

---

## Lesson 10      The Query Wizard and Query Operators

### Lesson Topics

- The Simple Query Wizard
- Using Comparison Operators
- Using Other Operators
- Review Exercise

### Lesson Objectives

At the end of the lesson, you will be able to:

- Use the Query Wizard to create a simple query;
- Modify a query that was created with a wizard so that it includes criteria;
- Use the comparison operators (=, >, <, >=, <=, and < >) to specify criteria ranges;
- Combine comparison operators in queries;
- Use the Between and the Like operators in queries.

---

### Student Files Used

You will use the following file from your student folder:

- More Query

## The Query Wizard

You can use the *Query Wizard* to design simple queries. A wizard is a series of dialog boxes that assist you in the completion of a task. You may have already noticed that there are many different wizards in Access.

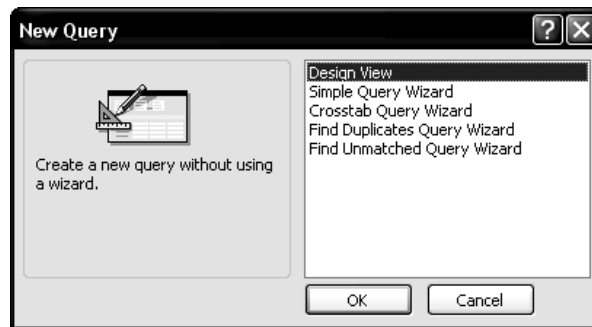
You are going to use the query wizard to help you design a simple query to display students in homeroom 362.

**1. Open the *More Query* database.**

The Database window should be visible.

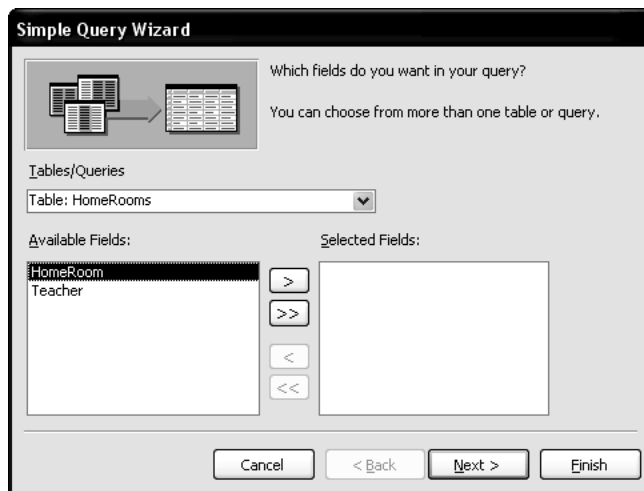
**2. On the Insert menu, choose *Query*.**

The New Query dialog box appears.



**3. On the dialog box, choose *Simple Query Wizard* and then click **OK**.**

The following dialog box appears.



This is the first step in the Simple Query wizard. As you go through any of the Access wizards, be sure to read each dialog box to understand each step.

4. The first dialog box asks for you to specify the fields to include in the query. First, you should make sure that the correct table is selected.

**Click the *Tables/Queries* drop-down list box and choose *Table: StudentInfo*.**

In the *Available Fields* list box, notice the fields that make up the selected table. To include a field in the query, you must move it to the Selected Fields list box. You can do this by selecting the field and then clicking the appropriate Move Field button located between the list boxes.

The Move Selected Field buttons (the single arrow buttons) will move the selected field to the list box towards which they are pointing. The Move All Fields buttons (the double arrow buttons) will move all of the fields to the list box towards which they are pointing.

5. **With the *StudentID* field highlighted, click the Move Selected Field button.**



The field is moved to the list box at the right.

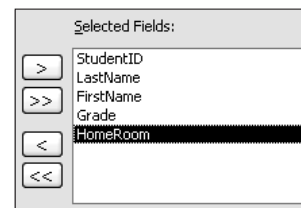
**Tip:** Instead of clicking the Move Selected Field button, you can simply double-click a field to move it from one list box to the other.

6. **In the same way, move the *LastName*, *FirstName*, *Grade*, and *HomeRoom* fields to the Selected Fields list box.**

The *Selected Fields* list box should look like the following:

**Note:** The fields appear in the *Selected Fields* list box in the order in

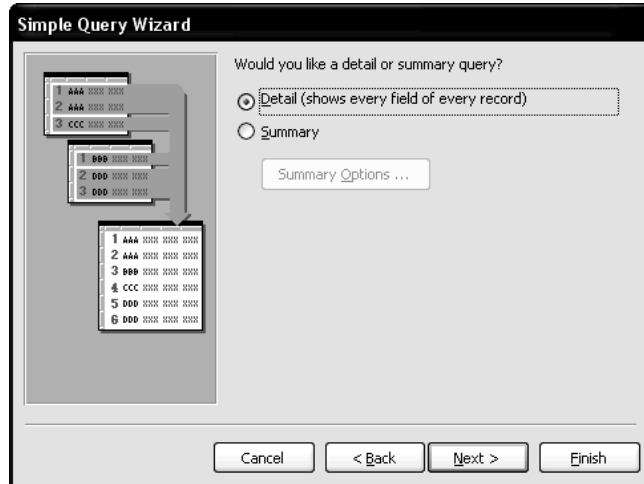
which they were moved. This, in turn, is the order in which they will appear in the query design grid.



7. The Next button takes you to the next dialog box.

**Click the Next button.**

Notice the second dialog box of the wizard.



Access is asking you if you want a detail or summary query. All of the queries that you have done have been detail queries — the query results list all of the selected fields for each record. A summary query is used to list only “summary values” of the records. These summary values can include totals, averages, or minimum/maximum values. You might use a summary query when working with a table of student test scores. You will use summary queries in the Intermediate Course.

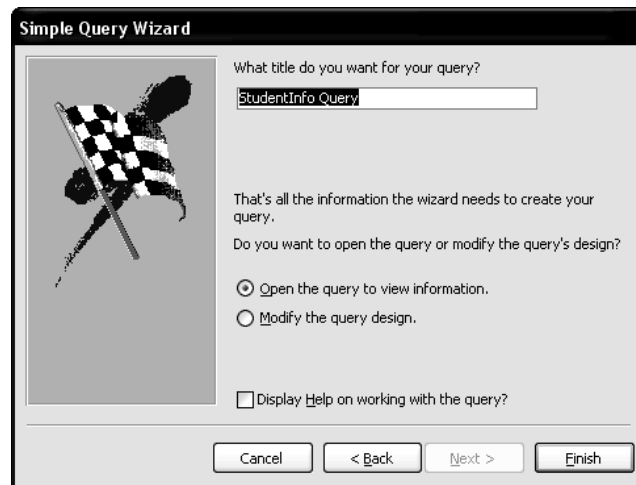
**Note:** This dialog box appears only if your table or query contains a *Number* or *Currency* data type field.

**8. On the dialog box, verify that *Detail* is selected.**

Notice the other buttons on the wizard dialog box. The Back button will take you to the previous dialog box. This is helpful if you need to adjust any of the options that you selected earlier. The Cancel button will close the wizard without making any changes to the database. The Finish button will finish the wizard by (automatically) choosing the default options on any of the remaining wizard dialog boxes. You can click this button at any time.

**9. Click the Next button.**

The next (and final) dialog box appears.



The last dialog box in a wizard usually contains a checkered flag. In this case, you are given the opportunity to name the query and to choose between running it or modifying it.

10. **Change the contents of the top text box to:**  
HomeRoom362
11. Although you are not done with the query yet (you must specify the criteria), you are going to run it from the wizard to see how it looks.

**Verify that *Open the query to view information* is selected and then click **Finish**.**

Notice the results of the query. All of the records in the table are listed, but only the fields that you specified in the wizard appear in the Query Datasheet.


Designing a query that includes all records but only some of the fields can be a useful way of getting an overview of essential information in a table without the clutter of many fields.

12. To display only the records of students in homeroom 362, you must set criteria in the query design grid.

**Click the View button on the toolbar.**



Notice the query in Design View. Notice that the records are not sorted and that no criteria are set. You are going to set criteria and sort the records by last name and then first name.

13. In the *Criteria* row of the *HomeRoom* column, type: 362
14. In the *Sort* row for the *LastName* and *FirstName* columns, choose *Ascending*.
15. On the **File** menu, choose *Save*.
16. On the toolbar, click the **Run** button. 

Notice the results of the query. Only students in homeroom 362 are listed and the records are sorted by last name and then first name.

	StudentID	LastName	FirstName	Grade	HomeRoom
▶	1042	Diaz	Maria	12	362
	1003	Feliciano	Jose	12	362
	1027	Tolino	Nicole	12	362
	1002	Williams	Latonya	12	362
*	0			0	

17. Close the **Query Datasheet** window.

## Using Comparison Operators

So far, all that you have typed into the *Criteria* rows of the query design grids have been examples of criteria to be matched exactly (e.g., the contents of the *HomeRoom* field equals 362). At times, however, you might need to enter criteria that are more general — such as students who entered the school after a certain date or employees who earn between \$20,000 and \$30,000 per year. To do this, you can use an *operator*.

First, you are going to work with *comparison operators*. Below are standard comparison operators used in most database programs.

Operator	Meaning
=	equals
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
<>	not equal to

You are going to use the Simple Query wizard to design a new query and then use different comparison operators with it.

(Without knowing, you have already used the *equals* operator. Access assumes this operator if no other is entered.)

1. **The Database window of *More Query* should be open.**
2. **On the Insert menu, choose *Query*.**
3. **On the New Query dialog box, select *Simple Query Wizard* and then click **OK**.**

The first dialog box of the wizard appears.

4. **On the dialog box, click the *Tables/Queries* drop-down list box and choose *Table: StudentInfo*.**
5. **Move the *StudentID*, *LastName*, *FirstName*, *DOE*, and *Grade* fields to the *Selected Fields* list box.**
6. **Click the **Next** button.**

The second dialog box in the wizard appears.

7. **Verify that *Detail* is selected and then click **Next**.**

The last step in the wizard appears.

8. **Name the query *EntryDate*.**
9. **Verify that *Open the query to view information* is selected and then click **Finish**.**

Notice the results of the query. Since no criteria were specified, all of the records in the table appear.

### The Greater Than Operator (>)

You are going to use the *greater than* operator (>) to locate the records of students who entered the school after February 1, 2003.

1. **Open the *EntryDate* query in Design View.**
2. **In the *Criteria* row of the *DOE* field, type: >2/1/03**

You have used the greater than operator to ask Access to locate the records of students who entered the school after 2/1/03. You can use any valid date format — you are asking Access to compare the date *value*, not the date *format*.

3. **Tap the **TAB** key to move out of the column.**

Notice that Access surrounds the date with pound signs (#).

DOE
StudentInfo
<input checked="" type="checkbox"/>
>#2/1/2003#

This is so the slashes in the date format are not confused with the division symbol. Notice that Access also expands dates from two digits (03) to four digits (2003).

**4. On the Query Design toolbar, click the Run button.**

Notice the results of the query. Only students who entered *after* (but not *on*) 2/1/03 are listed.

**5. Click the View button to return to the Design View window.**

---

### The Less Than Operator (<)

You are going to use the *less than* operator (<) to locate the records of students who entered the school *before* 2/1/03.

**1. You are going to use a different date format.**

**In the DOE column, change the contents of the Criteria row to: <Feb 1, 2003**

Remember, Access considers only the value of the date, not its format.

**2. Click the Run button.**

Notice the results of the query. Only students who entered before (but not *on*) 2/1/03 are listed.

**3. Click the View button to return to Design View.**

In the DOE field, notice that Access converted the longer date format to a short date format and then surrounded it with pound signs (#).

---

### The Greater Than or Equal To Operator (>=)

You are going to use the *greater than or equal to* operator (>=) to locate the records of students that entered the school *on* or *after* 2/1/03.

**1. In the DOE column, change the contents of the Criteria row to: >=2/1/03**

**2. Run the query.**

Notice the results of the query. Only students who entered the school on or after 2/1/03 are included.

**3. Return to Design View.**

---

**The Less Than or Equal To Operator (<=)**

You are going to use the *less than or equal to* operator (<=) to locate the records of students who entered the school on or before 2/1/03.

1. **In the *DOE* column, change the contents of the *Criteria* row to: <=2/1/03**

2. **Run the query.**

Notice the results of the query. Notice that the records of students who entered the school on or before 2/1/03 are included in the datasheet.

3. **Return to Design View.**

---

**The Not Equal To Operator (<>)**

You are going to use the *not equal to* operator (<>) to locate the records of all students who did not enter the school on 2/1/03.

1. **In the *DOE* column, change the contents of the *Criteria* row to: <>2/1/03**

2. **Run the query.**

Notice the results of the query. Most of the records are listed. Only the records of students who entered the school on 2/1/03 are not included.

3. **Return to Design View.**

---

**Combining Comparison Operators**

You can combine the comparison operators to further narrow the criteria for the query. As you might expect, these comparison operators can be combined using the logical operators *AND* and *OR*. You are going to combine comparison operators to list all students who entered the school in 2003.

1. To display the records of students who entered the school in 2003, you are going to use the *AND* operator to combine two comparison operators.

**Change the contents of the *Criteria* row of the *DOE* column to: >=1/1/03 and <=12/31/03 (The text will scroll to the left as you type.)**

For a record to be listed, its *DOE* field must meet both of those conditions. The only records that will satisfy both conditions are those of students who entered the school at some point in 2003.

## 2. Run the query.

Notice the results of the query.

	StudentID	LastName	FirstName	DOE	Grade
▶	1004	Ying	Jing	2/1/2003	11
	1010	Paulson	Chivone	9/8/2003	10
	1013	Tsyrlin	Sam	9/8/2003	12
	1015	McCoy	Shauna	9/8/2003	10
	1017	Cormican	Lorraine	2/1/2003	11
	1020	Harrigan	Lydell	9/8/2003	10
	1022	Ridley	Shantae	9/8/2003	10
	1036	Williams	Terrance	9/8/2003	10
	1037	Green	Katrina	2/1/2003	11
	1038	Hilton	Shakima	9/8/2003	10
	1043	Rosich	Melizza	9/8/2003	10
	1044	Reyes	Ednita	2/1/2003	11
	1045	Powell-Abreu	Akira	9/8/2003	10
	1049	Blake	Sha-Qona	9/8/2003	10
	1050	Nieves	Jose	9/8/2003	11
*	0				0

## 3. Return to Design View.

Combining comparison operators with logical operators provides you with another way of designing queries. You are going to design a query to list students who have entered school on 9/8/03 or 9/7/04. Earlier, you would have done this by using the *or* rows of the query design grid. There is another way.

### 1. Change the contents of the *Criteria* row of the *DOE* column to: 9/8/03 or 9/7/04

**Note:** Remember, when you do not specify a comparison operator, the equal operator is assumed.

### 2. Run the query.

Notice the results of the query.

### 3. Close the Query Datasheet without saving changes.

**Tip:** These same techniques work with filters.

## Using Other Operators

Access provides other operators that are related to comparison operators. Although some of these same results can be achieved by combining comparison operators with the logical operators AND and OR, these other operators can be easier to use.

You are going to use these operators on the *StudentIDList* query.

### 1. First, you are going to run the query.

**The Queries tab of the Database window should be selected.**

**2. Double-click the *StudentIDList* query to run it.**

Notice the results of the query. All records are listed because there are no criteria.

**3. Click the View button to return to Design View.**

---

## The Between Operator

The *Between* operator is used to determine if a value lies within a specified range. You are going to use *between* to list the records of students whose *StudentID* is between 1010 and 1025.

**1. Earlier, you used the Zoom window to edit text in your table. You can also use it when designing queries.**

**Click an insertion point in the *Criteria* row of the *StudentID* column.**

**2. Use SHIFT/F2.**

The Zoom Window appears.

**3. In the Zoom Window, type: Between 1010 and 1025**

**Note:** You are not required to capitalize *Between*.

**4. Click OK.**

In the design grid, notice the criteria Access inserts.

**5. Run the query.**

Notice the results. Only records with a *StudentID* from 1010 to 1025 (inclusive) are listed.

---

The *Between* operator can also be used on text data types. You are going to use it to locate students whose last name begins with *F*, *G*, *H*, *I*, or *J*.

**1. Click the View button to return to Design View.**

**2. First, you must delete the current criteria. A quick way to do this is to delete the row.**

**Click an insertion point anywhere in the criteria row.**

**3. On the Edit menu, choose *Delete Rows*.**

The row is removed from the query design grid.

4. You can enter the new criteria now. You are going to make a common “mistake” that you will later correct.

**In the Criteria row of the LastName column, type:**  
Between F and J

As you will see, this condition will not produce the desired results.

5. **Run the query.**

Notice the query results.

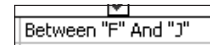
	StudentID	LastName	FirstName	DOE	Grade	HomeRoom
▶	1003	Feliciano	Jose	7/2001	12	362
	1014	Fischer	Colin	6/2002	11	364
	1018	Feliciano	Michelle	6/2002	11	363
	1020	Harrigan	Lydell	8/2003	10	255
	1037	Green	Katrina	1/2003	11	365
	1038	Hilton	Shakima	8/2003	10	255
*	0				0	

Although there are students whose names begin with *F*, there are no students listed with a last name beginning with *J* (even though there are two such records in the table). The reason for this is because when comparing text data types, Access views the text in alphabetical order — just as the words appear in a dictionary. Although you correctly told Access to begin with the letter *F* (the first “word” starting with *F* in the dictionary), you incorrectly told Access to end with *J*. Since this is the first “word” in the dictionary beginning with *J*, however, no last names beyond that are included in the results.

6. You are going to correct the criteria.

**On the View menu, choose Design View.**

In the query design grid, notice that Access used quotation marks to format the condition that you entered.



7. **In the LastName column, change the contents of the Criteria row to:** Between F and K

**Note:** Of course, if there were a student with a last name of *K*, the record would be listed as well.

8. **Run the query.**

Notice the results of the query. Students with last names beginning with *F*, *G*, *H*, *I*, and *J* are listed.

**Tip:** Alternately, you could use the following criteria:  $\geq F$  and  $< K$ . In this case, the record of a student with a last name of K would not be listed.

### 9. Click the View button to return to Design View.

## The Like Operator

The *Like* operator is used to determine if a text value begins with one or more specified characters. It is used in combination with the wildcard characters \* (which replaces any number of characters) or ? (which replaces exactly one character). You are going to use the *like* operator to list students whose last name begins with *Co*.

#### 1. Change the contents of the *Criteria* row of the *LastName* field to: like co\*

Once again, the asterisk (\*) is a substitute for one or more characters. You are telling Access to look for last names whose first two letters are *co*. Any number of characters can follow these letters — capitalization does not matter.

#### 2. Run the query.

Notice the results of the query.

	StudentID	LastName	FirstName	DOE	Grade	HomeRoom
▶	1017	Cormican	Lorraine	1/2003	11	364
	1030	Coleman	James	1/2004	10	255
	1039	Cordero	Eric	7/2004	9	251
*	0				0	

#### 3. Return to Design view.

At times, you might want to see if a particular word is found anywhere in a field — not just at the beginning. (You did this earlier when you used the *Filter By Selection* feature.) You can use the *Like* operator (along with the \* wildcard character) to search for text anywhere in a field. You are going to look for last names that contain the two consecutive letters *co*.

1. If you use a wildcard character, Access will automatically add the word *Like*, just as it adds hash marks to dates.

**Change the contents of the *Criteria* row of the *LastName* column to: \*co\***

You are telling Access that any number of characters can precede or follow the consecutive letters *co*.

**2. Tap the TAB key.**

Notice that Access changes the criteria to *Like "\*co\*"*.

**3. Run the query.**

Notice the last names that contain *co* as consecutive letters. In some cases, no characters preceded these two letters.

	StudentID	LastName	FirstName	DOE	Grade	HomeRoom
▶	1015	McCoy	Shauna	8/2003	10	255
	1017	Cormican	Lorraine	1/2003	11	364
	1026	Becote	Kiya	7/2004	9	252
	1030	Coleman	James	1/2004	10	255
	1039	Cordero	Eric	7/2004	9	251
*	0				0	

**4. Close the Query Datasheet window without saving changes.**

## Notes

If you enter the *Is Null* operator in the query design grid for a particular field, Access will locate records for which the field is empty (or blank).

The *Is Not Null* operator tells Access to test for fields that contain data (i.e., are not empty).

## Review Exercise

To review the concepts presented in this lesson, you are going to create a new query based on the *StudentInfo* table and then modify it in several ways.

- 1. The More Query Database window should be open.**
- 2. On the Insert menu, choose Query and then begin a Simple Query Wizard based on the StudentInfo table.**
- 3. The query should include the following fields (in the order listed): StudentID, LastName, FirstName, City, Grade, DOE, HomeRoom**
- 4. Name the query MyQuery and then run it to notice the listing of all records in the table.**

- 5. Modify the query so that only the records of students in homerooms 250 through 255 are displayed.**

*Tip:* This can be done in two different ways.

- 6. Run the query.**
- 7. Modify the query design so that only the records of students whose homeroom ends in a 0 or a 5 are displayed.**
- 8. Run the query.**
- 9. Modify the query design so that only the records of students whose last names are between the letters D and M are listed. Sort the records by StudentID.**
- 10. Save the changes to the query.**
- 11. Run the query and notice the changes.**

---

*End of Lesson 10*