

In Access Level 2, we learned how to perform basic data retrievals by using Search & Replace functions and Sort & Filter functions. For more complex tasks such as retrieving data records that meet particular criteria, you can use queries. **Queries are questions that you ask of your database.** By using queries, you can retrieve data from a single table or from multiple tables and display all data together. You can also save and store the queries as reusable Access objects. Open *LibraryDatabaseAccess3.accdb*.

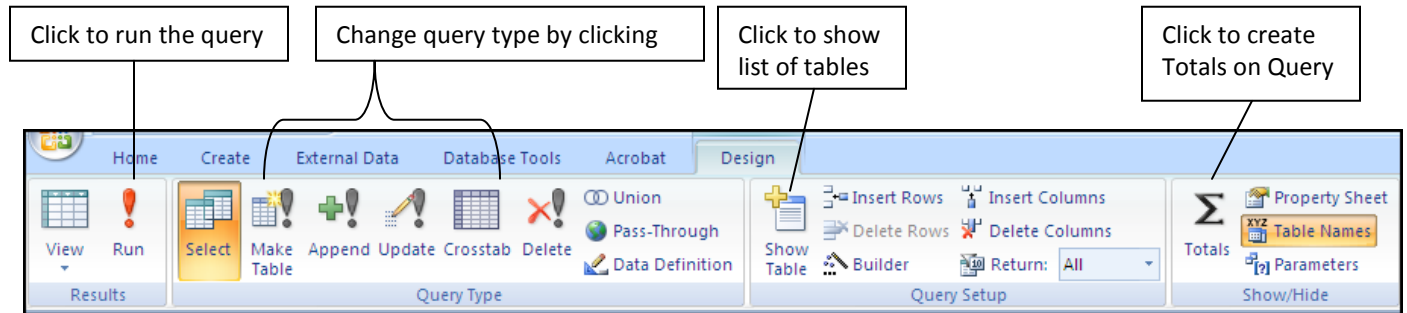
Let's take a look at the Query Design Feature. Click on Create>Query Design (in Other group). In the Show Table window on the right, double click on one or more tables to be included in your query. Or you can select each table by clicking once and then click on the Add button. Click on the Close button after finishing your selection. You will see the Query Design ribbon under the Query Tools tab and the new tab for your query opens up with two segments. The upper segment shows selected tables with existing relationships. Double-clicking on any particular field from the table list will include that field in the Lower Segment of the Query Design View window. You may also drag and drop field names to the lower segment. See below.



Click on a table in this segment then press delete key to delete from the query

Double-clicking on the asterisk above field names will include all the fields in that table.

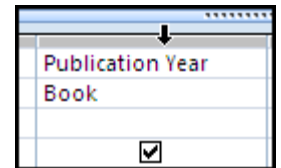
Let's explore the Query Design Ribbon.



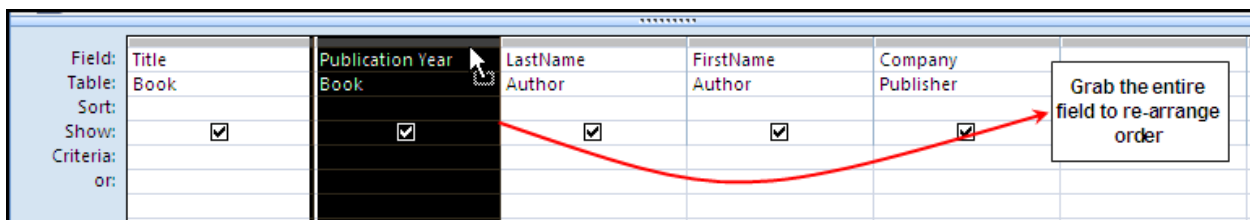
Include the following fields to run a test query.

- *Title* and *Publication Year* from the *Book* table
- *Last Name* and *First Name* fields from the *Author* table
- *Company* from the *Publisher* table

You can re-arrange the order of fields by grabbing the entire field and place it on an empty field holder. Position your mouse on the top of a field and click once to select the entire field just like in Excel as to select the entire column. The field will turn black. See sample.



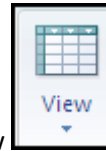
Once done, point to the gray bar and grab the entire field with your mouse to move to a new location. Click on the **Run** command on the ribbon to run your query.



Click on the Run command on the Design Ribbon. Your result query should appear in a new tab named Query1 and should look like this:

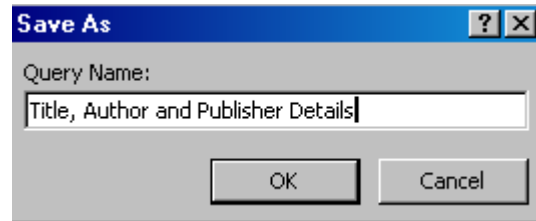


Title	LastName	FirstName	Company	Publication Year
In & Out Stories	Smith	Dick	Scrambler Publ	1987
Castaways	Smith	Dick	Arrows Publica	2004
The 2nd World War	Jackson	Marion	Carling & Sons	1988
Teach Yourself Access	Stephen	Moira	Hodder & Stou,	2007
Campfire Cooking	Watson	George	Beaver Books L	2005
Garden Shrubs	Ferguson	John	Beaver Books L	1996
African Drums	MacDonald	Donald	Hodder & Stou,	2002
Bites & Bytes	Stephen	Moira	Borthwick-Hen	2002
West Highland Way	Jackson	Allan	Hodder & Stou,	1982
Hamsters at Home	Smith	Dick	Arrows Publica	2009
Giant World Atlas	Watson	George	Outreach Colle	2001
Easy Internet	Ferguson	Alan	Borthwick-Hen	2006
The Night Sky	McDonald	Alastair	Outreach Colle	1991
Bread and Biscuits	Meunier	Luc	Harry Cousin Lt	1997
Crazy Comets	Williams	Peter	Outreach Colle	1989
Changing Skies	Borthwick	Anne	Borthwick-Hen	2000
Reading with the Under 5's	Smith	Dick	Hodder & Stou,	1991
Perfect Pizzas	Meunier	Luc	Trueform Press	2006
Outdoor Adventures	Allan	Isabelle	City Publicatio	1985
The Tortoise in the Corner	Adamson	Pauline	Arrows Publica	1991
*				

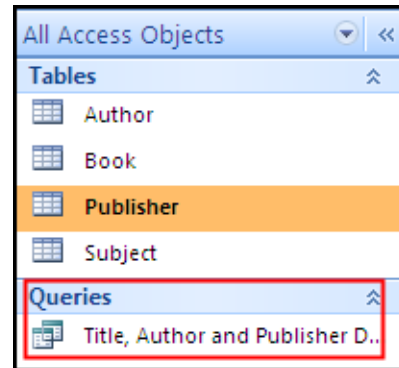


You can switch between the Design view and the Table view on the Ribbon to modify the query in the Design view even after you created it.

Saving the Query: Click the Save button on the Quick Access Tool bar or by closing out the query tab. That will prompt you to save your query. Give a name that reflects the nature of your query. See the example.



Your saved query should display in the Navigation Pane under the Queries object type.

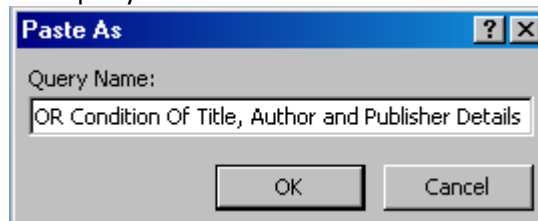


Remember: Any change you make in a query (such as adding, deleting, or editing data) will be changed in the tables too!!

Creating Subset of a Query:

OR Condition: To retrieve the subset of your records in the above query by using the “Or” condition, you can specify the criteria you want in the Query Design. Not to start from scratch, follow the steps below to copy the existing query and then create a subset out of it as a new query.

1. Right-click on the current query in the navigation pane and select “copy”.
2. Right-click on the empty area in the navigation pane and click on “paste”. Give a new name that is meaningful to your new query.



3. A new query with the new name will appear in the navigation pane. Double-click on it to open. Then click on the Design button to edit it in design view.
4. Assume you are only interested in retrieving titles in ascending order published by two companies: “Arrows Publication” and “Beaver Books Ltd” regardless of the publication year.
 - o Set the sort order to “ascending” in the *Title* column.
 - o In the *Company* column, enter “Arrows Publication” in the first criteria row and “Beaver Books Ltd” in the next criteria row below. Quotes will be automatically added when you click off. You must enter the entire name NOT a partial name.
 - o Deselect the **Show** checkbox in the *Publication Year* column. Run your query and note the results.

See the criteria in design view below.

Field:	Title	LastName	FirstName	Company	Publication Year
Table:	Book	Author	Author	Publisher	Book
Sort:	Ascending				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:				"Beaver Books Ltd"	
or:				"Arrows Publication"	

Your result should look like this.

Subset Of Title, Author and Publisher Details				
Title	LastName	FirstName	Company	
Campfire Cooking	Watson	George	Beaver Books Ltd	
Castaways	Smith	Dick	Arrows Publication	
Garden Shrubs	Ferguson	John	Beaver Books Ltd	
Hamsters at Home	Smith	Dick	Arrows Publication	
The Tortoise in the Corner	Adamson	Pauline	Arrows Publication	

Access returns records that have both names in *company* column

And Condition: You can also combine criterion in your query.

- Follow the first two steps above to create a new query out of our first query "Title, Author, and Publisher Details" and give a new name "And Condition of Title, Author and Publisher Details." See below.
- Enter "Outreach College Press" in the criteria row of *company* column and the year "1991" in the *Publication Year*. Run your query. You should retrieve one record that meets both of your criteria as shown below.

Paste As ? X

Query Name:

And Condition Of Title, Author and Publisher Details

Company	Publication Year
Publisher	Book
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
"Outreach College Press"	1991

And Condition Of Title, Author and Publisher Details				
Title	LastName	FirstName	Company	Publication Year
The Night Sky	McDonald	Alastair	Outreach College Press	1991

In both conditions above, Access assumes that you are using the comparison operator "=" (equal). You are looking for records where the text or value in the corresponding field in the datasheet is "equal" to the text or value that you have entered in the Criteria row in the Query Design grid. There are other comparison operators you can use. See below.

More on Operators

You use the comparison and other special operators in criteria rows to compare values and return a result that is True, False, or Null by using the following operator range.

Criteria	Definition	Example	Description & Result
<	less than	<3	Finds records where the field is less than a certain value – values less than 3.
>	greater than	>3	Similar to above except finding records that has field value is greater than 3.
<=	less than or equal to	<=#1/12/1936#	Finds records where the field value is less than or equal to a certain value – records will have dates on or before January 12, 1936. Note: Dates are enclosed between # signs.
>=	greater than or equal to	>=#1/12/1936#	Similar to above except finding records that has value greater than or equal to a certain value – January 12, 1936.
<>	not equal to	<>1994	Find records where the field value is not equal to a certain value – 1994.
=	equal to	=Ferguson Ferguson ="Ferguson" "Ferguson"	Finds records where the value in the field is Ferguson. Notes: In Access 2007, either equal sign or double quotes around your text value are not necessary.
Between..And..	between the first and the last value entered (including the values)	Between 1 and 3	Find records where the field is between the two endpoints including 1 and 3.
Is Null	is empty	Is Null	Find records where the field is empty.
Is Not Null	is not empty	Is Not Null	Find records where the field is not empty.
Like	Like	Like "ferguson"	Find records with "ferguson" in the field.

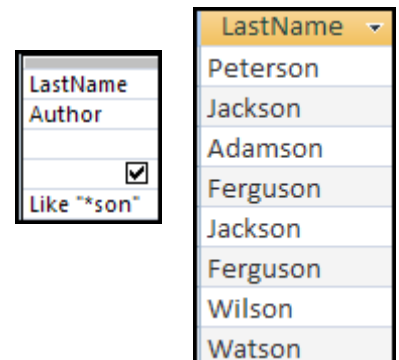
Wild card Characters

When using "Like" and "=" operators, you can include "wild cards" characters to represent either individual characters or a string of characters. Two wild card characters are:

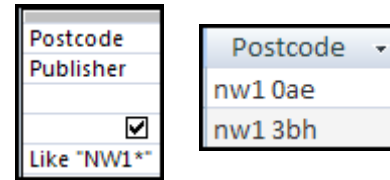
- * used to present a string of characters
- ? used to represent a single character

See examples below.

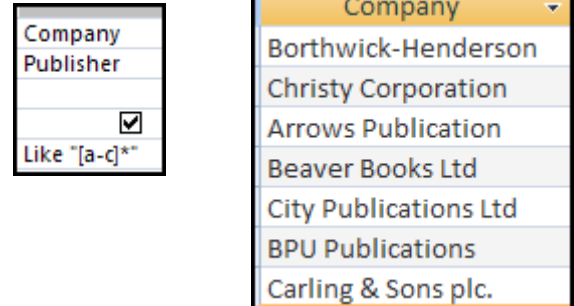
Like "*son" - Find records that end in the letters "son" as in Ferguson, Peterson, etc.



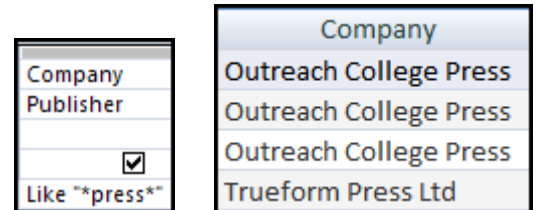
Like "NW1*" - Find records that begin with "NW1"



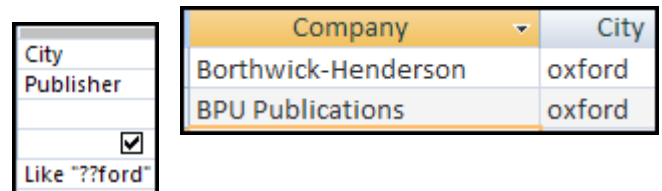
Like "[a-c]*" - Find records that begins with letters a, b, or c



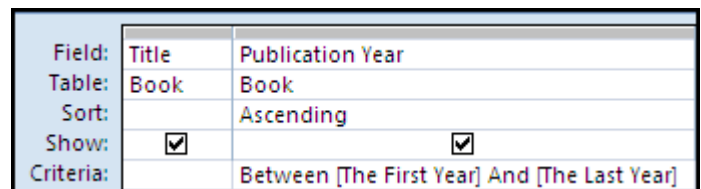
Like "*press*" - Find records that contain the word "press" in the field (beginning, middle, or end)



Like "??ford" - Find records that begins with two letters but ends with the letters "ford" in the field.



Between..And - This allows a user to put in a range in query, called a **parameter query**. A parameter query prompts the user for the value of a particular query field, rather than having the value built into the query itself. For example, if you desire to find titles published in a specified range of years, you can create a parameter query. Inside the brackets, type text telling the user what kind of information to enter.



Title	Publication Year
The Tortoise in the Corner	1991
Reading with the Under 5's	1991
The Night Sky	1991
Garden Shrubs	1996
Bread and Biscuits	1997
Changing Skies	2000

In this query, user will put in the first year and the last year of publication. Example shows the search result on titles published between 1990 and 2000 arranged in ascending order by the year.

Creating a Concatenated field: As in Excel, Access allows you to create a combined [Concatenated] field from more than one field in a new field [column]. Assume you want to combine the Last Names and First Names fields into one field called "FullName". Select the original query "Title, Author and Publisher Details", copy it and create a new query and give it the name "Concatenated field". In Design view, click in a blank column and type the following syntax in the field row.

FullName: [FirstName] & " " & [LastName]

Use double quotes to create a space between the FirstName and LastName fields. Use square brackets around the field names and "&" to combine between every two fields. Uncheck the LastName and FirstName columns not to show then run your query. The partial result should look like this:

Concatenated field

Title	Company	Publication Year	FullName
In & Out Stories	Scrambler Publications Ltd	1987	Dick Smith
Castaways	Arrows Publication	2004	Dick Smith
The 2nd World War	Carling & Sons plc.	1988	Marion Jackson
Teach Yourself Access	Hodder & Stoughton Ltd	2007	Moira Stephen
Campfire Cooking	Beaver Books Ltd	2005	George Watson

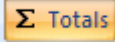
Creating a Calculated field: You can also create a calculated field in a new blank field in the Design form similar to creating a Concatenated field above. Assume you want to find out the total figure of all the copies of each title in the database. You can use normal mathematical operators (+ - * /). See more on "Operators" in the Appendix. Create a copy of the original query "Title, Author and Publisher Details", copy and create a new query and give the name "Calculate on Title, Author and Publisher Details". In Design view, delete the "LastName" and "FirstName" fields; add "Price" and "Number of Copies" fields from the Book table. Select a new field and type in the following syntax in the field row. Run the query.

Total Cost: [Number of Copies]*[Price]

The partial result should look like this. Access carried out the multiplication operator and provided results in the new calculated field "Total Cost" column.

Title	Company	Publication Year	Number of Copies	Price	Total Cost
In & Out Stories	Scrambler Publications Ltd	1987	3	\$3.99	\$11.97
Castaways	Arrows Publication	2004	2	\$4.50	\$9.00
The 2nd World War	Carling & Sons plc.	1988	3	\$12.50	\$37.50
Teach Yourself Access	Hodder & Stoughton Ltd	2007	2	\$8.99	\$17.98

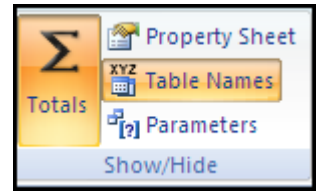
You can use the **Totals** function in your query (in datasheet view) by using

 on the Home ribbon. Simply select the column where you want to see Sum, Average, Count, Maximum, etc. For practice, select Count in Number of Copies column and Sum in Total Cost column or any column you want to see Math function results. Clicking on that Totals function again will turn off those Math function results.



None
Sum
Average
Count
Maximum
Minimum
Standard Deviation
Variance

You can also use the Totals function in your query (in the design grid) by using the Totals function key in the Show/Hide group. The aggregate functions e.g. are sum, average, min, max, and count. You can use this type of function to work out how much you spend on each subject in your library. Create a new query by pulling out the *Book* and *Subject* tables. Use the *Subject Name* field. Create a new field for Total Cost by typing:



Total Cost: [Number of Copies]*[Price]

Click on Totals function button. A new line called “Total” will be added. Select “sum” in the Total Cost field and select “Ascending” in the sort row of the Subject Name field. Save the query as “Totals in Query”. Run the query. The result shows how much in total you spend on each subject.

Subject Name	Total Cost: [Number of Copies]*[Price]
Subject	
Group By	Sum
Ascending	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Subject Name	Total Cost
Adult Fiction	\$12.50
Animals	\$4.50
Astronomy	\$109.48
Children's Fiction	\$33.97
Computing	\$57.94
Cooking	\$49.97
Education	\$13.98
Gardening	\$9.98
History	\$37.50
Travel	\$77.50

On the other hand, if you want to count how many books under each subject, just insert a “Number of Copies” field. Uncheck the Show checkbox for the Total Cost field and set it to group by the “count” function under the Number of Copies field.

Subject Name	Number of Copies
Subject	Book
Group By	Count
Ascending	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Subject Name	CountOfNumber of Copies
Adult Fiction	1
Animals	1
Astronomy	3
Children's Fiction	3
Computing	3
Cooking	3
Education	1
Gardening	1
History	1
Travel	3

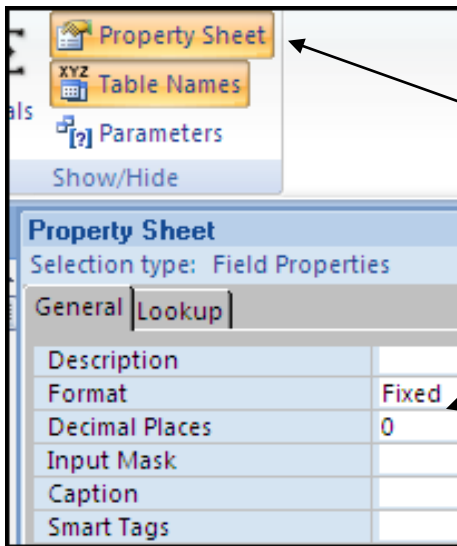
Is Null & Is Not Null: Useful in retrieving records that have blank or non-blank fields. For example, you want to retrieve those publishers with business phone numbers listed in the fields. Create a new query that includes only the Publisher Table. Pull in the Company and BusinessPhone fields in design grid. See partial results below. Save the Query as “Null Query”.

Field:	Company	BusinessPhone
Table:	Publisher	Publisher
Sort:		
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		Is Not Null

Company	BusinessPhc
Hodder & Stoughton Ltd	020 7738 6060
Borthwick-Henderson	01865 333545
Westward Lock Ltd.	020 7333 4454

Date Function: In Access, the Date() function returns the current date from your computer system as a number in *days* . You will have to convert *days* into *weeks* by using (/7) and into *years* by using (/365). Say that you wanted to calculate the ages of those living authors. Create a new query by using the *Author Table* alone. First, you need to pull out the authors into your query whose “DateofDeath” field is blank (Is Null) and whose “DateofBirth” field is not blank (Is Not Null). Then create a new blank field to calculate from today’s date. See example below.

Field:	LastName	FirstName	DateofBirth	DateofDeath	Age: (Date()-[DateofBirth])/365
Table:	Author	Author	Author	Author	
Sort:					
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			Is Not Null	Is Null	



Age=Today’s date minus [DateofBirth] then divide by 365 days to convert to years.

In the Property Sheet button while in the Age field, limit the field format to “Fixed” and run the Query. The age field will have two decimals. Go back to Design View and set zero to Decimal places.

The result should look like this. (Note: DateofDeath field is unchecked not to show in the result page.) save the query as “Author Age Query”.

LastName	FirstName	DateofBirth	Age
McDonald	Alastair	1 /12/1936	75
Jackson	Marion	6 /24/1955	56
Duncan	Wilma	7 /4 /1938	73
Schmit	Hans	12/12/1952	58
Allan	Isabelle	5 /10/1965	46
Williams	Peter	10/10/1930	80
Ferguson	Alan	12/4 /1945	65
Wilson	Peter	1 /3 /1930	81
Watson	George	10/22/1945	65

In level 2, we learned how to set relationships among tables. The lines between the tables that define relationships are called join lines. There are two types of join lines: inner and outer joins. In queries, you can define your joins based on what you need to ask of your database. Create a new query that includes the *Publisher* and *Book* Tables. Pull in Company from the *Publisher* Table and Title from the *Book* Table. Double-click on a join line to define join properties.

Inner Joins: These queries return only those records from both tables in the join that match on the joining field. For example, if you want to retrieve only the titles that have corresponding publisher company names, use inner joins - option 1.

Outer Joins: On the other hand, if you want to display all publisher company names, regardless of whether we own any titles from them, use outer joins - option 2 or 3.

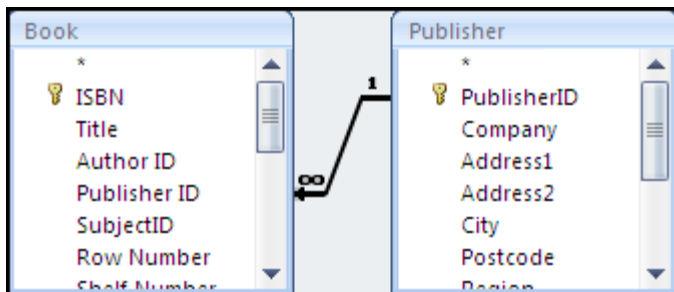
Title	Company
Hamsters at Home	Arrows Publication
Castaways	Arrows Publication
The Tortoise in the Corner	Arrows Publication
Campfire Cooking	Beaver Books Ltd
Garden Shrubs	Beaver Books Ltd
Changing Skies	Borthwick-Henderson
Easy Internet	Borthwick-Henderson
Bites & Bytes	Borthwick-Henderson
The 2nd World War	Carling & Sons plc.
Outdoor Adventures	City Publications Ltd
Bread and Biscuits	Harry Cousin Ltd
Teach Yourself Access	Hodder & Stoughton Ltd
African Drums	Hodder & Stoughton Ltd
West Highland Way	Hodder & Stoughton Ltd
Reading with the Under 5's	Hodder & Stoughton Ltd
The Night Sky	Outreach College Press

Inner joins with option 1

Title	Company
The Tortoise in the Corner	Arrows Publication
Castaways	Arrows Publication
Hamsters at Home	Arrows Publication
Garden Shrubs	Beaver Books Ltd
Campfire Cooking	Beaver Books Ltd
Bites & Bytes	Borthwick-Henderson
Easy Internet	Borthwick-Henderson
Changing Skies	Borthwick-Henderson
	BPU Publications
The 2nd World War	Carling & Sons plc.
	Christy Corporation
Outdoor Adventures	City Publications Ltd
	Darling Kinghorn Ltd
Bread and Biscuits	Harry Cousin Ltd

Outer joins with option 2

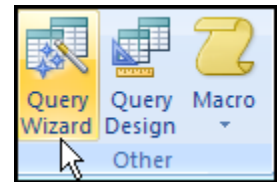
In the outer joins example, the direction of the arrow indicates that every row in the company field in the *Publisher* table will display whether or not a matching records exist in the title field of the *Book* table. You can save the query as *Join Properties Query*.



Field:	Title	Company
Table:	Book	Publisher
Sort:		Ascending
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		

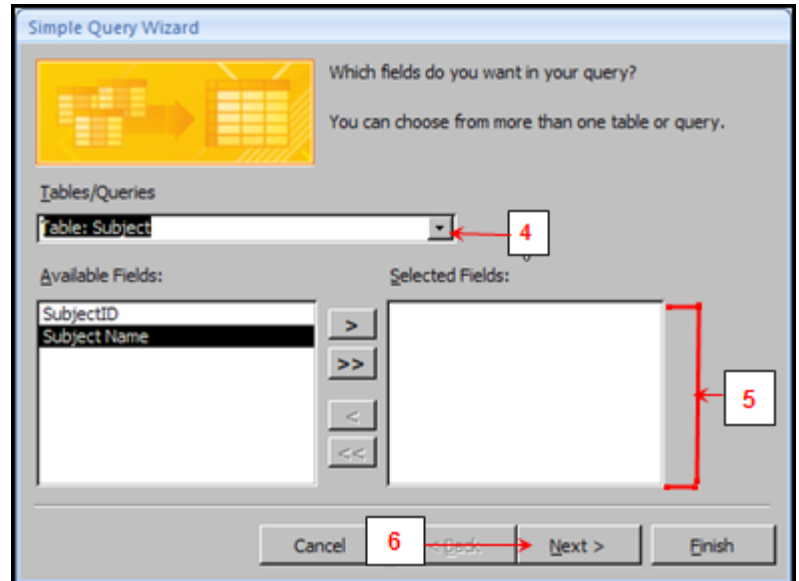
Creating a Query using a Wizard:

You can build a query by using the Query Wizard as follows:

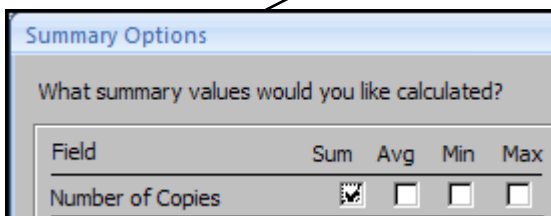
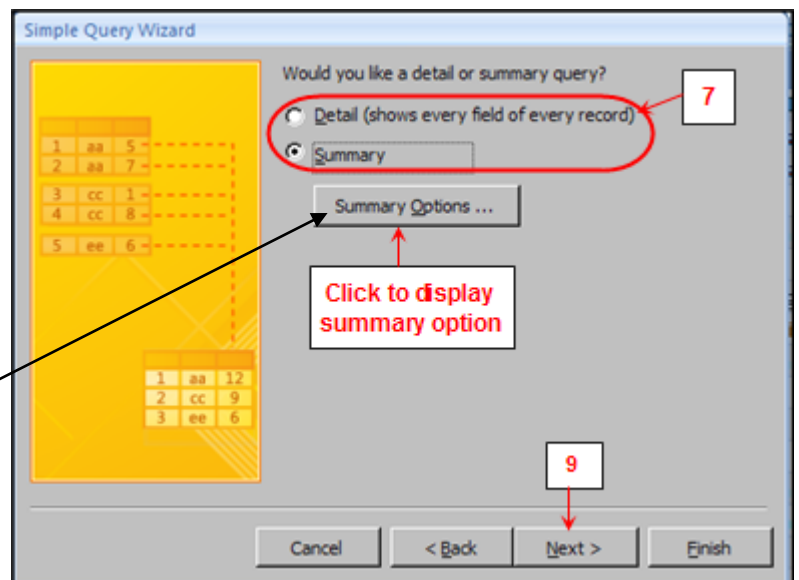


1. Click the *Create* tab.
2. Click the *Query Wizard* button.
3. Click *Simple Query Wizard*, and then click *OK*.

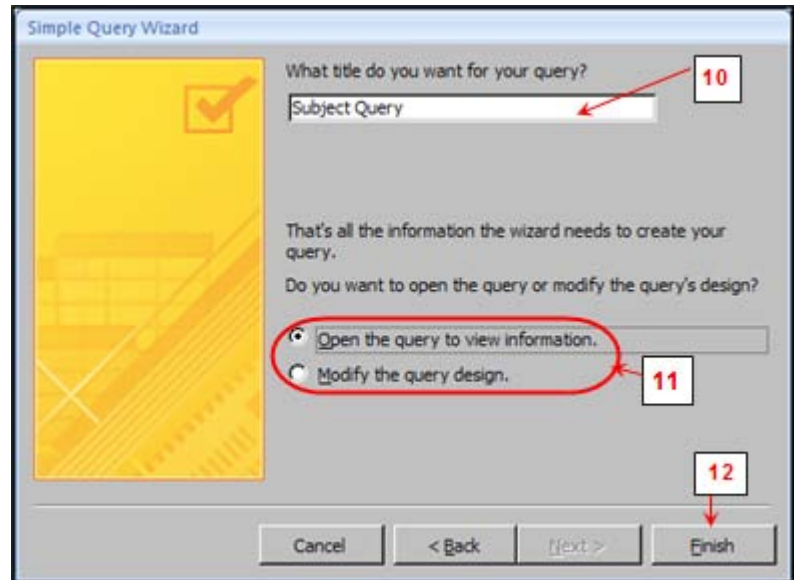
4. Select a table or existing query to include in your query. (When you use an existing query to create a new query, the existing one is known as a *subquery*.)
5. Click to select the fields that you want included in your query.
6. Click *Next* to continue.



7. If you selected numeric or date fields in step 5, indicate whether you want to see detail or summary information.
8. If you choose Summary, click *Summary Options* to specify the calculation for each field, and then click *OK*.
9. Click *Next* to continue.




10. In the final wizard dialog box, type the name of the query.
11. Select whether you want to view the results of the query or modify the query design in *Design View*.
12. Click *Finish*.



Backup Database: You should always create a back-up database before you create any action queries. To do so, follow the steps below.

Back up a database

1. Click the **Microsoft Office Button** , click the arrow next to **Manage**, and then click **Back Up Database**.

The **Save As** dialog box appears, and Access appends the current date to the file name. For example, if you have a database named Assets, Access creates the following type of file name: **Assets_2006-10-29**.

2. Accept the default name and location, or select another name or location, and then click Save. Access closes the original file, creates a backup, and then reopens the original file.

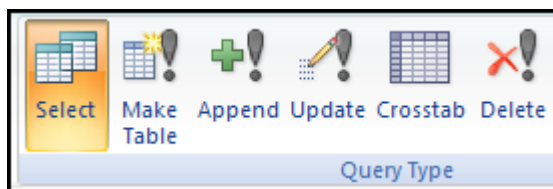
Appendix (For Self-Learning)

(Create a Back-up Database before trying anything)

Types of Queries: As you were going through the first step of the Query Wizard, you may notice that there are other types of queries.

1. A Select Query: Queries we have created so far either in Design View or by using the Query Wizard are Select queries. A select query retrieves and displays records in the Table window in Datasheet view.
2. A Crosstab Query: This query type displays summarized values (sums, counts, and averages, etc) from one field in a table, and groups them by one set of fields listed down the left side of the datasheet and by another set of fields listed across the top of the datasheet similar to Spreadsheet.
3. An Action Query: It performs operations on the records that match your criteria. There are four kinds of action queries:
 - a. Delete Queries: delete matching records (**Caution: Once done, non-reversible**)
 - b. Update Queries: make changes to matching records (**Caution: Once done, you cannot easily change them back**)
 - c. Append Queries: add new records to the end of a table
 - d. Make-table Queries: create new tables based on matching records
4. A Parameter Query: it allows you to prompt for a single piece of information to use as selection criteria in the query. You can enter the text of the prompt surrounded by square brackets. You can enter multiple criteria in a field by using AND, or use different fields. Example: We have created one by using "Between..And.." (see page 6).

The following picture shows different query types under the Design tab.



Make Table Query: The data that appears after you run a query appears in a table form and Access allows you to work with those results like tables. However, they are not tables. *Make Table* query is helpful when you want to save the query results to a new table in either the current or a different database. Try the following exercise. Open *LibraryDatabaseAccess3.accdb*.


1. In *Query Design* view, create a select query. Add all 4 tables: Subject, Book, Author, and Publisher.
2. Select Subject Name, Title, LastName, Price, and Company fields from the appropriate tables.
3. Set to sort the records - Ascending order by Subject Name.
4. Limit the Company records to "Borthwick-Henderson". (Hint: In the Criteria row in the Company field, type in **Like "borthwick*"**.)
5. Click on the *Data View* button to preview the list of records.
6. If you're satisfied that the appropriate records are displayed, click the *Make Table* button. Type the name of the table (BorthworkBooks) and indicate you want to place it in the current database. Click OK.
7. Click the *Run* button.
8. Click *Yes* when Access asks if you're sure you want to create the new table.
9. Open the new table to view the records resulting from the query.

- Close the Query in Design grid. (You actually do not have to save this query unless you need to run it again in future. If you do run it again, Access will delete the existing one first before it presents you with a new table.)

Append Query: You can use a query to add records to a table by creating an *append query*. If the fields you've selected have the same name in both tables, Access automatically fills the matching name in the *Append To* row in the design grid. If the fields in the two tables do not have the same name, enter the names of the fields in the *Append To* row in the design grid. If the table you are appending records to includes a primary key field, the records you are appending must have the same field or an equivalent field of the same data type. Access won't append any of the records if either duplicate or empty values would appear in the primary key field. Try the following example (Source: Steve Johnson, Perspection, Inc. *Microsoft Office Access 2007*.)

- Open *Vendor.accdb* and open the "Low Price CL" query in Design View. The query basically pulls *Price*, *Vendor Code*, and *Product* from *Products Table* and *Vendor Name* from *Vendor Table*. We are going to append some records from this query into another table by using *Append Table* query function.
- In Design View, click the *Append* button in Query Type group.
- Type the name of the table to which you want to append the records or click the list arrow and then choose one. Select *LightingElectric* table. Then click the *Current Database* option or click the *Another Database* option and type the name of another database (including the file path, if necessary.) Click OK.
- Specify which fields will contain the appended values by entering the field names in the *Append To* row of the design grid. Place "**Theil's Electric**" in the *Criteria* row under the *Vendor Name* column since you want to add this vendor to the other table.

Field:	Price	Vendor Code	Product	Vendor Name
Table:	Products	Products	Products	Vendors
Sort:				
Append To:	Price	Vendor Code	Product	Vendor Name
Criteria:				"Theil's Electric"
or:				

- Click the *Run* button. Click *Yes* when Access asks if you're sure you want to append records to the table.
- Open the *LightingElectric* table to view the appended records at the end. When you close the query, you will be asked to save the changes and if you said yes, your query now has the symbol of Append Query in front of the name.  Low Price CL

Delete Query: Use **Delete Query** to remove records from a table based on a criterion or criteria. The delete query searches the table you specify and removes all records that match your criteria. **Note: Access permanently deletes these records, so use with caution.** You can preview the results by clicking the Datasheet View button before you actually run the query. Let's try the example below.

- Open *Vendor.accdb*. Not to change the data in original table, copy the *Products Table* and paste it in the Table and give a name as "Copy of Products". (Hint: right-click to copy and right-click to paste in the Table section.)

Field:	Copy Of Products.*	Price
Table:	Copy Of Products	Copy Of Products
Delete:	From	Where
Criteria:		<0.99
or:		

2. Create a Select Query in Design View by selecting Copy of Products table.
3. Double-click on the * in the upper table grid to include every field. Double-click again on Price field alone to specify that you want to delete records from the table where price is less than \$0.99.
4. Click on the Design tab under Query Tools.
5. Click the Delete button. The Delete row in the lower grid will indicate a “Where” location. Type in <0.99 in the Criteria row to indicate all records with less than \$0.99 will be deleted.
6. Click the Datasheet View button to preview the list of deleted records. If you are satisfied that the appropriate records would be deleted, click the Design View again and then click the Run button.
7. Click Yes when Access asks if you’re sure you want to delete records from the table. See the criteria in design grid below.

Update Query: It allows you to make changes to a set of records that match your query’s criteria. Assume you want to increase all the prices by a particular vendor in your product table to 25%. Let’s try the example below.

1. Open *Vendor.accdb*. Not to change the data in original table, copy the Products Table and Vendor Table; paste them in the Table area and give names as “Copy of Products” and “Copy of Vendor”. (Hint: right-click to copy and right-click to paste in the Table section.)
2. Create a Select Query in Design View by selecting the Copy of Products table and Copy of Vendor. Create a relationship by connecting two Vendor ID fields from these two tables.
3. Select the Product, Price, and Vendor ID fields in the lower grid of the Design grid.
4. Click the Update button.
5. Enter the expression to update the selected field. (Under the Price field, update to price multiplied by 1.25 to increase prices to 25% and limit the criteria under Vendor ID field to “LT”.) See below.

Field:	Product	Price	Vendor ID
Table:	Copy Of Products	Copy Of Products	Copy Of Vendors
Update To:		[Price]*1.25	
Criteria:			"LT"
or:			

6. Click the Datasheet View button to preview the list of updated records. Remember once you run the query, records are updated and you cannot change them back.
7. Click the Run button.
8. Click Yes to update the records.

Other Specialized Queries: There are other queries that you can explore on your own in the Access help manual by clicking the Help button in the upper right corner of the screen.

Find Duplicate Query: Use this query to retrieve duplicate records in a table by displaying a field containing duplicate values alongside additional query fields for comparison purpose. The additional query fields will help you decide whether the record is really a duplicate or not.

Find Unmatched Query: Use this query to find the unmatched entries between two tables. For example, use this if you want to find out whether there are any products in Products Table that have no match in the Orders Table.

Crosstab Query: It allows you to summarize the contents of fields that contain numeric values, such as Date fields or Number fields. The results of the summary calculations are shown at the intersection of rows and columns. The value in a crosstab query cannot be changed in order to change the source data.

SQL-Specific Queries: SQL is a powerful database language used in querying, updating, and managing relational databases.

- **Union Query** – combines related fields from multiple tables into one field, thus combining the data from several tables.
- **Pass-Through** - sends SQL commands directly into an SQL database server. This allows you to work with tables on the server instead of linking the tables to your Access database.
- **Data Definition** – creates, deletes, or alters tables, or creates indexes in a database table.

Source: Microsoft.com

Operators

Arithmetic operators

You use the arithmetic operators to calculate a value from two or more numbers or to change the sign of a number from positive to negative or vice versa.

Operator	Purpose	Example
+	Sum two numbers.	[Subtotal]+[SalesTax]
-	Find the difference between two numbers or indicate the negative value of a number.	[Price]-[Discount]
*	Multiply two numbers.	[Quantity]*[Price]
/	Divide the first number by the second number.	[Total]/[ItemCount]
\	Round both numbers to integers, divide the first number by the second number, and then truncate the result to an integer.	[Registered]\[Rooms]
Mod	Divide the first number by the second number, and then return only the remainder.	[Registered] Mod [Rooms]
^	Raise a number to the power of an exponent.	Number ^ Exponent

Comparison operators

You use the comparison operators to compare values and return a result that is True, False, or Null.

Operator	Purpose	Example
<	Returns True if the first value is less than the second value.	Value1 < Value2
<=	Returns True if the first value is less than or equal to the second value.	Value1 <= Value2
>	Returns True if the first value is greater than the second value.	Value1 > Value2
>=	Returns True if the first value is greater than or equal to the second value.	Value1 >= Value2
=	Returns True if the first value is equal to the second value.	Value1 = Value2
<>	Returns True if the first value is not equal to the second value.	Value1 <> Value2

Note: In all cases, if either the first value or the second value is null, the result is then also null. Because null represents an unknown value, the result of any comparison with a null value is also unknown.

Logical operators

You use the logical operators to combine two Boolean values and return a true, false, or null result. Logical operators are also referred to as Boolean operators.

Operator	Purpose	Example
And	Returns True when Expr1 and Expr2 are true.	Expr1 And Expr2
Or	Returns True when either Expr1 or Expr2 is true.	Expr1 Or Expr2
Eqv	Returns True when both Expr1 and Expr2 are true, or when both Expr1 and Expr2 are false.	Expr1 Eqv Expr2
Not	Returns True when Expr is not true.	Not Expr
Xor	Returns True when either Expr1 is true or Expr2 is true, but not both.	Expr1 Xor Expr2

Concatenation operators

You use the concatenation operators to combine two text values into one.

Operator	Purpose	Example
&	Combines two strings to form one string.	string1 & string2
+	Combines two strings to form one string and propagates null values (if one value is Null, the entire expression evaluates to Null).	string1 + string2

Special operators

You use the special operators to return a True or False result as described in the following table.

Operator	Purpose	Example
Is Null or Is Not Null	Determines whether a value is Null or Not Null.	Field1 Is Not Null
Like "pattern"	Matches string values by using the wildcard operators ? and *.	Field1 Like "instruct*"
Between val1 And val2	Determines whether a numeric or date value is found within a range.	Field1 Between 1 And 10 - OR - Field1 Between #07-01-07# And #12-31-07#
In(val1,val2...)	Determines whether a value is found within a set of values.	Field1 In ("red","green","blue") - OR - Field1 In (1,5,7,9)