# TABLE OF CONTENTS

**QUERYING TABLES**
- BUILDING A QUERY IN DESIGN VIEW ................................................................. 4
- ADDING CRITERIA ................................................................................................. 6
- REARRANGING FIELDS IN THE QUERY .............................................................. 8
- ADDING FIELDS TO THE QUERY ........................................................................ 8
- REMOVING FIELDS FROM THE QUERY ............................................................. 9
- CHANGING COLUMN WIDTHS ............................................................................. 9
- SORTING THE QUERY ......................................................................................... 10
- SORTING ON MULTIPLE FIELDS ....................................................................... 10

**SAVING/OFFING A QUERY** ........................................................................... 11
- OPENING A QUERY ............................................................................................ 12

**USING WILDCARDS IN QUERIES** ................................................................ 13

**COMPOUND QUERIES** .................................................................................. 14
- CREATING AN "AND" QUERY ........................................................................... 14
- CREATING AN "OR" QUERY ............................................................................. 15

**QUERYING MULTIPLE TABLES** ................................................................. 17
- ADDING MULTIPLE TABLES TO A QUERY ......................................................... 18
- JOINING TABLES .................................................................................................. 19
- DELETING A JOIN ................................................................................................ 20

**CREATING CALCULATED FIELDS** ............................................................... 21
- USING SUBSTRINGS ........................................................................................... 22
- USING DATE SUBSTRINGS ............................................................................... 23

**USING SUMMARY OPERATORS** ................................................................. 24
- ADDING THE TOTAL CELL ................................................................................. 24
- TOTALING ALL RECORDS .................................................................................. 25
- TOTALS BY GROUPS ........................................................................................... 25

**PARAMETER QUERIES** ............................................................................... 28
- CREATING MULTIPLE PARAMETER PROMPTS ............................................... 29

**USING ACTION QUERIES** ......................................................................... 30
- CREATING AN UPDATE QUERY ....................................................................... 30
- MAKING A NEW TABLE ....................................................................................... 31
- CREATING A DELETE QUERY ......................................................................... 34
- CREATING AN APPEND QUERY ....................................................................... 35
QUERYING TABLES

Usage:

Database programs in general are used to enter huge amounts of information. Once the information has been entered, however, the program must have the capacity to narrow listings down based on criteria that you, the user, provide. Access refers to the process of questioning, selecting and manipulating data as Querying. For example, if you have a table consisting of thousands of company employees you may at some point want to view only those employees working in a given department. The Query would then consist of Access pulling out all records where the department is equal to the one you specify.

It is also possible to have Access select only specific fields to view. Using the example above, perhaps your table consists of fifteen fields but you are only interested in seeing the employee names and phone number. You can instruct Access to list only those two fields for all records meeting the department criteria.

You must create New queries and then select any/all tables and field(s) you wish included in your query listing. You will then use the Query Form to set your criteria.

Building a Query in Design View

Although the wizard can be useful for creating the basic design of a simple query, the query created by the wizard contains no selection criteria and simply displays a list of the entire table. You will need to modify the query design to include the criteria for selecting specific records.

If you are viewing the database query, click on . From the database window, click on the Design button.

Access will display all of the fields for the selected table in the top portion of the screen but none of the actual data will be seen, as illustrated in the diagram below:
Introduction to Access Queries

The top half of the query window contains a list of available tables and their associated fields. The bottom half of the query window lists each field that has been included in the query in a columnar format. If you need to add a field to the query, simply drag the field name (from the top half of the window) to a blank column (in the bottom half of the window).

The asterisk (*) can be used to indicate that all fields of a table should be included in the query.
The following sections are shown in the bottom half of the screen:

**Field**  
Used to enter the field name you wish to query. To add a field, click in this section and choose from the pull-down list. You can also drag the field name using the mouse from the table list.

To create a query that contains all of the fields, double-click on the title bar of the table. This highlights all fields, which you can then drag to this section. Access will place each field in a separate column. The asterisk (*) includes all current and future fields in the query. Use this to ensure you always have all the latest fields when the query is run.

**Table**  
If there is more than one table, use this section to specify the table from which the field will be derived.

**Sort**  
Use this to sort in ascending or descending order.

**Show**  
Check this box to actually show the data of the field in the final list. Fields can be used for criteria but do not have to be shown in the resulting dynaset.

**Criteria**  
This is where you actually type in what you are searching for. Be sure to place your criteria in the column of the field that would contain the data.

**Or**  
If you are looking for more than one item within the same column, place your criteria on multiple lines.

### ADDING CRITERIA

Once you have added the fields to view/search, your next step is to specify what condition(s) must be met for records to be displayed.

Move to the field where the condition is to be set and enter the criteria. The following operators may be used during a query:

- **=** Must be exactly equal to
- **<** Must be less than
- **>** Must be greater than
- **>=** Greater than or equal to
- **<=** Must be less than or equal to
Introduction to Access Queries

Example: If you were searching for those employees making more than $40,000.00 you would move to the salary field and enter the following condition:

\[ >40000 \]

Once you have entered all your criteria:

Click on this button to actually process the query.

The results of the query are stored in a recordset. A recordset looks similar to a table, but is not a real database table. However, any changes you make to the data in a recordset affect the actual tables from which the records were selected.

Recordsets can be saved and/or printed the same way as tables.

If you need to make a change to the query, click on the button to return to the query design and change your criteria.
REARRANGING FIELDS IN THE QUERY

Once you begin working with your queries, you may find that you want to rearrange the order in which the fields are displayed.

You can move fields by **dragging** the field column, as illustrated in the diagram below:

To move a field, click **once** on the column selector (the small gray bar at the top of the field column) to select the column.

Next, click on the column selector again and **drag** the column to the new location. As you drag the column left/right, a black vertical line will show up between the columns, indicating where the field will be placed once you let go of the mouse button.

ADDING FIELDS TO THE QUERY

If you want to quickly add a field to the query, simply **double-click** on the field name from the list. Access will add the field as the last column in the query. You could also move to the next blank column and click in the **Field** row. Next, click on the down arrow to select your field from the pull-down list.

If you want to place a new field in a specific column location other than at the end, you will need to insert the field. To insert a field into your query, click on the field name you want and drag the field name to the column where you want the field to be displayed. Access will place the new field in that location and shift the existing field to the right to make room.
REMOVING FIELDS FROM THE QUERY

Deleting a field column is easy - click on the column selector of the field that you want to remove from the query and press =. You will not be asked to confirm the deletion.

CHANGING COLUMN WIDTHS

If you need more room in a column to show all of your criteria or other information, you will need to adjust the width of the column.

NOTE: Changing the width of a field column while in the design mode does not adjust the width of the field when you view the data table.

To change the width of a field column, move the mouse pointer to the right border of the column you want to adjust. The mouse pointer must be on the right side of the gray column selector.

The mouse pointer will change to a two-way arrow, as shown in the diagram below:

Once the pointer has changed to a two-sided arrow, click and drag the column border left or right as needed.

TIP: If you double-click when the pointer changes to the two-way arrow, Access will automatically adjust the width to fit the widest entry in the column.
SORTING THE QUERY

Viewing the data in the resulting recordset will usually be easier if you have it sorted by at least one of the fields.

You can sort in **ascending** (alphabetically A-Z or numerical order 0-9) or **descending** (Z-A or 9-0) order.

Sorted queries are also useful when used with forms or reports. To sort the query, click in the **Sort** row under the column containing the field you want to sort. A small down arrow will appear to the right of the field.

Click on the down arrow to access a pull-down menu, as shown in the diagram below:

```
<table>
<thead>
<tr>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ascending</td>
</tr>
<tr>
<td>Descending</td>
</tr>
<tr>
<td>(not sorted)</td>
</tr>
</tbody>
</table>
```

Select **Ascending** or **Descending** as needed. If you no longer wish to sort on that field, choose **(not sorted)**.

You may also type the letter **A** (for ascending) or **D** (for descending) into the sort row instead of using the pull-down menu.

SORTING ON MULTIPLE FIELDS

Although you can sort on more than one field, you do need to take a couple of things into consideration. First, Access sorts from left to right. In other words, if you select more than one field for sorting, Access will sort on the leftmost field you selected first, followed by the next selected sort field to the right.

If you have the fields **City** and **State** in your query (in that order) and decide to sort on both the city and the state, Access will sort by the city first and duplicate cities will then be sorted by state.

If you want to sort by state first, then by city, you will have to move the state field to the left of the city field in the query. Make sure your fields are arranged from left to right in the order in which they should be sorted.
SAVING/OPENING A QUERY

USAGE:

As you know by now, querying a table or multiple tables may be complex and requires careful attention to detail as Access is sensitive to any and all errors. If you must perform the same type of query periodically, it becomes tedious to set up those same queries over and over again. Instead, you may wish to save the query for future use. Once a query has been saved, you can "open" the query and use it whenever you need to list those same records again.

⚠️ Click on the close button to close the query.

If haven’t saved the query, the following warning will appear:

![Warning dialog]

Click on **Yes** to save the query.

**NOTE:** If you had created the query manually (without the wizard), you will be asked to assign a name to it.
OPENING A QUERY

Once you have saved your queries, you can use them repeatedly as needed without having to recreate the conditions of the queries.

To use a previously saved query, you must **open** the query - which is done the same way you open tables - from the database window.

Click on the **Queries** icon to list the queries.

To open a query, simply double-click on its name or highlight it and click on the **Open** button to view the query results.

If you want to make changes to an existing query, click on **Design** to enter the design mode.
USING WILDCARDS IN QUERIES

USAGE:

There are many times when you cannot look for something based on exact spelling (because you are not sure) or you would like to look for a group of records that match a particular pattern. Access provides a method by which you can search for a "pattern" of characters using the asterisk (*) as a "wildcard" symbol. This is especially useful when you are unsure of how to spell a name.

Examples:

This pattern... ...would find:
S* Smith, sugar
g*t Giant, gross weight
*D Grand, Elm Road
*e*s Phillip Edward Willis, roses
7*5 7485, 70,005
6/*/01 All dates in June of 2001.

In addition to the asterisk, there are a few other wildcard characters that can be used for a query, as listed below:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>b?ll</td>
<td>finds ball, bell, and bill</td>
</tr>
<tr>
<td>#</td>
<td>1#3</td>
<td>finds 103, 113, 123</td>
</tr>
<tr>
<td>[ ]</td>
<td>b[ui]ll</td>
<td>finds bill, bull not ball</td>
</tr>
<tr>
<td>!</td>
<td>b![ui]ll</td>
<td>finds bell, ball not bull</td>
</tr>
<tr>
<td>-</td>
<td>b[a-c]d</td>
<td>finds bad, bbd, and bcd</td>
</tr>
</tbody>
</table>
COMPOUND QUERIES

USAGE:

Access allows you to combine your criteria to form more complex queries. These combinations can be used to narrow the search to the exact information you need. They can also be used to make your searches more flexible and include more options than a simple query.

CREATING AN "AND" QUERY

If you want more than one condition in different fields to be true, you may enter conditions in more than one field on the same row.

This specifies that all the conditions on that row must be true at the same time for a record to be selected and placed in the recordset.

To create an "AND" condition, you enter the criteria on one line but within different fields. For example, perhaps you want to find all hotels costing more than $500.00 in the city of San Francisco, as illustrated in the diagram shown below:

<table>
<thead>
<tr>
<th>Field: Hotel</th>
<th>City</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table: Hotels</td>
<td>Hotels</td>
<td>Hotels</td>
</tr>
<tr>
<td>Sort:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Criteria: or:</td>
<td>&quot;San Francisco&quot;</td>
<td>&gt;500</td>
</tr>
</tbody>
</table>

This would result in Access listing all records where the cost is greater than 500 and the city matches San Francisco.

Because the two conditions are located on the same criteria line, Access assumes both conditions must be met. When both must be met, it is considered to be an AND query!
If you want to look for a range of values within the same field, use the **And** operator command. Since you need to specify two criteria for the same field, this is entered a bit differently. For example, to display records for only those employees who make between $30,000 and $40,000 per year, move to the salary field and enter:

```
>30000 And <40000
```

In the scenario above, only records consisting of employees making more than $30,000 but less than $40,000 would be extracted. This would not include anybody making exactly 30,000 or exactly 40,000. To include those values, use `>=` and `<=`.

### CREATING AN "OR" QUERY

This type of query is performed when you want to locate records meeting **either** one condition or another. It allows you to specify several different options at once. If a record satisfies **any** one of these conditions, the record will be selected.

To create multiple conditions, place each condition on a separate line. Each line specifies a different condition. You can mix AND/OR conditions as much as you need.

If you want multiple choices from **within the same** field, the command **OR** is used. The **OR** tells Access that only one of the specified criteria needs to be met in order for a record to be extracted. For example, to find employees who work in the Admin or Accounting departments move to the "Dept" field and enter:

```
"Admin" Or "Accounting"
```

This would result in Access listing employees working in either the Admin or Accounting departments.
If you want to extract records based on an OR existing between different fields, you must use two lines for the query.

For example, if you are searching for employees working in the Admin department or those employees making more than $30,000 regardless of their department you would enter the criteria as shown below:

If both conditions are on the same row, Access assumes you want both conditions to be true. In other words, this is considered an AND statement whereas two separate lines are considered an OR statement.

You can be as selective as you want by using several lines to include several OR conditions.
QUERYING MULTIPLE TABLES

USAGE:

There may be times when you need to query more than one table in order to get the desired data. Since data can be spread across multiple tables, you may need to "combine" several tables to gain access to all of the fields you need for a particular report.

The tables need to have a connection/relationship between them so that you can "link" the data in all the tables together. For each two tables linked, there must be a common field between them.

If you query five tables at once, each table must be linked to another -- not all tables need to have the same field in common. For example, the first and the second table may have one field in common while the second and third have a different field in common which allows them to be linked. A simple example would be the following: To create an invoice for a client you would require the name and address for the client and the order information. The client's name and address might be stored in one table while the order is stored in another. You would need to connect the two tables together so that the order could be matched to the client.

To relate (connect) the two tables, you would need a field matched to the orders of each client. This field (e.g., client ID#) would be the "common" field between the two tables.

If you also needed the description and sell price, you would have to access the inventory table. In this case, you might "relate" the "item#" from the orders table to the "item#" of the inventory table. The result is a multiple table query with the three tables related to each other in a "daisy-chain" format, as shown below:

You could then create a query that contains the customer information, the order data, and the product details for each order.
ADDING MULTIPLE TABLES TO A QUERY

If you create a new query, you can simply add all of the tables from the start. However, if you are working with an existing query, you will need to add the additional tables, as shown below:

Click on this button located on the toolbar.

The following dialog box will be displayed:

Highlight the name of the table you need and click on Add.

As you add a new table to the query, Access will attempt to establish a relationship between the new table and the existing tables in the query. If an identical field name is found in both tables, Access assumes the tables are related and creates a link between the two common fields.

Continue adding all the tables and then click on Close.
JOINING TABLES

Once you have added all of the tables you require, you will need to connect the tables based on their common fields.

As mentioned, Access will have attempted to relate the tables, however, you should always verify the links between tables in case Access has created an incorrect link.

NOTE: The common fields do not have to have the same name, but they do need to be of the same type. For example, you cannot link a text field with a numeric field. Also, the field from the table that you are linking from needs to be a primary key.

Simply click on the field of the table you are linking from and drag the field on top of the matching field in the table you are linking to, as illustrated in the diagram below:

Once you let go of the mouse, Access will display the link with a line between the two common fields. Continue linking all tables required for your query.
Once the tables have been linked, add the fields (from any table) you would like to see in the query result table.

By combining fields from different tables, you are essentially creating a new table using information from multiple tables. This allows you to create more detailed reports.

You do not have to include the common fields in your query and you do not have to include all of the fields from all of your tables. The rest of the query preparation works the same as when creating a query from a single table.

**DELETING A JOIN**
To delete a link between tables, click on the join line and press the Delete key on your keyboard.
INTRODUCTION TO ACCESS QUERIES

CREATING CALCULATED FIELDS

USAGE:

In addition to the normal querying, Access allows you to create mathematical calculations based on numeric fields. For example, if you have an existing field containing car prices, you may want to see what a 5% increase would look like before actually editing each record and permanently changing its contents.

The following mathematical operators can be used:

+ Addition
- Subtraction
* Multiplication
/ Division

You can create calculations on numeric, $ and date fields.

You place your mathematical expression in the "Field" cell.

Also, if you are using other fields in the calculation, you must place square brackets [ ] around the field name of the other field so that Access knows where to get the data from, as shown below:

This would result in Access taking what is currently in the Price field and multiplying it by 5%. The result is placed in a newly created field, as illustrated in the diagram shown below:

<table>
<thead>
<tr>
<th>Stock No</th>
<th>Description</th>
<th>Qty</th>
<th>Price</th>
<th>Expr1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Chia Pet</td>
<td>12</td>
<td>$35.00</td>
<td>1.75</td>
</tr>
<tr>
<td>2</td>
<td>Pet Rock</td>
<td>10</td>
<td>$27.50</td>
<td>1.375</td>
</tr>
<tr>
<td>3</td>
<td>Ginsu Knife Set</td>
<td>22</td>
<td>$19.99</td>
<td>0.9995</td>
</tr>
<tr>
<td>4</td>
<td>PEZ Dispenser</td>
<td>5</td>
<td>$5.75</td>
<td>0.2875</td>
</tr>
<tr>
<td>5</td>
<td>Lava Lamp</td>
<td>14</td>
<td>$41.00</td>
<td>2.05</td>
</tr>
<tr>
<td>6</td>
<td>Chop Sticks</td>
<td>25</td>
<td>$35.00</td>
<td>1.75</td>
</tr>
</tbody>
</table>
USING SUBSTRINGS

In some cases, you may want to display only part of a field. For example, on a phone number, you may wish to only show the area code for a particular report. As long as the piece of text (substring) you are looking for is always in the same spot in the field, you can use one of the following functions to extract the piece of text that you want to display:

- **Left([[fieldname]],#)**
  - where `[fieldname]` is the name of the field containing the piece of text to be displayed and `#` is the number of characters to be displayed from the left edge (beginning) of the field.

- **Right([[fieldname]],#)**
  - where `[fieldname]` is the name of the field containing the piece of text to be displayed and `#` is the number of characters that should be displayed from the right edge (end) of the field.

- **Mid([[fieldname]],start,#)**
  - where `[fieldname]` is the name of the field containing the piece of text to be displayed, `start` is the position of the first character to display and `#` is the number of characters that should be displayed from the start.

**NOTE:** These functions work only on text fields.

**Example:** Assume you have a field called Phone that contains entries similar to (949)249-8081.

<table>
<thead>
<tr>
<th>Function</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left([Phone],5)</td>
<td>(949)</td>
</tr>
<tr>
<td>Right([Phone],4)</td>
<td>8081</td>
</tr>
<tr>
<td>Mid([Phone],6,3)</td>
<td>249</td>
</tr>
</tbody>
</table>
USING DATE SUBSTRINGS

There is a function that can pull out parts of a date similar to the function that pulls out parts of a text field. The following function will extract such items as the four-digit year, weekday, quarter and other such date parts.

\[ \text{DatePart("datepart",[fieldname])} \]

where \textit{datepart} is the part of the date you want to pull out and \textit{fieldname} is the name of the field containing the date(s).

Assume you have a field in a record called \textit{Order Date} containing the date 10/24/01. The following examples show the codes that can be used for the \textit{datepart} section.

<table>
<thead>
<tr>
<th>Code</th>
<th>Represents</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>Day</td>
<td>24</td>
</tr>
<tr>
<td>m</td>
<td>Month</td>
<td>10</td>
</tr>
<tr>
<td>q</td>
<td>Quarter</td>
<td>4</td>
</tr>
<tr>
<td>w</td>
<td>Weekday</td>
<td>7 where 1= Sunday 2 = Monday 3 = Tuesday 4 = Wednesday 5 = Thursday 6 = Friday 7 = Saturday</td>
</tr>
<tr>
<td>ww</td>
<td>Week of the year</td>
<td>43</td>
</tr>
<tr>
<td>y</td>
<td>Day of the year</td>
<td>297</td>
</tr>
<tr>
<td>yyyy</td>
<td>Four-digit year</td>
<td>2001</td>
</tr>
</tbody>
</table>

\textit{Examples:}  
\begin{align*}  
\text{DatePart("q",[OrderDate])} & \text{ returns the value “4”} \\
\text{DatePart("yyyy",[OrderDate])} & \text{ returns the value “2001”} \\
\text{DatePart("mmmm",[OrderDate])} & \text{ returns the value “October”} \\
\end{align*}

In addition, if you have a field that contains the time, you can use the following functions:

<table>
<thead>
<tr>
<th>Code</th>
<th>Represents</th>
</tr>
</thead>
<tbody>
<tr>
<td>h</td>
<td>Hour</td>
</tr>
<tr>
<td>n</td>
<td>Minute</td>
</tr>
<tr>
<td>s</td>
<td>Second</td>
</tr>
</tbody>
</table>
USING SUMMARY OPERATORS

Usage:

Access has some functions that help make certain types of calculations easier to perform without using formulas. This also allows you to break the calculations down by groups.

Since these are calculations, you can perform them only on numeric, currency or date fields. The count function is an exception since it is used to count the number of records.

The following are some of the main functions that may be used when creating group calculations:

- **SUM**: Calculates the total of all values within a field.
- **AVG**: Calculates the average value within a field.
- **MAX**: Calculates the highest value within a field.
- **MIN**: Calculates the lowest value within a field.
- **COUNT**: Calculates the number of occurrences of a field.

Adding the Total Cell

To total a field, click on this button (located on the toolbar) while working in the query design mode.

The bottom section of the query will expand to include another cell for specifying which fields should be used for groups and which fields will be used for creating total calculations.

When creating totals in a query, you should only include the field(s) on which you wish to group and the field(s) you are calculating on since Access will use each field for either grouping or totaling your records.

When you click in the Total section of a field, you see a variety of functions that can be performed.
The first step is to drag the fields you want to total to the query and choose the type of calculation you want to perform on the numeric fields you included in the query.

TOTALING ALL RECORDS

To total all records in a table, limit your query to only those fields you wish to total. Delete all other fields from the query form. Choose the type of function (sum, avg) for the selected field.

The example shown above would count the total number of records based on the Product ID and give the average unit price for all items.

TOTALS BY GROUPS

When you first clicked on the total button, each field was marked as a "Group By" field, meaning the query will break the results down into the different categories of this field.

Since every single field was marked as a group, Access would have to extract the same data in every single field for several records before it could actually have a group on which to total.

Therefore, it is up to you to rearrange the fields and to only include the fields that make the most sense for groups and totals.
For example, you might use a field called "Item #" to break down sales by each item number, as illustrated below:

<table>
<thead>
<tr>
<th>Field</th>
<th>Table</th>
<th>Total</th>
<th>Sort</th>
<th>Show</th>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item #</td>
<td>Orders</td>
<td>Orders</td>
<td>Ascending</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Group By</td>
<td>Sum</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The example above will calculate the number of items sold and break the sums down by "Item #" as shown below:

<table>
<thead>
<tr>
<th>Item #</th>
<th>SumOfQuant</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>899</td>
</tr>
<tr>
<td>2</td>
<td>152</td>
</tr>
<tr>
<td>3</td>
<td>73</td>
</tr>
<tr>
<td>4</td>
<td>296</td>
</tr>
<tr>
<td>5</td>
<td>234</td>
</tr>
</tbody>
</table>

Access always groups your results starting with the leftmost field and works to the right. That is why it is important that you arrange your fields from left to right in the order they should be grouped.

For example, to break the previous example report down by "Date" of sale first and then by "Item #", you would add the "Date" field to the beginning of the query, as shown below:

<table>
<thead>
<tr>
<th>Field</th>
<th>Table</th>
<th>Total</th>
<th>Sort</th>
<th>Show</th>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>Orders</td>
<td>Orders</td>
<td>Group By</td>
<td>Group By</td>
<td>Sum</td>
</tr>
<tr>
<td>Item #</td>
<td>Orders</td>
<td>Orders</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Quan</td>
<td>Group By</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>or:</td>
<td>Criteria</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Notice how the sales are grouped first by date. If there were multiple sales for the same date, the sales would then be broken down by item # for each date, as shown below:

<table>
<thead>
<tr>
<th>Date</th>
<th>Item #</th>
<th>SumOfQuant</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/25/09</td>
<td>5</td>
<td>14</td>
</tr>
<tr>
<td>1/16/91</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>3/1/91</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4/6/91</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>5/18/91</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>7/5/91</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>7/13/91</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>7/13/91</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>8/21/91</td>
<td>5</td>
<td>22</td>
</tr>
<tr>
<td>9/3/91</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>9/13/91</td>
<td>1</td>
<td>21</td>
</tr>
<tr>
<td>9/14/91</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>9/14/91</td>
<td>2</td>
<td>25</td>
</tr>
</tbody>
</table>

The final **SumOfQuantity** is only calculated once the items have been broken down by date.

This report will show how many of each product sold for each day of sales.
PARAMETER QUERIES

USAGE:

If you use a particular query often, but tend to change the criteria each time, you might find using that query becomes tedious since you have to constantly enter the design view and make changes. For example, you may create a sales list each day summarizing the sales of the previous day. Normally, you would have to edit the query and change the sales date every time the query is run.

Access provides a feature referred to as a Parameter query which allows you to run the query without having to manually change it each time. Instead, a box will pop up asking you to fill in the criteria to be searched for. If you have more than one criteria, you will get more than one dialog box (displayed in sequence).

You can also use these parameter queries for forms and reports. To create a parameter query, first create/edit the query you wish to use and make sure you are in the design view.

To create the prompt for the dialog box, simply enter your prompt, enclosed in square brackets[], in the Criteria cell. The prompt can contain the field name, but cannot be just the field name.

When you run the query, you will see your prompt in a dialog box:

When prompted, simply fill in the data and click on OK.
CREATING MULTIPLE PARAMETER PROMPTS

If you need to prompt for more than one value for a field, you can add multiple prompts to the Criteria cell.

This comes in handy whenever you need to ask for a range of values, such as a beginning and ending date.

To create a multiple parameter prompt for a range, use the Between operator in the criteria cell.

Example:

Between [Enter the first date:] AND [Enter the last date:] 

When using multiple parameter prompts, Access will display a separate prompt for each entry required.
Introduction to Access Queries

USING ACTION QUERIES

 USAGE:

Up to this point, you have been creating Select Queries, which are simply used to locate and display data based on your criteria. Access also has additional types of queries (referred to as Action Queries) that can be used to make new tables, update information, append records, or delete unwanted records from a table.

You can use these action queries to quickly make mass changes to your tables. For example, you may need to increase all of your prices by 10% and not want to do this manually. Instead, you could create an action query (using the “Append” query) to perform this change automatically.

CREATING AN UPDATE QUERY

To create a query that will automatically update your table, follow the steps outlined below:

1. Access the query form design window.

2. Click on the down arrow beside the “Query Type” button and choose Update Query from the list. The picture on this button will vary depending on the last type of query you selected.

3. A new row labeled Update To will be added to the query form. Add all fields required for your search criteria as well as the fields to be updated.

4. Enter the criteria for the records that are to be updated.

5. Enter the new data or an expression (formula) in the Update To section of the field(s) to be updated.
Click on this button to actually perform the query.

**TIP:** If you realize you have made a mistake while the query is still processing, you can stop it by pressing Ctrl + Break.

The example shown below could be used to update existing records by increasing the field labeled *Price* by 10%:

| Field: Price | Table: Products | Update To: [Price] * 1.1 |

**MAKING A NEW TABLE**

The “Make Table” query allows you to create a new table from all or part of the data in one or more existing tables.

You can use this type of query for such things as making backup copies, archiving data in separate tables, and creating specific tables for exporting data to other databases or applications.

To create a new table based on a sub-set of records contained within an existing table, follow these steps:

1. Access the query form design window.
Introduction to Access Queries

2 Click on the down arrow beside this button and choose **Make-Table Query**. The picture on this button will vary depending on the last type of query you selected.

A dialog box is presented asking for the name of the new table, as shown below:

![Make New Table dialog box](image)

3 Enter a name for the new table.

You will also be able to choose whether the new table should be added to the current database file or another database. When done, select **OK**.

4 Add all fields to the query that should be included in the new table and enter any criteria required for selecting records to be added.
Click on this button to actually perform the query.

A warning box appears listing the number of records (rows) that will be added to the new table:

![Warning Box]

Click on **Yes** to confirm the creation of the table.

**TIP:** Since the action cannot be undone, you may wish to review the number of records and data to be added to the new table before you actually create it.

To do so, click on **[Design]** (located on the toolbar).

When done reviewing, click on **[Design]**.

Even if the table is created and you are not satisfied with the results, you can always delete it.

**NOTE:** If you re-run the query, Access will delete the previous version of the table and recreate the table from your current query.

A box will appear warning you that the file will be deleted if you re-run the query.
CREATING A DELETE QUERY

The “Delete” query is used to delete a group of records from an existing table. For example, you can use this type of query to delete discontinued items from an inventory table. Be aware, however, that this process deletes entire records - not just the selected fields.

To use the “Delete” query, follow the steps outlined below:

1. Access the query form design window.
2. Click on the down arrow beside this button and choose Delete Query. A new row labeled Delete will appear in the design grid.
3. Drag the asterisk (*) from the field list of the table you wish to delete from to the design grid. Access will automatically place the word From in the “Delete” cell of that column.
4. If you wish to delete a specific group of records based on a set of criteria, add the fields needed for specifying the criteria to the design grid. Access adds the word Where in the “Delete” cell of each field you add to the grid.
5. Enter the Criteria for selecting the records to be deleted in the field column(s) containing the word “Where” in the “Delete” cell. Do not make any changes in the column containing the word “From” in the “Delete” cell.
6. Click on this button to actually perform the query. You will be asked to confirm the deletion.

TIP: Since the action cannot be undone, you may wish to review the number of records and data to be deleted. To do so, click on the button (located on the toolbar). When done reviewing, click on .

©2001 EZ-REF Courseware
CREATING AN APPEND QUERY

The "Append Query" adds a group of records from one or more tables to the end of another table. This is useful for combining like-tables or adding records from another user’s table.

To append records from one table to another, follow these steps:

1. Access the query form design window.

2. ![Down arrow] Click on the down arrow ![Check] beside this button and choose **Append Query**. A dialog box is presented asking for the name of the table to append to:

   ![Append To]

   - **Table Name**: Clients
   - **Current Database**
   - **Another Database**: File Name:

3. Select the name of the table. You will also be able to choose whether the table is stored with the current database file or another database. When done, select **OK**.

4. Add all fields to the query that are to be included with the appended records and any fields needed for setting criteria.

5. In the Query Design grid, use the **Append To** section to specify which fields in the new table the data should be added to. Access will attempt to match fields that have the same name in both tables. Otherwise, you will need to manually enter the field names in the table you are appending to.

6. ![Exclamation mark] Click on this button to perform the query. You will be asked to verify that you want to append the records.