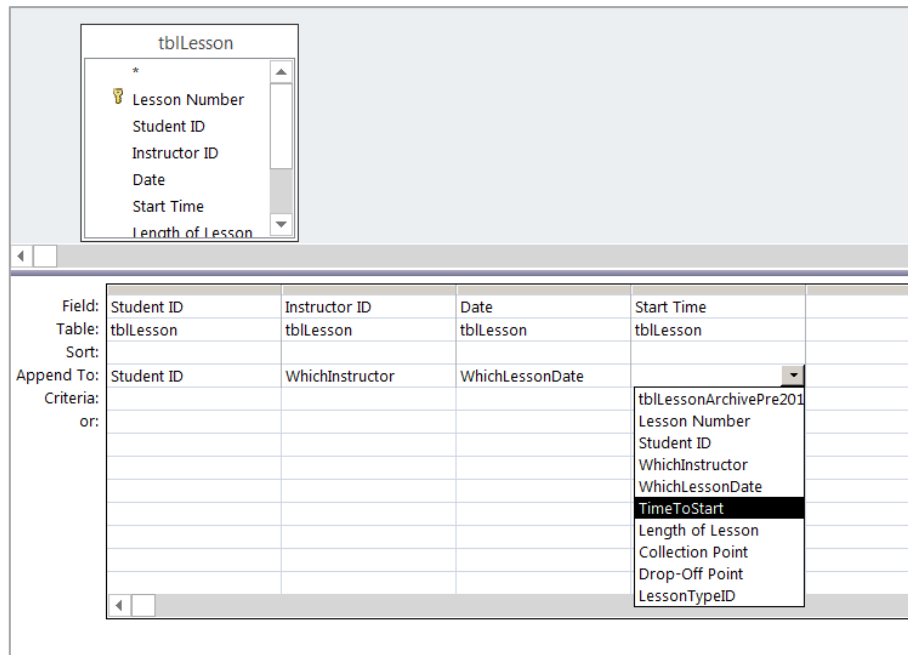


Figure 64 Nominating Fields for Data Appending

12.5 Running Action Queries from the Navigation Pane

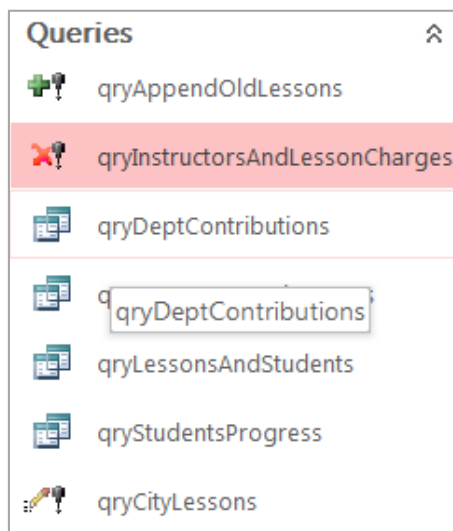


Figure 65 Action Queries in the Navigation Pane

Action queries are listed in the Navigation Pane when they are saved. Each type has a distinctive icon. The query can be run by double-clicking the query name from here. A warning dialog appears, emphasising that you are changing data values. Beware that if you run such a query repeatedly, the data will be amended repeatedly, potentially resulting in duplicate records or wrong data values.

Exercise 14: Action queries

Now look at this exercise (page 77).

13 What Next?

Now that you have some useful *Access* skills you may want to develop them further. IT Learning Programme offers a range of resources for study and teaching.

13.1 Downloadable Course Materials and More – the ITLP Portfolio

These course materials are available through the ITLP Portfolio, at <http://portfolio.it.ox.ac.uk>.

Each course pack includes the course handbook in pdf form and a zip folder of the exercise files that you need to complete the exercises. Archive versions of the course book may also be useful if you use an earlier version of the software.

The ITLP Portfolio helps you find articles, videos, resources and weblinks for further IT study. For some resources, you will be asked for your Oxford (SSO) username and password.

It may be possible for you to use the facilities at IT Services to work through the exercises in this booklet, or use any of the applications that are available. Contact us on courses@it.ox.ac.uk for details.

13.2 Database Courses Which Precede This

Databases: Concepts of database design

Databases: Building a database using *Access*

Databases: User-friendly databases using *Access*

Databases: Reporting data using *Access*

Read about the content of these courses in the IT Learning Programme Catalogue at www.it.ox.ac.uk/itlp/courses/catalogue

13.3 Word-processing and Spreadsheet Courses

A number of titles are available to improve your skills with word processing (*Microsoft Word*) and spreadsheets (*Microsoft Excel*). The schedule appears at www.it.ox.ac.uk/courses/.

A self-study guide to mailmerge can be downloaded from the ITLP Portfolio at <http://portfolio.it.ox.ac.uk>.

13.4 Course Clinics

We encourage everyone to work at their own pace. This may mean that you don't manage to finish all of the exercises for this session. If this is the case, and you would like to complete the exercises while someone is on hand to help you, come along to one of the Course Clinics that run during term time. More details are available from www.it.ox.ac.uk/courses/

13.5 IT Services Help Centre

The Help Centre is a good place to get advice about any aspect of using computer software or hardware. For Help Centre opening times, visit www.it.ox.ac.uk/help/gettinghelp/ and follow links to the General Helpdesk, or contact them by email on help@it.ox.ac.uk.

Appendix 1: Viruses and Security Levels in *Access*

1. Why Worry About Viruses?

A *virus* is malicious code that may arrive from another computer, and copy itself onto your hard disk. Depending on the taste of the person who devised it, it may be just a joke, or it may destroy data or system files. Some viruses send themselves on to other people via your email setup.

A virus may infect your computer while you are connected to the internet, or it may arrive attached to an email or on a contaminated removable disk, and it may take the form of a *macro* in an *Access* database file. It is essential to make provision for your computer to be scanned frequently for any viruses which may have arrived.

2. Scanning for Viruses

2.1. Virus Scanning Software

You should install and use virus-checking software such as *Sophos* (supported by IT Services). Other popular virus-checkers include McAfee *VirusScan* and Norton *AntiVirus*. Contact the Online Shop or visit www.it.ox.ac.uk for more information about obtaining *Sophos* at no cost or low cost for University members.

A typical virus-checker scans your computer disks according to a preset schedule. For instance, it may be set to check the hard disk every time you start *Windows*, or twice a week, or to scan every document on opening. The program detects any viruses, then alerts you and gives the options of deleting the file, putting it in quarantine or perhaps fixing it.

2.2. Keeping the List of Viruses Up-to-date

A virus-checking program must be kept up-to-date. It is important to connect frequently to the virus-checking centre (this is typically done via their internet website). The virus list on your computer is then updated with all viruses known to date, with any antidotes.

3. *Access 2013*¹ and Virus Security

Access 2013 considers any database file (such as **.accdb** or **.mdb**) as a potential route for virus infection. It starts by disabling any automatic content, and then asks you to decide whether to enable it.

This applies to content such as macros and some Control Wizards and ActiveX Controls.

Information about the other levels of protection, and further options for the way *Access* handles macros, is given in **Access Help**.

3.1. Security Warning Message

When you open a database file using *Access 2013*, a Security Warning message may appear in a white/yellow bar near the top of the window.

¹ Access 2007 and 2010 handle viruses in a similar way to Access 2013

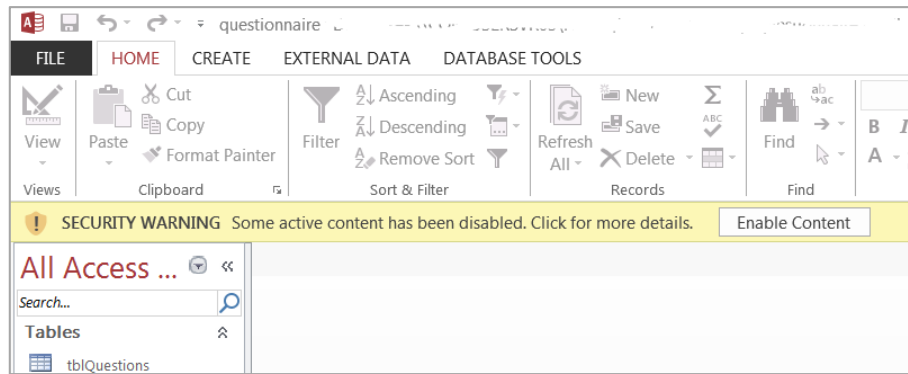


Figure 66 Security Warning Message When a Database File is Opened

If you do not expect to use any automatic features, you can safely ignore this message and continue work.

3.2. Enabling Automatic Content For Just One Visit

Enable Content on the Security Warning bar gives you the opportunity to enable any automatic content that may be present: only do this if you are sure that the database file comes from a safe source and does not contain any virus or other hostile code.

Note that the decision to **Enable this content** is effective for this session only. If the database is closed, then the decision will need to be repeated each time it is opened. This may be a convenient and cautious approach to take.

3.3. Trusted Locations in *Access 2013*

Some locations (folders, drives or devices) can be nominated as **Trusted Locations**: *Access* then considers any database files found there to be trusted, and their automatic content is permitted to run.

So you may find it convenient to nominate a folder or area on your computer or your network as being **Trusted by Access**. Then make sure that all your *Access* database files are saved there.

This decision may be reached in consultation with colleagues and other users of the database, and with your local IT support contact. Some departments have a policy about where computer files of different kinds should be saved.

For example, in our teaching rooms, the students' files are provided for you on a drive known as **H:**, and this has been set as a **Trusted Location** in the installed copies of *Access*. Computers at IT Services have *Sophos* virus protection software which is kept up-to-date. If you are in an IT teaching room, you can rely on *Sophos* to manage the virus scanning.

The program files for *Access* are placed in a **Trusted Location**, by default.

3.4. When to Trust a Location for *Access 2013*


You should only consider doing this if you do need to use automatic content such as macros, and if you have made very good arrangements to protect the computer from virus attack.

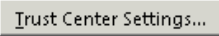
If you are using a University- or College-owned computer, take advice from your local IT Support Staff before making a location **Trusted**. Ensure that you have reputable virus scanning software installed, and that you keep its files of known viruses up-to-date (i.e. it checks at least weekly by connecting to the software

provider and downloading the latest lists). Then confirm that the software is configured to scan every file as it is opened.

Only once this is in place can you rely on the virus scanning software to do the virus scanning, and use *Access* to manage your database files.

3.5. Finding the Access Trust Center

You may decide to set *Access* to **Trust** a folder or computer storage area. This is done in the **Trust Center**, which is among the **Access Options** (found on the **File** menu) .

In the left-hand column of the **Access Options** dialog, choose the **Trust Center** category. This reveals the button  on the right.

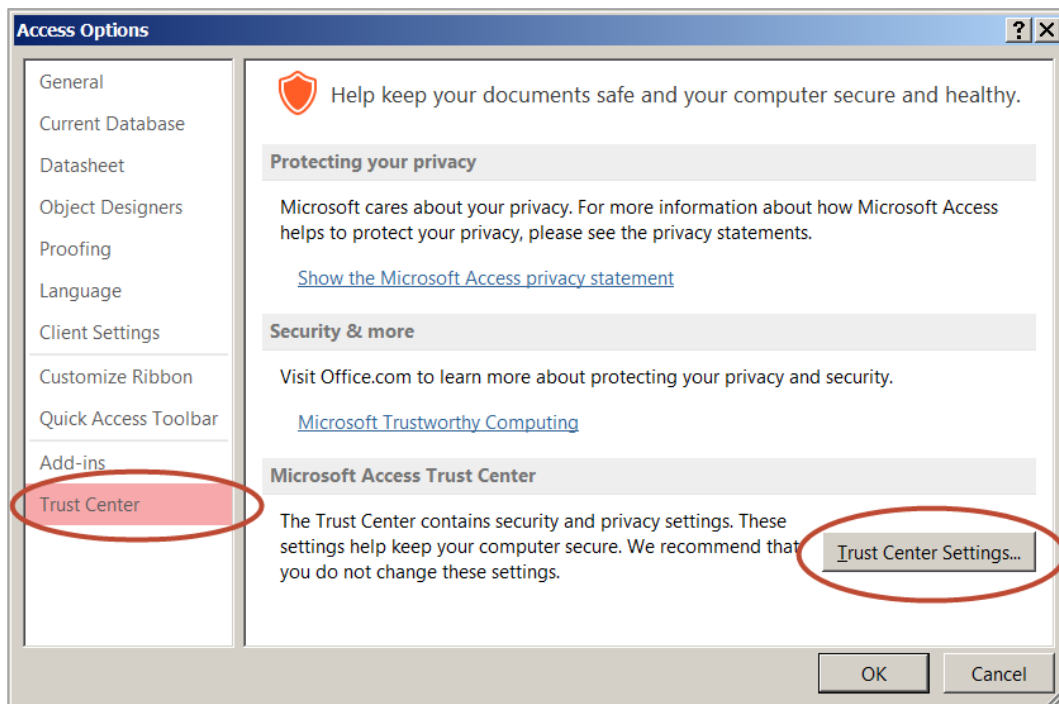


Figure 67 The Trust Center With the Trust Center Settings Button

3.6. How to Trust a Specific Location with *Access 2013*

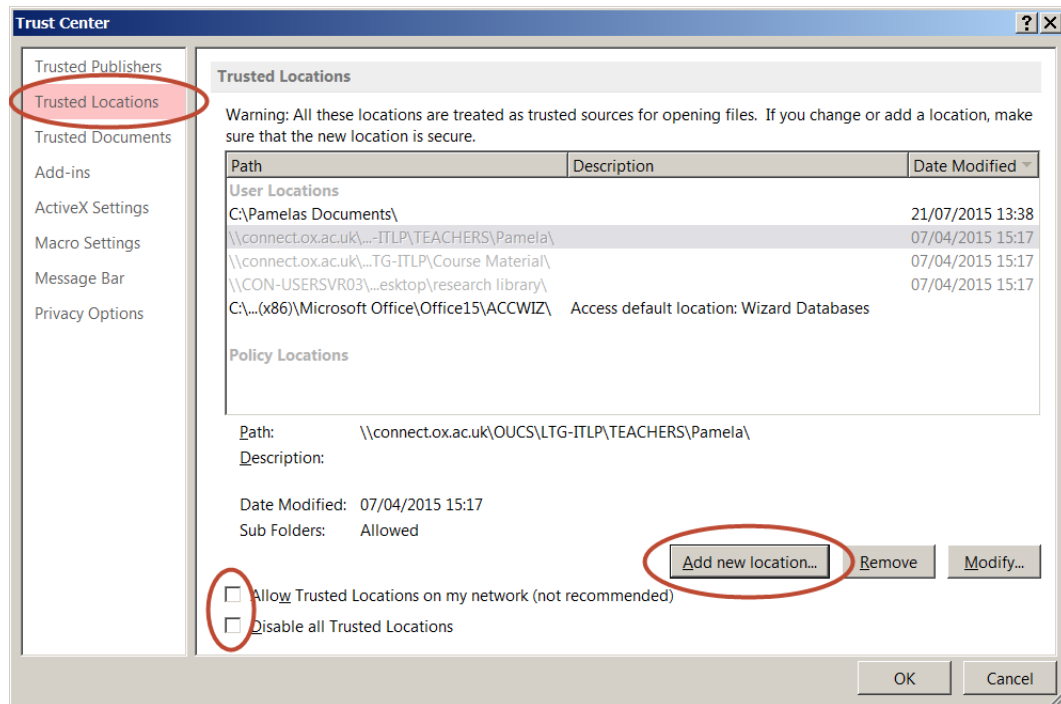


Figure 68 The List of Trusted Locations

Trusted Locations (found in the category list on the left) shows which locations have already been treated as trusted. Here you can **Add new location...** or **Modify...**. Beware that if you add a location to this list, you are no longer protected from active content saved in that location.

If the location where you plan to save your databases is on a network drive, check **Allow Trusted Locations on my network** in the same dialog. You should consult the IT support contact who is responsible for the network, before doing this.

You may also decide to trust subfolders within the chosen location – there is a check box for this in the **Trusted Location** dialog.

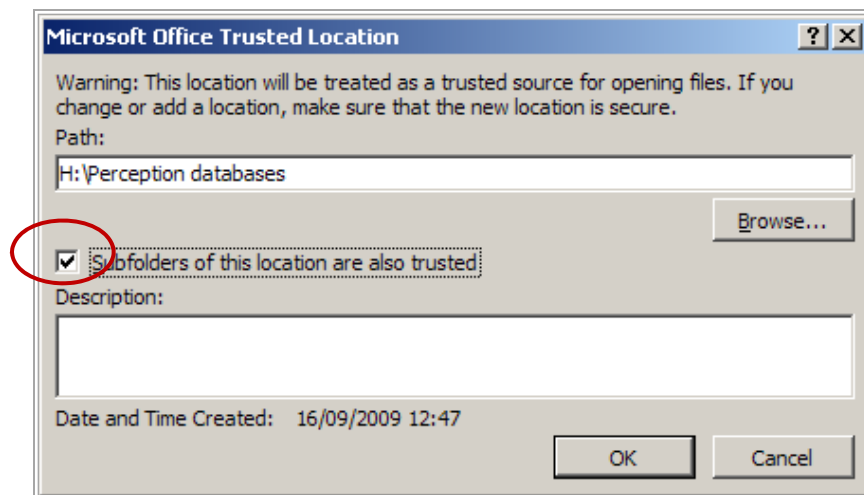


Figure 69 Trusting Subfolders in a Trusted Location

Once a folder has been Trusted, save all database **.accdb** files there.

3.7. Sandbox Mode

Note that these choices make *Access* run in a controlled and limited way, known as *sandbox mode*. In sandbox mode, *Access* will run a reasonable range of commands and functions. Occasionally, some macros created using older versions of *Access* may need to be re-written using revised commands which are now permitted.

In sandbox mode, some commands are disabled because they are deemed potentially unsafe, and could be used maliciously to damage your files or system. A user can only disable sandbox mode by changing keys in the *Windows* registry, which is not a task for the inexperienced to attempt (if you are not perfectly familiar with working in the *Windows* registry, don't experiment: ask someone who has experience, as there is a risk of making the computer unusable).

4. Further Help with Security and Trust Center Settings

There is a range of further settings available through the Trust Center. Read more about this in *Access*' Help.

Appendix 2: Student Exercises

Exercise 1 Opening a database file

- *Start Access*
- *Open an existing database file*
- *View the list of queries*

All the files for these exercises have been provided for you on a network drive. Your area of the drive is called drive H:\ This drive has been set up as an Access Trusted Location


Task 1

Start *Access* from the Start menu

Step 1

Start the computer if necessary

Step 2

Click  on the taskbar at the bottom of the screen

In the **Start** menu, explore the folders and look for the **Microsoft Office** folder

Choose *Access* 

Step 3

If you are prompted for any user information, just click **OK**

Step 4

(On your office or home computer you might otherwise start the program using an *Access* icon on the Desktop)

Task 2

Access 2013 can also be used to create an **app database**: where users work on the data via a web browser. This would require communication using *Office 365* or *SharePoint 2013*, and is not the subject of this course. We will work on a **desktop database**, which is saved locally on your computer or a network drive.

Task 3

Open *Inventory.accdb*

It has been placed in your network drive **H:** (or in another place as directed by your teacher)

Step 1

Choose **File| Open**

Step 2

Browse to the network drive called drive **H:**

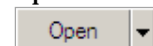
Alternatively, browse to a drive and folder as directed by your teacher




All the files you need for this course will be found here

Step 3

Locate the file called **Inventory.accdb**

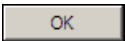
Open the file by selecting the filename then clicking

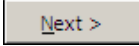
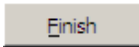







	<p>Step 4</p> <p>If a security warning bar appears, stating that certain content in the database has been disabled, then read Appendix 1</p>
<p>Task 4</p> <p>Look at the relationships diagram to see the tables and joins in this database –</p> <p>click  on the Database Tools tab of the ribbon</p> <p>In this database, an administrator is looking after a collection of equipment, each item being assigned to a department</p> <p>When ready, close the Relationships diagram</p>	
<p>Task 5</p> <p>Use the Navigation Pane in the database window</p>	<p>Step 1</p> <p>Select  next to the title on the Navigation Pane (on the left-hand side of the screen)</p> <p>Select Object Type (under Navigate To Category)</p> <p>Step 2</p> <p>Select  again and choose All Access Objects (under Filter By Group)</p>
<p>Task 6</p> <p>Look at the list of tables and queries in the Navigation Pane</p>	<p>Step 1</p> <p>Notice that the names of tables and queries that have already been created are listed in the Navigation Pane</p>

Exercise 2 Creating select queries using the wizard

- *Creating a select query using the wizard*
- *Saving the query*
- *Running the query and look at the results*
- *Switching between Design View and Datasheet View*




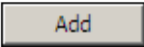
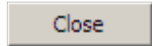
<p>Task 1</p> <p>Continue work in Inventory.accdb</p>	<p>Step 1</p> <p>Ensure that the file Inventory.accdb is open (if necessary, open it by choosing File Open)</p>
<p>Task 2</p> <p>Use the wizard to create a query based on tblAssets, including these fields: Asset Description, Make, Model and Date Acquired</p>	<p>Step 1</p> <p>Click  on the Create tab of the ribbon</p> <p>Select Simple Query Wizard and click </p> <p>Step 2</p> <p>In the list of available tables, choose tblAssets</p>





	<p>Step 3</p> <p>Double-click on the fields called AssetDescription, Make, Model and DateAcquired to place these fields in the right-hand list</p>
	<p>Step 4</p> <p>Click  to move through the wizard</p>
<p>Task 3</p> <p>Save the query</p> <p>Name it qryAssets</p> <p>Examine the results of qryAssets</p>	<p>Step 1</p> <p>When asked by the wizard, give a suitable name for the new query: qryAssets</p> <p>Make sure Open the query to view information is selected</p>
	<p>Step 2</p> <p>Click </p>
	<p>Step 3</p> <p>The query runs</p> <p>The results are presented in Datasheet View</p> <p>Notice that all records are included, but only those fields you requested</p>
<p>Task 4</p> <p>Switch between Design View and Datasheet View</p> <p>Close the query</p>	<p>Step 1</p> <p>Use  on the Home tab to switch to Design View</p>
	<p>Step 2</p> <p>In Design View, notice the table of assets with a list of all the available fields, in the top pane</p> <p>In the design grid (lower part), notice the fields that have been chosen and set out in columns</p>
	<p>Step 3</p> <p>Use  on the Home tab or the Query Tools Design tab, to switch to Datasheet View</p>
	<p>Step 4</p> <p>When ready, close the query using </p> <p>Leave the file open, showing just the Navigation Pane</p>
<p>Task 5 Practice</p> <p>Create another query based on the table tblDepartments</p> <p>List every department by name, along with the name of the Head of Dept and a contact phone number</p>	<p>Step 1</p> <p>Click  on the Create tab of the ribbon, select Simple Query Wizard and click </p>
	<p>Step 2</p> <p>Select the table tblDepartments</p> <p>Choose the fields DeptName, DeptHead, ContactTelNum</p>

Save the query with a suitable name (with the prefix qry)	Step 3 Save the query as qryContactingHeadsofDept Examine the resulting data Close the query
---	---

Exercise 3 Working on a query in Design View

- *Starting a new query*
- *Adding tables in Query Design View*
- *Adding fields*
- *Running a query based on two or more tables*
- *Saving and closing the query*

Task 1 Continue work in Inventory.accdb	Step 1 Ensure that Inventory.accdb is open Close any tables or queries that may be open using 
Task 2 Start a new query in Design View Add two tables tblAssets and tblDepartments These tables are already related by a join using DeptID	Step 1  Click  to display the Design View with the Show Table dialog Step 2 Choose tblAssets and tblDepartments by selecting the table name and clicking  or double-clicking a table name Repeat until you have both the required tables in the design grid behind (do not worry if you collect some extras, they can be deleted later) Step 3 Click  to use the design grid If an unwanted table has appeared, click it once to select it then press DELETE to delete Step 4 Notice that the two tables are shown in the top pane, with a line joining the DeptID fields The top pane can be resized by dragging the divider between the two panes
Task 3 Choose fields and set them up in the design grid: Asset ID	Step 1 Set up a field in the first column of the design grid: Select the field name AssetID in the list for tblAssets Drag the field name to the top of the first empty column in the grid

<p>Description</p> <p>Make</p> <p>Date for next maintenance</p> <p>Phone number for contacting the Dept</p>	<p>Step 2</p> <p>The next fields you need are AssetDescription, Make and NextMaintDate from the tblAssets</p> <p>Set these up in the design grid, perhaps by double-clicking a fieldname, or by clicking at the top of an empty column and picking a fieldname from the menu that drops down</p> <p>Step 3</p> <p>The final field, ContactTelNum, must be taken from the related table tblDepartments</p>
<p>Task 4</p> <p>Run the query</p> <p>Examine the results</p>	<p>Step 1</p> <p>Click  or  to switch to Datasheet View</p> <p>The query is run (<i>Access</i> interrogates the data in the tables at that moment)</p> <p>Step 2</p> <p>Any records which appear in both tables with corresponding DeptID values are presented</p> <p>Only those fields you selected in the query design are presented</p> <p>Step 3</p> <p>The data <i>can</i> be edited in the query</p> <p>For example, suppose that asset number 15, a Watkins bar code reader, should next be serviced on 1 January 2016</p> <p>Locate that record, move into the NextMaintDate field, and type the required date to change the date in this list</p> <p>You have changed the value in the underlying table, tblAssets</p>
<p>Task 5</p> <p>Close the query</p> <p>Save it, giving a suitable name</p>	<p>Step 1</p> <p>Close the query by clicking , saving any changes when asked</p> <p>When asked, give the query a name that conforms to the naming convention, such as qryMaintenanceList</p>
<p>Task 6 Practice</p> <p>Create another new query in Design View, which lists each asset with its description, serial number, date acquired and expected life, with the name of a responsible Head of Department</p>	<p>Step 1</p> <p>Click  to display the Design View with the Show Table dialog</p> <p>Add the tables tblAssets and tblDepartments</p> <p>Step 2</p> <p>Choose the fields: AssetDescription, SerialNumber, DateAcquired and DepreciableLife from one table</p> <p>Choose the DeptHead from the other table</p>

	Step 3 Run the query and examine the results
Task 7 Close the query, saving it as qryAssetLifeTimes	

Exercise 4 Setting criteria in a query

- *Criteria with equals*
- *Editing criteria*
- *Comparison criteria*
- *Criteria with wildcard symbols*
- *Criteria with Not and Is Null*
- *Multiple criteria with AND and OR*

Task 1

Continue work in **Inventory.accdb**

Task 2

Create a query listing assets with their description, make, model, date acquired and purchase price

Step 1

Close any tables or queries that may be open

Step 2

Create a new query in Design View, with **tblAssets** only

Step 3

Add these fields:

AssetDescription, Make, Model, DateAcquired, PurchasePrice

Step 4

Run the query

Task 3

Add criteria to show only laptops

Save the query so far

Step 1


Switch back to Design View for the same query

In the **AssetDescription** column, on the Criteria row, type **laptop**

Step 2

Run the query

Step 3

Save the query, by clicking  on the Quick Access Toolbar (a suitable name might be **qryAssetsWithPrices**)

Task 4

Step 1

In Design View of the same query, delete **laptop** from the Criteria row

<p>Edit the criteria so the query now shows only printers</p> <p>Then change the criteria to show all assets made by Watkins</p>	<p>Step 2 Enter printer as the Criteria for Description Run the query</p> <p>Step 3 Delete any previous criteria (look at <i>all</i> the columns) In the Make column, enter Watkins Run the query</p>
<p>Task 5 Edit the same query to show all assets which cost over £200</p>	<p>Step 1 Delete any previous criteria (look at <i>all</i> columns)</p> <p>Step 2 In the PurchasePrice column, enter >200 (note you do not need to type the £ symbol, as that is part of the format for this field)</p> <p>Step 3 Run the query</p>
<p>Task 6 Use this query to show all laptops which cost over £350</p>	<p>Step 1 Delete any previous criteria In the Description column enter the criteria laptop</p> <p>Step 2 In the PurchasePrice column, in the same row, enter >350</p> <p>Step 3 Run the query If you do not get 3 records, check both criteria, and confirm that each one is in the correct column and in the correct row</p>
<p>Task 7 Revise the query to show all assets made by Watkins, Wilkinson or Wood Associates (hint: use the * wildcard symbol)</p>	<p>Step 1 Delete any previous criteria In the Make column, enter the criteria w* <i>Access edits this to Like "w*"</i></p> <p>Step 2 Run the query</p>
<p>Task 8 Revise the query to show all machines whose model number is 180 to 190 (use Between)</p>	<p>Step 1 Delete any previous criteria In the Model column, enter the criteria Between 180 and 190</p> <p>Step 2 Run the query</p>
<p>Task 9 Revise the query to show all machines which are not laptops</p>	<p>Step 1 Delete any previous criteria In the AssetDescription column, enter the criteria not laptop</p>

	<p>Step 2 Run the query</p>
<p>Task 10 Find all records for machines that have not yet been sold (no value has been entered in the DateSold field)</p>	<p>Step 1 Delete any previous criteria Add the field DateSold to an empty column on the query design grid</p>
	<p>Step 2 Under DateSold, enter Is Null</p>
	<p>Step 3 Run the query This suggests that 22 of the machines have not yet been sold, as no DateSold has been entered</p>
<p>Task 11 Revise this query again, to show a list of laptops and PCs and printers</p>	<p>Step 1 Delete any previous criteria In the Description column, enter laptop</p>
	<p>Step 2 In the next row of the Description column, enter PC, and on the row below enter printer</p>
	<p>Step 3 Run the query</p>
<p>Task 12 Practice Make up some more queries, such as: all assets acquired since 1st January 1985 all assets which cost between £100 and £300</p>	<p>Step 1 Devise some more interesting query criteria, and try them out</p>
	<p>Step 2 For each new attempt, delete any previous criteria before entering new values Look carefully at criteria on the same row (AND) or on separate rows (OR)</p>
	<p>Step 3 – Entering criteria Enter any date criteria in the form 1/1/1985 (<i>Access</i> will insert # symbols as necessary when it recognises your date) If you use “between”, the criteria should be of the form: Between 100 And 300</p>
	<p>Step 4 Close the query, saving if necessary Close all tables and queries, but leave the Navigation Pane and Inventory.accdb still open</p>

Exercise 5 Rearranging a query

- *Removing a field*
- *Rearranging columns*
- *Adjusting column width*
- *Hiding and showing a field*
- *Sorting query results*
- *Deleting a query*

Task 1

Continue working in the file **Inventory.accdb**

Step 1

If necessary, open **Inventory.accdb**


Close any tables or queries that may be open, using 

Task 2

Open the query **qryAssetsAndPrices** which has already been made, in Design View

Step 1

Double-click the query **qryAssetsAndPrices** which has already been made

Click  on the **Home** tab to switch to Design View

Step 2

Look carefully at the design grid, noting which tables and which fields are included

Step 3

Run the query to look at the resulting dataset, then return to Design View

Task 3

Delete the field **DepartmentID**

Step 1

Select the column for **DepartmentID** (click at the top of the column until the whole column appears black)

Step 2

Press **DELETE** to delete this field from the query

Task 4

Drag the column **DeptName** to be the last on the right

Step 1

Click at the top of the **DeptName** column to select it (the column turns black)

Step 2

Use drag and drop to move it to an empty column to the right of other fields on the grid


This may be tricky and takes practice!

Task 5

Run the query



In Datasheet View, make the **DeptName** column wide enough to show all the text entries

Step 1

Click  to run the query and display fresh results.

Step 2

In Datasheet View, drag the divider at the right-hand edge of the top of the **DeptName** column, to adjust the column width

	<p>Step 3</p> <p>Make this column wider until the text values can be seen</p>
<p>Task 6</p> <p>In Design View, clear the Show checkbox under AssetID, then run the query to confirm that this field is no longer showing</p> <p>Then re-tick the checkbox to show AssetID values</p>	<p>Step 1</p> <p>Click  to return to Design View</p> <p>In the column for AssetID, click the checkbox in the Show row, until the tick disappears</p> <p>Step 2</p> <p>Run the query and note that the AssetID data is not displayed</p> <p>Step 3</p> <p>In Design View again, click the Show checkbox under AssetID to place a tick there, so these values will be shown</p>
<p>Task 7</p> <p>Sort the results in order of the date the asset was bought (acquired), oldest items first</p> <p>Then re-sort in alphabetical order of Department name, with a secondary sort of Purchase Price</p>	<p>Step 1</p> <p>In the DateAcquired column, in the Sort row, choose Ascending</p> <p>Step 2</p> <p>Run the query and notice the order the records appear</p> <p>Step 3</p> <p>Remove the previous sort setting: select Not Sorted in the drop-down list</p> <p>Move the DeptName column so it appears to the left of the PurchasePrice column</p> <p>Step 4</p> <p>Select Ascending in the Sort row in the DeptName column and also in the PurchasePrice column</p> <p>Step 5</p> <p>Run the query and check the sorting is as expected (alphabetical order by Department name with a secondary sort based on the purchase price)</p>
<p>Task 8</p> <p>Close the query, discarding changes</p>	<p>Step 1</p> <p>In either view, click  to close the query</p> <p>When asked, choose not to save the changes</p>

Exercise 6 Parameters in queries

- Create and use a parameter query
- Create and use a parameter that uses wildcard characters

Task 1

Continue work in **Inventory.accdb**

<p>Task 2 Create a new query based on tblAssets</p> <p>Set up criteria so that the user can ask, when they run the query, for only one kind of asset</p>	<p>Step 1 Create a new query in design view Base it on tblAssets</p>
	<p>Step 2 Choose the fields AssetDescription, Make, and DateAcquired and DateSold</p>
	<p>Step 3 In the Criteria row, enter [Which kind of machine?] in the AssetDescription field</p>
<p>Task 3 Test the query and save it as qrySingleMachineType</p>	<p>Step 1 Run the query In the Enter Parameter Value dialog, type laptop Only laptop records are displayed</p>
	<p>Step 2 Run the query again, requesting only printers</p>
	<p>Step 3 Save the query with the name qrySingleMachineType (do not close it yet)</p>
<p>Task 4 Change the criteria so that the user can ask for machines by make, just giving the first character</p>	<p>Step 1 Remove the AssetDescription criteria from the query design grid</p>
	<p>Step 2 In the Make field, enter the parameter criteria: Like[Which make do you want?]&"**"</p>
<p>Task 5 Test the query and save it as qryMachinesByMake</p>	<p>Step 1 Run the query</p>
	<p>Step 2 When asked, type a single letter such as s, so as to display only records whose makes begin with that letter</p>
	<p>Step 3 Think: how would you use this query to display only machines made by Watkins?</p>
	<p>Step 4 Choose File Save As, and give the query the name qryMachinesByMake</p>

Exercise 7 Special types of query

- *Top values queries*
- *Find duplicate records*
- *Find unmatched records*

Task 1

Continue work in **Inventory.accdb**


Task 2

Create a new query based on **tblAssets**

Include **Description**, **Serial Number** and **Purchase Price**

Set the query to display only the 5 most expensive items

Step 1

Click  on the **Create** tab

Step 2

Base the new query on **tblAssets**

Step 3

Choose the fields **Description**, **SerialNumber** and **PurchasePrice**

Step 4

Set a descending sort by **PurchasePrice**

Step 5

Click the **Return Top Values** control



Return: 25% and choose 5, so as to display the 5 items with the top purchase prices

Step 6

Run the query and observe the result

Task 3

Now display only the 10 cheapest items

Step 1

In Design View, sort the records in ascending order of purchase price

Step 2

Use the **Return Top Values** control and enter 10

Step 3

Run the query and observe that only the 10 cheapest items are listed

Task 4

Change the query to display the 10% of items bought most recently

Close the query

Step 1

Remove the sort and add a **DateAcquired** field to the grid

Step 2



Sort by this field in descending order

Step 3

Set the Top Values control to 10% (type 10% in the control, then press ENTER)

Step 4

Run the query

<p>Task 5</p> <p>Create a new query to find duplicate asset entries</p>	<p>Step 1</p> <p>Click  on the Create tab</p> <p>Choose Find Duplicates Query Wizard</p> <p>Choose to base the query on tblAssets</p> <p>Step 2</p> <p>Fields that might contain duplicate information include AssetDescription and SerialNum</p> <p>Step 3</p> <p>Additional fields to show should include Make, Model and DateAcquired</p> <p>Step 4</p> <p>Name the query qryFindDupAssets</p> <p>Step 5</p> <p>The query runs when you finish the wizard</p> <p>Step 6</p> <p>Note any duplicate records</p> <p>Here you might delete unwanted records</p> <p>Close the query</p>
<p>Task 6</p> <p>Create a new query to find any assets which have not yet been assigned to any Department (DeptID)</p>	<p>Step 1</p> <p>Click  then choose Find Unmatched Query Wizard</p> <p>Choose tblAssets</p> <p>Step 2</p> <p>tblDepartments contains related records</p> <p>The field DepartmentID (or DeptID) is in both tables</p> <p>Step 3</p> <p>In query results, show the fields AssetDescription, Make, Model and SerialNum</p> <p>Name the new query qryFindUnmatchedAssets</p> <p>Step 4</p> <p>The query runs when you finish the wizard</p> <p>Here you might edit the data to reconcile the values between the two tables</p> <p>Close the query</p>

Exercise 8 Crosstab query (Optional)

- *Build and use a crosstab query (using the wizard)*

Task 1

Continue work in **Inventory.accdb**

Task 2

Create a new crosstab query:

Base it on

qryPriceByDescrAndDept

Choose **Description** and **DeptName** as the row and column headings

Calculate the Sum of **PurchasePrice**

Step 1

Click  and choose **Crosstab Query Wizard**

Base the new query on a previous query **qryPriceByDescrAndDept**

Step 2

Choose **AssetDescription** for the row headings

Step 3

Choose **DeptName** for the column headings

Step 4

PurchasePrice is the field for calculations

Choose the **Sum** function

Step 5

Name the new query

qryCrossTabAssetsAndDepts

Task 3

Run the query

Look at the money spent by each department on purchasing each type of asset

Task 4

Close the query, saving if you wish

Close the database file, leaving *Access* open

Exercise 9 Calculations in queries

- *A calculated column*
- *Operators in expressions*
- *Concatenating text*
- *Totals in a query*

Task 1

In **Inventory.accdb**, create a new query based on **tblAssets**


Step 1

Open **Inventory.accdb**

Step 2

Create a new query in Design view

Base the query on **tblAssets**

Include asset descriptions and purchase price	Step 3 Choose the fields AssetDescription and PurchasePrice
Task 2 Add a calculated column to find the depreciated value of each asset (we are told this is 1/3rd of the purchase price) Edit the column heading to be Depreciated	Step 1 In an empty column on the query design grid, enter an expression in the Field row: Depreciated: [PurchasePrice]/3
	Step 2 Run the query and check that the calculated values are plausible
Task 3 Add more columns for dates when an asset was acquired and sold Calculate how many days each asset was kept; give a suitable column heading	Step 1 Add new columns for the fields DateAcquired and DateSold
	Step 2 In an empty column, enter an expression in the Field row: DaysKept: [DateSold] - [DateAcquired]
	Step 3 Run the query and check the calculated values are plausible
Task 4 Add a column to show the asset make and model concatenated into one text	Step 1 In an empty column in the query design grid, enter an expression in the Field row: AssetName: [Make] & " " & [Model] Note: make sure there is a space between the double quote marks
	Step 2 Run the query and check the calculated values are sensible
Task 5 Close the query, saving it as qryAssetDetails	
Task 6 Create a new query based on tblAssets Include AssetID and AssetDescription	Step 1 Create a new query Base it on tblAssets
	Step 2 Include the fields AssetID and AssetDescription
Task 7 Group records by Asset Description Count how many of each type of asset has been purchased Save and close the query	Step 1  Click Totals to show the Total row
	Step 2 Under AssetDescription , choose Group By
	Step 3 Under AssetID , choose Count

	Step 4 Run the query and inspect the result
Task 8 Save the query as qryCountEachAssetType , and close it	

Exercise 10 Calculations on a form

- *Add a calculated control on a form*
- *Add a concatenated text control on a form*
- *Using the Expression Builder*
- *Look at the results on the form*

Task 1

Continue work in **Inventory.accdb**

Open **frmAllAssets** in Design View

Task 2


These activities are described in Design View, but can also be carried out in Layout View if you prefer

Task 3

Add an unbound control

Enter an expression to calculate the VAT payable on the purchase price (this is 20% of the price paid)

Step 1

Click  then click in a space in the form, to add an unbound control

Step 2

In the unbound control, enter an expression:
`=0.2*[PurchasePrice]`


Step 3

Click in the associated label and replace **Text:** with **VAT payable**


Task 4

Add a calculated control which shows the head of department's name and phone number in this way:
Jim Hacker (72345) – by using the Expression Builder

Step 1

Click  then click in a space in the form to add an unbound control

Step 2


Click  **Property Sheet** to show the Property Sheet, and select the new unbound control so that its properties are listed

Step 3

Click in the **Control Source** row, then click  to open the Expression Builder

Step 4

Choose **frmAllAssets** in the left panel, then double-click **DeptHead** in the middle, to start building an expression

	<p>Step 5 Build up this expression, using a combination of typing characters and choosing Operators and field names from the list: <code>=[DeptHead]&" "&[ContactTelNum]&"")"</code></p> <p>Take care that the space and opening bracket are in quote marks, and the closing bracket is in quote marks</p> <p>Step 6 When ready, click  to insert the expression into the Control Source property of the text box Click on the grey desktop to finalise the calculated control</p>
<p>Task 5 You may now want to delete the separate controls for DeptHead and ContactTelNum from the report</p>	
<p>Task 6 View the form and inspect the two calculated controls on several records Close the form, saving changes</p>	

Exercise 11 Calculations in a report

- Add a calculated control in a report (in the Detail)
- Format the calculated control
- Add a summary control in the report Footer

Task 1

Continue work in **Inventory.accdb**

These activities are described in Design View, but can also be carried out in Layout View if you prefer

Task 2

In **rptAllAssets**, add an unbound text box in a space in the Detail

Insert an expression to calculate the (depreciated) end date of each asset - use the **DateAcquired** and the **DepreciableLife**


NB Multiply the number of years of depreciable life by 365 to find the number of days

NB2 We will neglect leap years for simplicity

Step 1

Open **rptAllAssets** - preview it then switch to Design View

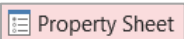
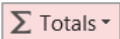
Step 2

Click  then click in a space in the **Detail** section of the report, to create an unbound control

Step 3

Click in the unbound control and type an expression for the end date of an asset:

`=[DateAcquired]+([DepreciableLife]*365)`

Task 3 Assign the Long Date format Give suitable text for the label	Step 1 With the calculated control selected, click  if necessary to display the Properties
	Step 2 On the Format tab of the Property Sheet, select the format Long Date
	Step 3 Delete the text in the associated label Type End date
Task 4 Add a summarising control in the Report Footer, to calculate the total of the purchase prices	Step 1 Select the Purchase Price control (in the Detail)
	Step 2 Click 
	Step 3 Choose Sum from the list of functions
	Step 4 A control appears in the Report Footer, with the expression: <code>=sum([PurchasePrice])</code>
Task 5 Format the control to show currency; give suitable label text	Step 1 With the calculated control selected, edit its properties: select the Currency format
	Step 2 Replace the label text with Total Spend
Task 6 Preview the report, and examine the calculations after each record and at the end of the report Name the report rptAssetDepreciation , and close it	

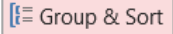
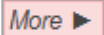
Exercise 12 Summary calculations on a grouped report (*Optional*)

- Add a summary control in the group footer of a grouped report
- Run the report to see the effect of the group summary

Task 1

Continue work in **Inventory.accdb**




Note these tasks are described using the **Group, Sort and Totals** pane, but you can achieve similar results using  in a grouped report


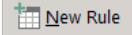



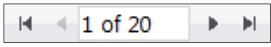
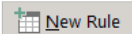
Task 2 Open the report rptAssetsByDepartment in Design View	Step 1 Preview rptAssetsByDepartment Inspect the way the records are laid out: assets for each department are grouped together
	Step 2 Switch to Design View Notice the different sections of the report: the Detail, the Report Header and Footer, the Page Header and Footer and the grouped DeptID Header and Footer
Task 3 Use the Group, Sort and Total pane to add a calculated control in the DeptID Footer, to show the average purchase price spent by each department	Step 1 Click 
	Step 2 In the Group, Sort and Total pane, click the GroupID bar Click  for more options
	Step 3 Click the arrow beside with no totals to show some further options
	Step 4 Set Total on PurchasePrice Set Type to be Average Check Show Grand Total and Show in Group Footer
	Step 5 A control appears in the DeptID Footer, with the expression: <p style="text-align: center;">=average([PurchasePrice])</p> Another control with the same expression appears in the Report Footer
Task 4 Format and label both of the new controls	Step 1 With the calculated control in the group footer selected, apply Currency format
	Step 2 Replace the label with the text Average Purchase for the Department You may need to resize or move the label
	Step 3 Repeat this for the grand total in the Report Footer





Task 5 Add another summarising control, which shows how many assets each department holds	Step 1 Use the Totals option on the DeptID Group bar (in the Group, Sort and Total pane) to set: Total by Asset Description Type is Count Records Show Grand Total Show in Group Footer
Task 6 Run the report and look at the values calculated for each department Close the report, saving changes	
Task 7 Close the Inventory database file, leaving <i>Access</i> open	

Exercise 13 Conditional formatting for controls in a form

- *Formatting text in a field box in a form*
- *Formatting that is conditional upon a field value*
- *Formatting that is conditional upon a function or formula*
- *Multiple conditional formatting on the same control*

Task 1 Open New Dentists.accdb Look at the Relationships diagram to discover the tables and their links	Step 1 Display the Open dialog eg. using  on the File menu
	Step 2 Find and open New Dentists.accdb
	Step 3  Click  on the Database Tools tab, to display the Relationships diagram
	Step 4 Examine the tables and their fields, and the links between them In this database, a dental surgery is keeping track of patients and their appointments Close the diagram
Task 2 Open frmPatientsPlain in Design View	Step 1 Open frmPatientsPlain in Design View, by right-clicking the form name in the Navigation Pane then selecting Design View
	Step 2 Examine the form, noticing the controls that are in place

<p>Task 3</p> <p>Set conditional formatting so that if a patient was born before 1st January 1960 their Date Of Birth control is emphasised</p>	<p>Step 1</p> <p>Select the Date Of Birth control (the text box control where the data values appear, not the label beside it)</p> <p>Step 2</p> <p>On the Format tab of the ribbon, click </p> <p>In the Conditional Formatting dialog, confirm that the DOB field control is selected</p> <p>Step 3</p> <p>Click </p> <p>Step 4</p> <p>The new rule will Check values in the current record</p> <p>Set it to Format only cells where the field value is less than 1/1/1960</p> <p>Step 5</p> <p>In the Preview, choose a bright fill colour and a contrasting font colour</p> <p>Step 6</p> <p>Click  to create the rule</p> <p>Click  to close the dialog</p>
<p>Task 4</p> <p>Examine several records, to see whether the conditional formatting is working as expected</p>	<p>Step 1</p> <p>Switch to Form View by clicking  on the Home tab</p> <p>The records are displayed one at a time</p> <p>Step 2</p> <p>Use the navigation buttons at the bottom of the form to page between all the records </p> <p>Step 3</p> <p>When you reach a patient who was born before 1st January 1960, confirm that the DOB field control has changed colour</p>
<p>Task 5</p> <p>Add another rule to the DOB field control, to draw attention if a date is accidentally entered that is after today's date</p> <p>Hint: Look among the pre-defined functions for the computer's date</p>	<p>Step 1</p> <p>Switch to Design View</p> <p>With the Date Of Birth control selected, display the Conditional Formatting dialog again</p> <p>Step 2</p> <p>Click  to add a new rule</p> <p>The new rule is to Check values in the current record</p>

	<p>Step 3</p> <p>Set up to Format only cells where the field value is more than today's date:</p> <p>Choose Field Value Is More Than</p>
	<p>Step 4</p> <p>Click </p> <p>In the Expression Builder:</p> <p>In the first column choose Functions then Built-In Functions</p> <p>In the middle column choose Date/Time</p> <p>In the last column choose Date</p> <p>Click </p>
	<p>Step 5</p> <p>Check that the expression box now contains Date ()</p> <p>In the Preview, choose another eye-catching fill colour and a contrasting font colour</p>
	<p>Step 6</p> <p>Click  to create the rule</p> <p>This control now has 2 conditions set: if the Date Of Birth is pre-1960 it appears with a bright colour to remind the staff , and if it is after today it appears with another colour to point out that this is likely to be a mis-type</p> <p>Click  to close the dialog</p>
<p>Task 6</p> <p>Run the form</p> <p>Confirm that the conditional formatting is appearing as expected</p> <p>Save changes and close the form</p>	

Exercise 14 Action queries

- *Update values*
- *Delete records*
- *Make a table*
- *Append records*

Task 1





Continue working in the database **New Dentists.accdb**





Task 2



Create a query based on **tblStaffContactInfo**

Step 1

Click  in the **Create** tab

<p>Include all fields</p> <p>Name it qryStaffContactInfo</p>	<p>Step 2</p> <p>In the Add dialog, choose tblStaffContactInfo</p>
	<p>Step 3</p> <p>Set up columns for the fields ID, StaffMember, email and jobtitle</p>
	<p>Step 4</p> <p>Save the query as qryStaffContactInfo</p> <p>Run the query and look briefly at the data</p>
<p>Task 3</p> <p>All the Odd-job Men are now to be known as Senior Administrative Supervisors</p>	
<p>Task 4</p> <p>Create an update query to make this change</p> <p>Run the update query and save it as qryOddJobUpgrade</p>	<p>Step 1</p> <p>In the jobtitle field, enter Odd-job man in the Criteria row</p> <p>Run the query to confirm that only the 2 Odd-job men are included</p>
	<p>Step 2</p> <p>Return to Design View</p> <p>Click  Update</p> <p>The Update To row appears in the design grid</p>
	<p>Step 3</p> <p>In the Update To row, in the Jobtitle field, enter Senior Administrative Supervisor</p>
	<p>Step 4</p> <p>Click  Run to run the update query</p> <p>Accept the warning about updating 2 rows</p>
	<p>Step 5</p> <p>Close the query, saving it as qryOddJobUpgrade</p>
	<p>Step 6</p> <p>Open tblStaffContactInfo and check that the odd-job men now have new job titles</p>
<p>Task 5</p> <p>As a precaution, make a backup copy of tblAppointment, naming it tblApptCopy</p>	<p>Step 1</p> <p>Select tblAppointment and copy it (e.g. use  Copy)</p>
	<p>Step 2</p> <p>Paste  Paste a copy into the list of tables</p> <p>Give the new table the name tblApptCopy</p> <p>This copy of the data may be needed in case of a mistake with an action query</p>

<p>Task 6</p> <p>In tblAppointment, use a Delete query to remove all records for appointments which were not kept</p>	<p>Step 1</p> <p>Create a new query based on the table tblAppointment (use Create query in Design View)</p>
	<p>Step 2</p> <p>Include the fields AppointmentID and AppointmentKept</p>
	<p>Step 3</p> <p>Set criteria 0 under AppointmentKept (recall that value zero 0 represents “no” in a Yes/No field)</p> <p>Run the query</p>
	<p>Step 4</p> <p>Use  and convert this to a Delete query</p>
	<p>Step 5</p> <p>Click  to run the Delete query</p> <p>Accept the warning about deleting 21 records</p>
<p>Task 7</p> <p>Save this Delete query as qryFailedAppointments</p>	<p>Step 1</p> <p>Close the query, saving it when asked as qryFailedAppointments</p>
	<p>Step 2</p> <p>Open tblAppointments again, and notice that the records have been reduced to 59</p> <p>Close the table</p>
<p>Task 8</p> <p>For those patients who go to school, make a list of the patient’s name, their school and a contact phone number at the school</p> <p>Use a query to make a new table with this information called tblSchoolContacts</p> <p>Save this query as qrySchoolContacts</p>	<p>Step 1</p> <p>Create a new query using tblPatient and tblSchool</p> <p>Include FirstName and LastName from tblPatient</p> <p>Include SchoolName and ContactPhone from tblSchool</p>
	<p>Step 2</p> <p>Run the query and look at the 7 records</p>
	<p>Step 3</p> <p>Use  to convert the query to a make-table query</p> <p>Specify the new table name tblSchoolContacts, in the current database file</p>
	<p>Step 4</p> <p>Click  to run the make-table query</p> <p>Accept the warning about pasting 7 records</p>

	<p>Step 5 The new table is seen in the Navigation Pane</p> <p>Step 6 Close and save the query as qrySchoolContacts</p>										
<p>Task 9 Create a query based on tblMorePeople Include all the fields except the ID number Append the details of these patients to tblPatient Choose corresponding fields where the appended data should go Run the query Save this query as qryAppendMorePeople</p>	<p>Step 1 Create a new query based on tblMorePeople Include all fields except ID</p> <p>Step 2  Use Append and convert this to an Append query Select the table tblPatient in the current database</p> <p>Step 3 In the Append To row, match up the field names:</p> <table><tr><td>Surname</td><td>LastName</td></tr><tr><td>Forename</td><td>FirstName</td></tr><tr><td>Address4</td><td>PostCode</td></tr><tr><td>TelNo</td><td>HomePhoneNum</td></tr><tr><td>DateOfBirth</td><td>DOB</td></tr></table> <p>Step 4  Click Run to run the append query Accept the warning about appending 43 records</p> <p>Step 5 Save the query as qryAppendMorePeople, and close it</p> <p>Step 6 Open tblPatient and confirm that the records have been appended</p>	Surname	LastName	Forename	FirstName	Address4	PostCode	TelNo	HomePhoneNum	DateOfBirth	DOB
Surname	LastName										
Forename	FirstName										
Address4	PostCode										
TelNo	HomePhoneNum										
DateOfBirth	DOB										
<p>Task 10 Close all queries and tables Close the database, leaving the computer at the <i>Windows</i> desktop</p>											

Databases: Querying and analysing data using Access



IT Learning Programme

Today's arrangements

Your teacher is:

Your demonstrators are:

We finish at:

12:15

Your safety and comfort are important

Where is the fire exit?

Please report any equipment faults to us

The toilets are along the corridor outside the teaching rooms

The rest area is where you registered;
it has vending machines and a water cooler



The course handbook

Tasks for you to practice during today's course

Work at your own pace!

Divided into tasks and small steps

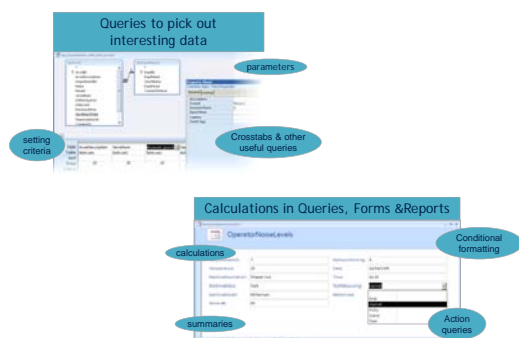
Be selective

Follow-up work

Continue with exercises after the session

Course Clinics

Road map for "Databases: Querying and analysing data"



Getting Started

Getting started with Access

Start Access



Use a desktop icon or Start menu etc.



Access version 2013 in teaching rooms

Enabling active content?
(see appendix in the course book)



Trusting the File Location (Optional)

On the **File** menu, click

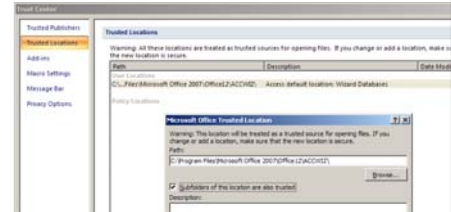
Options

Select the **Trust Center** and

Trust Center Settings...

Select **Trusted Locations** and

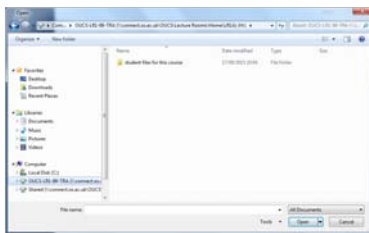
Add new location...



Opening a database file

Open

Today, the exercise files are in your
network drive H:\



Creating a Query

A select query collects data from tables

Combine data from one or more related tables

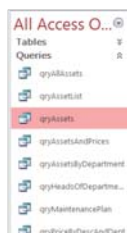
Sort records

Select only some fields

Select only some records

Apply criteria

Queries are listed in the Navigation Pane



Creating a query using the wizard



on **Create** tab

Choose a table (or another query)

Choose some fields

Use > and < buttons

Give the query a name

Query names begin with **qry**

No spaces

Query results

Results are presented in Datasheet View

Only selected fields and selected records are shown

Beware: editing the data here *is changing* the source data in the tables

Query Design View

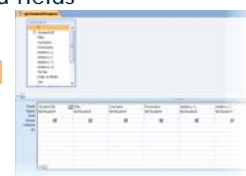
Switch between Design and Datasheet Views



Diagram shows tables, fields and joins

Design grid lists the selected fields

Close and save the query



Using a query

Available queries are listed in the Navigation Pane

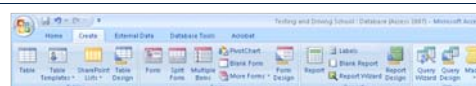
Select a query name and press <Enter>

Or double-click the query name

When query is run, Access presents the *latest* values from the tables

Student ID	Title	Surname	Forename	Address 1	Address 2
1 Mr	Brammer	Robert	10 Plymouth Drive	Stenson Field	
2 Mr	Jenkins	Steven	37 Woodfield Close	Etwell	
3 Miss	Fowler	Sarah	19 Sea View Road	Mullacree	
4 Mr	Barneswood	Michael	25 Lunde Close	Alfreton	
5 Miss	Williams	Charlotte	21 Church Street	Liffesey	
6 Mr	Wendler	David	86 Millford Road	Alfreton	
7 Mrs	Trueman	Mary	156 Station Road	Alfreton	
8 Mr	Spencer	Victoria	73 Mayfield Road	Stenson	
9 Mr	Watson	Greg	7 Denwent Close	Etwell	
10 Mr	Jones	Lyle	121 Ulminster Road	Alfreton	

Creating a query in Design View



Choose a table

Choose some fields

Drag & drop a field name

or double-click a field name

or choose from drop-down list at the top of a column

Wildcard * represents all fields

A query based on two tables

In Query Design View

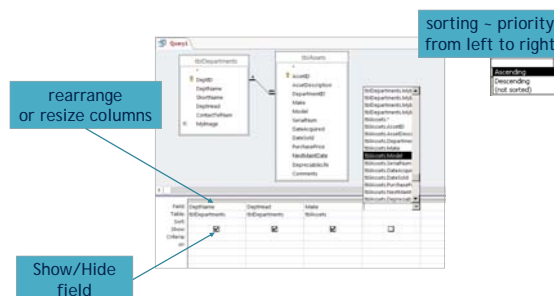


Add all the tables needed for this enquiry

Joins show any relationships which had been set up previously

Any record which appears in both tables will be shown in the results dataset

Changing a query design



Queries for Selecting Data

Query criteria



Limit the records included in the results

Enter a value under one field

All records which exactly match will be included

AND - 2 or more criteria in the same row must all be satisfied

OR - 2 or more criteria on separate rows permits any one to be satisfied

Expressions in criteria

Exact match

Comparisons using > < >= <= Between

Wildcard symbols * ? #

Not

Is Null

Yes/No criteria: use -1 or 0

Databases: Querying and analysing data

Look at Exercise 1 to 5

Drinks and food in the refreshment area only, please

Restart at 10:10

Demonstrators:



More Queries

Parameter queries

A more flexible query

The user sets different criteria to be applied each time

In Criteria, type prompt text

in [square brackets]

Using wildcards in parameter criteria

Like [Which subject]&"*"

Mistakes in criteria may appear as parameter queries



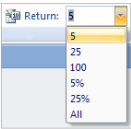
Top or bottom values (Optional)

Selecting top or bottom few from query results

Sort data *first*

Choose:

- Ascending or descending
- A top or bottom range
- Counting or percent



Some special types of query

Create, using the Query Wizard



Find duplicates

12 Personnel	1012	KL809890543	George	H	Carter
13 Personnel	1021	KL809890543	George		Carter

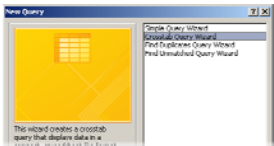
Find unmatched

StudentFirstName	StudentLastName
Lucy	Trueman
	Bowler

Crosstab queries

Used to examine a many-to-many relationship

- Collect records required, using a select query
- Run the Crosstab wizard
- Choose row and column headings
- Choose a field for the Value, and a calculation function



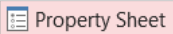
Calculation columns in a query

Add further columns to query design grid

- Expressions using + - * / ^ **And Or Not ()**
- Text concatenation using &

Label text before the colon :

Formats in the Property Sheet



Grouping records in a query
"Totals"

Summarising calculations over all records

Show the **Totals** row



Group by one field

Choose an aggregate functions for every column

Field:	CompanyName	PurchaseOrderNumber
Table:	tblCustomers	tblOrders
Total:	Group By	Count
Sort:		
Show:	<input checked="" type="checkbox"/>	
Criteria:		

Company Name	Count
Fluted Crystal Ltd	2
HiroToko Enterprises	4
Jennie's Jellies	2
Roses Potteries	4
Stitch in Time Ltd	1

Databases: Querying and analysing data

Look at Exercise 6 to 9

Drinks and food in the refreshment area only, please

Restart at 11:15

If you want to continue with the Exercises,
you could ...



Copy the Exercise files to a
memory stick

Download the files (and
more) from
the ITLP Portfolio at
<http://portfolio.it.ox.ac.uk>

Calculations on Forms and Reports

Calculated controls on a form or report

Unbound text control

Insert it in the Detail

Type or build an expression in the Control Source

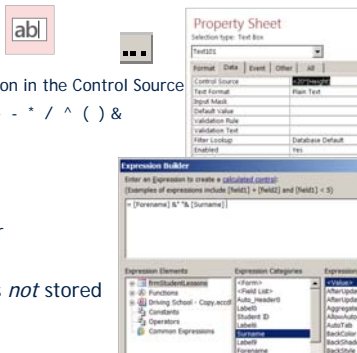
= field values functions + - * / ^ () &

Appearance

Format it

Add a label in the header

The calculated result is *not* stored



Summary calculations on a report

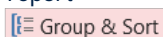
May have set up summary calcs in the wizard

Add simple summary functions

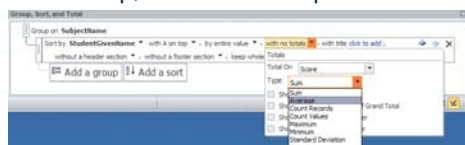
Select a field in the Detail
SUM, AVG, COUNT, MAX etc



Summary calculations on a grouped report (Optional)



Use the Group, Sort and Totals pane



Click **More**

Choose a field and a function

Control is inserted in the appropriate Header/Footer

Running sum (Optional)

A running total of detail

Add a field control into the Detail, bound to the
data field you want to sum

In Properties, Data tab

set Running Sum = Over Group or Over All

Conditional formatting (*Optional*)

Object appearance changes, depending on its value

Set a rule

Set several rules

Compare values

Or show a gradient bar



Action Queries

"Update Values" query



Replaces some data values with new text/numbers

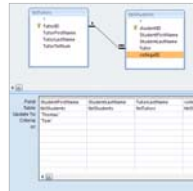
First make a select query

Change it to an Update Query

In the **Update To** row, enter a new value

Run the Query

All selected records are edited



Running action queries



Action queries can be saved

Special icons show in the Navigation Pane

Run a query from the Navigation Pane

Check warnings about altering/removing data

Changes are made *every time* you run an action query

"Make-Table" query



Useful for archiving or re-structuring

Create a Select Query to collect the fields and records as required

Change it into a make-table query

Data is *copied* to the new table

"Delete Records" query



Selected records are deleted

First build a Select Query to select only the intended records

Convert it to a Delete Query

Run the delete query

"Append Records" query



Selected records from the current query are appended to an existing table

Select query collects the fields and records as required (from one or more tables)

Convert to an append query

Confirm the *field names* do match

Nominate the target table - may be in this database or another

Think about the *primary key*

More About Databases

Further courses

Further work with Access:

Databases: User-friendly databases

Databases: Reporting data using Access

MySQL:

Introduction

Further work

If you want to continue with the Exercises, you could ...



Copy the Exercise files to a memory stick

Download the files (and more) from the ITLP Portfolio at <http://portfolio.it.ox.ac.uk>

Databases: Querying and analysing data using Access

Now look at Exercises 10 to 14

Finish at 12:15

Problems?

Come to Course Clinics

help@it.ox.ac.uk

Appointments or phone 8:30am-8:30pm Monday-Friday





This presentation is made available by Pamela Stanworth
under a Creative Commons licence:

Attribution-NonCommercial-ShareAlike
CC BY-NC-SA

pamela.stanworth@it.ox.ac.uk