Access 2007
Queries - One
# Contents

Creating a query (on a single table) ................................................................. 3
Adding and removing additional tables ......................................................... 4
Adding fields ....................................................................................................... 5
Sorting data in a query ....................................................................................... 5
Displaying or hiding fields in a query .............................................................. 6
Defining criteria in a query .............................................................................. 6
Using a query ...................................................................................................... 7
Criteria ............................................................................................................... 9
Parameters ........................................................................................................ 10

Queries on Multiple Tables ............................................................................. 12

Calculations in a query ..................................................................................... 13
  Aggregate functions ....................................................................................... 13
  Calculated fields ........................................................................................... 14

Queries with the wizard ................................................................................... 18
  Find Duplicates queries ................................................................................ 18
  Find Unmatched queries .............................................................................. 19

Creating Queries in Access: Tasks ................................................................. 20
  Task 1: simple select queries ...................................................................... 20
  Task 2: using wildcards ............................................................................... 20
  Task 3: Using AND and OR ........................................................................ 21
  Task 4: using date and number criteria ...................................................... 21
  Task 5: null values and excluding data ...................................................... 21
  Task 6: Parameter queries ......................................................................... 22
  Task 7: Grouping and aggregate functions ................................................. 22
  Task 8: Grouping and aggregate functions ................................................. 22
Select Queries

Select queries find specific records matching your criteria and can retrieve data from one or more tables. When you ‘run’ a select query, the query results are displayed as a datasheet that looks and behaves like a table. As with all types of query they can be saved for future use as an object in your database.

This datasheet is called a recordset. Each time it is run, the query interrogates the table according to the criteria you have set and displays the current result of the query. The recordset only exists when the query is run. When you save a query you are saving the criteria and not the recordset; i.e. effectively you are saving the question and not the answer.

Steps to creating and defining a query

The following steps need to be followed to create a select query in Design View using the Query Design window. Details of how to carry out each step are described in the rest of this section.

- Create a new query and add any relevant tables to it.
  See Creating a query.

- Add any fields you want to include in your query.
  See Adding and removing additional tables.

- Define the sort order for your query results.
  See Sorting data in a query.

- Decide which fields to include in or exclude from your recordset.
  See Displaying or hiding fields in a query on page 6.

- Define the criteria on which to select the records to be displayed.
  See Defining criteria in a query.

Creating a query (on a single table)

1. Click on Query Design on the Create tab.
   A new query opens in Design View and the Show Table dialog box appears:

2. Select the table you wish to query and click on Add.
   The table is added to the top half of the Query Design window as shown overleaf.

3. Click on Close.

The Design View window

In Access 2007 the Design View window crops up when designing of all types of database objects: tables, queries, forms, reports, macros, and data access pages. As well as creating new queries by using this window you can also modify the design of existing ones.

The window is split into two sections. The top section contains a field list for each table in the query. The lower section is called the Design Grid – this is where you define your query criteria. By entering criteria expressions in the design grid it is possible to restrict the records in the datasheet to a subset of records.

Each column is used to define the query criteria for a single field.
The grid is made up of a number of rows, each with a special function:

- **Field:** Displays the name of the fields included in the query. Each column represents one field and the cells in the rows below it define the way in which the field will be used in the recordset (see below).
- **Sort:** Displays ascending or descending in any field included in the sort order. The sort order is applied from left to right. If the box is left blank no sorting takes place.
- **Show:** Displays a check mark (tick) if the field is to be displayed in the recordset (i.e. displayed when the query is eventually “run”).
- **Criteria:** Displays criteria to be imposed on that field.
- **Or:** Displays any alternative criteria by which records can be selected.

### Adding and removing additional tables

You can add further tables to your query from the *Show Tables* dialog box used to create the query:

1. Click on the Show Table button on the Design tab.
2. Add tables as described in Creating a query on page 3.
3. To remove unwanted tables:
   1. Click on the Field List of the table in the upper pane of the Design window.
   2. Press the Delete key on the keyboard.

**Helpful hint**

Any fields you have added to the grid below will also be removed.

For information about using multiple tables in queries see page 12.
Adding fields

You will need to include all the fields you require to define criteria and sort the records, as well as any additional fields you would like to display in the recordset.

Single field

1. Click on the required field, hold down the left mouse button and drag the pointer to the first cell in the query grid, or
2. Double-click on the field name to be included in the query, or
3. Click on the drop-down arrow in the field row of the query grid and click on the field you want to include.

Multiple fields

1. Click on the first required field.
2. Hold down the Ctrl key and click on any additional fields you wish to include. Release the Ctrl key.
3. Click anywhere on the selected fields, hold down the left mouse button and drag the pointer to the first cell in the query grid.

Helpful Hint
Double click on the Field List title bar to selects all fields.

Additional fields

To add fields between fields you have already added to the grid, select the required fields as above and then drag them to the column to the right of where you want them to go.

Rearranging and deleting fields

Field columns can be rearranged in the same way as columns in a datasheet.

To remove a field column from your query, select the column by clicking on the column header and then press the Delete key.

Sorting data in a query

Sorting your data allows you to view your data in alphabetical, numerical or chronological order.

1. Click in the Sort row within the column of the field to be sorted.
2. Click on the drop-down arrow and select Ascending or Descending.

More than one field can be sorted in a table. Microsoft Access sorts in order from left to right.

Helpful Hint
Blanks will appear at the beginning in ascending order or at the end in descending order.
Displaying or hiding fields in a query

You can choose whether or not particular fields (columns) are displayed in your query results. It is possible to enter query criteria in a field and choose not to display the field.

For example, if a query finds all doctors in the specialism *Cardiac*, you might choose not to display the Specialism field because you would know that all the records, by definition, would have *Cardiac* in that field.

**To display/hide a field in a query:**

Click in the check box in the *Show* row of a field to add or remove the check mark from the box. Only fields with checked boxes will be displayed in the recordset.

Defining criteria in a query

Criteria are the expressions you enter beneath a field to determine which records are included in the results of the query depending on what is in that particular field.

Defining criteria in a field

To define which records are to be displayed, click in the Criteria row of a field and type in the criterion required.

There are numerous possible criteria that and a list of some of these is displayed under *Criteria* on page 9.

The word **or** can be used within a particular field to allow for more than one possible criterion. For example, *Cardiac or Geriatric* in the Specialism field will display records in the record set which have either the word *Cardiac* or the word *Geriatric* in that field.

Defining criteria in multiple fields

Criteria can be defined in the Criteria row in one or more of your field columns.

If criteria are added to more than one field, when the query is run, it will only display records in the record set which match all the criteria.

Defining criteria in multiple rows

Criteria can also be added to one or more of the field columns in the *Or* row. If the *Or* row is used, when the query is run it will display records in the recordset which match all the criteria in the Criteria row or all the criteria in the *Or* row. Additional *Or* rows can be used to create more complex queries.

**Design grid example one**

<table>
<thead>
<tr>
<th>Field</th>
<th>Table</th>
<th>Sort</th>
<th>Show</th>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>StaffNumber</td>
<td>tbl_doctor</td>
<td>Ascending</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>DoctorName</td>
<td>tbl_doctor</td>
<td></td>
<td></td>
<td>Specialism</td>
</tr>
<tr>
<td>Position</td>
<td>tbl_doctor</td>
<td></td>
<td></td>
<td>DateOfBirth</td>
</tr>
<tr>
<td>Specialism</td>
<td>tbl_doctor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DateOfBirth</td>
<td>tbl_doctor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Salary</td>
<td>tbl_doctor</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The Design Grid shown above will display all doctors who are Registrars and in Cardiac. It will sort them in ascending alphabetical order of DoctorName. Only StaffNumber, DoctorName, Position, Specialism, DateOfBirth and Salary are included in the query and Position will not be displayed.
Design grid example two

The Design Grid shown above will display all doctors born in the 1970s and all doctors not in Cardiac. Doctors who are both not in Cardiac and born in the 1970s will also be displayed.

Using a query

Running a query

To see the results of your query it is necessary to run the query:

- Either click on Run on the Design Tab to run the query
- Or click on View to open the query in Datasheet view

Running a select query selects data from your database so both of these methods will display the query in Datasheet view, i.e. it will show a record set of the matching data.

Helpful Hint

If a query is taking too long to run, you may want to cancel it. Press Ctrl+Break to cancel a query whilst it is running.

Working with query recordsets

Although the data isn’t stored in a query, the data you can see in Datasheet view is the actual data from your tables. If you edit this data you are actually modifying the data in your tables. By default, when you create a query, it will usually be created as a Dynaset which allows data to be edited in this way. However, it is possible to change the properties of the query to make the recordset a Snapshot. This prevents users being able to edit the data in Datasheet view.

To create a Snapshot query:

1. In Design view, click on any blank part of the upper pane.
2. Click on Property Sheet on the Design Tab.

   The Property Sheet will appear (see right)
3. From the recordset Type drop-down list choose Snapshot.
4. Click on the close button at the top right-hand corner of the window.

Helpful Hint

You can Filter, Sort and Find records in a recordset in the same way as you can in tables.
Editing the query

You can toggle back to the Query Design view from the Datasheet view and vice versa using the View icons

- Click on the Design View symbol to switch from Datasheet to Design view.
- Click on the Datasheet View symbol to switch from Design to Datasheet view.

Saving a query

1. Click on Save on the Quick Access Toolbar
   or
   close the Query and you will be prompted to save the query before closing it.
2. Enter a suitable name for the query and click OK.

Helpful Hint
A query cannot have exactly the same name as a table in the same database.

Printing the record set

Given that the query results, or record set, change whenever data in the table or tables it is based on change, you may find it useful to print the record set to preserve a record of it:

1. Open the query in Datasheet view.
2. From the Office Button select Print.
Criteria

Given below are a range of some of the possible criteria that can be used in queries and filters. This is by no means an exhaustive list – use Microsoft Access Help for more information.

Helpful Hint

In certain fields it is only possible to define criteria matching the data type in that field. For example, it is only possible to specify a particular date or range of dates in a date field and to specify a particular number or range of numbers in a number field. Access will warn you if you attempt to enter an inappropriate criterion in a field.

Dates and numbers

To find records matching a specific date or number, or a range of dates or numbers:

<table>
<thead>
<tr>
<th>To find:</th>
<th>Type:</th>
<th>Example</th>
<th>Displays:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact match</td>
<td>[date or number]</td>
<td>14/02/03</td>
<td>14/02/03</td>
</tr>
<tr>
<td>Greater than</td>
<td>&gt;[date or number]</td>
<td>&gt;100</td>
<td>101 and up</td>
</tr>
<tr>
<td>Less than</td>
<td>&lt;[date or number]</td>
<td>&lt;100</td>
<td>up to 99</td>
</tr>
<tr>
<td>Greater than or equal to</td>
<td>[=date or number]</td>
<td>&gt;=14/02/03</td>
<td>after and including 14/10/03</td>
</tr>
<tr>
<td>Less than or equal to</td>
<td>&lt;=[date or number]</td>
<td>&lt;=100</td>
<td>up to and including 100</td>
</tr>
<tr>
<td>Within an inclusive range</td>
<td>between [date or number] and [date or number]</td>
<td>Between 1 and 5</td>
<td>1, 2, 3, 4, 5</td>
</tr>
</tbody>
</table>

Text

To find records which contain a particular word or words, or a combination of letters.

<table>
<thead>
<tr>
<th>To find:</th>
<th>Type:</th>
<th>Example</th>
<th>Displays:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact match</td>
<td>[text]</td>
<td>Cat</td>
<td>Cat</td>
</tr>
<tr>
<td>Text begins with [text] followed by any letters.</td>
<td>[text]*</td>
<td>Ca*</td>
<td>Cat and Canary</td>
</tr>
<tr>
<td>Text occurs anywhere within the Field</td>
<td><em>[text]</em></td>
<td><em>Sales</em></td>
<td>Assistant Sales Agent, Sales Manager etc.</td>
</tr>
</tbody>
</table>

The wildcard asterisk * represents any combination of characters of any length. Replace this with a question mark ? to find a single character for each ? you type.
E.g. B??ch will find Beech or Birch but not Bach.

Null fields

To find records for which a certain field is blank:
Type Is Null in the relevant field.
To find only records for which a certain field contains data (isn’t blank): Type **Is Not Null** in the relevant fields.

**Exclusion**

To find Fields which don’t match certain criteria: Add **Not** or **< >** to the front of the criteria.

This can be used for text, dates or numbers.

**Not *Manager***: finds records for which a field doesn’t contain *Manager* anywhere.

**Not 14/10/03**: finds records for which a field doesn’t exactly match the date 14/10/03.

**Acceptable syntax**

Access automatically places double quote marks “ around text and hash symbols # around dates entered in the criteria line, but numbers are not changed. Access will also add the word Like to any expressions including a wildcard (asterisk * or question mark ?). To save time when creating queries it is more usual to leave out these symbols and allow Access to add them.

To ensure that Access recognises dates correctly, they should always be written in an acceptable format. Access recognises dates in a number of formats, but it is recommended to use dd/mm/yy (e.g. 25/12/03) or dd/mm/yyyy (e.g. 25/12/1795) as standard.

**Parameters**

Parameters, which allow the user to specify criteria each time the query is run, can be added to a query in place of specific criteria. When the query is run, a dialog box appears to prompt the user to type in the criteria. For example, a query which lists all the students visiting a particular city, could have a parameter which prompts for a particular city name each time the query is run. In this way, only one query needs to be created to view student going to any city, rather than one for each city.

To create a parameter:

1. In the **Criteria** row of the field, type the instructions that will prompt the user for what they should enter in the dialog box. Place this text in square brackets. This can be anything except the exact text of the field name of any fields in your query. For example, type the following: **[Enter Exchange City]**
2. Run the query. A dialog box will appear containing the text you typed within square brackets and an empty text box. The example above would appear as right:
3. Type the appropriate text into the box (in this example, type a city name from the Exchange City field) and click **OK**.
4. The results of the query will be displayed.

**Helpful Hint**

Clicking **OK** without entering any text in the box is the equivalent of removing the criterion from the field. In this example, to leave the text box blank means that records will be displayed for all cities.
You can use several parameters in a query, alongside other criteria. They can be placed in the Criteria row or the Or row(s) in the same way as other criteria. Access will display a dialog box for each parameter in turn each time the query is run.

You can also combine other criteria symbols with parameters and use the words or or and to include other criteria (including other parameters) alongside parameters.

Examples

- [Enter Town] or “Exeter”
  This will show the students who live in the Town entered, PLUS all the student who live in Exeter
- >= [Enter the earliest date required]
- Between [First Number] and [Last Number]
- [Enter Town Name] or Is Null
  Will include any Town fields left blank
- not [Enter Subject you wish to exclude]

Using wildcards with parameters

To include wildcards such as the asterisk * in a parameter, type the criterion in the following way:

Like [Parameter] & *
(This will prompt the user for the beginning of the field value)

Or

Like * & [Parameter]
(This will prompt the user for the end of the field value).

Example

To find nurses with surnames beginning with a certain letter, type the following criteria into the Surname Field:

Like [Enter first initial of Student's Surname] & *

Helpful Hint

Access will add quote marks “ around the asterisk.
Queries on Multiple Tables

Up to now we have been applying queries to single tables. However the real power of queries lies in being able to bring together or perform an action on data from more than one table (or even another query). For example, you might want to view a customer's information with the orders the customer placed. To see this information, you need data from the Customers and Orders tables.

When you add more than one table or query to a query, you need to make sure their field lists are joined to each other with a join line so that Microsoft Access knows how to connect the information.

Joins defined in relationships

If you have created relationships between tables in the Relationships window, Access automatically displays join lines when you add related tables in Design view. If Referential Integrity is enforced (an option when defining relationships), Access also displays a “1” above the join line to show which table is on the one side of a one-to-many relationship and an infinity symbol to show which table is on the many side.

Joins not defined in relationships

Even if you haven't created relationships Access automatically creates joins if you add two tables to a query and the tables each have a field with the same or compatible data type and if one of the join fields is a primary key. The one and many symbols are not displayed in this case, because referential integrity is not enforced.

Helpful Hint

By default tables are joined using an Inner Join. This means that only records where there are related records will be shown. For example, in a query based on Exchange City (in the Pupils table) and School Name (in the School table) - where there is a linked field School Number which appears in both tables- only the School records which have a matching record in the Pupils table will be visible ie Records from the Schools table will only show if there is a record in the pupils table (a pupil attending that particular school).
Calculations in a query

There are two types of calculations you can perform using a Select query. You can use the Total row to make calculations across all records or groups of records in your data using aggregate functions or you can add calculated fields to your query which make calculations within each record in your data.

Aggregate functions

Using the Total row in a Select query allows you to use certain built-in functions known as aggregate functions to calculate Totals, Averages, Maximum or Minimum value or to count the number of values in a particular field across all your records or in groups of records.

Adding the Total row to your query:

1. Click on the Totals button on Query Tools Design Tab.
   A new row appears on the Query Grid named Total. The default option Group By appears in all fields.
2. Click on the drop-down arrow in the Total row and select the required option from the list.
3. To remove the Total row from the Query Grid, click the Total button on the toolbar to deselect it.

Total row options

All fields included in your query must have an option selected in the Total row. Depending on your choices, you may not be able to include all fields from all tables.

Group by

This option groups records so that only unique values are shown.

This can be used on any field which has repeated values and only one row in your query results will be displayed for each.

An example of this would be a query based on two tables with a one-to-many relationship. In this situation, any field from the primary table (on the One side of the relationship) can have a Group By value in the Total row.

Aggregate functions

The aggregate functions will operate on each group if grouping has been added using the Group By option, or to all records if no grouping has been added:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Counts all occurrences in a field, including duplicate values (but not empty fields). This option is mainly useful if records have been grouped by a particular field, to count how many records have been grouped together.</td>
</tr>
<tr>
<td>Sum</td>
<td>Find the sum of all the values in that particular field.</td>
</tr>
<tr>
<td>Average</td>
<td>Finds the average of all the values in that particular field.</td>
</tr>
<tr>
<td>Min</td>
<td>Finds the lowest value in the field.</td>
</tr>
<tr>
<td>Max</td>
<td>Finds the highest value in the field</td>
</tr>
</tbody>
</table>

Helpful Hint

When the query is run, the column headings are changed to reflect the aggregate function used.
**Where**

This option can be used to add criteria to a specific field. The field will not be included in the query results. Select the **Where** option and then add criteria in the **Criteria** row.

**Helpful Hint**

To include criteria for a field and also use it in an aggregate function or to use a field in more than one aggregate function, add the field to the query grid more than once. Criteria can be added to a field with **Group By** selected.

**Aggregate query example**

In the example below, a query has been created based on tbl_ward and tbl_nurse. There is a one-to-many relationship between the two tables.

From tbl_ward:
WardNumber, WardName and NumberOfPatients have been added with the **Group By** option selected, so only four rows appear in the query results – one for each ward with nurses in it.

From tbl_nurse:
- **StaffNumber** has a Count function to show the number of nurses in each ward.
- Salary has been added twice, once with a Min and once with a Max function to show the highest and lowest salaries in each ward.
- **Grade** has been added using the **Where** option to include criteria which exclude trainees from the query results. This field is not displayed in the results.

**Calculated fields**

A calculated field can be added to a query to calculate a value based on one or more fields in each record. The calculation is saved in the query but the results of the calculation are not stored and will change depending on the data in the database.

**To create a calculated Field:**

1. Start a new query in **Design** view.
2. Place the fields needed to create the new calculated field on the query grid.
3. In the next column, enter the name for the newly created field.
4. Type in a colon [: ] and then create the calculation. Remember that field names must be enclosed within square brackets.
**Helpful Hint**

You can change the column heading for any field by typing the new heading to the left of the field name in Design view followed by a colon.

**Using the Expression Builder**

The *Expression Builder* is a tool which facilitates the process of creating a calculated field. It is particularly useful for adding elements from your database such as Field Names because it helps to avoid making errors in your spelling or syntax.

1. Click in a blank column of the query grid.
2. Click on the Builder button on the toolbar.
   The Expression Builder dialog box will appear as shown.

The Expression Builder consists of
- an Expression box for creating the expression,
- a set of three panes in the lower part of the builder for adding functions or elements from your database objects to your expression.

For more details, click on the Help button or press F1.

3. Add your expression to the Expression box and click OK to enter it in the query grid column.

**Example**

To create a calculated expression to add 10% to the Salary field from *tbl_doctor*:

1. Click on the + symbol next to *clinic-example.mdb* in the lower left pane of the Expression Builder.
   A list of object types will appear.
2. Click on the + symbol next to **Tables**.
   A full list of Tables in clinic-example.mdb will appear.
3. Click on *tbl_doctor* to select it.
   A list of the fields in *tbl_doctor* will appear in the middle pane.
4. Double click on **Salary** in the middle pane.
   The Salary field ([tbl_doctor][Salary]) will appear in the Expression box.
5. Complete the expression by adding the necessary calculation to the *Salary* field and typing in the field name of this new column to the left of the expression.
   The completed expression should look something like this:
   
   **SalaryIncrease**: `[tbl_doctor]!Salary*1.1`

6. Click **OK**.

**Expression syntax**

As you can see in the above example:

- field Names are represented in expressions in the following format:
  
  `[Table name]![Field name]`

- Mathematical operators use symbols that are common to all Microsoft applications:
  + Add
  - Subtract
  * Multiply
  / Divide

- Brackets can be used in the conventional mathematical way.

There are also specific Functions that can be used in calculations. Some examples are given in the table below.

### Examples of calculations

<table>
<thead>
<tr>
<th>Calculation</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>([tbl_ward]![NoOfPatients]/[tbl_ward]![NoOfBeds])*100</code></td>
<td>Finds the percentage occupancy of each Ward</td>
</tr>
<tr>
<td><code>[tbl_doctor]!Salary + [tbl_doctor]!Bonus</code></td>
<td>Adds the value in the <strong>Salary</strong> field of <strong>tbl_doctor</strong> to the value in the <strong>Bonus</strong> field of <strong>tbl_doctor</strong>.</td>
</tr>
<tr>
<td><code>DateDiff(&quot;yyyy&quot;,[tbl_doctor]!EmployedDate,Date())</code></td>
<td>Find the total amount of time in years a doctor has worked at the clinic to date.</td>
</tr>
<tr>
<td><code>Date()</code></td>
<td>Today’s date</td>
</tr>
</tbody>
</table>

**Calculated fields in total queries**

To include a calculated field in a query with aggregated records (i.e. that includes a *Total row*), select **Expression** from the *Total row* drop-down list for the calculated field. To include a field with an aggregate function in the calculation, you will need to use the column heading used by Access in the query results.

**Example**

A calculated field to find the difference between the highest and lowest salaries as shown in the Aggregate query example on page 14:

**Salary Difference**: `[MaxOfSalary] – [MinOfSalary]`
Formatting fields

Fields with a data type of date, number, currency or yes/no can be formatted to reflect how you wish the data to be displayed.

1. Click on the field you wish to format. Click on the **Property Sheet** button on the **Query Tools Design** tab.
   The Properties window will appear (as on the right).
2. Select the required format from the drop-down list.
Queries with the wizard

Find Duplicates queries

Using the Find Duplicates wizard, you can determine whether a table contains duplicate values in one or more fields. You can then use the results to determine if there are duplicate records in the table, or to determine which records in the table share the same value. For example, you might search for duplicate values in an address field to determine if you have duplicate records for the same patient, or you might search for duplicate values in a Doctor field to see which patients have the same doctor.

1. Click on Query Wizard on the Create tab.
2. In the New Query dialog box, click Find Duplicates Query Wizard, and then click OK.
3. Select the table or query that you want to check for duplicate values and click Next.
4. Select the field(s) that are likely to contain the duplicate information and click the right arrow button to place the field (or fields) in the Duplicate-value Fields box on the right.

Helpful Hint
If you select two or more fields, the query will only find duplicate combinations of all the fields selected i.e. where records have the same values in all the fields.

In the example below the query will look in tbl_nurse and find nurses who are in the same ward and also on the same grade:

5. Click Next.
6. Select any other fields you want to appear in the query result and click the right arrow button to place the field (or fields) in the Additional Query Fields box on the right.
7. If you choose not to show any additional fields, the query results will count the number of instances of each duplicate value and place this number in a NumberOfDups column.

With an additional field (NurseName):

<table>
<thead>
<tr>
<th>WardNumber</th>
<th>Grade</th>
<th>NurseName</th>
</tr>
</thead>
<tbody>
<tr>
<td>w2</td>
<td>Nurse</td>
<td>Zuzupak</td>
</tr>
<tr>
<td>w2</td>
<td>Nurse</td>
<td>Galvin</td>
</tr>
<tr>
<td>w3</td>
<td>Nurse</td>
<td>Shar</td>
</tr>
<tr>
<td>w3</td>
<td>Nurse</td>
<td>Swan</td>
</tr>
<tr>
<td>w1</td>
<td>Trainee</td>
<td>Kawamura</td>
</tr>
<tr>
<td>w1</td>
<td>Trainee</td>
<td>Taylor</td>
</tr>
</tbody>
</table>
Without additional fields:

<table>
<thead>
<tr>
<th>WardNumber Field</th>
<th>Grade Field</th>
<th>NumberOfDups</th>
</tr>
</thead>
<tbody>
<tr>
<td>w1</td>
<td>Trainee</td>
<td>2</td>
</tr>
<tr>
<td>w2</td>
<td>Nurse</td>
<td>2</td>
</tr>
<tr>
<td>w3</td>
<td>Nurse</td>
<td>2</td>
</tr>
</tbody>
</table>

8. Click **Next**.

9. In the last dialog box, you can choose to run the query or see the query's structure in **Design view**.

10. Type in a name for your query.

11. Click **Finish** to display the result of your query.

### Find Unmatched queries

Using the Find Unmatched Query wizard, you can find records in one table that don't have related records in another table. For example, you can find patients that do not have any admissions.

1. Click on **Query Wizard** on the Create tab.

2. In the **New Query** dialog box, click **Find Unmatched Query Wizard**, and then click **OK**.

3. Select the table or query that contains the data that may have no matching records (e.g. *tbl_patients*) and click **Next**.

4. Select the table or query with which you want to compare the first table or query (e.g. *tbl_admissions*) and click **Next**.

5. Select the field that links the two tables and appears in both the related tables and click **Next** (this may already be selected for you):

6. Select the fields from the first table (*tbl_patients*) that you want to appear in the query result and click **Next**.

7. In the last dialog box, you can choose to run the query or see the query's structure in **Design view**. Type in a name for your query and then click **Finish**.

The query results will include all patients with no admissions.
Creating Queries in Access: Tasks

All the following tasks use the database called Exercises.accdb

**Task 1: simple select queries**

1. Create a query based on the tbl_Employees table.
2. Display the following fields: First_Name, Last_Name, Job_Title, Telephone, Nationality, Dept_ID.
3. Select all employees who are American.
4. Sort by Last_Name.
5. Save your query with the name qry_17-1.

**Additional tasks:**

6. Create a second query based on the tbl_Employees table.
7. Display the following fields in this order: First_Name, Last_Name, Job_Title, Telephone.
8. Select all employees who work in the Sales Department (Dept_ID = 100).
9. Do not display the Dept_ID field in the query results.
10. Sort by Job_Title and then by Last_Name, but do not change the order of the fields.
11. Save your query with the name qry_17-2.

**Task 2: using wildcards**

1. Create a query based on the tbl_Employees table. Select all employees whose surnames begin with the letter “W”. Decide for yourself which fields to display. Save your query with the name qry_18-1. How many records are displayed in the query results?

2. Create a query based on the tbl_Employees table. Select all employees with Manager in their job title. Decide for yourself which fields to display. Save your query with the name qry_18-2. How many records are displayed in the query results?
Task 3: Using AND and OR

1. Create a query based on the tbl_Employees table. Show details for all employees who are either managers or who work in the Sales Department (Dept_ID = 100). Save your query with the name qry_19-1.

2. Create a query based on the tbl_Employees table. Show details for all employees who are managers in the Sales Department. Save your query with the name qry_19-2.

3. What type of logical operator does each of the above two queries use? Which of the two queries returns the most records and why?

Additional tasks:

4. Create a query based on the tbl_Employees table. Show details for all employees who are officers in either the Finance Department (Dept_ID = 300) or the Personnel Department (Dept_ID = 500). How many records are displayed in the query results? Save your query with the name qry_19-3.

Task 4: using date and number criteria

1. Create a query based on the tbl_Employees table. Select all employees who were born before 1950. Decide for yourself which fields to display. Save your query with the name qry_20-1.

2. Create a second query based on the tbl_Employees table. Select all employees who were born during the 1960s. Decide for yourself which fields to display. Save your query with the name qry_20-2.

Additional tasks:

3. What criterion did you use in step 2 above? How many different ways can you find of displaying the same results?

4. Create a third query based on the tbl_Employees table. Select all employees who were born in the month of April. Decide for yourself which fields to display. Save your query with the name qry_20-3.

Task 5: null values and excluding data

Create a query which includes all the fields from tbl_Employees. For each of the following tasks, add criteria to find the appropriate employees, view the results in Datasheet view and then modify the Criteria rows to carry out the next task.

1. All employees who do have a disability. (qry_23-1)

2. All employees who do not have a disability. (qry_23-2)

3. All employees who do not have a telephone number. (qry_23-3)

4. All employees who do have a telephone number. (qry_23-4)

5. All employees who are not in the Sales department. (qry_23-5)

6. All employees who are not managers. (qry_23-6)

Additional tasks:

7. All employees who are either officers or assistants. (qry_23-7)

8. All employees who are neither officers nor assistants. (qry_23-8a and b) Which of the two answers given is correct?
Task 6: Parameter queries

1. Create a query based on the `tbl_Employees` and `tbl_Departments` tables. Display the following fields: `Dept_ID`, `Dept_Name`, `First_Name`, `Last_Name`, `Job_Title`, `Telephone`, `Dept_Town`. Enter a parameter criterion in the `Dept_Town` field. Save your query with the name `qry_22-1`. Display results for different towns.

2. Create a query based on the `tbl_Employees` and `tbl_Departments` tables. Decide for yourself which fields to display. Enter a parameter criterion in the `Dept_Name` field. Save your query with the name `qry_22-2`. Display results for different departments.

Additional tasks:

3. Create a query based just on the `tbl_Employees` table. Decide for yourself which fields to display. Enter parameter criteria in the DOB field to find employees born between two dates (you will need two parameters). Save your query with the name `qry_22-3`. Display results for different date ranges.

Task 7: Grouping and aggregate functions

1. Create a query based on the `tbl_Employees` and `tbl_Departments` tables. Display the following fields: `Dept_ID`, `Dept_Name` and `Staff_ID`.

2. Run the query to view the results. Which values are repeated?

3. Add grouping and aggregate functions to the query to enable you to view the number of employees in each department:
   - Add grouping to the fields from `tbl_Departments`
   - Add a Count function to `Staff_ID`

4. Display the results of the query.

5. Save the query as `qry_25-1`.

Task 8: Grouping and aggregate functions

1. Create a query based on `tbl_Employees` and `tbl_Salaries`

2. Select appropriate fields and appropriate grouping and aggregate functions to display the `Salary_Date` when each employees salary was last increased.

3. Add criteria so that only salaries under £20 000 are displayed.

4. Change the column heading of the `Salary_Date` field to ‘Last Salary Increase’.

5. Save the query as `qry_25-2`. 