



# Microsoft Access 2016

## Advanced Queries

IN FOCUS COURSEWARE

Designed to fast-track you through the process of learning about computers and information technology, the *In Focus* range is a unique and innovative concept in learning.

A quick reference summary of key procedures is provided at the bottom of each page together with handy tips and additional information.

Each title in the *In Focus* series can be used as:

- a classroom workbook for instructor-led teaching and training;
- a self-study guide for self-paced learning;
- a tutorial guide for distance education programs;
- a resource collection of just-in-time support and information for help desk users and support staff;
- a handy, desk-side reference for computer users.

This publication has been created using EngineRoom Desktop™ document management and publishing software developed by Watsonia Publishing.

Microsoft Access 2016  
Advanced Queries

**wp:**  
WATSONIA PUBLISHING

# MICROSOFT ACCESS 2016

## ADVANCED QUERIES

---

Copyright © 2017 Melbourne  
by Watsonia Software Pty Ltd (ABN 64 060 335 748)  
Published by Watsonia Publishing



47 Greenaway Street  
Bulleen Vic Australia 3105  
Telephone: (61 3) 9851 4000  
Facsimile: (61 3) 9851 4001  
Web site: [www.watsoniapublishing.com](http://www.watsoniapublishing.com)

Product Code AccAdvQueries

Build: 29/06/17

---

### Trademark Acknowledgments

All terms mentioned in this manual that are known to be trademarks or service marks have been appropriately acknowledged or capitalised. Watsonia Software cannot attest to the accuracy of this information. Use of a term in this manual should not be regarded as affecting the validity of any trademark or service mark.

Screen Shots © 1983-2017 Microsoft. All rights reserved.

### Disclaimer

Every effort has been made to provide accurate and complete information. However, Watsonia Software assumes no responsibility for any direct, indirect, incidental, or consequential damages arising from the use of information in this document. Data and case study examples are intended to be fictional. Any resemblance to real persons or companies is coincidental.

### Copyright Notice

This publication is protected in accordance with the provisions of the Copyright Act. Apart from permissions expressed in the Copyright Act pertaining to copying for study, review, or research, no part of this publication may be reproduced in any form, or stored in a database or retrieval system, or transmitted or distributed in any form by any means, electronic, mechanical photocopying, recording, or otherwise without written permission from Watsonia Software Pty Ltd.

# READ ME FIRST

In case you're not familiar with the terminology, *Read Me First* is quite often the name given to a computer file that contains important information for people to know prior to using an application.

This section contains some important information to help you use this book so we thought we'd start with a *Read Me First* section.

---

## **What skills and knowledge you will acquire...**

The skills and knowledge acquired in Microsoft Access 2016 - Advanced Queries are sufficient to be able to use and operate the software effectively.

## **What you'll need to know before beginning this course...**

Microsoft Access 2016 - Advanced Queries assumes little or no knowledge of the software. However, it would be beneficial to have a general understanding of personal computers and the Windows operating system environment.

## **The objectives of this guide...**

At the completion of this course you should be able to:

- create simple and effective queries
- create queries based on one or more tables
- create and use parameter queries
- create calculated queries
- create and work with aggregation queries
- create and use a series of action queries

## **What you get in a chapter...**

Each chapter begins with a summary page listing the topics covered in that chapter. The chapter then consists of single-page topic sheets pertaining to the theme of the chapter.

## **What you'll need to have before commencing this course...**

Many of the topics in this learning guide require you to open an existing file with data in it. These files can be obtained from your instructor and need the product code for this course which is AccAdvQueries.

## **As you work through this guide...**

It is strongly recommended that you close all open files, if any, prior to commencing each new chapter in this learning guide. Each chapter, where relevant, has its own set of exercise files and any from a previous chapter are no longer required.

## **Where to from here...**

Have a look at the next page which explains how a topic page works, ensure that you have access to the exercise files (see above), and you're ready to make a start.

# WORKING WITH TOPIC SHEETS

The majority of this book comprises single-page topic sheets. There are two types of topic sheets: **task** and **reference**. The layout of both is similar – an overview at the top, *detail* in the centre and

*additional reference* (optional) material at the bottom. Task sheets contain a *Try This Yourself* step-by-step exercise panel in the detail area as shown below.

1 OPENING A DOCUMENT

Although there are a number of different ways to open a Word file, which include using the Start menu or clicking directly on an icon of the file, perhaps the best and simplest way to do it is from within the Word program itself using the File > Open command. The Open dialog box has tools that help you to identify file types and location.

2 Try This Yourself:

Before you begin ensure that Word 2000 has started...

- 1 Select File > Open to display the Open dialog box
- 2 Click on the drop arrow for Look in to display a list of possible locations available to your computer where documents may be found
- 3 Click on Drive C (C:) or its equivalent on your computer
- 4 The contents of drive C: will now be displayed in the Open dialog box...
- 5 Double-click on Course Files For Word 2000 – this is the folder where files for this course can be found
- 6 The contents of the folder Course Files For Word 2000 will now be displayed...
- 7 Click on W002 Document Essentials\_1.doc to select it as the file that you wish to open, then click [Open] to open the document on the screen

3 For Your Reference...

To open a document in Word:

1. Select File > Open to display the Open dialog box
2. Locate the file and folder (if necessary)
3. Click on [Open]

4 Handy To Know...

There is more than one way to open a document in Word. Alternatively you could:

- Click on the Open tool
- Select a recently opened file from the File menu.

5 Skillgate Learning Centres

Page 10

Chapter 2: Working With A Document

- 1 Topic name
- 2 General topic overview provides an introduction to the topic
- 3 *Try This Yourself* (Task-based topic sheets) is a detailed step-by-step practice exercise for you to work through. In *Reference* topic sheets this is usually replaced by a box with reference information.
- 4 In *Task* topic sheets screen shots and graphics provide a visual clue as to what will happen when you work through the *Try This Yourself* practice exercise. In *Reference* topic sheets the screen shots and graphics are used to visually represent information and concepts.
- 5 The *For Your Reference* (optional) element provides a quick summary of the steps required to perform a task. These usually only appear in Task-based topic sheets.
- 6 The *Handy To Know* (optional) element provides additional information such as alternate ways of accomplishing a task or further information providing handy tips.



# CONTENTS

---

<b>Chapter 1 Creating Queries.....</b>	<b>1</b>
Understanding Queries .....	2
Creating A Query Design .....	3
Working With A Query .....	4
Changing A Query Design .....	5
Applying Record Criteria .....	6
Clearing Selection Criteria .....	7
Saving A Query .....	8
Running Queries From The Navigation Pane.....	9
Deleting A Query.....	10
Creating Additional Queries .....	11
<b>Chapter 2 Multi-Table Queries .....</b>	<b>13</b>
Understanding Relational Queries .....	14
Creating A Relational Query Design .....	15
Filtering A Relational Query.....	16
Filtering Related Fields .....	17
Adding More Tables And Fields.....	18
Utilising Hidden Fields .....	19
Understanding Query Joins .....	20
Creating An Inner Join .....	21
Creating A Left Outer Join .....	22
Creating A Right Outer Join .....	23
<b>Chapter 3 Parameter Queries.....</b>	<b>25</b>
Creating A Parameter Query .....	26
Displaying All Records .....	27
Using Parameters To Display A Range .....	28
Using Parameters In Expressions.....	29
Using Parameters With Wildcards .....	30
<b>Chapter 4 Calculations In Queries.....</b>	<b>31</b>
Creating A Calculated Field .....	32
Formatting Calculated Fields .....	33
Summarising Data Using A Query .....	34
Changing The Grouping.....	35
Calculating With Dates.....	36
Using Criteria In Calculations.....	37
Concatenating String Fields.....	38
<b>Chapter 5 Aggregation Queries .....</b>	<b>39</b>
Creating An Aggregation Query.....	40
Working With Aggregation Queries.....	41
Multiple Aggregations .....	42

Modifying Aggregation Headings .....	43
Aggregating Calculated Fields .....	44
Applying Criteria To Aggregates .....	45
Understanding Nested Queries.....	46
Creating An Assembly Query.....	47
Creating The Nesting Query .....	48
<b>Chapter 6 Action Queries .....</b>	<b>49</b>
Creating A Make Table Query .....	50
Using A Make Table Query.....	51
Understanding Expressions And Update Queries .....	52
Preparing An Update Query.....	53
Running An Update Query.....	54
Updating Using Expressions.....	55
Running An Expression-Based Update .....	56
Creating A Delete Query.....	57
Running A Delete Query .....	58
Creating An Append Query.....	59
Running An Append Query .....	60
Turning Action Messages Off.....	61

## CHAPTER 1

# CREATING QUERIES

 INFOCUS

You can easily sort and locate data using table sorting and filtering options. However, these are relatively lightweight when compared to using queries.

Queries are sometimes known as **Query By Example**, or **QBE**. A query is like a report based on the data in a table. With a query you are required to specify which fields to see in the query, and which records to display.

This is done by giving Access **criteria** to search for, much the same as what is done for a simple filter. For example, listing all of the records that have **Sales** in the **Department** field, or all of the records that occurred on **12/3/2016**, are examples of queries.

Once the **criteria** and **output fields** are nominated, Access will search through the data and produce a table of matching records.

**In this session you will:**

- ✓ gain an understanding of how queries work
- ✓ learn how to create a new query design
- ✓ learn how to work with a query
- ✓ learn how to modify a query design
- ✓ learn how to apply record criteria for a query
- ✓ learn how to clear selection criteria
- ✓ learn how to save a query
- ✓ learn how to run queries from the **Navigation** pane
- ✓ learn how to delete a query from a database file
- ✓ gain an understanding of creating additional queries.

# UNDERSTANDING QUERIES

A **select query** is like a filter that you place on your data so that you see only the information that is relevant to you. Select queries can be used, for example, to produce a list of customers

from Tasmania, or all of the items that you've purchased in the last six months valued at \$300 or more. Select queries are so named because they *select* records according to your query design.

Select queries are created using the **Create** tab of the ribbon, and are run and modified as a **Query object** in the **Navigation pane**.

Select queries are based on a **Query Design**. The upper part of the design is known as the **Field List**, while the lower portion is known as the **Query Grid**.

*Field list  
Query grid*

The screenshot shows the Microsoft Access Query Design View. At the top is the Field List, which displays the fields from the 'Employees' table: EmpNo, FirstName, LastName, Department, and PhoneNo. Below the Field List is the Query Grid, which is a table with columns for Field, Table, Sort, Show, and Criteria. The 'Field' column lists 'EmpNo', 'LastName', 'FirstName', 'Department', and an empty cell. The 'Table' column lists 'Employees' five times. The 'Sort' and 'Show' columns have checkboxes. The 'Criteria' row contains criteria for 'EmpNo' (containing '107') and 'Department' (containing 'Administration').

The records displayed in the query are determined by the sample data that you enter into the **Criteria row** in the **Query Grid** – this is why the process is sometimes referred to as *query by example*.

*Criteria row*

This screenshot shows the same Query Design View as above, but with a red arrow pointing to the 'Criteria' row in the Query Grid. The 'Criteria' row now includes the value '107' in the 'EmpNo' column and 'Administration' in the 'Department' column.

The easiest way to see the data is to switch to **Datasheet** view. In **Datasheet** view the data that matches the query criteria is displayed in a special **dynaset** table. A **dynaset** is a subset of the full table of data – however, it is still a live set of data and any changes made to data here will be reflected back in the full table later on.

The screenshot shows the Microsoft Access Datasheet view for 'Query1'. It displays a table with columns: EmpNo, LastName, FirstName, and Department. The data shows employees from the 'Administration' department. The first record is highlighted with a yellow background. The status bar at the bottom indicates 'Record: 14 of 26'.

EmpNo	LastName	FirstName	Department
107	Millson	Augustine	Administration
108	Bennet	Amanda	Administration
109	Samuelson	George	Administration
110	Smith	Neville	Administration
111	Henricks	Petra	Administration
112	Clark	Vivienne	Administration
113	Hancock	Jerry	Administration
114	Brown	Victor	Administration
115	Kendall	Sandra	Administration
117	Morris	Charles	Administration
118	Williams	Lance	Administration
181	Rellote	Leigh	Administration
183	Chapman	Michael	Administration
184	Ivanson	Natalie	Administration
185	Andronikos	Pavlos	Administration
186	Taron	Steve	Administration
188	Cummings	John	Administration
189	Isaac	Ajith	Administration
190	Alexopoulos	Aris	Administration
191	Thurst	Brett	Administration

# CREATING A QUERY DESIGN

Queries are created from the **Create** tab on the ribbon. Like table structures, there is a **design** view where the layout, criteria, and the like, required for the query are specified, and a **run**

view where the data is brought into the design layout structure from the relevant table. The first step in creating a query, therefore, is to create a query design structure.

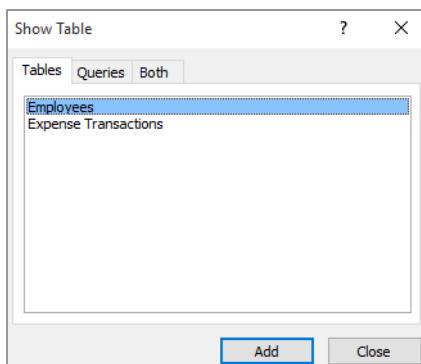
## Try This Yourself:

**Open File**

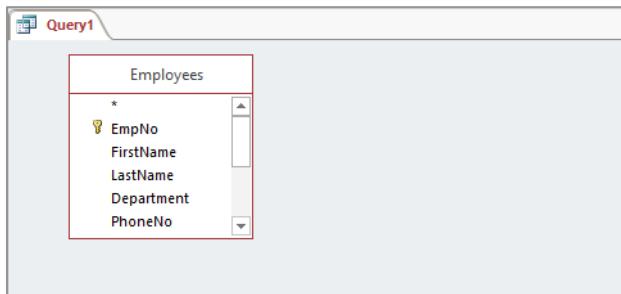
Before starting this exercise you **MUST** open the file *Creating Queries\_1.accdb*...

- 1 Click on the **Create** tab, then click on **Query Design** in the **Queries** group to display a new query design and the **Show Table** dialog box
- 2 Click on **[Add]** to add the **Employees** table fields to the design, then click on **[Close]** to close the dialog box
- 3 In the field listing double-click on **EmpNo**, **LastName**, **FirstName** and **Department** to add these fields to the grid in this order
- 4 Click on **Save** in the QAT to display the **Save As** dialog box
- 5 Type **qryEmployees** in **Query Name**, then click on **[OK]**  
*The name of the query will now appear in the Navigation pane under the Queries header...*
- 6 Close the query

1

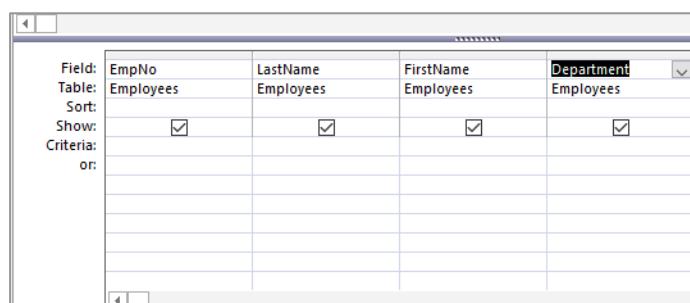


2



Double-click on the entries here to add them to the table below

3



## For Your Reference...

To **create a query design**:

1. Click on the **Create** tab
2. Click on **Query Design** in the **Queries** group
3. Add the table and select the fields
4. Save the query

## Handy to Know...

- The **Show Table** dialog box, displayed when creating a new query design, lists all of the tables and queries in the current database file.

# WORKING WITH A QUERY

Queries offer you the ability to see snapshots of your data – a particular view or representation of your data at a point in time. There are three main views within a query: the **design** view where you

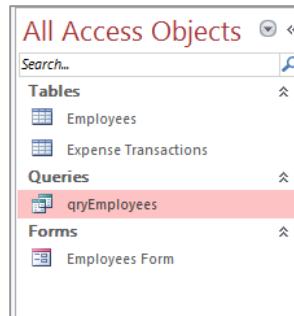
specify what data you wish to see in the snapshot; the **datasheet** view where the data based on the design is displayed; and **SQL** view which shows the programming behind the query.

## Try This Yourself:

**Same File**

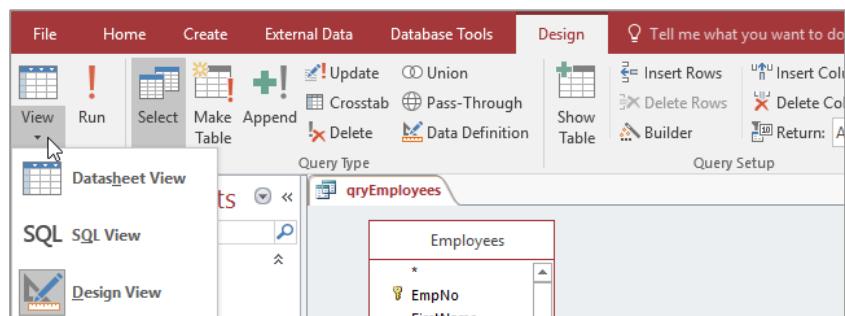
Continue using the previous file with this exercise, or open the file *Creating Queries\_2.accdb...*

- 1 Double-click on **qryEmployees** to see the query in **Datasheet** view, displaying the data
- 2 On the **Home** tab, click on the top half of **View** in the **Views** group to toggle between **Design** and **Datasheet** views
- 3 Click on the bottom half of **View** to display a menu of options
- 4 Select **SQL View** to see the **SQL** code required behind the scenes to create the query
- 5 Close the query

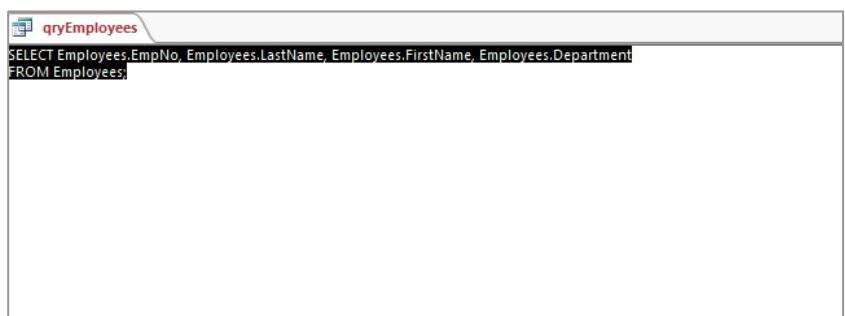


The screenshot shows the 'All Access Objects' dialog box. In the center, there is a table titled 'qryEmployees' with columns: EmpNo, LastName, FirstName, and Department. The first row (EmpNo 101) is highlighted. To the left of the table, there is a list of objects: Tables (Employees, Expense Transactions), Queries (qryEmployees), and Forms (Employees Form). The 'qryEmployees' query is currently selected.

1



2



The screenshot shows the 'qryEmployees' query in 'SQL View'. The screen displays the SQL code: 'SELECT Employees.EmpNo, Employees.LastName, Employees.FirstName, Employees.Department FROM Employees;'. The 'Design' tab is still visible in the ribbon.

3

## For Your Reference...

To **see different aspects** of a **query**:

1. Double-click on the query to see it in **Datasheet** view
2. On the **Home** tab, click on the top half of **View** in the **Views** group to toggle between **Design** and **Datasheet** views

## Handy to Know...

- Until you seriously get into programming, you won't use the **SQL View** option for queries all that often. SQL is pronounced "sequel" or simply S.Q.L.

# CHANGING A QUERY DESIGN

Most **query designs** are not as critical as table designs and can therefore be changed randomly and when the need arises. **Select queries**, where you are trying to extract matching data,

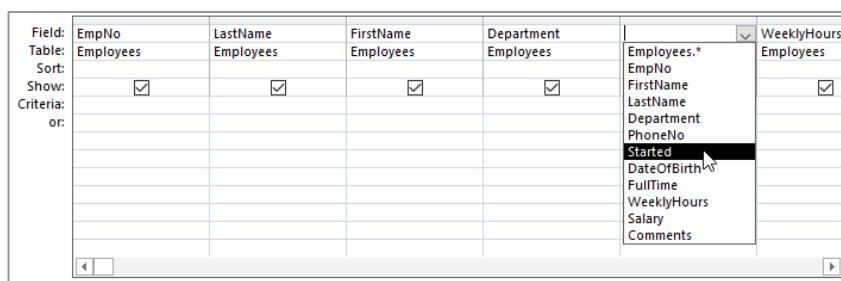
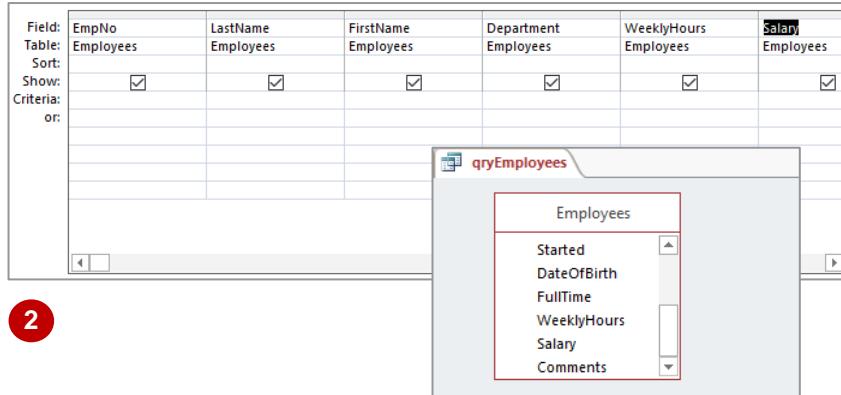
are often run using a trial and error approach where the query design is experimented with and modified until the perfect solution is found.

## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file Creating Queries\_3.accdb...

- 1 In the **Navigation** pane, right-click on **qryEmployees** to display a menu of options, then select **Design View** to see the query in **Design** view
- 2 Scroll down the list of fields in the **Employee** field listing and double-click on **WeeklyHours** and then **Salary** to place both fields at the end of the grid
- 3 Click on **WeeklyHours** in the grid then, on the **Query Tools: Design** tab, click on **Insert Columns** in the **Query Setup** group  
*A new, blank column will appear...*
- 4 Click on the drop arrow in the new column and select **Started**
- 5 Click on the **Home** tab, then click on the top half of **View** in the **Views** group to run the query and see the data presented in the modified design
- 6 Click on **Save** in the **QAT**, then close the query



4

qryEmployees							
EmpNo	LastName	FirstName	Department	Started	WeeklyHou	Salary	
101	Kerr	Julianne	Executive	28-Jun-10	40	\$250,000.00	
102	Jones	Harry	Executive	19-Jul-10	40	\$140,000.00	
103	Harrington	Angel	Executive	19-Jul-10	40	\$145,000.00	
104	Dawson	Peter	Executive	19-Jul-10	40	\$140,000.00	
105	Jones	Mark	Executive	19-Jul-10	40	\$132,000.00	
106	Grayson	Maureen	Occupational S	06-Sep-10	40	\$85,000.00	
107	Millson	Augustine	Administratior	06-Sep-10	40	\$85,000.00	
108	Bennet	Amanda	Administratior	06-Sep-10	40	\$87,000.00	
109	Samuelson	George	Administratior	06-Sep-10	40	\$98,000.00	

5

## For Your Reference...

To **insert more fields** into a **Design grid**:

- Double-click on the field name in the field listing, or
- Click in the grid, then click on **Insert Columns** in the **Query Setup** group

## Handy to Know...

- You can delete a field from a query grid by clicking on it and then clicking on **Delete Columns** in the **Query Setup** group on the **Home** tab.

# APPLYING RECORD CRITERIA

The real power of a query lies in its ability to display a filtered list of records in a ***dynaset***. To filter the records and see only the ones that you want, you will need to enter search criteria in the

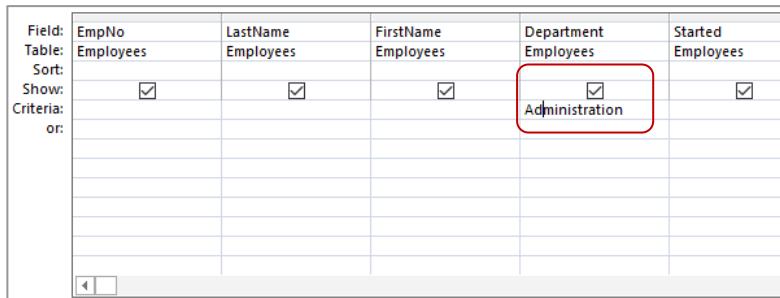
criteria row in the query grid. You simply type an example of the data that you want to see in the criteria cell and run the query to display all records from the original table that match the criteria.

## Try This Yourself:

Same File

Continue using the previous file with this exercise, or open the file *Creating Queries 4.accdb*...

- 1 Right-click on **qryEmployees** to display a menu of options and select **Design View**
  - 2 Click in the **Criteria** cell for **Department** and type **Administration**
  - 3 On the **Query Tools: Design** tab, click on the top half of **View** in the **Views** group to see only those records with **Administration** in the **Department** field
  - 4 Repeat step 3 to switch back to **Design** view
  - 5 Type **40** in the **Criteria** cell for **WeeklyHours**, then click on **View** to display only those people who work **40** hours in the **Administration** department
  - 6 Switch to **Design** view and type **>=80000** in the **Criteria** cell for **Salary**, then click on **View** to see all **Administration** people who work **40** hours and earn **\$80,000** or more
  - 7 Save and close the query



2

qryEmployees						
EmpNo	LastName	FirstName	Department	Started	WeeklyHou	
107	Millson	Augustine	Administration	06-Sep-10	40	
108	Bennet	Amanda	Administration	06-Sep-10	40	
109	Samuelson	George	Administration	06-Sep-10	40	
110	Smith	Neville	Administration	06-Sep-10	40	
111	Henricks	Petra	Administration	06-Sep-10	40	
112	Clark	Vivienne	Administration	06-Sep-10	40	
113	Hancock	Jerry	Administration	06-Sep-10	40	
114	Brown	Victor	Administration	06-Sep-10	40	
115	Kendall	Sandra	Administration	06-Sep-10	40	

3

Last Name	First Name	Department	Started	Weekly Hours	Salary
Millson	Augustine	Administration	06-Sep-10	40	\$85,000.00
Bennet	Amanda	Administration	06-Sep-10	40	\$87,000.00
Samuelson	George	Administration	06-Sep-10	40	\$98,000.00
Henricks	Petra	Administration	06-Sep-10	40	\$82,000.00
Clark	Vivienne	Administration	06-Sep-10	40	\$80,000.00
Brown	Victor	Administration	06-Sep-10	40	\$81,000.00
Kendall	Sandra	Administration	06-Sep-10	40	\$88,000.00
Morris	Charles	Administration	06-Sep-10	40	\$84,000.00
Williams	Lance	Administration	23-Sep-10	40	\$83,000.00
				0	\$0.00

6

## **For Your Reference...**

To **select records** in a **query**:

1. Click in the **Criteria** cell for a field and type the desired search criteria
  2. On the **Query Tools:Design** tab, click on **View** in the **Views** group to run the query.

### **Handy to Know...**

- When creating queries, if you add more criteria across fields, you are creating what is known as an **AND** query – you want records that have this AND this AND this...
  - When constructing queries, use **>** for greater than and **<** for less than situations.

# CLEARING SELECTION CRITERIA

You do need to exercise a little care when running queries. If you leave residual criteria from an earlier query in the query grid (which is easy to do if you have more fields than can be seen on

the screen), you may end up with incorrect results. It is a good idea therefore to clear the selection criteria after you have performed a query and found the data that you want.

## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file *Creating Queries\_5.accdb*...

- 1 In the **Navigation** pane, right-click on **qryEmployees** to display a menu of options and select **Design View**
- 2 Point to the left of the first criteria cell until the mouse pointer changes to a black horizontal arrow ➔
- 3 Click once to select the entire criteria row
- 4 Press **Del** to delete all of the criteria in the row
- 5 Save and close the query

Field:	EmpNo	LastName	FirstName	Department	Started
Table:	Employees	Employees	Employees	Employees	Employees
Sort:					
Show:	<input checked="" type="checkbox"/>				
Criteria:	>0			"Administration"	
or:					

2

Field:	EmpNo	LastName	FirstName	Department	Started
Table:	Employees	Employees	Employees	Employees	Employees
Sort:					
Show:	<input checked="" type="checkbox"/>				
Criteria:				"Administration"	
or:					

3

Field:	EmpNo	LastName	FirstName	Department	Started
Table:	Employees	Employees	Employees	Employees	Employees
Sort:					
Show:	<input checked="" type="checkbox"/>				
Criteria:					
or:					

4

## For Your Reference...

To **clear selection criteria**:

1. Point to the left of the row and click once to select it
2. Press **Del** to delete the criteria in the row

## Handy to Know...

- When working with a query design, you can delete the contents of a single cell in the **Criteria** row by double-clicking on the value in the cell and pressing **Del**.

# SAVING A QUERY

There are two main types of select queries: those that you create as a one-off search of the data; and those that you create for repeated and on-going use. If you are going to use a query on

a regular basis it should be saved. You can then also use it as a template to create other queries with variations perhaps to the criteria or the field grid.

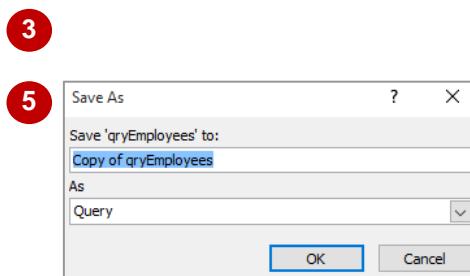
## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file Creating Queries\_6.accdb...

- 1 Right-click on **qryEmployees** to display a menu of options and select **Design View**
- 2 Type **Administration** in the **Criteria** cell for **Department**
- 3 On the **Query Tools: Design** tab, click on **View** in the **Views** group to see the results – there should be 26 records
- 4 Click on the **File** tab, then click on **Save As** to display the **Save As** area
- 5 Click on **Save Object As** in **File Types**, then click on **[Save As]** to display the **Save As** dialog box
- 6 Type **qryEmployeesAdmin** and click on **[OK]**  
*The new query appears in the Navigation bar...*
- 7 Switch to **Design** view then repeat steps **2** to **6** to create another query that only displays employees from the **Executive** department – save this query as **qryEmployeesExec**
- 8 Close the query

189	Isaac	Ajith	Administrator	16-Dec-10	40
190	Alexopoulos	Aris	Administrator	27-Nov-10	40
191	Thurst	Brett	Administrator	16-Dec-10	40
192	Ahlund	Christof	Administrator	09-Dec-10	40
193	Zylinski	David	Administrator	20-Nov-10	32
194	Hurst	Ellinor	Administrator	27-Nov-10	40
203	Hutchins	Philip	Administrator	27-Nov-10	40
204	Baker-Smith	Susan	Administrator	16-Dec-10	40
205	Abelseth	Trond	Administrator	02-Dec-10	25
*					0



EmpNo	LastName	FirstName	Department	Started	Week
101	Kerr	Julianne	Executive	28-Jun-10	
102	Jones	Harry	Executive	19-Jul-10	
103	Harrington	Angel	Executive	19-Jul-10	
104	Dawson	Peter	Executive	19-Jul-10	
105	Jones	Mark	Executive	19-Jul-10	
*					

## For Your Reference...

To **save a query**:

1. Create the query
2. On the **File** tab, click on **Save As**, then click on **Save Object As** and click on **[Save As]**
3. Type a name and click on **[OK]**

## Handy to Know...

- It is important to give your queries meaningful names so that you remember what they are for. Using a prefix, such as **qry**, will tell you at a glance that you are looking at a list of queries and make the queries easier to distinguish from tables, forms and reports.

# RUNNING QUERIES FROM THE NAVIGATION PANE

Queries store the layout, fields, criteria and other information required to produce the list of data that you want. Given that they can be time consuming to create, especially in the case of

complex queries, it makes sense to save them and then run them as often as you require. Queries can be run directly from the object listing in the **Navigation** pane, as often as you like.

## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file *Creating Queries\_7.accdb*...

- In the **Navigation** pane under **Queries**, double-click on **qryEmployees**, then double-click on **qryEmployeesAdmin**, then double-click on **qryEmployeesExec**

Notice how the names of the three queries appear in three separate tabs at the top of the window. The last query opened is the one that is currently seen and is known as the “active” query...

- Click on the tab for **qryEmployeesAdmin** to see the employees in the **Administration** department
- Click on the tab for **qryEmployees** to see all of the employees
- Close each query

EmpNo	LastName	FirstName	Department	Started	WeeklyHou	Salary
101	Kerr	Julianne	Executive	28-Jun-10	40	\$250,000.00
102	Jones	Harry	Executive	19-Jul-10	40	\$140,000.00
103	Harrington	Angel	Executive	19-Jul-10	40	\$145,000.00
104	Dawson	Peter	Executive	19-Jul-10	40	\$140,000.00
105	Jones	Mark	Executive	19-Jul-10	40	\$132,000.00
*					0	\$0.00

1

EmpNo	LastName	FirstName	Department	Started	WeeklyHou	Salary
107	Millson	Augustine	Administratior	06-Sep-10	40	\$85,000.00
108	Bennet	Amanda	Administratior	06-Sep-10	40	\$87,000.00
109	Samuelson	George	Administratior	06-Sep-10	40	\$98,000.00
110	Smith	Neville	Administratior	06-Sep-10	40	\$78,000.00
111	Henricks	Petra	Administratior	06-Sep-10	40	\$82,000.00
112	Clark	Vivienne	Administratior	06-Sep-10	40	\$80,000.00
113	Hancock	Jerry	Administratior	06-Sep-10	40	\$79,000.00

2

EmpNo	LastName	FirstName	Department	Started	WeeklyHou	Salary
101	Kerr	Julianne	Executive	28-Jun-10	40	\$250,000.00
102	Jones	Harry	Executive	19-Jul-10	40	\$140,000.00
103	Harrington	Angel	Executive	19-Jul-10	40	\$145,000.00
104	Dawson	Peter	Executive	19-Jul-10	40	\$140,000.00
105	Jones	Mark	Executive	19-Jul-10	40	\$132,000.00
106	Grayson	Maureen	Occupational S	06-Sep-10	40	\$85,000.00
107	Millson	Augustine	Administratior	06-Sep-10	40	\$85,000.00

3

## For Your Reference...

To run a query from the **Navigation** pane:

- In the **Navigation** pane, double-click on the name of the query from the **Query** object list

## Handy to Know...

- Queries do not contain data. Each time a query is opened in **Datasheet** view, Access retrieves the latest data from the table upon which the query is based and uses the query design to display the relevant records and information.

# DELETING A QUERY

Queries often work with data that is stored in tables or that results from other queries. They can be used to create data by performing calculations and can be used as a source of data

for other queries, forms and reports. Therefore, you should be especially careful when deleting queries – make sure that the query is not used by any other objects in the database first.

## Try This Yourself:

**Same File** Continue using the previous file with this exercise, or open the file *Creating Queries\_7.accdb*...

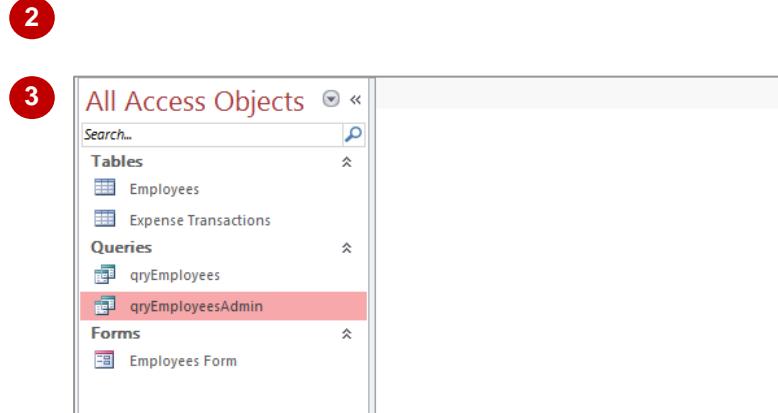
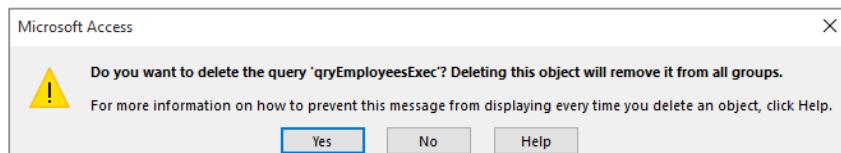
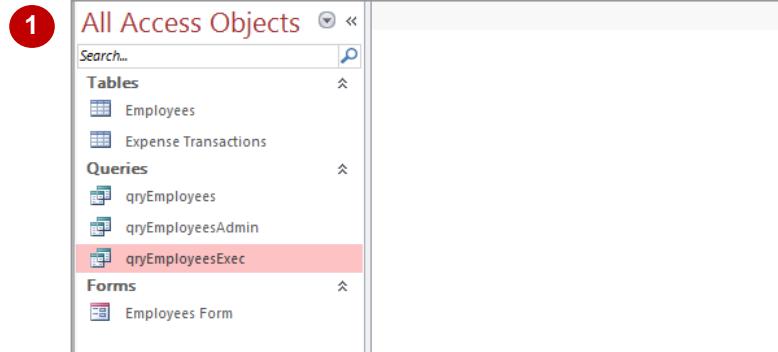
**1** Click on **qryEmployeesExec** in the **Navigation** pane to select it

**2** On the **Home** tab, click on **Delete** in the **Records** group

A warning message will appear, seeking your confirmation to delete the query...

**3** Click on **[Yes]** to confirm the deletion

The query no longer appears listed under **Queries** in the **Navigation** pane



## For Your Reference...

### To delete a query from a database file:

1. Click on the name of the query in the **Navigation** pane
2. On the **Home** tab, click on **Delete** in the **Records** group

## Handy to Know...

- You can delete a query by clicking on it in the **Navigation** pane and pressing **Del**.

# CREATING ADDITIONAL QUERIES

Select queries are by far the most common type of query that you will create and use. In this assignment you will have the opportunity to put your understanding of queries to use by creating

a wide range of different queries, including those that show records that match specific criteria, and those that fit within specific ranges of dates.

## Sub Heading

Use the qryEmployees query to run the various queries as shown. Note that you will have to clear the criteria from time to time. Also, we won't need these queries so there is no need to save them.

Save the final query design as *qryEmployeesNew* then close it.

The datasheet that shows the results is exactly like a table and you can therefore use the Print commands on the **File** tab to print the result once the datasheet is displayed.

Access automatically places quotation marks around criteria based on text. The quotation marks are programming symbols that tell the computer to treat the data as character strings rather than numbers.

## NOTES:



CHAPTER 2

## MULTI-TABLE QUERIES

IN FOCUS

The real benefits associated with relational databases and queries become clear when you need to create a query that draws on data from several tables.

With single tables you can create filters that essentially filter out unwanted data and display only what you need to see.

To draw data from several tables, however, you are better advised to create a query. Queries that span multiple tables actually use the underlying table join settings to extract data quickly and easily for either one-off informational queries or full scale report-destined queries.

**In this session you will:**

- ✓ gain an understanding of how relational queries work
- ✓ learn how to create a relational query design
- ✓ learn how to filter a relational query
- ✓ learn how to filter related fields and some of the associated problems
- ✓ learn how to add more tables and fields to a query design
- ✓ learn how to work with hidden fields in the query grid
- ✓ gain an understanding of the three join types in **Access**
- ✓ learn how to create an inner join query
- ✓ learn how to create a left outer join
- ✓ learn how to create a right outer join.

# UNDERSTANDING RELATIONAL QUERIES

A query is like a filter that you can place on your data so that you can see or work with only the information that is relevant. Queries are used to display data (**Select** query) or to change data

(**Update** query). **Relational queries** involve one or more linked tables and are the ideal way to create reports comprising of data from several different tables.

## How Relational Queries Work

All queries, including relational queries, are based on a **Query Design**.

The upper part of the design is known as the **Field List**, where Access displays the table or tables used for the query. If more than one table is chosen and the chosen tables are related, relational links will show how the tables are connected.

The lower part of the query design is known as the **Query Grid**, where Access displays the fields from the tables that need to be used in the query operation. You can choose all fields to display, or a subset of the fields, which is more common. The **Query Grid** is also used to enter examples of the data for the query to search for. This sample data is known as **criteria**.

The screenshot shows the Microsoft Access Query Design View. At the top, there are two tables in the Field List: 'Employees' and 'Expense Transactions'. The 'Employees' table has fields: EmpNo, FirstName, LastName, DeptID, and PhoneNo. The 'Expense Transactions' table has fields: ExpTransNo, ExpDate, Amount, Employee, and Expense Type. A relational link is shown between the 'Employee' field in the 'Employees' table and the 'Employee' field in the 'Expense Transactions' table. Below the Field List is the Query Grid, which contains four columns corresponding to the joined tables. The first column is 'Field:' with dropdown menus for 'Table:' and 'Criteria:'. The second column is 'Table:' with dropdown menus for 'Table:' and 'Criteria:'. The third column is 'Sort:' with dropdown menus for 'Table:' and 'Criteria:'. The fourth column is 'Criteria:' with dropdown menus for 'Table:' and 'Criteria:'. The grid itself has rows for 'Field:', 'Table:', 'Sort:', and 'Criteria:'.

*The Field List displays the related tables – note how the links provide information about the type of join.*

*The Query Grid contains the settings that determine the data that will be presented.*

In the example above, fields from all of the tables have been pulled into the query grid. Because of the nature of the joins (one-to-many) the resultant query will use the data from the transaction table (**Expense Transactions**) and supplement it with information from the two lookup tables (**Employees** and **Expense Type**).

The criteria above is asking Access to present us with all transactions on or after March 1, 2015, and for either meals or postage greater than or equal to \$50. When run, the resultant query would appear as:

EmpNo	FirstName	LastName	ExpTransNo	ExpDate	Amount	Employee	Expense Type
104	Peter	Dawson	1	2/01/2015	\$132.00	Dawson	Accommodation
101	Julianne	Kerr	2	2/01/2015	\$145.00	Kerr	Accommodation
134	Syed	Ali	3	2/01/2015	\$27.06	Ali	Gifts
120	Belinda	Moore	4	2/01/2015	\$3.59	Moore	Postage
117	Charles	Morris	5	2/01/2015	\$16.99	Morris	Postage
112	Vivienne	Clark	6	2/01/2015	\$154.50	Clark	Accommodation
107	Augustine	Millson	7	2/01/2015	\$125.50	Millson	Accommodation
158	Elizabeth	Dangaard	8	2/01/2015	\$48.39	Dangaard	Other
114	Victor	Brown	9	2/01/2015	\$18.26	Brown	Subsistence
153	Victoria	McDonald	11	2/01/2015	\$123.44	McDonald	Accommodation
118	Lance	Williams	12	2/01/2015	\$237.66	Williams	Accommodation
107	Augustine	Millson	13	2/01/2015	\$52.86	Millson	Subsistence
107	Augustine	Millson	14	16/01/2015	\$155.60	Millson	Accommodation
*				(New)			

# CREATING A RELATIONAL QUERY DESIGN

Irrespective of whether you are creating a query for a single table or for multiple tables, the query is originally created from options on the **Create** tab on the ribbon. The key to multiple-table

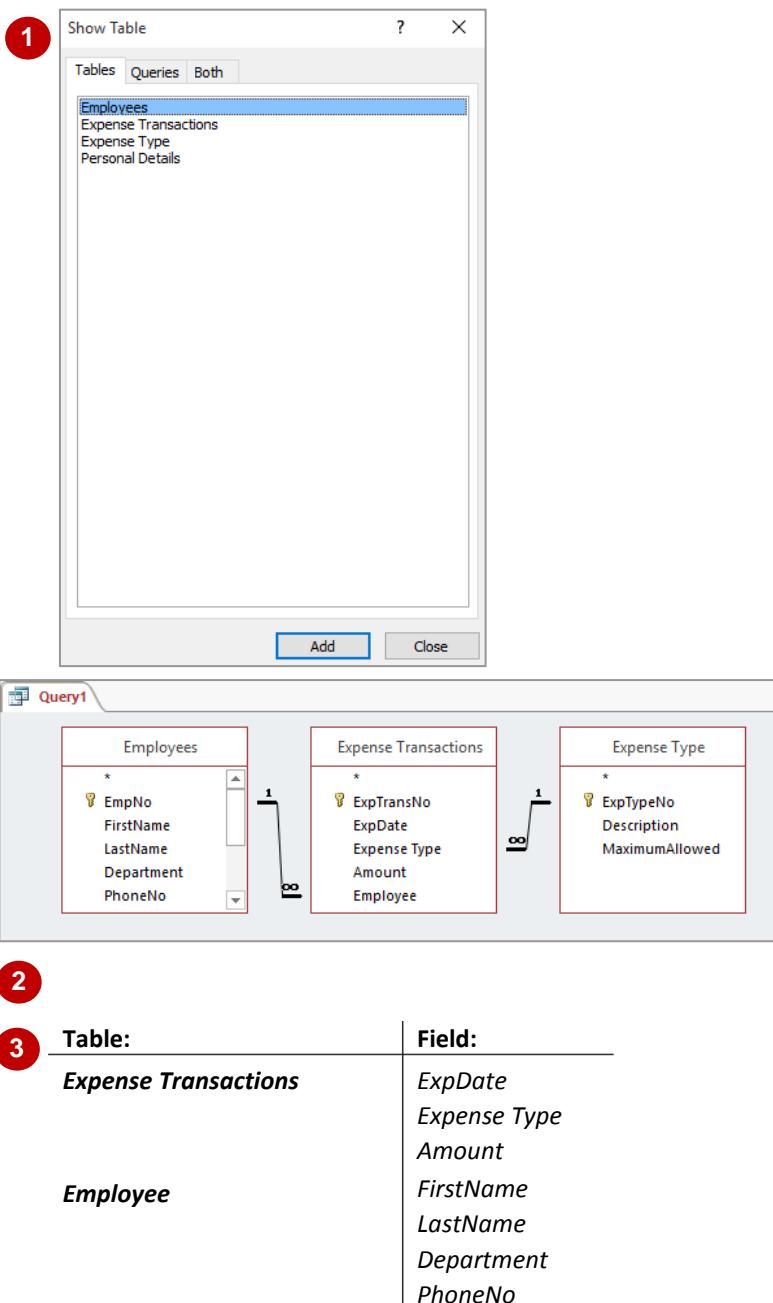
queries is selecting the tables to use when the **Show Table** dialog box appears. Here you can select which tables to pull into the **Field List** on the query design.

## Try This Yourself:

**Open File**

Before starting this exercise you **MUST** open the file *Multi Table Queries\_1.accdb*...

- 1 Click on the **Create** tab, then click on **Query Design** in the **Queries** group to display the **Show Table** dialog box
- 2 Double-click on the tables: **Employees**, **Expense Transactions** and **Expense Type**, then click on **[Close]**
- 3 Double-click on the fields, as listed, to populate the query grid
- 4 Click on **Save** in the QAT to display the **Save As** dialog box
- 5 Type **Employee Expenses** in **Query Name**, then click on **[OK]**  
*The new query is listed in the Navigation pane...*
- 6 On the **Query Tools: Design** tab, click on **Run** to run the query
- 7 Close the query



## For Your Reference...

To *create a relational query*:

1. Click on the **Create** tab and click on **Query Design** in the **Queries** group
2. Double-click on the tables to add
3. Double-click on the fields to add to the query grid

## Handy to Know...

- It doesn't matter in which order you place tables in the **Field List** in the query because how the data is presented when the query is run is determined by the relationship joins.

# FILTERING A RELATIONAL QUERY

A relational query is filtered in the same way as a single table query by providing examples of the data in the criteria cells displayed in the query grid. Text field examples need to be enclosed in

quotation marks, but Access will do this for you. Numeric and date fields can make use of operators such as greater than, less than, equal to, and so on.

## Try This Yourself:

Same  
File

*Continue using the previous file with this exercise, or open the file Multi Table Queries 2.accdb...*

- 1 Open the query **Employee Expenses** in **Design View**
  - 2 Click in the **Criteria** cell for **Amount** and type **>150**
  - 3 On the **Query Tools: Design** tab, click on **Run** in the **Results** group to display all of the transactions greater than **150**
  - 4 On the **Home** tab, click on the top half of **View** in the **Views** group to return to **Design View**, then delete the previous criteria and try each of the examples as shown (delete the previous criteria after each one)
  - 5 Close the query – click on **[No]** to close without saving the design changes

2

Employee Expenses					
ExpDate	Expense Type	Amount	FirstName	LastName	Department
2/01/2015	Accommodation	\$154.50	Vivienne	Clark	Administration
16/01/2015	Accommodation	\$155.60	Augustine	Millson	Administration
2/02/2015	Accommodation	\$254.42	Peter	Dawson	Executive
2/02/2015	Accommodation	\$281.36	Julianne	Kerr	Executive
4/02/2015	Accommodation	\$295.81	Vivienne	Clark	Administration
4/02/2015	Accommodation	\$239.10	Augustine	Millson	Administration
6/02/2015	Accommodation	\$154.11	Victoria	McDonald	Research & Development
18/02/2015	Accommodation	\$267.48	Augustine	Millson	Administration
2/03/2015	Accommodation	\$458.88	Peter	Dawson	Executive

3

4	Field:	Criteria:
	<i>Department</i>	<i>Executive</i>
	<i>ExpDate</i>	<i>&gt;01/07/08</i>
	<i>ExpDate</i>	<i>Between 01/06/08 And 30/06/08</i>

## **For Your Reference...**

To *filter* a *relational query*:

1. Open the query in **Design View**
  2. Type a relevant example or expression in the **Criteria** cell of the appropriate fields

### **Handy to Know...**

- If you save a query as it is closed, the last criteria example used will be saved with the query. If you want to keep the **Criteria** cells clear, don't save the query.

# FILTERING RELATED FIELDS

The real strength of ***related queries*** is not only being able to choose fields from different tables for the query, but also to be able to filter them and provide them with criteria values. While this

works well for most fields, there are a few problems when trying to filter on the lookup fields that have been used to create the relationship between the tables.

## Try This Yourself:

Same  
File

*Continue using the previous file with this exercise, or open the file Multi Table Queries 2.accdb...*

- 1 Open the query **Employee Expenses** in **Design View**
  - 2 Click in the **Criteria** cell for **Expense Type** and type **Meals**
  - 3 On the **Query Tools: Design** tab, click on **Run** in the **Results** group

*An error message will appear indicating a data mismatch. This has occurred because Expense Type is actually a numeric field which looks up the Expense Type table using the ExpTypeNo field...*
  - 4 Click on **[OK]** to close the message
  - 5 Select the value in **Criteria** for **Expense Type** and type **6**

*This is the ExpTypeNo for Meals...*
  - 6 On the **Query Tools: Design** tab, click on **Run** in the **Results** group to display all of the transactions for **Meals**
  - 7 Close the query – click on **[No]** to discard the changes

An error message will appear indicating a data mismatch. This has occurred because Expense Type is actually a numeric field which looks up the Expense Type table using the ExpTypeNo field...

Click on **[OK]** to close the message

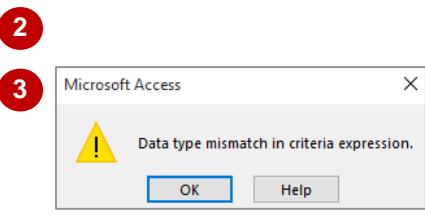
Select the value in **Criteria** for **Expense Type** and type 6

*This is the ExpTypeNo for Meals...*

On the **Query Tools: Design** tab, click on **Run** in the **Results** group to display all of the transactions for **Meals**

Close the query – click on **[No]**  
to discard the changes

Field:	ExpDate	Expense Type	Amount	FirstName	LastName
Table:	Expense Transactions	Expense Transactions	Expense Transactions	Employees	Employees
Sort:					
Show:	<input checked="" type="checkbox"/>				
Criteria:	Meals				
or:					



Employee Expenses				
ExpDate	Expense Type	Amount	FirstName	LastName
2/01/2015	Meals	\$52.86	Augustine	Millson
6/02/2015	Meals	\$60.13	Augustine	Millson
6/03/2015	Meals	\$60.59	Augustine	Millson
6/04/2015	Meals	\$60.77	Augustine	Millson
22/04/2015	Meals	\$61.03	Augustine	Millson
22/05/2015	Meals	\$61.62	Augustine	Millson
6/06/2015	Meals	\$61.93	Augustine	Millson

## **For Your Reference...**

To *filter related fields*:

1. Type examples into the **Criteria** cell of any field in the query grid
  2. Take care to compensate for lookup fields

### **Handy to Know...**

- Using incorrect syntax when entering criteria can generate a data type mismatch error. For instance, if you place numerical or date criteria in quotation marks, rather than cross hatch symbols (#), Access interprets the data as text, rather than numbers.

## ADDING MORE TABLES AND FIELDS

Query designs do not need to be static in design – you can add (or remove) tables and fields as dictated by your reporting needs. In our case study we have a *Personal Details* table that can

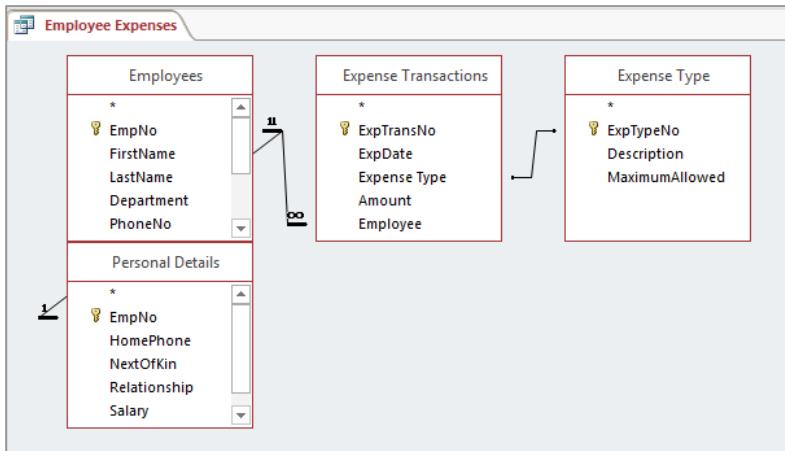
be added to the design to provide us with extra fields for our employees, such as *Salary*.

## Try This Yourself:

Same File

*Continue using the previous file with this exercise, or open the file Multi Table Queries 2.accdb...*

- 1 Open the query **Employee Expenses** in **Design View**
  - 2 On the **Query Tools: Design** tab, click on **Show Table** in the **Query Setup** group to display the **Show Table** dialog box
  - 3 Double-click on **Personal Details** to add it to the design, then click on **[Close]**
  - 4 Double-click on **Salary** to add the field to the grid
  - 5 Click on **Save** in the **QAT** to save the changes
  - 6 Click in the **Criteria** cell for **Salary** and type **>=85000**
  - 7 On the **Query Tools: Design** tab, click on **Run** in the **Results** group, to display expense transactions for employees with a salary greater than or equal to 85,000
  - 8 Close the query – click on **[No]** to discard the criteria changes



3

6

Employee Expenses					
ExpDate	Expense Type	Amount	FirstName	LastName	Department
2/01/2015	Accommodation	\$145.00	Julianne	Kerr	Executive
2/02/2015	Accommodation	\$281.36	Julianne	Kerr	Executive
2/03/2015	Accommodation	\$349.54	Julianne	Kerr	Executive
2/04/2015	Accommodation	\$352.65	Julianne	Kerr	Executive
19/04/2015	Accommodation	\$355.50	Julianne	Kerr	Executive
19/05/2015	Accommodation	\$357.79	Julianne	Kerr	Executive
2/06/2015	Accommodation	\$358.44	Julianne	Kerr	Executive

7

## **For Your Reference...**

To **add more tables to a query design:**

1. Open the query in **Design View**
  2. On the **Query Tools: Design** tab, click on **Show Table** in the **Query Setup** group
  3. Double-click on the table(s) to add
  4. Click on **[Close]**

## **Handy to Know...**

- Save a query design after making a structural change to the design, and before you add criteria, to then be able to close the table later without the criteria but with the structural alteration.

# UTILISING HIDDEN FIELDS

Earlier we encountered a problem when we tried to enter text into a field that was numeric even though, because of its lookup arrangement, it displayed text results. The problem with our

solution was that we have to enter the criteria as a numeric code. Utilising a **hidden field**, we can enter text values and have the correct result displayed without incurring a data mismatch error.

## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file Multi Table Queries\_3.accdb...

- 1 Open the query **Employee Expenses** in **Design View**
- 2 In the query grid, click on the **Show** box for **Expense Type** so it appears without a tick – this will hide the field when the query is run
- 3 In the **Field List**, double-click on **Description** in the **Expense Type** table to add it to the end of the query grid
- 4 Click on **Save** in the **QAT** to save these changes
- 5 Click in the **Criteria** cell for **Description** and type **Meals**
- 6 On the **Query Tools: Design** tab, click on **Run** in the **Results** group to display all of the *Meals* transactions
- 7 Close the query – click on **[No]** to discard the changes since the last save

Field: Table: Sort: Show: Criteria: or:	ExpDate Expense Transactions	Expense Type Expense Transactions	Amount Expense Transactions	FirstName Employees	LastName Employees
	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

2

Field: Table: Sort: Show: Criteria: or:	LastName Employees	Department Employees	PhoneNo Employees	Salary Personal Details	Description Expense Type
	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
					Meals

5

Employee Expenses					
FirstName	LastName	Department	PhoneNo	Salary	Description
Augustine	Millson	Administration	61022	\$85,000.00	Meals
Augustine	Millson	Administration	61022	\$85,000.00	Meals
Augustine	Millson	Administration	61022	\$85,000.00	Meals
Augustine	Millson	Administration	61022	\$85,000.00	Meals
Augustine	Millson	Administration	61022	\$85,000.00	Meals
Augustine	Millson	Administration	61022	\$85,000.00	Meals
Augustine	Millson	Administration	61022	\$85,000.00	Meals
Augustine	Millson	Administration	61022	\$85,000.00	Meals
Augustine	Millson	Administration	61022	\$85,000.00	Meals

6

## For Your Reference...

To **hide a field from a query when it is run**:

- In the query grid, click on the **Show** box for the field so it appears without a tick

## Handy to Know...

- Hiding a field from the query grid is better than removing it, as it still enables you to search the records based on that field.

# UNDERSTANDING QUERY JOINS

In addition to selecting specific fields and specifying criteria for a query, the data that is displayed in a multi-table query is also influenced by the type of **join** that has been specified

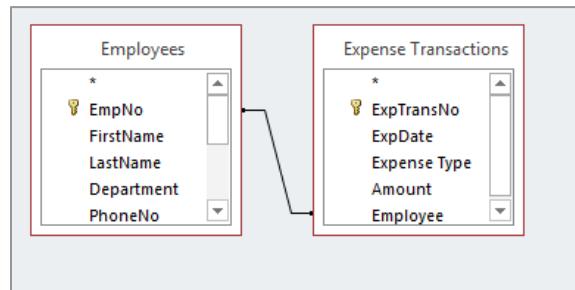
between the tables. Basically there are three types of joins at your disposal: an **inner join**, a **left outer join** and a **right outer join**. In Access these joins are numbered 1 to 3 respectively.

## A Classic Join Example

The easiest way to understand the different joins is through the use of a simple example. Let's assume you have two tables, one for **Employees** and the other for their expense transactions. The **Employee** table would have one record per employee, while the **Expense Transactions** table could have many transactions per employee.

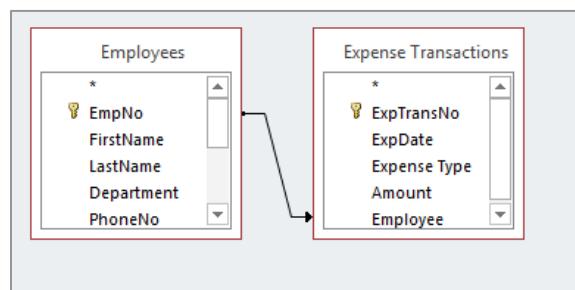
### Join 1: An Inner Join

This is the default join type in Microsoft Access and is the one most commonly used. In this type of join the query will display only employees who have transactions and the transactions recorded against them. Employees who do not have any transactions will not be shown, nor will transactions that are orphaned and do not have an associated employee (there really shouldn't be any of these).



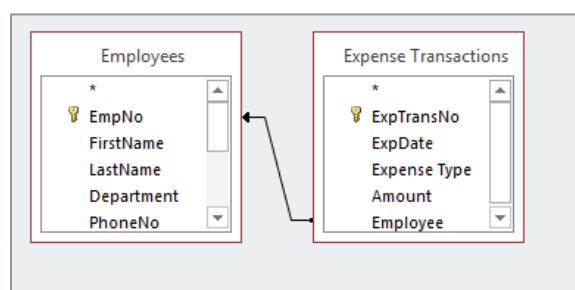
### Join 2: An Outer Left Join

Let's suppose you want to see all employees, irrespective of whether they do or don't have transactions, as well as their transactions. An **Outer Left Join** achieves this. Again, orphaned transactions that do not have any associated employee will not appear.



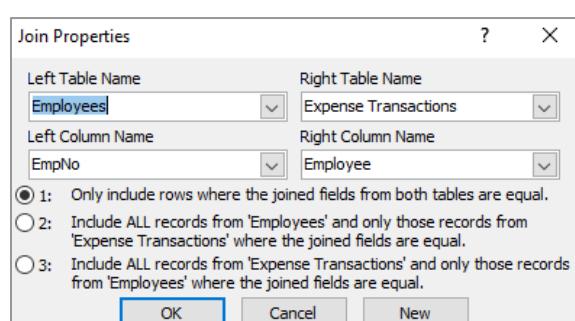
### Join 3: An Outer Right Join

Let's suppose you want to see all of the transactions, both those that have associated employees and those that are orphaned, then use the **Outer Right Join**. This join focuses on the transaction side of the join showing all transactions, but only employees from the left side that have transactions.



## Join Properties

When you right-click on a join between tables you can open the **Join Properties** dialog box. In this dialog box you can specify the type of join to work with.



# CREATING AN INNER JOIN

**Inner Joins** are the default join type and are automatically created for you when you have two joined tables in a query. Essentially an inner join shows only records from either table that are

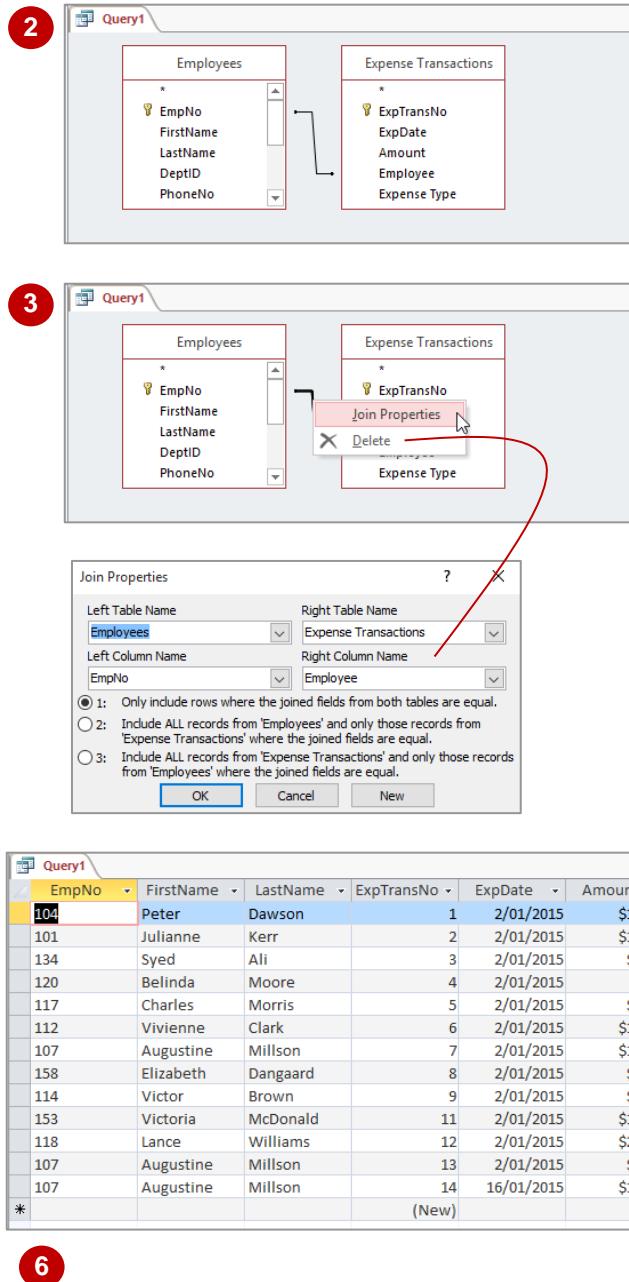
matched in the other table. Records in the left table that do not have matches in the right table won't appear, nor will records in the right table that aren't matched to records in the left table.

## Try This Yourself:

**Open File**

Before starting this exercise you **MUST** open the file *Multi Table Queries\_4.accdb*...

- 1 Click on the **Create** tab, then click on **Query Design** in **Queries**
- 2 Double-click on **Employees** and then on **Expense Transactions**, then click on **[Close]**
- 3 Right-click on the line that joins both tables and select **Join Properties** to display the **Join Properties** dialog box
- 4 Ensure option **1:** is selected, then click on **[OK]**
- 5 Double-click on the **EmpNo**, **FirstName** and **LastName** fields in the **Employees** table, and the asterisk (\*) in the **Expense Transactions** table
- 6 On the **Query Tools: Design** tab, click on **Run** in the **Results** group to run the query and note the records
- 7 Click on the **File** tab, select **Save As**, then click on **Save Object As** and click on **[Save As]**
- 8 Type **qryInnerJoin**, then click on **[OK]**
- 9 Close the query



## For Your Reference...

To *create* an **inner join query**:

1. In query **Design View**, right-click on the join line, then select **Join Properties**
2. Ensure option **1:** is selected and click on **[OK]**

## Handy to Know...

- With an inner join query, which is the default join type, only records from the left table that have matches in the right table are joined. Orphans from the right table are not shown when the query is run.

# CREATING A LEFT OUTER JOIN

With **left outer joins** the table on the left side of the join has precedence and all records of that table, irrespective of whether they have matching records in the right table, will appear. Records

from the right table will only appear if they are matched to records in the left. With a left outer join it is the left table that is 'dominant'.

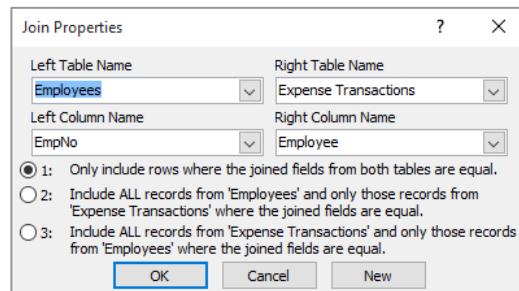
## Try This Yourself:

**Same File**

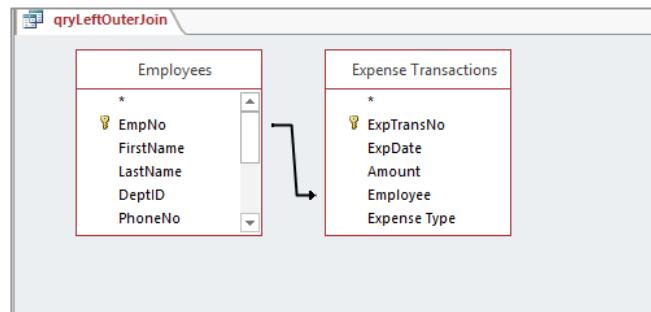
Continue using the previous file with this exercise, or open the file *Multi Table Queries\_5.accdb*...

- 1 Open the query *qryInnerJoin* in **Design View**
- 2 Click on the **File** tab, select **Save As**, click on **Save Object As**, then click on **[Save As]**
- 3 Type *qryLeftOuterJoin*, then click on **[OK]**
- 4 Right-click on the join line and select **Join Properties**
- 5 Click on option **2**; then click on **[OK]**
- 6 On the **Query Design: Tools** tab, click on **Run** in the **Results** group to run the query  
*This time all employees, whether they have transactions or not, will be displayed...*
- 7 Save and close the query

4



5



6

EmpNo	FirstName	LastName	ExpTransNo	ExpDate
101	Julianne	Kerr	2	2/01/2015
102	Harry	Jones		
103	Angel	Harrington		
104	Peter	Dawson	1	2/01/2015
105	Mark	Jones		
106	Maureen	Grayson		
107	Augustine	Millson	14	16/01/2015
107	Augustine	Millson	13	2/01/2015
107	Augustine	Millson	7	2/01/2015
108	Amanda	Bennet		

## For Your Reference...

To *create a left outer join query*:

- 1 In query **Design View**, right-click on the join line, then select **Join Properties**
- 2 Click on option **2**: and click on **[OK]**

## Handy to Know...

- Follow the arrow in a join between tables in a query to get an idea of what is happening. If an arrow points to the right table, then all of the records from the left will appear but only matching ones from the right will show up.

# CREATING A RIGHT OUTER JOIN

In a **right outer join** the table on the right, which is usually the transaction table, is the dominant table. In this type of join, all of the transactional records will appear irrespective of whether they

are orphaned (don't have a match on the left) or not. In a good database design there should never be unmatched transactions – so this type of query is ideal for tracking down data problems.

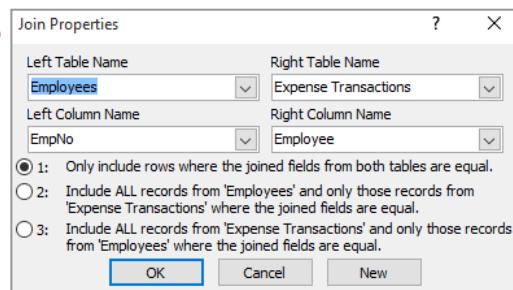
## Try This Yourself:

**Same File**

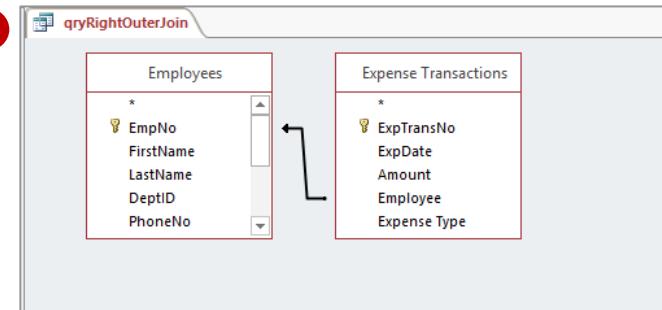
Continue using the previous file with this exercise, or open the file *Multi Table Queries\_6.accdb*...

- 1 Open the query *qryInnerJoin* in **Design View**
- 2 Click on the **File** tab, select **Save As**, click on **Save Object As**, then click on **[Save As]**
- 3 Type **qryRightOuterJoin**, then click on **[OK]**
- 4 Right-click on the join line, then select **Join Properties**
- 5 Click on option **3**; then click on **[OK]**
- 6 On the **Query Tools: Design** tab, click on **Run** in the **Results** group to run the query  
*This time all transactions, whether they have associated employees or not, will be displayed...*
- 7 Save and close the query

4



5



6

EmpNo	FirstName	LastName	ExpTransNo	ExpDate
104	Peter	Dawson	1	2/01/2015
101	Julianne	Kerr	2	2/01/2015
134	Syed	Ali	3	2/01/2015
120	Belinda	Moore	4	2/01/2015
117	Charles	Morris	5	2/01/2015
112	Vivienne	Clark	6	2/01/2015
107	Augustine	Millson	7	2/01/2015
158	Elizabeth	Dangaard	8	2/01/2015
114	Victor	Brown	9	2/01/2015
			10	2/01/2015

## For Your Reference...

To *create a right outer join*:

- 1 In query **Design View**, right-click on the join line, then select **Join Properties**
- 2 Click on option **3**: and click on **[OK]**

## Handy to Know...

- Databases with sound rules of referential integrity should never present unmatched transactions. For example, in a good system you should not be able to delete an employee if there are still open transactions for that employee.

## NOTES:



## CHAPTER 3

 INFOCUS

# PARAMETER QUERIES

A **query** selects data based on a field list and criteria that are entered into the query grid. This is fine if the criteria and field list never change. However, you may find that you need to run a query on a regular basis with different criteria depending on particular situations or events.

A **parameter query** can save you the hassle of entering **Design View** and manually changing the criteria of a saved query. A parameter in computer language refers to some information or data that is passed from one object to another. In a parameter query, the parameter is simply the example that needs to be entered into the criteria cell of the query grid.

When you run a parameter query, you are prompted for the criteria by a user-friendly dialog box. The criteria you type are placed temporarily in the query grid to create the query.

**In this session you will:**

- ✓ learn how to create a parameter query
- ✓ learn how to display all records using a parameter query
- ✓ learn how to use parameters to display a range of values
- ✓ learn how to use parameters in expressions
- ✓ learn how to use wildcards with parameter queries.

# CREATING A PARAMETER QUERY

A **parameter query** is created in the same way as a normal query, except that you place square brackets around text in a criteria cell in lieu of an example of the search data. Access uses this text

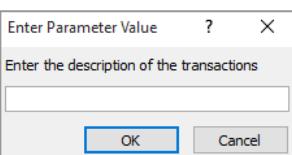
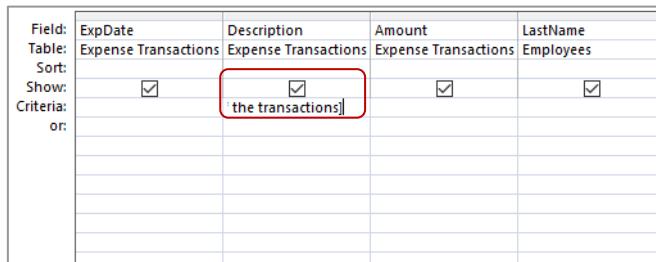
as a prompt in a dialog box that appears when the query is run. Whatever the user types as a response to this prompt then becomes the criteria (passed as a *parameter*) in the query.

## Try This Yourself:

Open  
File

*Before starting this exercise  
you MUST open the file  
Parameter  
Queries 1.accdb...*

- 1** Open *qryEmployeeExpenses* in Design View
  - 2** Click in the **Criteria** cell for **Description** and type [Enter the description of the transactions]  
*The square brackets tell Access that this is a parameter query and the text serves as the prompt...*
  - 3** On the **Query Tools: Design** tab, click on the top half of **View** in the **Results** group to run the query  
*You will be prompted for a description...*
  - 4** Type **Postage**, then click on **[OK]** to see all of the postage transactions (52 records)
  - 5** Click on the **File** tab, click on **Save As**, then click on **Save Object As** and click on **[Save As]**
  - 6** Type **Query – Parameter Query Example** in **Save to**, then click on **[OK]**
  - 7** Close the query



qryEmployeeExpenses				
ExpDate	Description	Amount	LastName	FirstName
2/01/2015	Postage	\$3.59	Moore	Belinda
2/01/2015	Postage	\$16.99	Morris	Charles
2/02/2015	Postage	\$6.24	Moore	Belinda
4/02/2015	Postage	\$18.32	Morris	Charles
2/03/2015	Postage	\$15.70	Moore	Belinda
4/03/2015	Postage	\$21.28	Morris	Charles
2/04/2015	Postage	\$15.83	Moore	Belinda
4/04/2015	Postage	\$21.31	Morris	Charles
19/04/2015	Postage	\$15.96	Moore	Belinda
20/04/2015	Postage	\$21.46	Morris	Charles



## **For Your Reference...**

To **create** a *parameter query*:

1. Create the query in **Design View**
  2. In the **Criteria** cell, type a prompt enclosed in square brackets that will be used for the query

## **Handy to Know...**

- When creating queries, you can enter parameters for more than one field. Access will stop the query for each one and prompt for criteria. Too many stops and prompts can become very frustrating. As an alternative you could create a form to capture the data required for the query.

# DISPLAYING ALL RECORDS

When you leave a parameter prompt box empty you would expect that all of the records are displayed, however they're not. Instead, you get nothing. If you want the option to see all of the

records when you leave the prompt box empty, you can use an **OR** expression to display a repeat of the prompt and test to see if it hasn't been used – in other words, if it is **Null**.

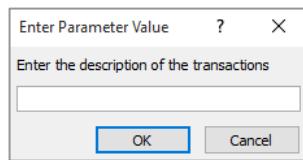
## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file *Parameter Queries\_2.accdb*...

- 1 Double-click on **Query – Parameter Query Example** to display the **Enter Parameter Value** dialog box
- 2 Leave the dialog box empty and click on **[OK]**  
*Nothing appears to have happened...*
- 3 On the **Home** tab, click on the top half of **View** in the **Views** group, then click in the **or** cell for **Description** and type **[Enter the description of the transactions] Is Null**
- 4 Click on the top half of **View** to display the dialog box again
- 5 Leave the dialog box empty and click on **[OK]**  
*All records are displayed...*
- 6 Click on the **File** tab, click on **Save As**, then click on **Save Object As** and click on **[Save As]**
- 7 Type **Query – Displays All Example** in **Save to**, then click on **[OK]**
- 8 Close the query

1



2

Query - Parameter Query Example				
ExpDate	Description	Amount	LastName	FirstName
*				

3

Field:	ExpDate	Description	Amount	LastName	FirstName
Table:	Expense Transactions	Expense Transactions	Expense Transactions	Employees	Employees
Sort:					
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		[Enter the description of the transactions] Is Null			
or:					

5

Query - Parameter Query Example				
ExpDate	Description	Amount	LastName	FirstName
2/01/2015	Accommodation A	\$132.00	Dawson	Peter
2/01/2015	Accommodation A	\$145.00	Kerr	Julianne
2/01/2015	Gifts	\$27.06	Ali	Syed
2/01/2015	Postage	\$3.59	Moore	Belinda
2/01/2015	Postage	\$16.99	Morris	Charles
2/01/2015	Accommodation A	\$154.50	Clark	Vivienne
2/01/2015	Accommodation A	\$125.50	Millson	Augustine
2/01/2015	Other Expenses	\$48.39	Dangaard	Elizabeth

## For Your Reference...

To **display all records** in a **parameter query**:

1. In query **Design View**, click in the **or** cell and re-type the parameter prompt
2. Add the text **Is Null** to the end of the second parameter prompt

## Handy to Know...

- **Is Null** is not an easy concept to understand – basically it translates as “is nothing there”. By duplicating parameter text and adding **Is Null** to the second parameter, the query is run with valid criteria **or** if there is nothing in the criteria, then it ignores the parameter request and shows everything.

# USING PARAMETERS TO DISPLAY A RANGE

There is no rule in Access to say that you can only use one parameter in a query. Two parameters are quite acceptable when working with criteria that are based on a *range*. A typical

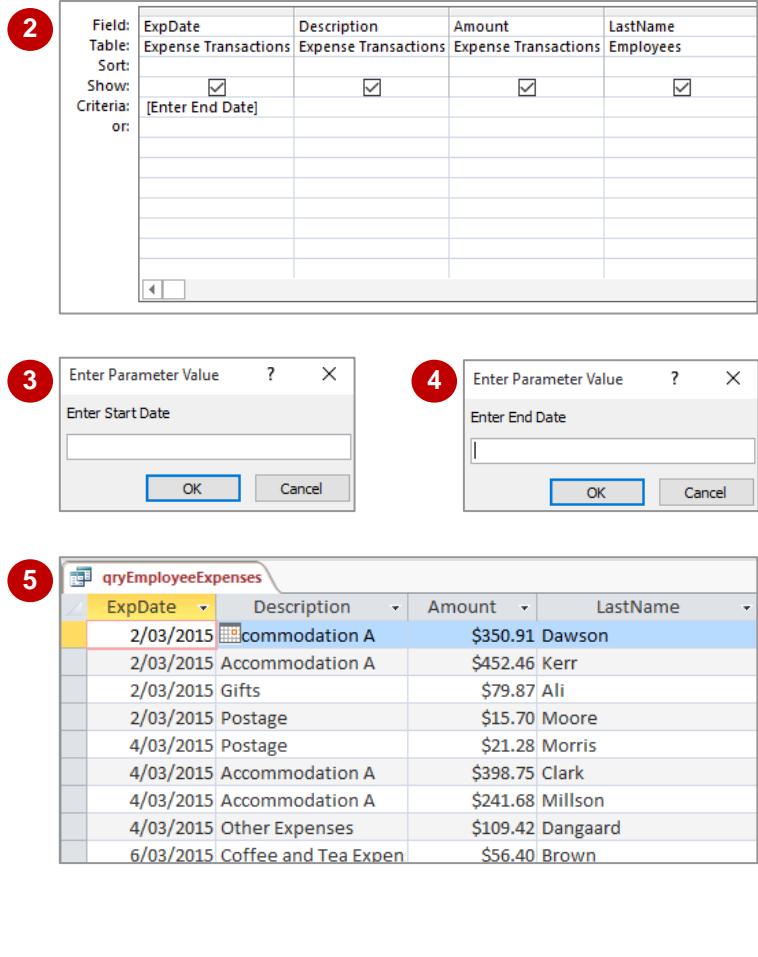
use for parameters is to capture dates where you want to specify a start date and an end date. The parameters will cause the query to extract all of the records between these two dates.

## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file *Parameter Queries\_3.accdb*...

- 1 Open *qryEmployeeExpenses* in Design View
- 2 Click in the **Criteria** cell for **ExpDate** and type **Between [Enter Start Date] AND [Enter End Date]**
- 3 On the **Query Tools: Design** tab, click on the top half of **View** in the **Results** group to display the first prompt box
- 4 Type **1/3/15**, then click on **[OK]** to display the second prompt box
- 5 Type **31/3/15** and click on **[OK]** to display all of the transactions for **March**
- 6 Click on the **File** tab, click on **Save As**, then click on **Save Object As** and click on **[Save As]**
- 7 Type **Query – Parameter Range Example** in **Save to**, then click on **[OK]**
- 8 Close the query



## For Your Reference...

To *create parameters* for a *range*:

1. In query **Design View**, click in the **Criteria** cell and type **Between**
2. Type the first parameter
3. Type **AND**
4. Type the second parameter

## Handy to Know...

- When constructing queries to display a range, the **Between...And** operators can also be used without parameters. For example, you can type **Between 1/3/13 And 31/3/13** to create a hard-coded select query.

# USING PARAMETERS IN EXPRESSIONS

**Expressions** (*formulas*) are quite often used in field cells to create calculated fields in a query. Access allows you to place a parameter within an expression in lieu of a value or field. All that you

need to do is create a prompt, enclose it within square brackets, and place it in the expression at the place where you want the value to appear. The value is entered when the query runs.

## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file *Parameter Queries\_4.accdb*...

- 1 Open *qryEmployeeExpenses* in Design View
- 2 Click on the grey bar above **LastName** to select the column, then on the **Query Tools: Design** tab click on **Insert Columns** in the **Query Setup** group to insert a new field
- 3 Click in the **Field** cell for the new column and type **Amount+[Enter a handling fee]**
- 4 Click on the top half of **View** in the **Results** group to trigger the handling fee prompt
- 5 Type **2.50**, then click on **[OK]** to display the transactions with an extra \$2.50 added to each
- 6 Click on the **File** tab, click on **Save As**, then click on **Save Object As** and click on **[Save As]**
- 7 Type **Query – Parameter Expression Example** in **Save to**, then click on **[OK]**
- 8 Close the query

2

3

4

5

Field:	ExpDate	Description	Amount	a handling fee]
Table:	Expense Transactions	Expense Transactions	Expense Transactions	
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:	or:			

## For Your Reference...

To **create an expression with a parameter**:

1. In query **Design View**, click in a free **Field** cell
2. Type the expression but include a parameter prompt where you would normally type a value or field name

## Handy to Know...

- When creating queries, expressions that contain parameters are ideal for "what-if" analyses where you run the query with different values.

# USING PARAMETERS WITH WILDCARDS

**Parameter queries** are ideal for allowing the user to specify the selection criteria without the need for hard-coding samples. **Wildcards** are great for using a scatter-gun approach to

searching for records. The two can be combined using string **concatenation** techniques where the parameter statement is joined to the wildcard using an ampersand character (&).

## Try This Yourself:

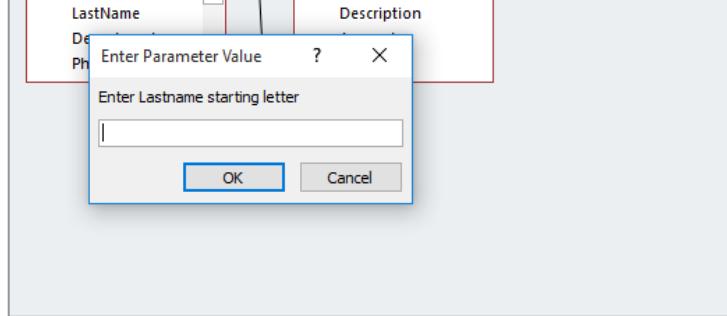
**Same File** Continue using the previous file with this exercise, or open the file Parameter Queries\_5.accdb...

- 1 Open **qryEmployeeExpenses** in **Design View**
- 2 Click in the **Criteria** cell for **Lastname** and type **Like [Enter Lastname starting letter] & “\*\*”**
- 3 Click on the **File** tab, click on **Save As**, then click on **Save Object As** and click on **[Save As]**
- 4 Type **Query – Parameters with Wildcards** in **Save to** and click on **[OK]**
- 5 On the **Query Tools: Design** tab, click on the top half of **View** in the **Results** group  
*You should now be prompted for a starting letter...*
- 6 Type **M**, then click on **[OK]** to see all records where the last name begins with the letter M
- 7 Close the query

2

Field:	ExpDate	Description	Amount	LastName
Table:	Expense Transactions	Expense Transactions	Expense Transactions	Employees
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> ting letter] & “**”
Criteria:				
or:				

5



6

Query - Parameters with Wildcards				
ExpDate	Description	Amount	LastName	
2/01/2015	Stage	\$3.59	Moore	
2/01/2015	Postage	\$16.99	Morris	
2/01/2015	Accommodation A	\$125.50	Millson	
2/01/2015	Accommodation A	\$123.44	McDonald	
2/01/2015	Meals	\$52.86	Millson	
16/01/2015	Accommodation A	\$155.60	Millson	
2/02/2015	Postage	\$6.24	Moore	
4/02/2015	Postage	\$18.32	Morris	
4/02/2015	Accommodation A	\$128.44	Millson	

## For Your Reference...

To use a **wildcard** in a **parameter query**:

1. In query **Design View**, type the parameter criteria
2. Type an ampersand (&) after the criteria
3. Type the wildcard character or expression enclosed in quotation marks

## Handy to Know...

- When constructing parameter queries, if the wildcard is at the start of the expression, you can actually use more than one letter to further refine the search. For example, typing **MO** in a prompt box asking for the letter that a last name begins with, will display only last names that begin with MO.

## CHAPTER 4

 INFOCUS

# CALCULATIONS IN QUERIES

Queries can be used to combine data from different tables to create detailed lists. Data columns can be also added to a query by typing formulas in the field cells of a query grid. These formulas are known as **expressions**, and the new fields are known as **calculated fields**. Expressions can include arithmetic signs, numbers, field names, and special built-in functions.

The typical use for **calculated fields** is to add or multiply the values in existing fields to produce a new value. For example, in an invoice table you may have a field for **quantity** and a field for **price**. Using a calculated query you can multiply the two together to create a calculated field showing the **total**.

**In this session you will:**

- ✓ learn how to create a calculated field
- ✓ learn how to rename expressions and format calculated fields
- ✓ learn how to summarise data in a query
- ✓ learn how to change the grouping in a summary query
- ✓ learn how to perform calculations in queries using dates
- ✓ learn how to use criteria in calculated fields
- ✓ learn how to concatenate fields together.

## CREATING A CALCULATED FIELD

A **calculated field** is created by typing an expression (or formula) into a new field column in a query grid. The expression normally contains a reference to one or more existing fields in the

query and these are shown enclosed in square brackets. Calculations conform to the standard mathematical rules of BODMAS.

## Try This Yourself:

Open File

*Before starting this exercise you  
MUST open the file Calculation  
Queries 1.accdb...*

## 1 Open *qryExpenses* in *Design View*

*Let's calculate the value of tax, where tax is 10% of the total...*

**2** Click in the blank **Field** cell to the right of **Amount** and type [Amount]/1.1

*This will create a new field...*

3 Click in the next blank **Field** cell and type **[Amount]-([Amount]/1.1)** then press **Tab**

*By default, Access names the new fields Expr1 and Expr2...*

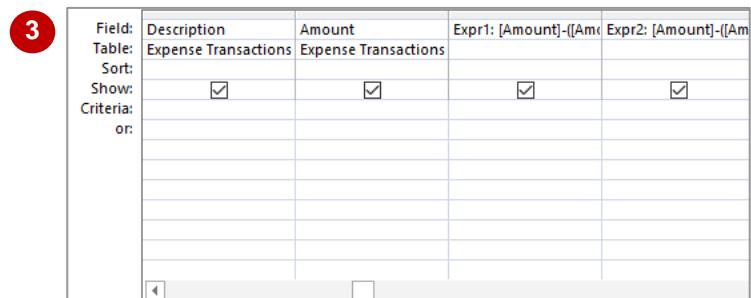
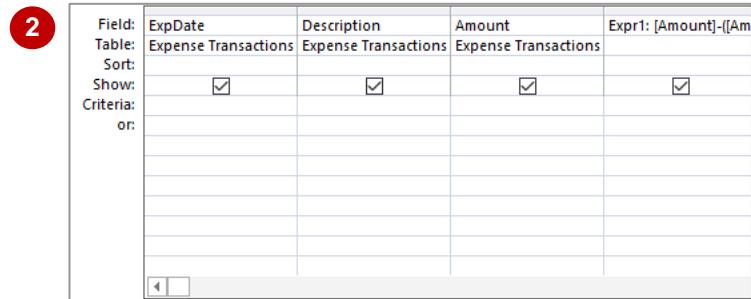
4 On the **Query Tools: Design** tab, click on the top half of **View** in the **Results** group to run the query and see the calculations

If hash signs appear,  
double-click on the field heading  
borders to perform a best fit...

**5** Click on the **File** tab, click on **Save As**, then click on **Save Object As** and click on **[Save As]**

**6** Type **Query – Calculated Fields Example** in **Save to**, then click on **[OK]**

## 7 Close the query



## **For Your Reference...**

To **create** a **calculated field** in a **query grid**:

1. In query **Design View**, click in the first blank column
  2. Type a **calculation expression** (reference existing fields by enclosing the field name in square brackets)
  3. Run the query

### **Handy to Know...**

- Access will assign a default field name to a calculation in a query. It appears as **Expr** (which is an abbreviation for *Expression*) followed by a sequential number. Calculations often appear unformatted.

# FORMATTING CALCULATED FIELDS

A **calculated field** appears with a default column heading. Obviously, it would be more aesthetically pleasing to nominate a more descriptive heading for the column. In addition,

the data is displayed without any formatting. This can be changed using the **Field Properties** of the column which allow you to change the format so that the values will appear with currency symbols.

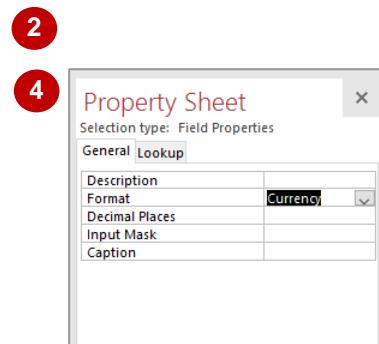
## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file Calculation Queries\_2.accdb...

- 1 Open **Query – Calculated Fields Example** in **Design View**
- 2 In the query grid select the text **Expr1**, then type **NetAmount**
- 3 On the **Query Tools: Design** tab, click on **Property Sheet** in the **Show/Hide** group to display the **Property Sheet** pane
- 4 Click in the cell for **Format**, then click on the drop arrow and select **Currency**
- 5 In the query grid select the text **Expr2**, then type **Tax**
- 6 Repeat step 4 to apply the **Currency** format
- 7 Click on the top half of **View** in the **Results** group to run the query
- 8 Click on the **File** tab, click on **Save As**, then click on **Save Object As** and click on **[Save As]**
- 9 Type **Query – Formatted Fields Example** in **Save to** and click on **[OK]**, then close the query

Field:	ExprDate	Description	Amount	Net Amount: [Amount]	Expr2: [Amount]-[Amount]
Table:	Expense Transactions	Expense Transactions	Expense Transactions		
Sort:					
Show:	<input checked="" type="checkbox"/>				
Criteria:					
or:					



Field:	Net Amount: [Amount]	Tax: [Amount]-[Amount]
Table:	Expense Transactions	
Sort:		
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		
or:		

ExpDate	Description	Amount	Net Amount	Tax
2/01/2015	Accommodat	\$132.00	\$120.00	\$12.00
2/01/2015	Accommodat	\$145.00	\$131.82	\$13.18
2/01/2015	Gifts	\$27.06	\$24.60	\$2.46
2/01/2015	Postage	\$3.59	\$3.26	\$0.33
2/01/2015	Postage	\$16.99	\$15.45	\$1.54
2/01/2015	Accommodat	\$154.50	\$140.45	\$14.05
2/01/2015	Accommodat	\$125.50	\$114.09	\$11.41
2/01/2015	Other Expense	\$48.39	\$43.99	\$4.40
2/01/2015	Coffee and Te	\$18.26	\$16.60	\$1.66
2/01/2015	Coffee and Te	\$7.72	\$7.02	\$0.70
2/01/2015	Accommodat	\$123.44	\$112.22	\$11.22

7

## For Your Reference...

To **change** the **name** of an **expression**:

1. Double-click on the name to select it
2. Type a new name for the expression

To **format** an **expression**:

- Use the **Property Sheet** pane settings to make appropriate changes

## Handy to Know...

- When formatting calculated fields, the formatting options in the **Property Sheet** pane will change slightly depending upon the data type of the field. For example, **Currency** is only available for fields of a currency or numeric data type.

# SUMMARISING DATA USING A QUERY

Queries can also be used to summarise your data. This is done by adding a **Total** row to the query grid. When this is done you can use some of the special operators available for

summarising data including: **sum**, **avg**, **min**, **max**, **count**, **StDev**, and **var**. There are also options to find the value of the *first* or *last* records.

## Try This Yourself:

**Open File**

Before starting this exercise you **MUST** open the file *Calculation Queries\_3.accdb*...

- 1 Double-click on **qrySummedExpenses** to run the query
- 2 On the **Home** tab, click on the top half of **View** in the **Views** group to display the query in **Design View**
- 3 On the **Query Tools: Design** tab, click on **Totals** in the **Show/Hide** group to add a **Total** row to the query grid
- 4 Click on **Group By** for the **Amount** field, then click on the drop arrow and select **Sum**  
*This will sum (total) this field...*
- 5 Repeat step 4 for the **Net Amount** and **Tax** fields
- 6 Click on the top half of **View** in the **Results** group to see the totals grouped by **Description**
- 7 Click on the **File** tab, click on **Save As**, then click on **Save Object As** and click on **[Save As]**
- 8 Type **Query – Summing Data Example** and click on **[OK]**
- 9 Close the query

3

Field:	Description	Amount	Net Amount: [Amount]-([Amount]-[Amount])
Table:	Expense Transactions	Expense Transactions	Group By
Total:	Group By	Group By	Group By
Sort:			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			
or:			

5

Field:	Description	Amount	Net Amount: [Amount]	Tax: [Amount]-([Amount]-[Amount])
Table:	Expense Transactions	Expense Transactions	Sum	Sum
Total:	Group By	Sum	Sum	Sum
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:				
or:				

6

qrySummedExpenses				
Description	SumOfAmount	et Amount	x	
Accommodation	\$37,891.81	34447.1036363636	3444.7033	
Accommodation	\$8,430.96	7664.51245454545	766.4496	
Coffee and Tea	\$5,606.71	5097.00545454545	509.6981	
Gifts	\$2,069.60	1881.45590909091	188.1444	
Meals	\$5,292.14	4811.03718181818	481.1022	
Other Expense	\$4,050.89	3682.63145454545	368.2619	
Postage	\$4,959.09	4508.25918181818	450.8233	

## For Your Reference...

To **summarise data** in a **query**:

1. In query **Design View**, on the **Query Tools: Design** tab, click on **Totals** in the **Show/Hide** group
2. Change **Group By** to **Sum** for the numeric fields

## Handy to Know...

- Using the **Sum** value in the **Total** field of a query works by totalling (summing) each category from the **Group By** field.

# CHANGING THE GROUPING

The key to understanding summary queries lies in the **Group By** field. In a summary query each example of the data in the **Group By** field is calculated and the result is displayed in the

query. You can easily change the **Group By** field to calculate on other fields.

## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file *Calculation Queries\_4.accdb*...

- 1 Open **Query – Summing Data Example** in **Design View**

Let's change the query so that we see the daily totals for expenses rather than for each Description...

- 2 In the query grid, click on the drop arrow for the **Description** field to see a list of fields from the **Expense Transactions** table, then click on **ExpDate**

- 3 On the **Query Tools: Design** tab, click on the top half of **View** in the **Results** group to see the daily totals for expenses

- 4 Click on the top half of **View** in the **Views** group to return to **Design View**

- 5 Repeat steps 2 and 3 to see the expenses totalled by **Employee**

- 6 Click on **Save** to save the changes to the query

- 7 Close the query

2

A screenshot of the Microsoft Access Query Designer window. The 'Field' column has a dropdown arrow next to 'ExpDate'. The 'Table' column shows 'Expense Transactions'. The 'Total' column has 'Sum' selected. The 'Group By' column has 'Group By' selected. The 'Sort' column has a checked checkbox. The 'Show' column has a checked checkbox. The 'Criteria' column has 'or:' followed by a blank line.

3

A screenshot of the Microsoft Access 'Query - Summing Data Example' results. The table has columns: ExpDate, SumOfAmor, et Amount, and x. The data shows daily expense totals for January 1st, 2009, through January 10th, 2009.

ExpDate	SumOfAmor	et Amount	x
1/01/2009	\$16.46	14.9648181818182	1.4964
2/01/2009	\$48.57	44.1574545454545	4.4157
3/01/2009	\$213.70	194.272090909091	19.4272
4/01/2009	\$329.68	299.710909090909	29.971
5/01/2009	\$137.02	124.562727272727	12.4562
6/01/2009	\$58.31	53.0105454545455	5.301
7/01/2009	\$10.34	9.40063636363636	0.94
8/01/2009	\$253.44	230.398363636364	23.0398
9/01/2009	\$440.86	400.783363636364	40.0783
10/01/2009	\$81.22	73.8386363636364	7.3838

5

A screenshot of the Microsoft Access 'Query - Summing Data Example' results. The table has columns: Employee, SumOfAmor, et Amount, and x. The data shows the total expenses for each employee.

Employee	SumOfAmor	et Amount	x
Kerr	\$7,119.20	6472.00245454545	647.1994
Dawson	\$6,823.20	6202.91009090909	620.2902
Millson	\$18,286.81	16624.3742727273	1662.4327
	\$2,294.00	2085.45736363636	208.5446
Clark	\$6,386.38	5805.80409090909	580.5793
Brown	\$3,312.70	3011.54809090909	301.1535
Morris	\$2,694.28	2449.34127272727	244.9327
Williams	\$8,430.96	7664.51245454545	766.4496
Moore	\$2,264.81	2058.91790909091	205.8906
Ali	\$2,069.60	1881.45590909091	188.1444

## For Your Reference...

To **change** the **grouping** in a **summary query**:

- 1 In query **Design View**, click on the field to change and click on the drop arrow
- 2 Select the desired field
- 3 Run the query

## Handy to Know...

- You can have more than one **Group By** field in a summary query. If you do, Access will cascade them. For example, if you have **Department** as the **Group By**, and then **Lastname** as a **Group By**, Access will group all of the records first by **Department** and then by **Lastname** within each **Department**.

# CALCULATING WITH DATES

Dates in computers are really just numbers. Each day of the year, the century, the millennium, has a unique number attached to it. Since these numbers run sequentially they can be used to

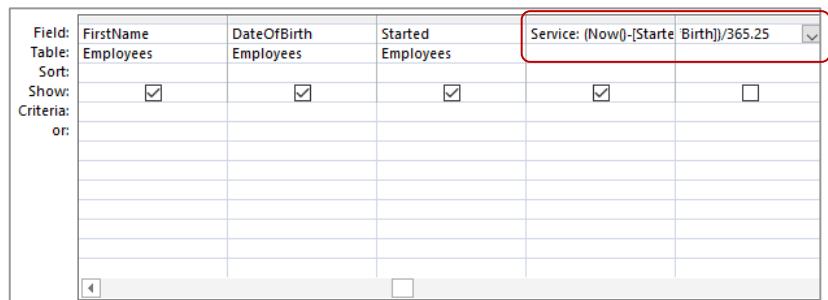
perform date arithmetic. Using this principal we can perform a great variety of arithmetic on dates to calculate ages, seniority, and the like.

## Try This Yourself:

Open  
File

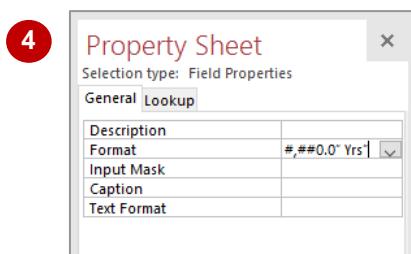
*Before starting this exercise  
you MUST open the file  
Calculation  
Queries 5.accdb...*

- 1 Open **qryEmployees** in **Design View**
  - 2 Type the two calculated fields into blank columns, as shown
  - 3 Click on the first calculated date then, on the **Query Tools: Design** tab, click on **Property Sheet** in the **Show/Hide** group to display the **Property Sheet** pane
  - 4 Click in **Format** and type **#,##0.0" Yrs"**
  - 5 Repeat step **4** for the second calculated date, then close the **Property Sheet** pane
  - 6 Click on the top half of **View** in the **Results** group to see the calculations
  - 7 Click on the **File** tab, click on **Save As**, then click on **Save Object As** and click on **[Save As]**
  - 8 Type **Query – Date Calculation Example** in **Save to**, then click on **[OK]**
  - 9 Close the query



- 2** **Service:** `(Now() - [Started]) / 365.25`  
Calculates the years of Service by calculating the number of days between the serial number for the date Started (must be in square brackets) and Now, then divides the result by the number of days in a year.

**Age:** `(Now() - [DateOfBirth]) / 365.25`  
Calculates Age by calculating the number of days between the serial number for the DateOfBirth and Now, then divides the result by the number of days in a year.



qryEmployees				
FirstName	DateOfBirth	Started	Service	Age
Julianne	05-Feb-60	28-Jun-10	6.1Yrs	56.5Yrs
Harry	13-Apr-65	19-Jul-10	6.1Yrs	51.3Yrs
Angel	19-Aug-58	19-Jul-10	6.1Yrs	58.0Yrs
Peter	12-Jul-54	19-Jul-10	6.1Yrs	62.1Yrs
Mark	06-Aug-63	19-Jul-10	6.1Yrs	53.0Yrs
Maureen	23-Oct-74	06-Sep-10	5.9Yrs	41.8Yrs
Augustine	07-Dec-78			37.7Yrs
Amanda	04-May-59			57.3Yrs
Neville	07-Aug-54	06-Sep-10	5.9Yrs	62.0Yrs

6

#### **For Your Reference...**

To *create calculated dates*:

1. In query **Design View**, click in the first free column
  2. Type a formula – a date field must be enclosed in square brackets
  3. Run the query

## **Handy to Know...**

- In queries, when calculating with dates, date fields from the table must be shown in square brackets ([ ]). You can also refer to a specific date by enclosing it between hash signs. For example, if you want to reference *March 10, 2016* you would type it as #10/3/2016#

# USING CRITERIA IN CALCULATIONS

Once a calculated field has been created, it is much the same as any other field in the query grid. Just like the other fields it also has a **Criteria** cell and you can add selection criteria to

it. For example, you can use the criteria field to find out how many employees have more than three years of service.

## Try This Yourself:

Same File

*Continue using the previous file with this exercise, or open the file Calculation Queries 6.accdb...*

- 1 Open **Query – Date Calculation Example** in **Design View**
  - 2 Click in the **Criteria** cell for **Age** and type **<=50**
  - 3 On the **Query Tools: Design** tab, click on the top half of **View** in the **Results** group to see all employees who are 50 years or younger
  - 4 Click on the top half of **View** in the **Views** group to return to **Design View**
  - 5 Clear the **Criteria** in **Age** and try the other examples, as shown
  - 6 Click on the **File** tab, click on **Save As**, then click on **Save Object As** and click on **[Save As]**
  - 7 Type **Query – Criteria in Calculations Example** in **Save to**, then click on **[OK]**
  - 8 Close the query

- 2
- 3

Query - Date Calculation Example					
FirstName	DateOfBirth	Started	Service	Age	
Maureen	23-Oct-74	06-Sep-10	5.9Yrs	41.8Yrs	
Augustine	07-Dec-78			37.7Yrs	
Petra	03-Apr-81	06-Sep-10	5.9Yrs	35.4Yrs	
Jerry	09-Oct-75	06-Sep-10	5.9Yrs	40.9Yrs	
Victor	02-Apr-73	06-Sep-10	5.9Yrs	43.4Yrs	
Sandra	06-Nov-78	06-Sep-10	5.9Yrs	37.8Yrs	
Charles	20-Dec-77	06-Sep-10	5.9Yrs	38.7Yrs	
Lance	03-May-75	23-Sep-10	5.9Yrs	41.3Yrs	
Antony	15-Aug-68	02-Dec-10	5.7Yrs	48.0Yrs	

*Note: your screen will vary from this one because the Age and Service calculations are based on the current date at the time that you run the query*

- ## 5 Criteria

	<p><b>Will show...</b></p> <p><i>Employees 50 years or over</i></p> <p><i>Employees in their 20s</i></p> <p><i>Trick one! Returns all employees – can you figure out why?</i></p> <p><i>Employees either in their 20s or 40s</i></p>
--	--

## **For Your Reference...**

To **apply criteria** to a **calculated field**:

1. In query **Design View**, click in the **Criteria** cell of the calculation field
  2. Type the criteria
  3. Run the query

### **Handy to Know...**

- In a query, if you have a numerical field and you apply the criteria **>=20 OR <30** you will return *all* records because each record is either greater than (or equal to) 20 or less than 30. The two dates therefore negate one another.

# CONCATENATING STRING FIELDS

**Concatenation** refers to joining things together. There may be times in an application when you want to have two fields concatenated. This can be done using an expression in a calculated

query and combining the two fields together using a special character. The character used to join fields is the **ampersand** (&).

## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file *Calculation Queries\_7.accdb*...

- 1 Open *qryEmployees* in **Design View**
- 2 Click in the first blank **Field** cell and type the field as shown
- 3 On the **Query Tools: Design** tab, click on the top half of **View** in the **Results** group to run the query  
*Notice how the fields have been joined to the text in quotation marks, using an ampersand character...*
- 4 Click on the **File** tab, click on **Save As**, then click on **Save Object As** and click on **[Save As]**
- 5 Type **Query – Concatenation Example** in **Save to**, then click on **[OK]**
- 6 Close the query

Field:	LastName	FirstName	DateOfBirth	Started	[Department]
Table:	Employees	Employees	Employees	Employees	
Sort:					
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:					
or:					

- 2 Submitted by: [FirstName] & " " & [LastName] & " from " & [Department]

qryEmployees	FirstName	DateOfBirth	Started	Submitted by
Julianne	05-Feb-60	28-Jun-10	Julianne Kerr	from Executive
Harry	13-Apr-65	19-Jul-10	Harry Jones	from Executive
Angel	19-Aug-58	19-Jul-10	Angel Harrington	from Executive
Peter	12-Jul-54	19-Jul-10	Peter Dawson	from Executive
Mark	06-Aug-63	19-Jul-10	Mark Jones	from Executive
Maureen	23-Oct-74	06-Sep-10	Maureen Grayson	from Executive
Augustine	07-Dec-78		Augustine Millson	from Administration
Amanda	04-May-59		Amanda Bennett	from Administration
Neville	07-Aug-54	06-Sep-10	Neville Smith	from Administration
Petra	03-Apr-81	06-Sep-10	Petra Henricks	from Administration
Vivienne	22-Nov-61	06-Sep-10	Vivienne Clark	from Administration
Jerry	09-Oct-75	06-Sep-10	Jerry Hancock	from Administration

- 3

## For Your Reference...

To **concatenate fields**:

1. In query **Design View**, click in the first available blank column
2. Type an expression joining fields together using the ampersand character (&)
3. Run the query

## Handy to Know...

- When concatenating string fields in a query, you can use a plus sign (+) to join fields. However, this can be a little confusing as it normally denotes arithmetic additions. It is best, therefore, to use the ampersand (&) for concatenation.

## CHAPTER 5

 INFOCUS

# AGGREGATION QUERIES

**Aggregation** queries are another form of calculation query where the values in one or more fields are aggregated to produce totals, averages, counts, and the like.

In its simplest form an aggregation query allows you to sum the values in a column (field). Unlike a spreadsheet where you have to enter a formula or function to do this, Access takes care of all of the formula building for you – all you need to do is to specify the settings.

**In this session you will:**

- ✓ learn how to create an aggregation query
- ✓ learn how to work with aggregation queries
- ✓ learn how to create more than one aggregation in a query
- ✓ learn how to modify aggregation headings
- ✓ learn how to aggregate calculated fields
- ✓ learn how to apply criteria to aggregations
- ✓ gain an understanding of nesting queries
- ✓ learn how to create an assembly query for nesting
- ✓ learn how to create nesting queries.

# CREATING AN AGGREGATION QUERY

An aggregation query begins as a select query. You start by choosing the tables or queries that the new query will be based on and the fields to include in the query. You then specify a

calculation operation for at least one field in its Total row. There are a number of operations that can be used including Sum, Count, Average, Min, and Max.

## Try This Yourself:

**Open File**

Before starting this exercise you **MUST** open the file *Aggregate Queries\_1.accdb*...

- 1 Click on the **Create** tab, then click on **Query Design** in the **Queries** group to display the **Show Table** dialog box
- 2 Double-click on each of the tables to add them to the query, then click on **[Close]**
- 3 Double-click on the fields as shown to place them in the grid
- 4 On the **Query Tools: Design** tab, click on **Totals** in the **Show/Hide** group, to add the **Totals** row to the grid
- 5 Click on **Group By** for **Amount**, click on the drop arrow, then select **Sum**
- 6 On the **Query Tools: Design** tab, click on **Run** in the **Results** group to run the query
- 7 Click on the **File** tab, select **Save As**, click on **Save Object As**, then click on **[Save As]**
- 8 Type **Expenses Summary**, then click on **[OK]**
- 9 Close the query

3 Table:	Field:
<i>Employees</i>	<i>Department</i>
<i>Expense Type</i>	<i>Description</i>
<i>Expense Transactions</i>	<i>Amount</i>

Field:	Department	Description	Amount	
Table:	Employees	Expense Type	Expense Transactions	
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:				
or:				

5

6	Query
	Department Description SumOfAmo!
	Administration Accommodation \$20,964.38
	Administration Accommodation \$9,198.53
	Administration Coffee and Tea \$5,711.12
	Administration Meals \$5,382.60
	Administration Postage \$3,064.31
	Executive Accommodation \$14,842.68
	Research & Development Accommodation \$4,992.34
	Research & Development Other Expense \$4,578.75
	Sales & Marketing Gifts \$2,021.73
	Sales & Marketing Postage \$2,485.98

6

## For Your Reference...

To *create* an **aggregation query**:

1. Create the query design
2. On the **Query Tools: Design** tab, click on **Totals** in the **Show/Hide** group
3. Change the **Group By** to an aggregation operation for the appropriate fields

## Handy to Know...

- **Group By** is the real key to the operation in an aggregation query. If a **SUM** operation is performed for example, the **Group By** translates as “group by this field and sum each of the categories in the group”.

# WORKING WITH AGGREGATION QUERIES

If you have only one Group By field in an aggregation query, the data will be grouped by each unique entry in the Group By field. If you have more than one Group By field, the records

will be grouped in a nested configuration where the first field forms the first group, the second field is then grouped within the first group, the third field is grouped within the second group, and so on.

## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file *Aggregate Queries\_2.accdb*...

- 1 Open the query **Expenses Summary** in **Design View**
- 2 Rework the query grids as shown – save after each one, using the names as shown
- 3 Close the last query you worked on

2

Field:	<b>LastName</b>	Description	Amount	
Table:	Employees	Expense Type	Expense Transactions	
Total:	Group By	Group By	Sum	
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:	or:			

Save as: **Total By Employee**

Field:	<b>Description</b>	Amount		
Table:	Expense Type	Expense Transactions		
Total:	Group By	Sum		
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteria:	or:			

Save as: **Total By Type**

Field:	<b>ExpDate</b>	Amount		
Table:	Expense Transactions	Expense Transactions		
Total:	Group By	Sum		
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteria:	or:			

Save as: **Total By Date**

Field:	<b>ExpDate</b>	Description	Amount	
Table:	Expense Transactions	Expense Type	Expense Transactions	
Total:	Group By	Group By	Sum	
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:	or:			

Save as: **Total By Date And Type**

## For Your Reference...

To use **Group By** in **aggregation queries**:

1. Double-click on database fields to add them to the query grid and ensure the **Total** row is visible
2. Use **Group By** to determine the order and grouping of the records, or select a summary function if required

## Handy to Know...

- When working with aggregation queries, don't be tempted to have too many groups because the resulting report may be quite meaningless. It is better to have two or three simple reports than a highly detailed one that is difficult to interpret.

# MULTIPLE AGGREGATIONS

An **aggregate query** can use multiple fields for calculating operations. They are not limited to one summary field. Aggregate queries can create group summaries using a wide variety of

mathematical functions, so you can summarise the same field in several different ways, as well as summarising more than one field in the same query.

## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file **Aggregate Queries\_3.accdb**...

**1** Open the query **Expenses Summary** in **Design View**

**2** Double-click on **Amount** in the **Expense Transactions** table three times to add the field three more times to the grid

**3** Click on **Group By** for the second **Amount** field, click on the drop arrow, then select **Count**

**4** Repeat step **3** and change the third **Amount** field to **Avg**

**5** Repeat step **3** and change the fourth **Amount** field to **Max**

**6** Click on the **File** tab, select **Save As**, click on **Save Object As**, then click on **[Save As]**

**7** Type **Expenses Statistics**, then click on **[OK]**

**8** Run the query to see the data

**9** Close the query

Field:	Description	Amount	Amount	Amount	Amount
Table:	Expense Type	Expense Transactions	Expense Transactions	Expense Transactions	Expense Trans.
Total:	Sum	Group By	Group By	Group By	Group By
Show:	<input checked="" type="checkbox"/>				
Criteria:					
or:					

**2**

Field:	Description	Amount	Amount	Amount	Amount
Table:	Expense Type	Expense Transactions	Expense Transactions	Expense Transactions	Expense Trans.
Total:	Sum	Count	Avg	Max	
Show:	<input checked="" type="checkbox"/>				
Criteria:					
or:					

**5**

Expenses Statistics					
Department	Description	SumOfAmo	CountOfAm	AvgOfAmou	MaxOfAmo
Administration	Accommodation	\$20,964.38	88	\$238.23	\$438.39
Administration	Accommodation	\$9,198.53	36	\$255.51	\$448.79
Administration	Coffee and Tea	\$5,711.12	56	\$101.98	\$436.55
Administration	Meals	\$5,382.60	36	\$149.52	\$434.24
Administration	Postage	\$3,064.31	26	\$117.86	\$486.38
Executive	Accommodation	\$14,842.68	52	\$285.44	\$476.04
Research & Development	Accommodation	\$4,992.34	28	\$178.30	\$375.15
Research & Development	Other Expense	\$4,578.75	28	\$163.53	\$482.18
Sales & Marketing	Gifts	\$2,021.73	26	\$77.76	\$319.37
Sales & Marketing	Postage	\$2,485.98	26	\$95.61	\$441.74

**8**

## For Your Reference...

To use **multiple aggregations** in a query:

1. Display the **Total** row in the query grid
2. Add the field(s) to aggregate
3. Click on the drop arrow for **Group By** for each field and select an aggregate function

## Handy to Know...

- When running multiple aggregates, empty rows (null values) will not be included in calculations. To convert null values to zero, use the **Nz** function. For example, the value of the calculated field **Nz([Amount],0)** will be the amount if the field **Amount** is not null, or zero if it is null.

# MODIFYING AGGREGATION HEADINGS

When you create aggregate queries, names are automatically assigned to the headings. Access uses a combination of the field name and the aggregate function, resulting in names like

SumOfAmount, AvgOfAmount, and MaxOfAmount. Fortunately, you can modify the headings to provide something more descriptive and meaningful for your users.

## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file **Aggregate Queries\_4.accdb...**

- 1 Open the query **Expenses Statistics** in **Design View**
- 2 In the field list, click to the left of the first **Amount** and type **Total:** then press **Tab** to move to the next field
- 3 Repeat step 2 and add the following text to the remaining aggregation **Amounts:**  
**Count Number:**  
**Avg Average:**  
**Max Maximum:**
- 4 Save the changes, then run the query to display the modified headings
- 5 Close the query

Field:	Description	Total: Amount	Amount	Amount	Amount
Table:	Expense Type	Expense Transactions	Expense Transactions	Expense Transactions	Expense Transactions
Total:	Sum	Count	Avg	Max	
Sort:					
Show:	<input checked="" type="checkbox"/>				
Criteria:					
or:					

2

Field:	Description	Total: Amount	Number: Amount	Average: Amount	Maximum: Amount
Table:	Expense Type	Expense Transactions	Expense Transactions	Expense Transactions	Expense Transactions
Total:	Sum	Count	Avg	Max	
Sort:					
Show:	<input checked="" type="checkbox"/>				
Criteria:					
or:					

3

Department	Description	Total	Number	Average	Maximum
Administration	Accommodation	\$20,964.38	88	\$238.23	\$438.39
Administration	Accommodation	\$9,198.53	36	\$255.51	\$448.79
Administration	Coffee and Tea	\$5,711.12	56	\$101.98	\$436.55
Administration	Meals	\$5,382.60	36	\$149.52	\$434.24
Administration	Postage	\$3,064.31	26	\$117.86	\$486.38
Executive	Accommodation	\$14,842.68	52	\$285.44	\$476.04
Research & Develop	Accommodation	\$4,992.34	28	\$178.30	\$375.15
Research & Develop	Other Expense	\$4,578.75	28	\$163.53	\$482.18
Sales & Marketing	Gifts	\$2,021.73	26	\$77.76	\$319.37
Sales & Marketing	Postage	\$2,485.98	26	\$95.61	\$441.74

4

## For Your Reference...

To **modify aggregation headings:**

- In query **Design View**, click in the **Field** cell and type the new name before the field name, followed by a colon e.g.  
**Total:[Amount]**

## Handy to Know...

- When you double-click on the **border** between headings in a displayed query, it automatically sizes the columns to fit the data displayed in the query. You can do the same in the query grid to check detailed criteria or to display more fields at once.

# AGGREGATING CALCULATED FIELDS

Access lets you **aggregate** calculated fields. Calculated fields are fields that have been created in the query. For example, in an invoice application you might create a Totals field by

multiplying the Quantity and the Price fields. Once created, a calculated field becomes like any other field and can therefore be aggregated to create a summary figure.

## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file *Aggregate Queries\_5.accdb*...

- 1 Open the query **Expenses Summary** in **Design View**
- 2 Click in the first available blank **Field** cell and type **Tax:[Amount]/11** then press **Tab** to move to the next field
- 3 Type **Net Amount:[Amount]-([Amount]/11)**
- 4 Click on the **Group By** cell for **Tax**, click on the drop arrow, then select **Sum**
- 5 On the **Query Tools: Design** tab, click on **Property Sheet** in the **Show/Hide** group to display the **Property Sheet**
- 6 Click in **Format**, then click on the drop arrow and click on **Currency**
- 7 Repeat steps **4** to **6** for the **Net Amount** field
- 8 Close the **Property Sheet** pane, then save and run the query
- 9 Close the query

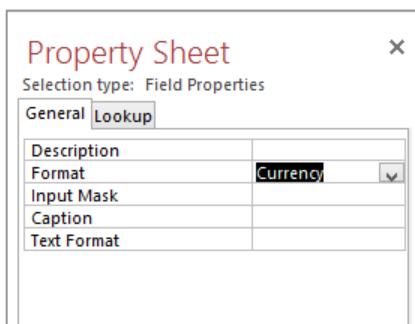
2

Field:	Department	Description	Amount	Tax: [Amount]/11
Table:	Employees	Expense Type		
Total:	Expense Transactions		Expense Transactions	
Group By:	Group By	Group By	Sum	Group By
Sort:				
Show:				
Criteria:				
or:				

3

Amount	Tax: [Amount]/11	Net Amount: [Amount]-([Amount]/11)
Expense Transactions		
Sum	Group By	Group By

6



8

Department	Description	SumOfAmount	Tax	Net Amount
Administration	Accommodation A	\$20,964.38	\$1,905.85	\$19,058.53
Administration	Accommodation B	\$9,198.53	\$836.23	\$8,362.30
Administration	Coffee and Tea Expenses	\$5,711.12	\$519.19	\$5,191.92
Administration	Meals	\$5,382.60	\$489.33	\$4,893.27
Administration	Postage	\$3,064.31	\$278.57	\$2,785.74
Executive	Accommodation A	\$14,842.68	\$1,349.33	\$13,493.34
Research & Development	Accommodation A	\$4,992.34	\$453.85	\$4,538.49
Research & Development	Other Expenses	\$4,578.75	\$416.25	\$4,162.50
Sales & Marketing	Gifts	\$2,021.73	\$183.79	\$1,837.94
Sales & Marketing	Postage	\$2,485.98	\$226.00	\$2,259.98

## For Your Reference...

To **aggregate calculated fields**:

1. Create a calculated field
2. Click in the **Total** row of the calculated field and select an appropriate aggregation operation (e.g. **Sum**, **Avg**, **Count**, etc)

## Handy to Know...

- You can create calculated fields that use aggregation functions. Change the **Total** row for a field to **Expression**, then type an expression in the **Field** row that uses aggregate functions e.g. **Sum([Amount])**.

# APPLYING CRITERIA TO AGGREGATES

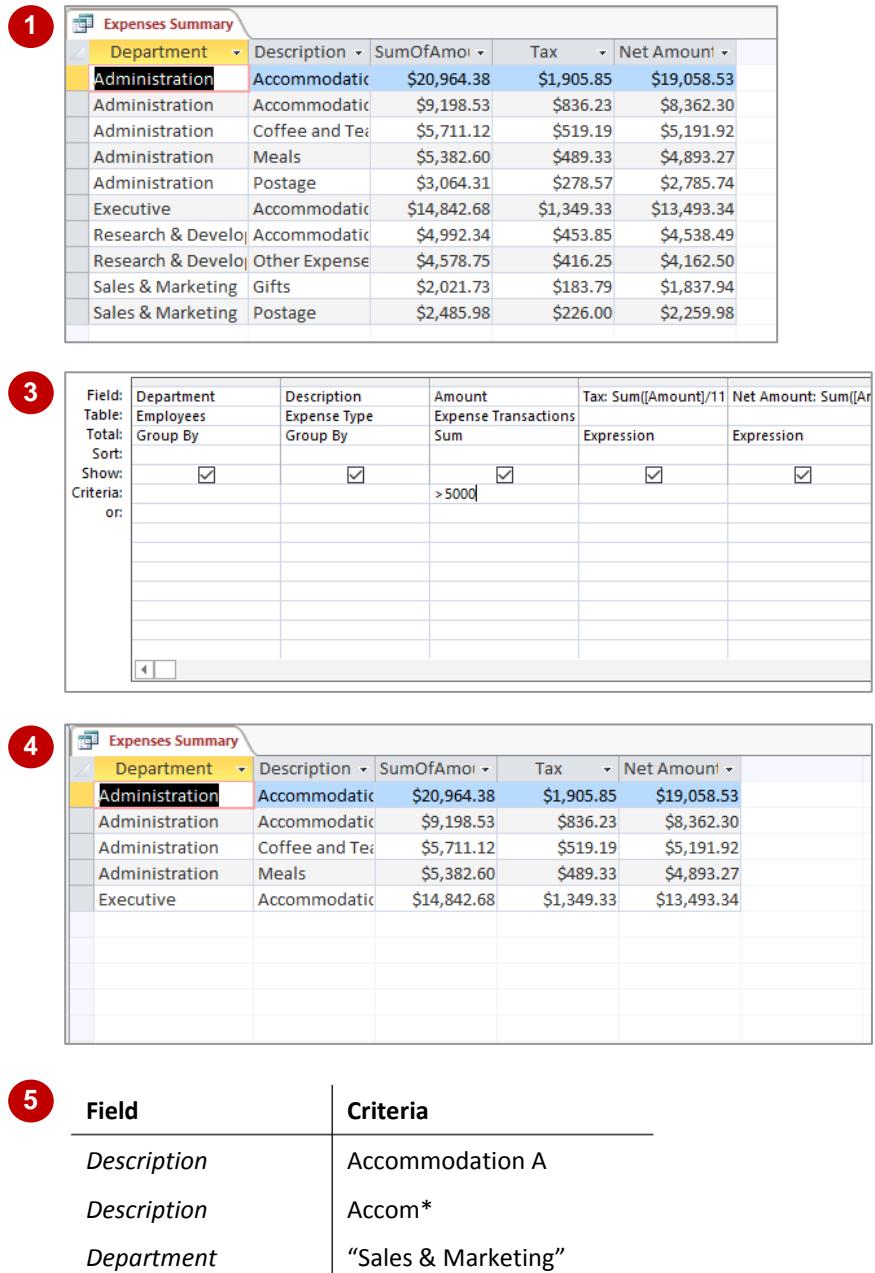
An aggregate query has many of the same attributes as a standard select query including the ability to be able to apply criteria to the aggregation. This is useful when you want to

select and aggregate specific records. For example, you could create an aggregate query that calculates the expenses for a department which has been specified by the user via a parameter.

## Try This Yourself:

**Same File** Continue using the previous file with this exercise, or open the file Aggregate Queries\_6.accdb...

- 1 Double-click on **Expenses Summary** to see a summary of expense transactions
  - 2 On the **Home** tab, click on the top half of **View** in the **Views** group to switch to **Design View**
  - 3 Click in the **Criteria** cell for **Amount** and type **>5000**
  - 4 On the **Query Tools: Design** tab, click on **Run** in the **Results** group, to see only summaries that have a total greater than 5000
  - 5 On the **Home** tab, click on the top half of **View** in the **Views** group, to return to **Design View** and try the criteria as shown – clear the previous criteria before trying each
  - 6 Close the query – click on **[No]** to discard the changes



## **For Your Reference...**

To **apply criteria** to **aggregation queries**:

1. In **Design View**, click in the appropriate **Criteria** cell and type the criteria
  2. On the **Query Tools: Design** tab, click on **Run** in the **Results** group

## **Handy to Know...**

- When applying criteria to aggregate queries, the criteria can either be fixed or involve a parameter, like **[Enter the Description]**, so that the user can provide the information each time the query runs.

# UNDERSTANDING NESTED QUERIES

Sometimes the scope of what you are trying to achieve with your queries is just too much for one query to handle. Sometimes you may be able to achieve very difficult operations but because of

the complexity of expressions, the whole thing becomes difficult to interpret later on. When this occurs you should consider ***nesting queries*** where one query calls on another to produce results.

---

## Queries That Use Queries

Normally when you create a new query you choose one or more tables upon which to base the query design. This is done through the **Show Table** dialog box.

However, the **Show Table** dialog box has a tab specifically for queries, where you can base your new query not on a table but on a pre-existing query in the database. When a new query is based on an existing query the process is referred to as ***nesting*** queries.

In our case study database we want to be able to produce a summary of transactions by transaction type, showing both net of tax and tax inclusive amounts, for a range of dates that can be specified by the user when the query is run. This will involve:

- the creation of calculated fields for tax, and net of tax amounts
- the use of parameters to allow the user to choose a date range
- the aggregation of the amounts.

If we tried to create this in one aggregation query we would end up with two **Group By** fields: one for the date and the other for the type of transaction. While we might be able to achieve this with convoluted expressions and criteria, we are better off creating two distinctly different queries:

- the first to calculate the tax and net of tax amounts and to provide the parameters where the user can specify a date range
- the second to aggregate the data from the first query into totals.

## The Assembly Query

The first query is usually where the majority of the work takes place and the raw data is assembled for further manipulation in the next query. The first query is known as an ***assembly query*** because the appropriate data is gathered, assembled, and extracted ready to pass on to the next query.

This query normally has:

- one or more related tables, forming the basis of the field listings
- calculated fields based on the fields from the tables
- parameters to allow the user to choose a data range.

## The Nesting Query

The ***nesting query*** uses the ***assembly query*** rather than tables from the database. This allows further manipulation of the data in the desired result. Nesting queries are used either because it would not be possible to place all of the manipulations into one query, or because to do so would require too high a level of complexity in expressions, criteria, parameters, formulas, and the like.

It is the ***nesting query*** that is used to display the final, desired results. When a nested query is run it first runs the assembly query upon which it is based. All of the necessary calculations and machinations from the assembly query are placed into a temporary ***dynaset*** (kind of like a results table), which then becomes the source data for the nesting query to further manipulate.

You can also run the assembly query as you would any other query, but it will stop when it has done its work as there is no direct two-way link between it and the nesting query.

# CREATING AN ASSEMBLY QUERY

When creating a nested query scenario the first query is usually where most of the manipulation and calculations take place. The first query, which we'll refer to as the **assembly query**,

usually utilises data from one or more tables and may include calculated fields and even parameters, allowing the user more choice in the data content.

## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file **Aggregate Queries\_7.accdb...**

- 1 Click on the **Create** tab, then click on **Query Design** in the **Queries** group
- 2 Double-click on **Expense Transactions** and **Expense Type**, then click on **[Close]**
- 3 Double-click on the fields as shown, to place them in the grid
- 4 Create two new calculation fields as shown
- 5 On the **Query Tools: Design** tab, click on **Property Sheet** in the **Show/Hide** group, then format the calculated fields as **Currency**
- 6 Click in the **Criteria** cell for **ExpDate** and type **Between [Enter Start Date] and [Enter End Date]**
- 7 Save the query as **Expenses (exTax)**
- 8 Close the query – try running the query using **1/3/15** and **31/3/15** as parameters

3 Table	Field
<i>Expense Transactions</i>	ExpDate
<i>Expense Type</i>	Description
<i>Expense Transactions</i>	Amount

Field:	ExpDate	Description	Amount	
Table:	Expense Transactions	Expense Type	Expense Transactions	
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:				
or:				

- 4 **Tax:**  $[Amount] - ([Amount]/1.1)$   
and

**Net Amount:**  $[Amount]/1.1$

Field:	Description	Amount	Tax: [Amount]-([Amount]/1.1)	Net Amount: [Amount]/1.1
Table:	Expense Type	Expense Transactions		
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:				
or:				

Field:	ExpDate	Description	Amount	Tax: [Amount]-([Amount]/1.1)
Table:	Expense Transactions	Expense Type	Expense Transactions	
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:	<input type="text" value="Between [Enter Start Date] and [Enter End Date]"/>			
or:				

## For Your Reference...

To *create* an **assembly query**:

- Create a new query incorporating fields, calculated fields, and parameters as required

## Handy to Know...

- You can use an assembly query for nested queries or as a standalone query. It does not have any formal links to a nested query. Even when the assembly query is used by a nested query there is no link within the assembly query to the nesting – all of the links are in the nested query.

# CREATING THE NESTING QUERY

A **nesting query** is one that uses a query as a data source rather than a table. Nesting queries are often relatively sparse as much of the real work has been done in the assembly query.

Nesting queries therefore often perform specialised operations such as aggregating data that has been pre-prepared in the assembly query.

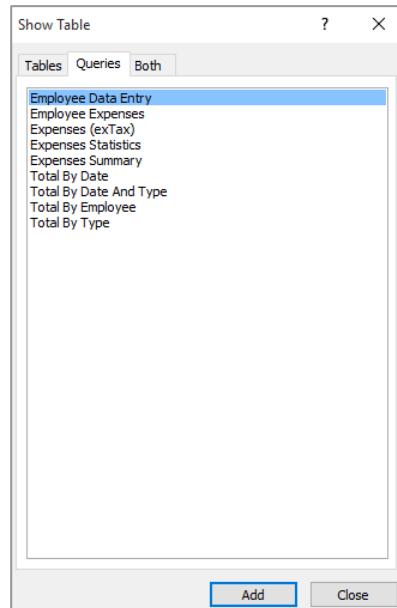
## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file **Aggregate Queries\_8.accdb...**

- 1 Click on the **Create** tab, then click on **Query Design** in the **Queries** group to display the **Show Table** dialog box
- 2 Click on the **Queries** tab to see the list of queries
- 3 Double-click on **Expenses (exTax)**, then click on **[Close]**
- 4 Double-click on **Description, Amount, Tax** and **Net Amount** to add them to the query grid
- 5 On the **Query Tools: Design** tab, click on **Totals** in the **Show/Hide** group, then change the **Group By** operation for **Amount, Tax** and **Net Amount** to **Sum**
- 6 Save the query as **Expenses Summary (exTax)**
- 7 Run the query and use **1/3/15** and **31/3/15** as the parameters
- 8 Close the query

2



5

Field:	Description	Amount	Tax	Net Amount
Table:	Expenses (exTax)	Expenses (exTax)	Expenses (exTax)	Expenses (exTax)
Total:	Sum	Sum	Sum	Sum
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:				
or:				

7

Expenses Summary (ex Tax)	Description	SumOfAmoi	SumOfTax	SumOfNet Amount
Accommodation A	\$2,040.32	185.4835	1854.83854545455	
Accommodation B	\$334.62	30.4199	304.199909090909	
Coffee and Tea Expenses	\$74.46	6.7691	67.692	
Gifts	\$50.67	4.6066	46.0669090909091	
Meals	\$60.59	5.5079	55.0795454545455	
Other Expenses	\$153.55	13.9586	139.586545454545	
Postage	\$57.51	5.2283	52.2837272727273	

## For Your Reference...

To *create a nesting query*:

1. Click on the **Create** tab, then click on **Query Design** in the **Queries** group
2. Click on the **Queries** tab of the **Show Table** dialog box and select the query
3. Add fields, etc. as normal

## Handy to Know...

- Formatting stays with the query in which it was created. You will need to re-format data in a nesting query even though it may have been previously formatted in the assembly query.

## CHAPTER 6

 INFOCUS

# ACTION QUERIES

**Action queries** are used to make large scale changes to the data in one or more tables. There are four action queries available for use.

- Make Table queries allow you to create a new table from an existing table or query. This type of query is ideal for archiving old data or making backup copies.
- Update queries allow you to update data in existing tables. A classic use is to perform a large-scale update of pay scales or prices. For example, if you need to add a 10% sales tax to a range of products, you can use an **Update** query to increment the current prices by 10%.
- Append queries allow you to add records from one table to another. For example, if you need to consolidate the sales of several salespeople, you can use an **Append** query to append the records from the sales people into one table.
- Delete queries allow you to delete records based on criteria. For example, you can use a **Delete** query to delete all sales records that are more than 12 months old.

**In this session you will:**

- ✓ learn how to create a **Make Table** query
- ✓ learn how to run a **Make Table** query
- ✓ gain an understanding of using expressions in **Update Queries**
- ✓ learn how to create an **Update Query**
- ✓ learn how to run an **Update Query** and change data
- ✓ learn how to use an expression to update records in an **Update Query**
- ✓ learn how to run an **Update Query** that is based on an expression
- ✓ learn how to create a **Delete** query
- ✓ learn how to run a **Delete** query
- ✓ learn how to create an **Append** query
- ✓ learn how to run an **Append** query
- ✓ learn how to turn action messages off.

# CREATING A MAKE TABLE QUERY

As the name suggests a Make Table query creates a new table from existing data. You can use a Make Table query to export records to another database, to start an archive table for

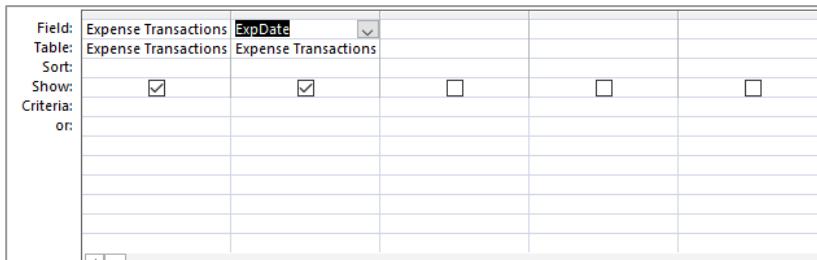
redundant records, to export related information from several tables to Excel or Word, or to create a smaller single entity recordset from a large, related recordset to facilitate further analysis.

## Try This Yourself:

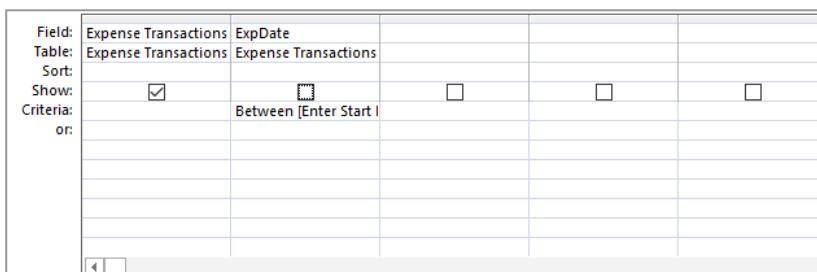
**Open File**

Before starting this exercise you **MUST** open the file *Action Queries\_1.accdb*...

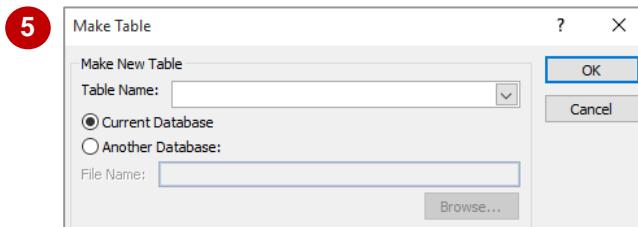
- 1 On the **Create** tab, click on **Query Design** in the **Queries** group to display the **Show Table** dialog box
- 2 Double-click on **Expense Transactions**, then click on **[Close]**
- 3 Double-click on the asterisk to add all fields to the query grid, then double-click on **ExpDate** to add a copy of this field
- 4 Click in the **Criteria** cell for **ExpDate**, then type **Between [Enter Start Date] and [Enter End Date]** and click on **Show** so it appears without a tick
- 5 On the **Query Tools: Design** tab, click on **Make Table** in the **Query Type** group to display the **Make Table** dialog box
- 6 Type **Monthly Expenses** in **Table Name**, then click on **[OK]**
- 7 Click on **Save**, type **Make Monthly Expenses Table**, then click on **[OK]**
- 8 Close the query



- 3 Double-click on the asterisk to add all fields to the query grid...



- 4



The idea with these settings is to create a Make Table query that can be used repeatedly. Let's say we need to produce a list of transactions on a monthly basis. With the query above, we could run it monthly, specify the dates that we are interested in, and only those transactions that match the dates will be outputted into the new table called *Monthly Expenses*. We could then perform further analysis, produce reports and do whatever we like with this table without fear of corrupting the data in the original transactions table.

## For Your Reference...

To **create a Make Table query**:

- 1 Create a new query, specifying the fields you want in the table
- 2 On the **Query Tools: Design** tab, click on **Make Table** in the **Query Type** group and give the output table a name

## Handy to Know...

- Access doesn't allow duplicate fields when a **Make Table** query is run.

# USING A MAKE TABLE QUERY

When you run a **Make Table** query it actually creates a new table of data based on the parameters and criteria that you have specified in the query design. The new table can then be

used as any other table in your database and can be exported, queried and reported on as you see fit. Each time you run the query the table is re-created with the latest data.

## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file Action Queries\_2.accdb...

- Double-click on the query **Make Monthly Expenses Table** to run it

A message advises that you are about to run a Make Table query...

- Click on [Yes] to proceed and display the first parameter prompt

- Type **1/3/15**, then click on [OK] to display the second prompt

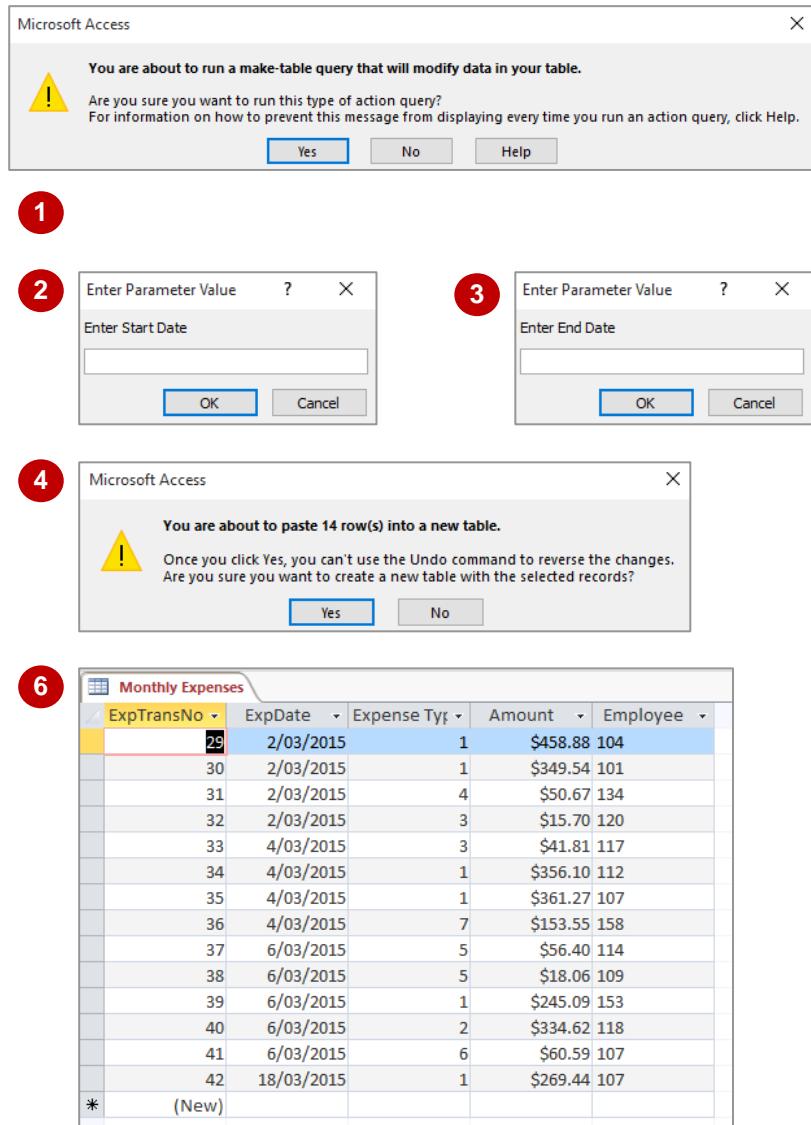
- Type **31/3/16**, then click on [OK]

A message advises that you are about to paste records into a table...

- Click on [Yes] to create a new table called **Monthly Expenses**

- Double-click on the table **Monthly Expenses** and spend a few moments examining the data

- Close the table



## For Your Reference...

To run a **Make Table** query:

- Double-click on the **Make Table** query in the **Navigation** pane
- Click on [Yes]
- Continue to follow the prompts

## Handy to Know...

- If you have already run the **make table** query and created a new table, that table will be deleted and replaced each time the query is run.

# UNDERSTANDING EXPRESSIONS AND UPDATE QUERIES

**Expressions** are used throughout Microsoft Access for a variety of purposes, such as setting properties, creating parameters, defining calculated controls in forms and reports, defining

calculated fields in queries, and setting validation rules. They are also essential in **Update queries** because they define the changes to be made to the records. They are discussed in more detail below.

## What is an Expression?

An **expression** is a combination of symbols – identifiers, operators, and values – that produce a result. For example, you can use the following expression in a control on a form or report to display the sum of values in the *Subtotal* and *Freight* controls: = [**Subtotal**] + [**Freight**].

## Expressions and Update Queries

Expressions are used in **Update Queries** to modify the value in a particular field for each record. For example, if you wanted to increase the price of all of your products by 3%, you could modify each record manually, or you could create an Update Query to do the job for you. If the field that held the price was called *[UnitPrice]*, you would use the expression *[UnitPrice]\*1.03* in the Update Query. When the Update Query was run, Access would take the existing value in *UnitPrice* in each record, and replace it with a value that was 3% higher.

## Examples of Expressions

Update expressions are entered in a special Update To row that appears in the query grid. The following table lists some examples of expressions and describes how they would work in an update query.

Field Type	Expression	Effect in an Update Query
Currency	[Quantity]*[Price]	Multiplies the value in the <i>Quantity</i> field by the value in the <i>Price</i> field to calculate the cost for each record
Currency	[Salary]*1.05	Increases the <i>Salary</i> for each record by 5%
Currency	DSum("[Quantity]*[UnitPrice]","Order Details","[ProductID]="" & [ProductID])	Where the <i>Product IDs</i> in the current table match the <i>Product IDs</i> in the <i>Order Details</i> table, updates the sales totals based on the product of <i>Quantity</i> and <i>UnitPrice</i>
Short Text	[LastName] & ", " & [FirstName]	Joins the <i>LastName</i> and the <i>FirstName</i> fields together, separating them with a comma so that John Smith will appear as: Smith, John
Short Text	"Salesperson"	Changes value to <i>Salesperson</i>
Yes/No	Yes	Changes all of the records to <i>True</i> (Yes)
Date	#1/1/2010#	Changes all of the dates to January 1 2010
Date	[Due]+30	Adds 30 days to the current <i>Due</i> date

## Testing Expressions

Given that you can use expressions to create calculated fields, you may like to create an interim select query to test that your expression gives the result you expect. Use the expression to create a calculated field in a select query on the table you want to update. Examine a good sample of the results to make sure the calculation is working as expected. You can then transfer your calculation to the update query, and make the changes required.

# **PREPARING AN UPDATE QUERY**

An Update **query** allows you to update the data in a table. You can update the data in one or more fields and even in one or more tables at once. You can also elect to update only some of

the records in a table. The first step is to create a select query that lists the records and fields you want to update. In this example, we will update a field in *Expense Transactions*.

## Try This Yourself:

Open  
File

*Before starting this exercise  
you MUST open the file  
Action Queries 3.accdb...*

- 1 Click on the **Create** tab, then click on **Query Design** in the **Queries** group
  - 2 Double-click on **Expense Transactions**, then click on [Close]
  - 3 Double-click on **ExpDate**, **Expense Type**, **Amount** and **Processed**
  - 4 Click in the **Criteria** cell for **ExpDate** and type [Enter the processing date]
  - 5 On the **Query Tools: Design** tab, click on **Update** in the **Query Type** group and notice how the **Sort** row changes to **Update To**
  - 6 Click in the **Update To** cell for **Processed** and type **Yes**
  - 7 Click on **Save**, type **Processing Completed**, then click on [OK]
  - 8 Close the query

6	Field: <input type="text" value="ExpDate"/>	Expense Type	Amount	Processed
Table:	<input type="text" value="Expense Transactions"/>			
Update To:				
Criteria: or:	<input type="text" value="Enter the processing"/>			

*In our case study, expense transactions are processed on a daily basis – this involves examining the claim and passing it on to the employee's manager for approval.*

*In the update query, we want a logical field (called Processed) in our Expense Transactions table to be updated to reflect that transaction records have been processed. We could do this manually by simply editing each record but that's very time-consuming. Since transactions in our case study are processed daily, we want the query to prompt for a date and, when the date is supplied, we want the update query to mark all of the records for that date as "processed" by placing a "Yes" in the Processed field.*

## **For Your Reference...**

To **create** an *Update Query*:

1. Create a new query, specifying the fields you want in the table
  2. On the **Query Tools: Design** tab, click on **Update** in the **Query Type** group, then enter a value/expression in **Update To**

## **Handy to Know...**

- In an **Update Query** you can enter a value directly into an **Update To** cell if the field only accepts true/false or yes/no values. If dealing with dates or numeric values, you would most likely have to create an expression to perform a calculation.

# RUNNING AN UPDATE QUERY

**Update queries** are used to update records in tables. In our case study some employees are authorised to claim their expenses back through their pay. The **Update** query that we've created

will update transactions and mark them as processed. It does this by prompting for the transaction date and then marking the *Processed* field for all records found for this date.

## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file **Action Queries\_4.accdb...**

- Double-click on the query **Processing Completed** to run it

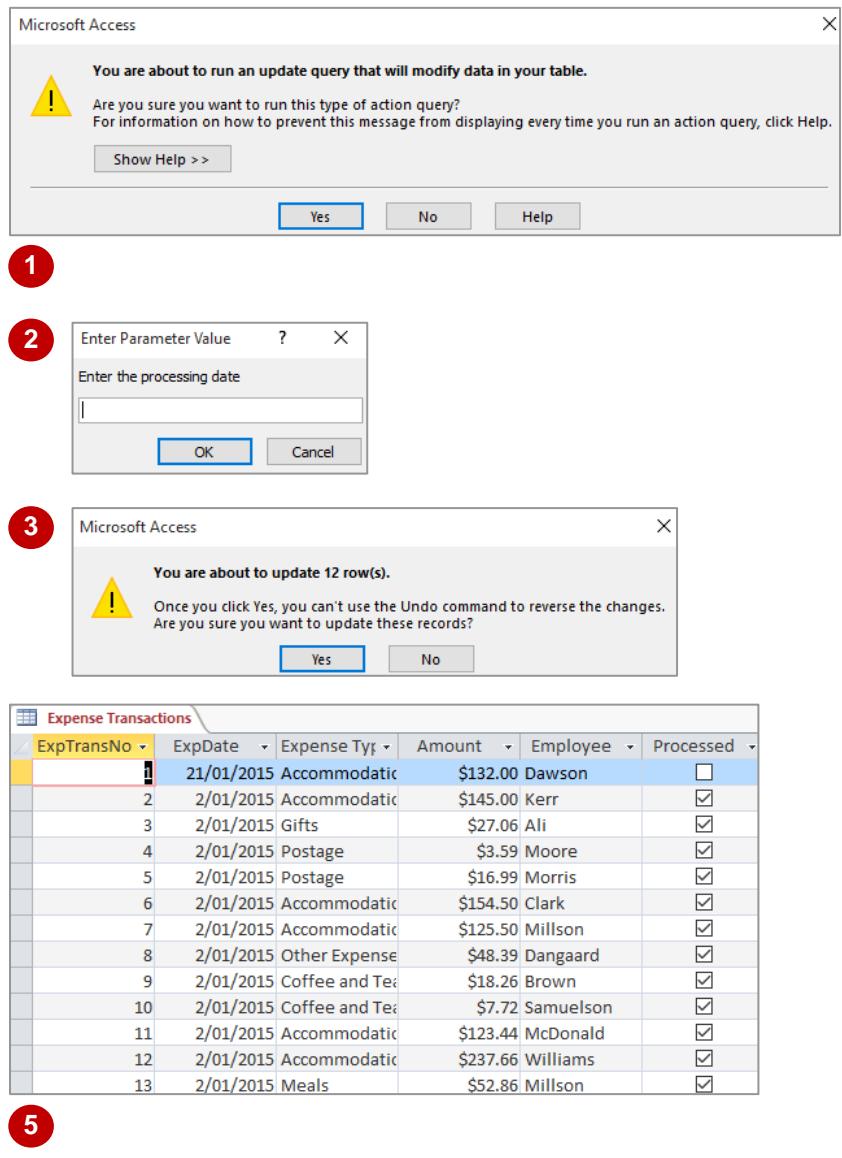
Since this is an update query you will be notified that data may be modified...

- Click on [Yes] – the parameter box will appear requesting a processing date
- Type **2/1/15**, then click on [OK]

You will now be advised of the number of records that will be updated...

- Click on [Yes] to perform the update
- Double-click on **Expense Transactions** to open the table – all records for 2/1/15 should show a tick in the **Processed** field

- Close the query



## For Your Reference...

To run an **Update query**:

- Double-click on the update query in the **Navigation** pane
- Click on [Yes]
- Continue to follow the prompts

## Handy to Know...

- When running update queries where a lot of data is updated, it may be handy to also create a sibling update query that reverses the changes.

# UPDATING USING EXPRESSIONS

**Expressions** can be used in the **Update To** cell of an **Update Query** rather than using hard-coded values. Expressions are often used to increment or decrement a date or a numeric value.

value. For example, you can update the pay date to the next date the pays are processed, give your employees a 5% pay rise, and more by using expressions.

## Try This Yourself:

Same  
File

*Continue using the previous file with this exercise, or open the file Action Queries 5.accdb...*

- 1 Click on the **Create** tab, then click on **Query Design** in the **Queries** group
  - 2 Double-click on **Employees** and **Personal Details**, then click on **[Close]**
  - 3 Double-click on **LastName** and **Salary** to add these fields to the grid
  - 4 Click in the **Criteria** cell for **LastName** and type **[Last name of employee:]**
  - 5 On the **Query Tools: Design** tab, click on **Update** in the **Query Type** group to display the **Update To** row
  - 6 Click in the **Update To** cell for **Salary** and type **[Employees.Salary] \* ((1 + [Percentage Increase]) / 100)**
  - 7 Click on **Save**, type **Salary Increase**, then click on **[OK]**
  - 8 Close the query



Field:	LastName	Salary		
Table:	Employees	Employees		
Date To:				
Criteria:	<b>Last name of employ</b>			
or:				
<input type="button" value="&lt;"/> <input type="button" value="&gt;"/>				

*In this query a parameter is used to prompt for the LastName of the employee to update. A second parameter within the expression is then used to capture the percentage increase that will be made to the employee's salary.*

For example, if the user types 5, the salary will be increased by 5% – the formula to do this reads “take the current salary [Salary] and multiply it by 1, plus whatever the user specifies as [Percentage Increase] divided by 100”.

If the user types 5, meaning 5%, this formula will be:  
[Salary] \* 1.05.

## **For Your Reference...**

To **use** an *expression* to *update records*:

1. Create a new query, specifying the fields you want in the table
  2. On the **Query Tools: Design** tab, click on **Update** in the **Query Type** group, then enter an expression in the **Update To** cell

## **Handy to Know...**

- Expressions can be tricky to create in update queries and you should take care before running the query, ensuring that you have a backup of the original data.

# RUNNING AN EXPRESSION-BASED UPDATE

Running an update query based on an expression is no different to running any other update query. The only real difference is in determining what changes have been made and

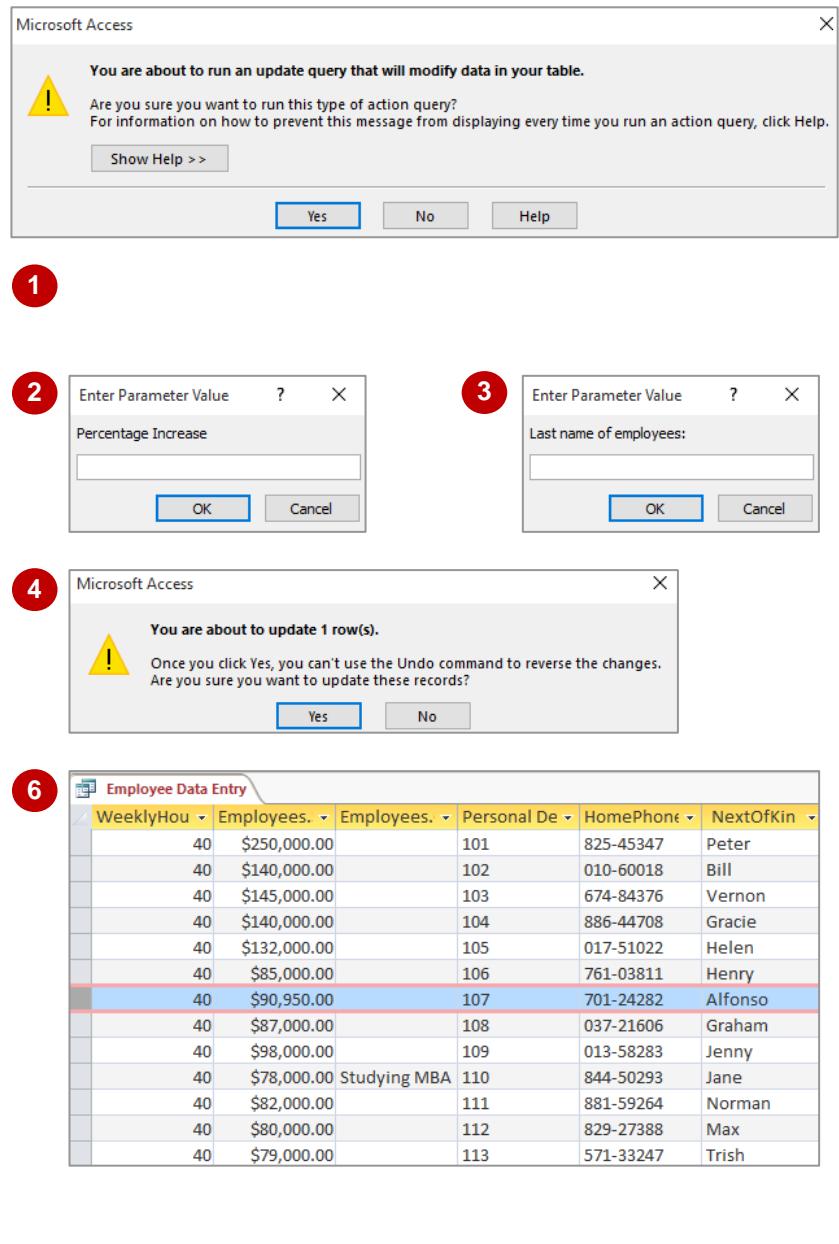
where they have occurred. Expression-based updates are usually more subtle than their full-replacement siblings so you really need to know your data before running an expression update.

## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file Action Queries\_6.accdb...

- 1 Double-click on the query **Salary Increase** to run it – you will receive a warning that an update will occur
- 2 Click on [Yes] – the first parameter prompt requesting the amount of increase appears
- 3 Type 7 (i.e. 7%), then click on [OK] to display the prompt for the employee's name
- 4 Type **Millson**, then click on [OK]
- Access will now advise that one record will be updated...
- 5 Click on [Yes] to perform the update
- 6 Double-click on the query **Employee Data Entry**, click on the record for **Millson** (#107), then scroll right to see how the salary shows **\$90,950** – an increase of 7% over \$85,000
- 7 Close the query



## For Your Reference...

To run an **expression-based update query**:

1. Double-click on the name of the query in the **Navigation** pane
2. Follow the prompts

## Handy to Know...

- When using parameters in an update query, be sure to use one or more fields with unique results to indicate specific records or you might update more records than intended.

# CREATING A DELETE QUERY

As its name suggests, the Delete query is used to delete records from a table. The deletion is based on criteria in the query grid – any record that matches the criteria will be deleted. Be careful

though – if you have enforced referential integrity and Cascade Delete Related Records is ticked, Access will delete matching records in any related table even if it is not included in the query.

## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file *Action Queries\_7.accdb*...

- 1 Click on the **Create** tab, then click on **Query Design** in the **Queries** group
- 2 Double-click on the **Expense Transactions** table, then click on **[Close]**
- 3 Double-click on **ExpDate** and **Processed** to place these fields into the query grid
- 4 Click in the **Criteria** cell for **ExpDate** and type **<=[Delete processed records on or earlier than]**
- 5 Click in the **Criteria** cell for **Processed** and type **Yes**
- 6 On the **Query Tools: Design** tab, click on **Delete** in the **Query Type** group to change the **Sort** row to a **Delete** row
- 7 Click on **Save**, then type **Delete Processed Transactions**, then click on **[OK]**
- 8 Close the query

4

Field:	ExpDate	Processed		
Table:	Expense Transactions	Expense Transactions		
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

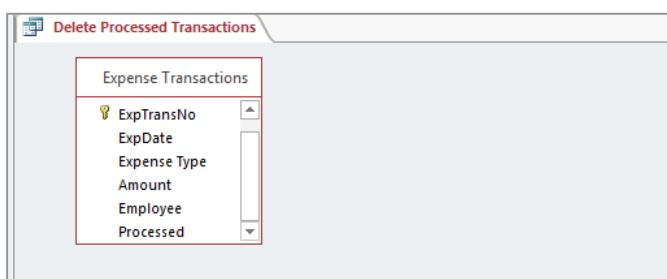
Criteria:  
or:  
on or earlier than]

6

Field:	ExpDate	Processed		
Table:	Expense Transactions	Expense Transactions		
Delete:			Where	
Criteria:	<=[Delete processed	Yes	Where	

or:

7



## For Your Reference...

To *create a delete query*:

1. Create a new query, specifying the fields you want in the table and criteria for record selection
2. On the **Query Tools: Design** tab, click on **Delete** in the **Query Type** group

## Handy to Know...

- Before changing a query to a delete query, run it as a select query and check which records will be deleted. Once deleted, records cannot be recovered, except from a backup file.

# RUNNING A DELETE QUERY

**Delete queries** are potentially very dangerous queries as they will delete data from your database. While we won't do it here, some precautions you should take before running a

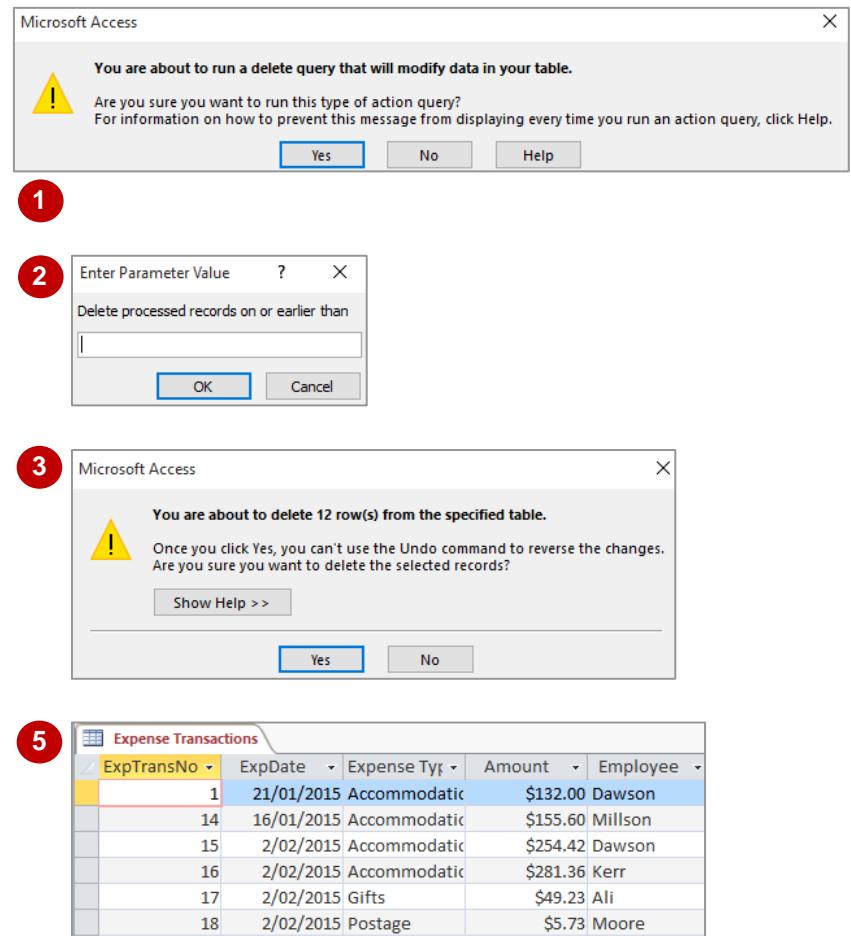
delete query include making a full backup of your database file and also running a **Select query** with the same criteria to see exactly what records will be deleted by the **Delete query** when it is run.

## Try This Yourself:

Same File

Continue using the previous file with this exercise, or open the file Action Queries\_8.accdb...

- 1 Double-click on the query **Delete Processed Transactions** to run it – a warning message indicates that data will be changed
- 2 Click on [Yes] to display the transaction date prompt
- 3 Type **2/1/15**, then click on [OK]  
*The number of records flagged for deletion will be displayed in a final prompt...*
- 4 Click on [Yes] to delete the records
- 5 Double-click on the table **Expense Transactions** and notice that there are no records for 2/1/2015
- 6 Close the table



## For Your Reference...

To run a **delete query**:

1. Double-click on the delete query in the **Navigation** pane
2. Follow the prompts

## Handy to Know...

- Delete queries are often used in conjunction with **Make Table** queries where the data that would be deleted is first put into a new table as an archive.

# CREATING AN APPEND QUERY

An Append **query** allows you to add records from one table into another. The table that holds the records you want to add is known as the **source** table. The table that you want to add them to is

the **target** table. The first step is to create a query in the source table and select the records you want to append. You then convert it to an Append query and nominate the target table.

## Try This Yourself:

**Open File**

Before starting this exercise you **MUST** open the file *Action Queries\_9.accdb*...

- 1 Click on the **Create** tab, then click on **Query Design** in the **Queries** group
- 2 Double-click on the **Expense Transactions** table, then click on **[Close]**
- 3 Double-click on each of the fields to place them all in the query grid
- 4 Click in the **Criteria** cell for **ExpDate** and type **Between [Enter the start date] And [Enter the end date]**
- 5 On the **Query Tools: Design** tab, click on **Append** in the **Query Type** group to display the **Append** dialog box
- 6 Click on the drop arrow for **Table Name**, select **Transactions Archive**, then click on **[OK]**
- 7 Click on **Save**, then type **Transactions Backup** and click on **[OK]**
- 8 Close the query

Step 1: Query Grid showing fields from the 'Expense Transactions' table.

Step 2: 'Append' dialog box.

Step 3: Query Grid with criteria for 'ExpDate'.

Step 4: 'Transactions Backup' query in the 'Transactions Archive' table.

Step 5: 'Transactions Backup' query in the 'Expense Transactions' table.

## For Your Reference...

To *create* an **append query**:

1. Create a query design using all of the fields from the source table and add the criteria
2. On the **Query Tools: Design** tab, click on **Append** in the **Query Type** group, then nominate the target table

## Handy to Know...

- An **Append** query is similar in operation to a **Make Table** query except that instead of re-creating a new table each time it is run, it simply adds (i.e. appends) the specified records from the source table to the nominated target table.

# RUNNING AN APPEND QUERY

An **Append query** is run from the Navigation pane the same as any other form of query or action query. You must, however, ensure that the nominated target table in the **Append** query

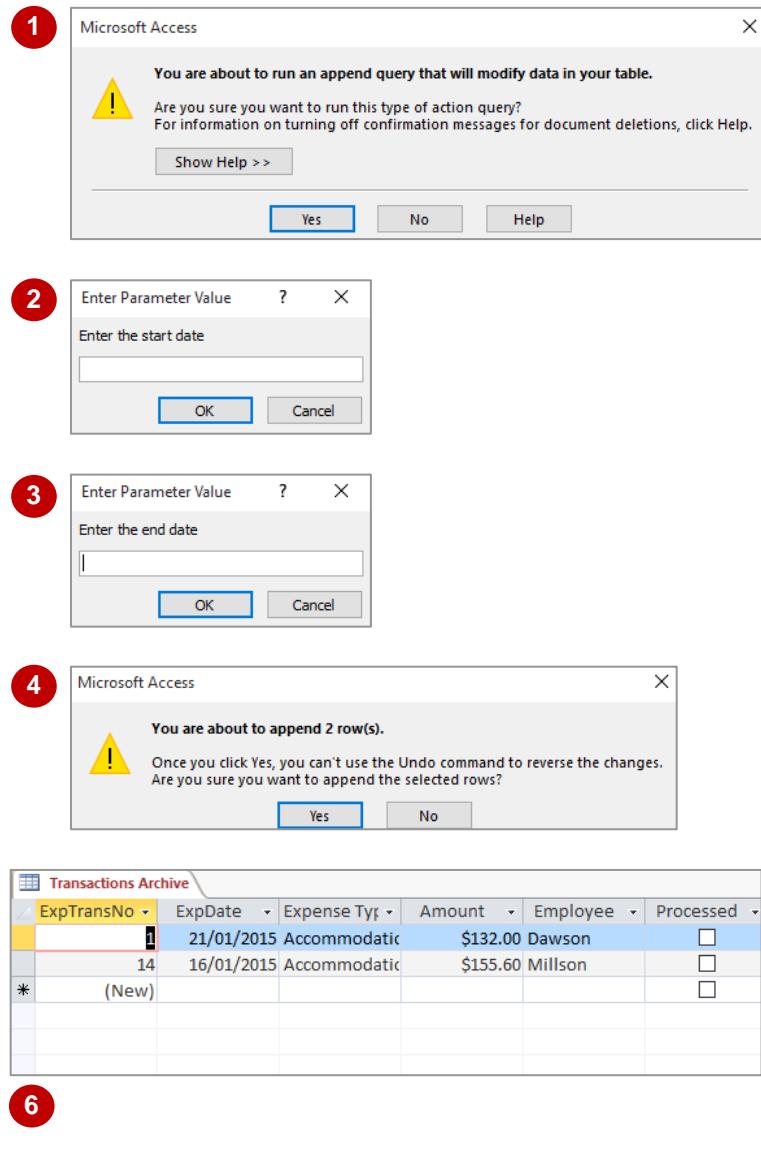
exists otherwise you will receive an error. Also, you can't run the **Append** query more than once because by adding the same records again you will be violating data integrity rules.

## Try This Yourself:

**Same File**

Continue using the previous file with this exercise, or open the file Action Queries\_10.accdb...

- 1 Double-click on the query **Transactions Backup** to display a warning about changing data in the database
- 2 Click on [Yes] to display the prompt for a start date
- 3 Type 1/1/15, then click on [OK] to display the prompt for the end date
- 4 Type 31/1/15, then click on [OK]
- 5 You will now be advised how many records will be appended to the target table...
- 6 Click on [Yes]
- 7 Double-click on the table **Transactions Archive** to display the records
- 8 Close the table



## For Your Reference...

To run an **append query**:

1. Double-click on the query in the **Navigation** pane
2. Follow the prompts

## Handy to Know...

- The target table must exist prior to running the **Append** query. This will most likely be the case because the target table is selected from a list in the **Query Design**. However, take care not to delete or modify the target table before running an **Append** query.

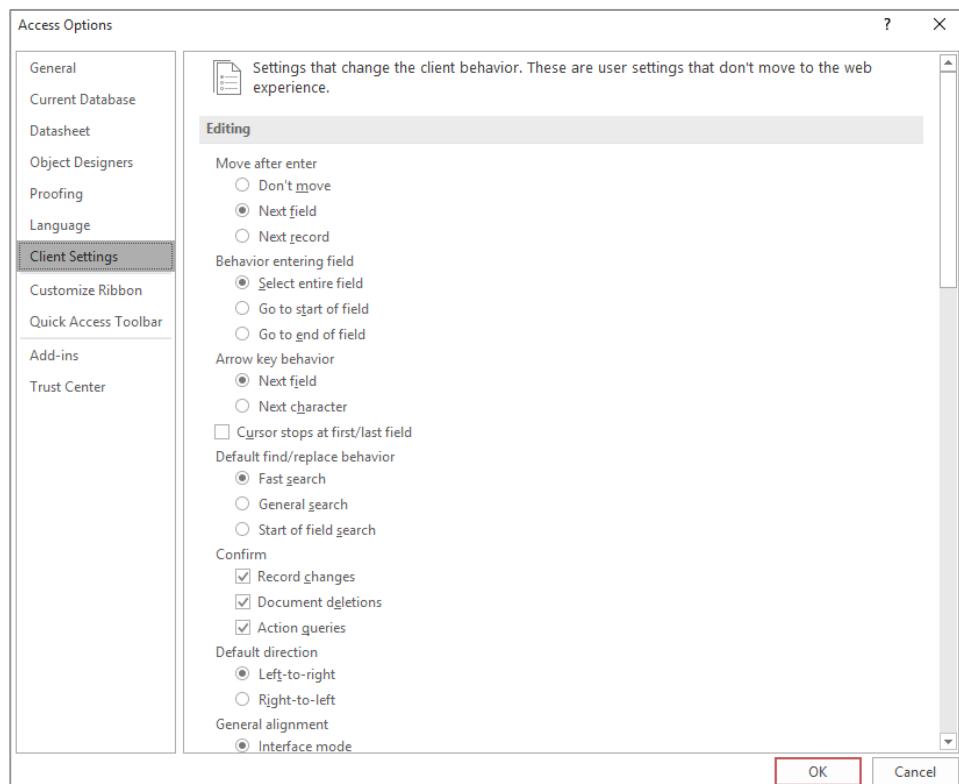
# TURNING ACTION MESSAGES OFF

By default, Microsoft Access displays two messages when performing action-based queries. The first appears when the query is first run and warns that data is about to be modified.

The second appears just before the action is performed and advises how many records will change. These are known as ***action messages*** and can be turned off if desired.

## The Steps to Turn Action Messages Off

Action messages can be turned off using a tick box found within the ***Client Settings*** category in the ***Access Options*** dialog box.



To turn action messages off, the ***Action queries*** option needs to appear without a tick. This is done using the following procedure:

1. Click on the ***File*** tab to display the ***Backstage***
2. Click on ***Options*** to display the ***Access Options*** dialog box
3. Click on the ***Client Settings*** category in the list on the left
4. Scroll down until you can see ***Action queries*** in ***Confirm***, under the ***Editing*** heading
5. Click on the tick box until the option appears without a tick
6. Click on ***[OK]***

Repeating the above steps and placing a tick back into the box will turn the action warnings back on.

**NOTE:** Turning ***action warnings*** on and off is a system-wide setting. This means that the setting is made for all of Microsoft Access irrespective of which database you are currently working on. Switching action messages off in a classroom environment can cause confusion, so that is why we have provided the above steps for reference only.

## NOTES:





## Congratulations!

You have now completed Microsoft Access 2016 - Advanced Queries. Microsoft Access 2016 - Advanced Queries was designed to get you to the point where you can competently perform a variety of operations.

We have tried to build up your skills and knowledge by having you work through specific tasks. The step by step approach will serve as a reference for you when you need to repeat a task.

## Where To From Here?

The following is a little advice about what to do next:

- Spend some time playing with what you have learnt. You should reinforce the skills that you have acquired and use some of the application's commands. This will test just how much of the concepts and features have stuck! Don't try a big task just yet if you can avoid it - small is a good way to start.
- Some aspects of the course may now be a little vague. Go over some of the points that you may be unclear about. Use the examples and exercises in these notes and have another go - these step-by-step notes were designed to help you in the classroom and in the work place!

Here are a few techniques and strategies that we've found handy for learning more about technology:

- read computer magazines - there are often useful articles about specific techniques
- if you have the skills and facilities browse the Internet, specifically the technical pages of the application that you have just learnt
- take an interest in what your work colleagues have done and how they did it - we don't suggest that you plagiarise but you can certainly learn from the techniques of others
- if your software came with a manual (which is rare nowadays) spend a bit of time each day reading a few pages. Then try the techniques out straight away - over a period of time you'll learn a lot this way
- and of course, there are also more courses and books for you to work through.

## Hungry for More?

We live in an ever-changing world where we all need to review and upgrade our skills.

If you have received this course book on a training course why not ask the tutor or trainer for other courses that may be of benefit to you. If you are attending a college ask for one of their brochures.

Alternatively, if you've enjoyed using this course book you can find others that cover a wide range of topics at our web site [www.watsoniapublishing.com](http://www.watsoniapublishing.com).

# INDEX

---

## A

- Access Options ..... 61
  - action messages ..... 61
  - adding tables ..... 18
  - additional queries ..... 11
  - aggregating calculated fields ..... 44
  - aggregation headings ..... 43
  - aggregation queries ..... 41, 45
  - aggregation query ..... 40
  - ampersand character ..... 38
  - AND queries ..... 6
  - Append query ..... 59, 60
  - assembly queries ..... 46
  - assembly query ..... 47, 48
  - AVG ..... 42
- 

## B

- backup tables ..... 50
- 

## C

- calculated fields ..... 32, 33, 44
  - calculating with dates ..... 36
  - clearing criteria ..... 7
  - concatenation ..... 38
  - creating a query design ..... 3
  - criteria ..... 2, 7, 14, 45
  - criteria cell ..... 16
  - criteria in calculations ..... 37
- 

## D

- data mismatches ..... 17
  - Datasheet view ..... 4
  - date fields ..... 36
  - dates ..... 36
  - Delete query ..... 57, 58
  - deleting queries ..... 10
  - dynaset ..... 46
- 

## E

- expressions ..... 29, 32, 33, 37, 52, 55, 56
- 

## F

- field list, queries ..... 14
- filtering a query ..... 16
- filtering records ..... 6
- filtering related fields ..... 17
- formatting expressions ..... 33
- formulas ..... 52

---

## G

- Group By ..... 34, 35, 41, 42
  - grouping ..... 35
- 

## H

- hidden fields ..... 19
- 

## I

- inner join query ..... 21
  - inner joins ..... 20
- 

## J

- join types ..... 20
  - joining string fields ..... 38
  - joins in queries ..... 23
- 

## L

- left outer join ..... 22
  - left outer joins ..... 20
- 

## M

- make table query ..... 51
  - Make Table query ..... 50
  - MAX ..... 42
  - multiple aggregations ..... 42
- 

## N

- nested queries ..... 47
  - nesting queries ..... 46, 48
- 

## O

- outer joins ..... 20
- 

## P

- parameter queries ..... 26, 27, 28, 29, 30
- 

## Q

- QBE ..... 2

- queries ..... 2, 4
  - query criteria ..... 6
  - query design ..... 4, 15
  - Query Design ..... 5
  - query grid ..... 2, 14, 19
  - query joins ..... 23
  - query, Append ..... 59, 60
  - query, Delete ..... 57, 58
  - query, Make Table ..... 50
  - query, nesting ..... 48
  - query, Update ..... 54, 55, 56
- 

## R

- relational queries ..... 14
  - relational query design ..... 15
  - renaming expressions ..... 33
  - right outer join ..... 23
  - right outer joins ..... 20
  - running queries ..... 4, 9
- 

## S

- Save Object As ..... 8
  - saving ..... 8
  - saving queries ..... 8
  - select queries ..... 2
  - Show Table dialog box ..... 3, 18
  - showing all records ..... 27
  - string fields ..... 38
  - Sum ..... 34
  - SUM ..... 40
  - summarising queries ..... 34
  - summary query ..... 35
- 

## T

- Total row ..... 41
  - totalling queries ..... 40
  - Totals command ..... 34
  - typing dates ..... 36
- 

## U

- update queries ..... 52
  - update query ..... 54
  - Update Query ..... 53, 55, 56
  - Update To cell ..... 53
- 

## W

- wildcards ..... 30

