Using Microsoft Access

Advanced Queries

Parameter Queries

In the last section you created queries with various criteria. If you wanted to keep a query with specific criteria you could save it. For example, if you wanted a query that shows all students who live in Dianella, you could create a query with appropriate criteria and save it with a name such as *students in dianella*. It would get a little complicated, however, if you wanted a query for every suburb, especially if your database contained a dozen different suburbs. Parameter queries are a solution to this problem.

A parameter query allows you to create a query that prompts the user for the criteria when the query is run. You could set up a query so that when it is run, it asks the user to enter a suburb and uses whatever they enter as the criteria. You could use the same query repeatedly without having to modify the design if you want to change the criteria.

Creating a parameter query is like creating a regular select query. The only difference is that where you would normally enter your query criteria, you instead enter square brackets [ ]. This tells access that the user will be prompted for the criteria in this field. If you want to customise the message that is used to prompt the user you can type your own custom message between the square brackets.
Using Microsoft Access

Exercise 1. Creating a Parameter Query

1) Make sure your Student List database is open and make sure you are in the queries section of the Database Window (your test query should be the only query in the query list).

2) Double-click the option which says Create a query in Design view.

3) Double-click, STUDENTS in the Show Table dialog to add it to the query design and then click Close to move to the design window.

4) Adjust your design window as shown in the previous section so that you have plenty of room for selecting fields from your table.

5) Add the following fields to the QBE grid:
   - Last Name
   - First Name
   - Address
   - Suburb
   - Postcode
   - State

6) In the criteria row for the Suburb field, add two square brackets as shown.

7) Click the View icon to view the results of the query. A prompt will appear asking you to enter the criteria for the field.

8) Enter Dianella and click OK. The query results will show students from Dianella.

9) Click the View icon to return to design view.

We will put a custom message in the criteria prompt. Enter a custom message between the square brackets so that it appears as the following. [Enter a suburb]

10) Click the View icon again. This time the prompt will include your custom message.

11) Enter Yokine and click OK (or press Enter). This time you will see all students from Yokine.

12) Return to design view.

13) Click the Save icon. Name the query parameter: students by suburb.

14) Close the query when done.

Note: If you try running a parameter query and get a message about Replication IDs, it is probably because you uses curly brackets {} instead of square brackets [].
Exercise 2. Using a Saved Parameter Query

Your parameter query should now appear in the query list.

1) Double-click the parameter: students by suburb query to run it.

2) Enter Bedford as the suburb and click OK. You will see all of the students who live in Bedford.
3) Close the query. You should be returned to the Database Window.
4) Double-click the parameter: students by suburb query to run it again.

5) Enter Morley as the suburb and click OK. You will see all of the students who live in Morley.

You can run the query with any suburb being used as the criteria without changing the design of the query.
Exercise 3. Creating a Parameter Query with Multiple Criteria

In regular select queries, you can have criteria on as many fields as you like. This applies for Parameter queries as well. If you have more than one parameter criteria in a query then a prompt will appear for each criterion, one after the other before the query results appear. You can also use parameters for range criteria as the following exercise demonstrates. We will create a query that allows the user to enter two amount, and see all students with marks between those amounts.

1) Create a new query that uses the following fields:
   - Last Name
   - First Name
   - Gender
   - Mark
   - Comment

2) For the Mark field, enter the following for the criteria
   \( \geq [\text{Enter a minimum mark}] \text{ and } \leq [\text{Enter a maximum mark}] \)

   **Tip** When you are working with long criteria like this, it can be difficult to edit because of the limited space. To make it easier, right-click on the criteria and click **Zoom**. You will then have a nice large window to edit your criteria. When you are done you can click **OK** to confirm the changes.

3) Click the **View** icon \( \text{View} \) to view the results of the query. A prompt will appear asking you to enter a minimum mark.

4) Enter 50 and click **OK**. A second prompt will appear asking you to enter a maximum mark.

5) Enter 70 and click **OK**. The results will show all students with marks from 50 to 70.

6) Click the **View** icon \( \text{View} \) to return to design view.

7) Save the query as **parameter: students marks between two amounts**.

8) Close the query when it is saved.
Exercise 4. Using Wildcards with Parameters

1) Open the parameter: students by suburb query.
2) When the parameter prompt appears, enter Dian* for the criteria.
   If this criterion were used in a query design, it would result in all students where the suburb name begins with dian. This won’t work with parameters though and you will get no results at all. For parameters, the wildcard needs to be built in to the query design along with the parameter.
3) Close the query to return to the Database Window.
4) Create a new query using the following fields.
   - Last Name
   - First Name
   - Date of Birth
   - Phone
   - Gender
   - Mark
5) For the Last Name field, enter the following for the criteria.
   Like [Enter the last name] & *
   The & symbol will join the user’s typed criteria on to a * wildcard. This will mean that the user will only need to enter the first few letters of a last name to get the results.
6) Click the View icon to view the results of the query.

7) When the prompt appears enter S for the criteria and click OK. You will see all students with a last name beginning with the letter S. This makes the query easier for the user.
8) Click the View icon to return to design view.
9) Save the query as parameter: students by last name.
10) Close the query when it is saved.

Some variations on the above method are shown below.

<table>
<thead>
<tr>
<th>Field</th>
<th>Criteria</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Name</td>
<td>Like * &amp; [Last Name] &amp; *</td>
<td>Adds a wildcard before and after what the user types</td>
</tr>
<tr>
<td>Phone</td>
<td>Like * &amp; [Phone number]</td>
<td>Phone numbers ending with the numbers the user types</td>
</tr>
</tbody>
</table>

11) Create queries to test each of the two examples shown.
Calculated Fields

Queries can include additional fields which automatically calculate a result based on existing fields. The calculations in these fields are similar to the calculations in Excel formulae. In the following exercise we will create a basic calculated field to work out a student’s age from their Date of Birth. The following guidelines apply to calculated fields.

- Like Excel, calculations in Access follow the order of operations rules. I.e.
  - Brackets calculated first
  - Indices (^) calculated next E.g. $4^2$ means 4 to the power of 2
  - Multiplication and division calculated next
  - Addition and subtraction calculated last.

- When you are referring to a field in your calculation, the field name must be enclosed in square brackets. E.g. [Mark].

- You can’t combine different data types in a calculation. E.g. You can’t add a number field to a text field.

Exercise 5. Creating a Calculated Field

1) Create a new query with the following fields:
   - Last Name
   - First Name
   - Date of Birth

2) Click in the blank field after Date of Birth as shown.

   The calculation is typed where the field name would normally go. In an Excel calculation, you would begin with an equal sign. In a calculated field, you begin with a name for the calculated field followed by a colon. This calculated field is to calculate a student’s age.

3) Enter the following in the Field row (you may like to use the Zoom option shown previously when you are editing the calculation).
   Age: (Date() - [Date of Birth])/365

4) This calculation will subtract their date of birth from the current date date() to calculate their age in days. The result is then divided by 365 to get the age in years.

5) Click the View icon to view the results of the query.

The calculation will appear as an additional field, with the result appearing for each student.

6) Click the View icon to return to design view.

7) Save the query as student listing with age.
Exercise 6. Formatting a Calculated Field

In the previous exercise, the resulting calculated field contained ages with a large number of decimal places. You can customise fields in a query by editing properties that are very similar to the ones found in table design.

1) Make sure you still have your student listing with age query open in design view.
2) Right-click on the Age field and choose Properties (you can also get to properties from the View menu, by clicking the properties icon or by pressing [Alt][Enter]).

The Properties window will show properties for the age calculated field (you can also edit properties for any regular field in the same way).

3) Click in the Format property and change the format to Fixed.
4) Click in the Decimal Places property and enter 1.

5) Close the Properties window.
6) View the results of the query again. The ages will now be formatted with one decimal place.
7) Save and close the query.
Exercise 7. Using a Function in a Query

Functions in Access are similar to functions in Excel and can be used to simplify complex calculations. The following example demonstrates the use of the Immediate If function which is similar to Excel’s If function.

1) Create a new query with the following fields:
   - Last Name
   - First Name
   - Mark

2) In the next blank field, enter the following.
   \[ \text{Pass: IIf([Mark]\geq 50, "Pass", "Fail")} \]
   
   The first part will be the name of the new field (Pass).
   This is followed by the IIF. Like all functions, its components are enclosed in brackets. This type of function has three parts separated by commas.
   The first part specifies the criteria. In this case \([Mark]\geq 50\) meaning are the contents of the Mark field greater than or equal to 50.
   The second part specifies what the answer will be when the condition is true (display the text “Pass”). When test is being referred to in a calculation it always need to be enclosed in “”.
   The third part specifies what the answer will be when the condition is false (display the text “fail”).

3) Click the View icon to view the results of the query.

Any student with a mark less than 50 will have \text{Fail} in the \text{Pass} field.

4) Click on the mark for Laura Davies as shown above.

5) Change the mark to 48. Ass soon as you move on to a different record, the \text{Pass} column will update.

6) Make sure you are still in the \text{Mark} column and click the Sort Descending icon so that the highest marks are at the top. Changes to the sort order will be saved as part of the query design.

7) Save the query as \text{student marks} and close the query when done.

Note: when you are viewing the results of a query you are actually viewing a selection from the table. When you change data in a query’s datasheet view those same changes are being made to the table data. Any formula depending on that data will automatically update.
Summary Queries

Unlike other queries you have done, a summary query won’t show individual records. It will only show a summary in the form of averages, totals and other available calculations. Summary Queries are useful for reporting where you are only interested in a summary of the information without the detail.

Exercise 8. Creating a Query with a Total

1) Create a new query with only the Mark field.
2) To change to a Summary Query, select Totals from the View menu or click the Σ icon on the toolbar.
3) Your QBE grid will now include a Total row.
4) Change the Total figure for the Mark field from Group By to Sum.
5) View the results of the query. Because of this change the query result will now show the Sum of the Mark field instead of showing each record.

6) Return to Design View.
7) Add the Mark field to the QBE grid three more times. Change the total row for each one to Avg, Min and Max as shown below.

8) View the results of the query. The results will now show the Sum of the Mark field, the Average of the mark field, the Minimum amount in the Mark field and the Maximum amount in the Mark field.

9) Return to Design View.
10) Save the query as summary: student marks.
Exercise 9. Using Grouping in a Summary Query

When using Summary Queries it is common to group the results by a particular field instead of viewing the totals for the entire query.

1) Drag the Gender field on to the first Mark field. This will insert the Gender field to the left of the first Mark field.

2) Leave the Total row for the Gender field as Group By.

3) View the results of the Query. Now you will see totals for the Female students and totals for the Male students.

4) Return to Design View.

5) Add the Suburb field as shown below.

6) View the results of the query. The results will now show the totals grouped by Gender and grouped by Suburb.

7) Save and close the query.