

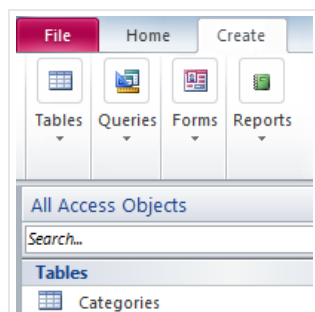
# Access 2010

## Introduction to Objects



Page 1

## Introduction



Databases in Access 2010 are composed of four objects: **tables**, **queries**, **forms**, and **reports**. Together, these objects allow you to enter, store, analyze, and compile your data as you wish.

In this lesson, you will learn about each of the four **objects** and come to understand how they interact with each other to create a fully functional relational database.

Page 2

## The Access objects

Tables, queries, forms, and reports are the framework for any database you create in Access. Understanding how each of these objects works will help you create a database that will be useful and help you retrieve the information you need.

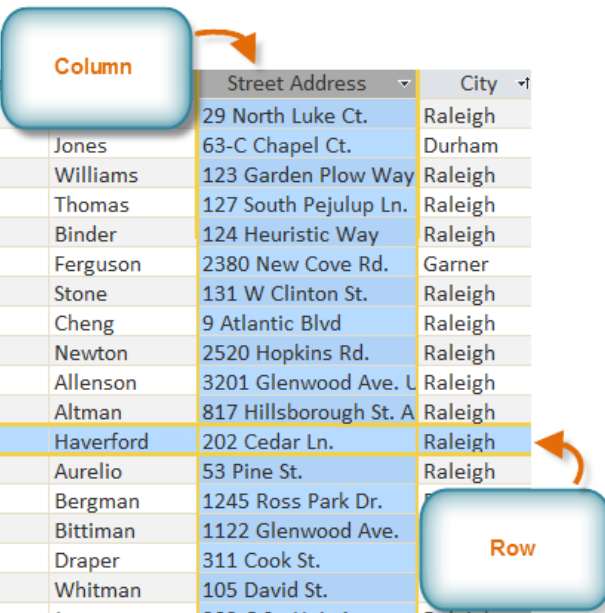
### Video: Introduction to Objects



## Tables


By this point, you should already understand that a database is a collection of data organized into many connected **lists**. In Access, all data is stored in **tables**, which puts tables at the heart of any database.

You might already know that tables are organized into vertical **columns** and horizontal **rows**.



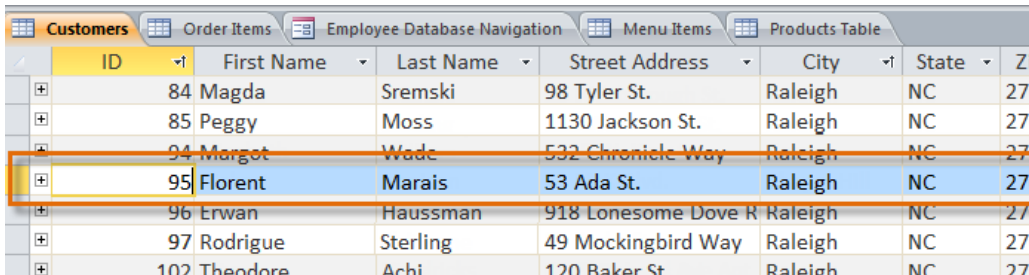
ID	First Name	Last Name	Street Address	City
20	Barbara		29 North Luke Ct.	Raleigh
29	Bob	Jones	63-C Chapel Ct.	Durham
30	Juanita	Williams	123 Garden Plow Way	Raleigh
31	Sara	Thomas	127 South Pejulup Ln.	Raleigh
32	Larry	Binder	124 Heuristic Way	Raleigh
33	Samantha	Ferguson	2380 New Cove Rd.	Garner
34	Jamie	Stone	131 W Clinton St.	Raleigh
35	Patti	Cheng	9 Atlantic Blvd	Raleigh
36	Greg	Newton	2520 Hopkins Rd.	Raleigh
37	Carol	Allenson	3201 Glenwood Ave. W	Raleigh
38	Zoey	Altman	817 Hillsborough St. A	Raleigh
39	Danny	Haverford	202 Cedar Ln.	Raleigh
40	Vig	Aurelio	53 Pine St.	Raleigh
41	Jeffery	Bergman	1245 Ross Park Dr.	
42	William	Bittiman	1122 Glenwood Ave.	
43	Megan	Draper	311 Cook St.	
44	Dick	Whitman	105 David St.	
45	Marjan	Jameson	202 C St. Unit A	Raleigh
46	Colin	Honkins	321 E. Edenton St.	Raleigh

In Access, rows and columns are referred to as **records** and **fields**. A **field** is more than just a column; it's a way of organizing information by the **type** of data it is. Every piece of information within a field is of the same **type**. For example, every entry in a field called "First Name" would be a name, and every entry in field called "Street Address" would be an address.



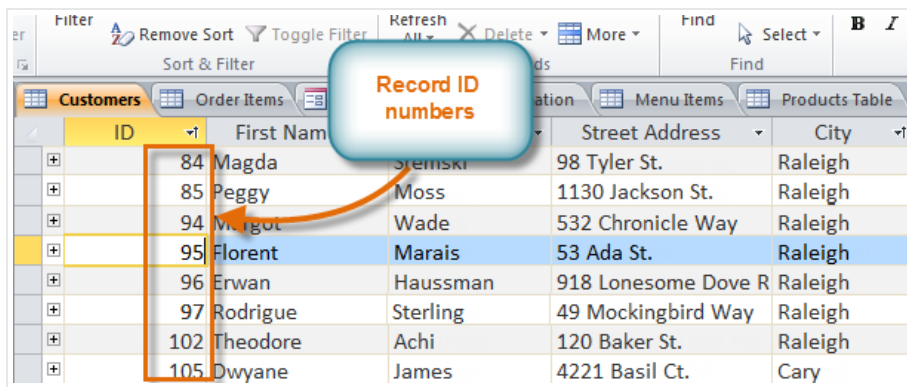
ID	First Name	Last Name	Street Address
67	Joy	Zachman	807 Lee St.
68	Frances	Trenton	901 Kenan Rd.
69	Latavia		McIver Ct.
70	Kurtis		3 Cobb Rd.
71	Lashaunda		1 Hinton St.
72	Lieselotte		2 Spencer Ave.
73	Sula	Smart	56 Dey Rd.
74	Jude	Smith	929 Greenlaw Dr.
75	Katharine	Kellerman	76 Murphy Ave.
76	Ruiari	O'Brien	100 Aycock St.
77	Tyra	Kirby	8700 Stacey Rd.
78	Michiko	Akiwana	901 Glenwood Ave.
79	Betty	Potter	80 Greene St.
80	Elizabeth	Loges	44 Steven Rd.

Likewise, a **record** is more than just a row; it's a unit of information. Every cell in a given row is part of that row's record.



ID	First Name	Last Name	Street Address	City	State	Zip
84	Magda	Sremski	98 Tyler St.	Raleigh	NC	276
85	Peggy	Moss	1130 Jackson St.	Raleigh	NC	276
94	Margot	Wade	532 Chronicle Way	Raleigh	NC	276
95	Florent	Marais	53 Ada St.	Raleigh	NC	276
96	Erwan	Hausman	918 Lonesome Dove R	Raleigh	NC	276
97	Rodrigue	Sterling	49 Mockingbird Way	Raleigh	NC	276
102	Theodore	Achi	120 Baker St.	Raleigh	NC	276

Notice how each record spans several fields. Even though the information in each record is organized into fields, it belongs with the other information in that record. See the **number** at the left of each row? That's the **ID number** that identifies each record. The ID number for a record refers to every piece of information contained in that row.



ID	First Name	Last Name	Street Address	City
84	Magda	Stemski	98 Tyler St.	Raleigh
85	Peggy	Moss	1130 Jackson St.	Raleigh
94	M. Igot	Wade	532 Chronicle Way	Raleigh
95	Florent	Marais	53 Ada St.	Raleigh
96	Erwan	Hausman	918 Lonesome Dove R	Raleigh
97	Rodrigue	Sterling	49 Mockingbird Way	Raleigh
102	Theodore	Achi	120 Baker St.	Raleigh
105	Dwyane	James	4221 Basil Ct.	Cary

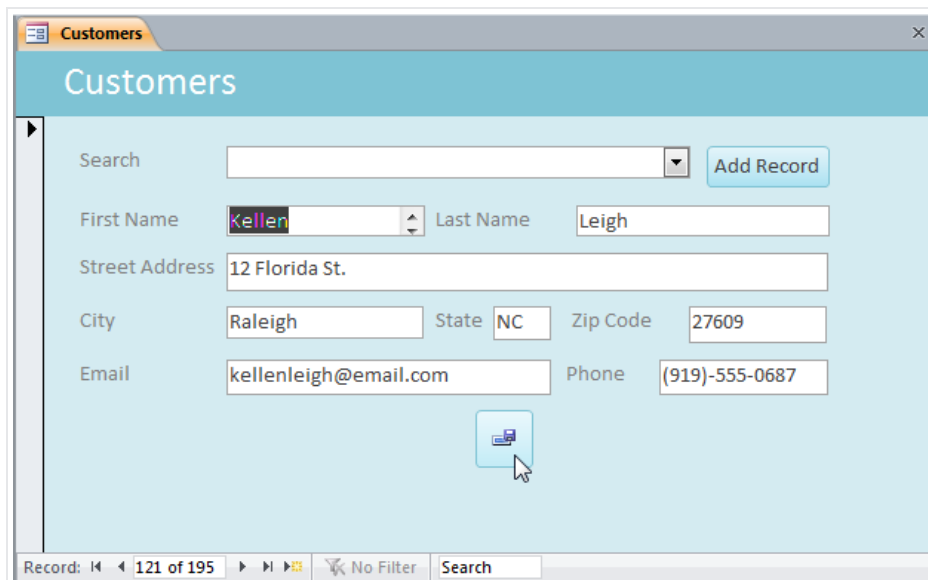
Tables are good for storing **closely related information**. Let's say you own a bakery and have a database that includes a table with your customers' names and information—their phone numbers, home addresses, and email addresses. Since these pieces of information are all details about your customers, you'd include them all in the same **table**. Each customer would be represented by a unique **record**, and each type of information about those customers would be stored in its own field. If you decided to add any more information—say, a customer's birthday—you would simply create a new field within the same table.

## Forms, queries, and reports

Although **tables** store all of your data, the other three objects offer you ways to work with it: **forms**, **queries**, and **reports**. Each of these objects interacts with the **records** stored in your database's tables.

### Forms

**Forms** are used for **entering**, **modifying**, and **viewing** records. You probably have had to fill out forms on many occasions, like when visiting a doctor's office, applying for a job, or registering for school. The reason forms are used so often is that they're an easy way to guide people into entering data correctly. When you enter information into a form in Access, that data goes exactly where the database designer wants it to go—into one or more related tables.

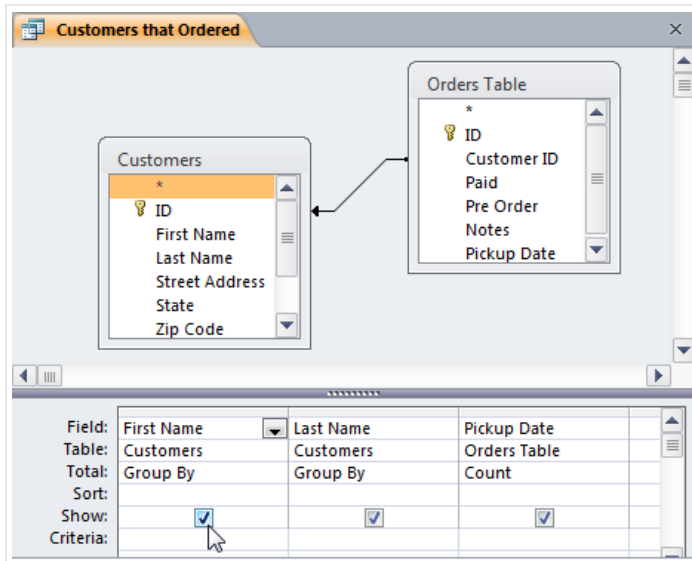


Forms make entering data easier. Working with extensive tables can be confusing, and when you have connected tables you might need to work with more than one at a time to enter a set of data. However, with forms, it's possible to enter data into multiple tables at once, all in one place. Database designers can even set restrictions on individual form components to ensure all of the needed data is entered in the correct format. All in all, forms help keep data consistent and well-organized, which is essential for an accurate and powerful database.

## Queries

Queries are a way of **searching** for and **compiling** data from one or more tables. Running a query is like asking a detailed **question** of your database. When you build a query in Access, you are **defining specific search conditions** to find exactly the data you want.

Queries are far more powerful than the simple searches you might carry out within a table. While a **search** would be able to help you find the name of one customer at your business, you could run a **query** to find the name and phone number of every customer who's made a purchase within the past week. A well-designed query can give information you might not be able to find out just by looking through the data in your tables.



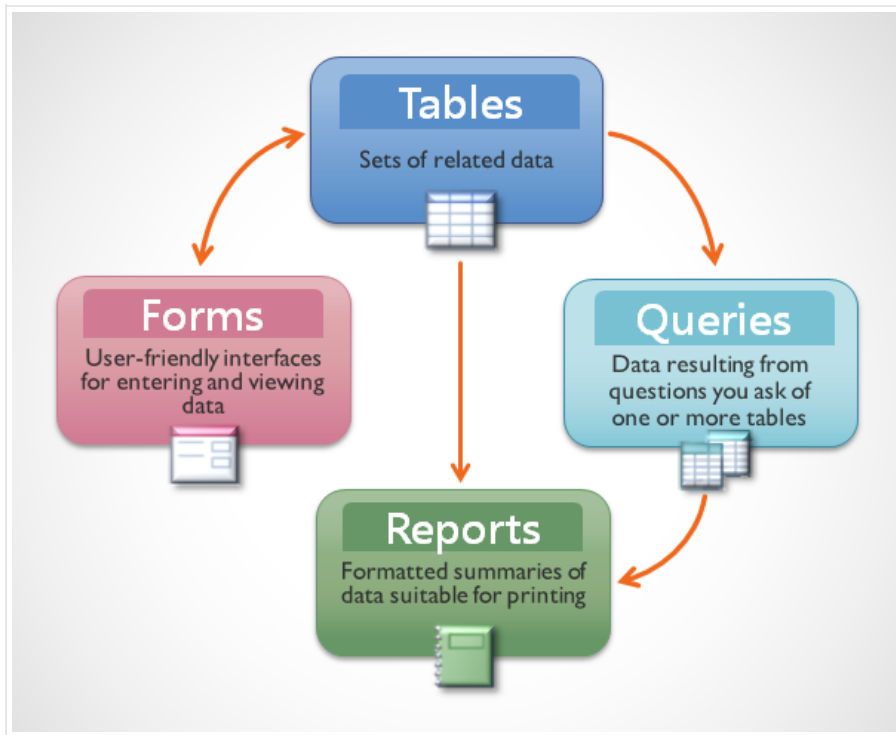
## Reports

**Reports** offer you the ability to **present** your data **in print**. If you've ever received a computer printout of a class schedule or a printed invoice of a purchase, you've seen a database report. Reports are useful because they allow you to present components of your database in an easy-to-read format. You can even customize a report's appearance to make it visually appealing. Access offers you the ability to create a report from any **table** or **query**.



## Putting it all together

Even if you have a good idea of how each object can be used, it can initially be difficult to understand how they all work together. It helps to remember that they all work with the same data. Every piece of data that a **query**, **form**, or **report** uses is stored in one of your database **tables**.

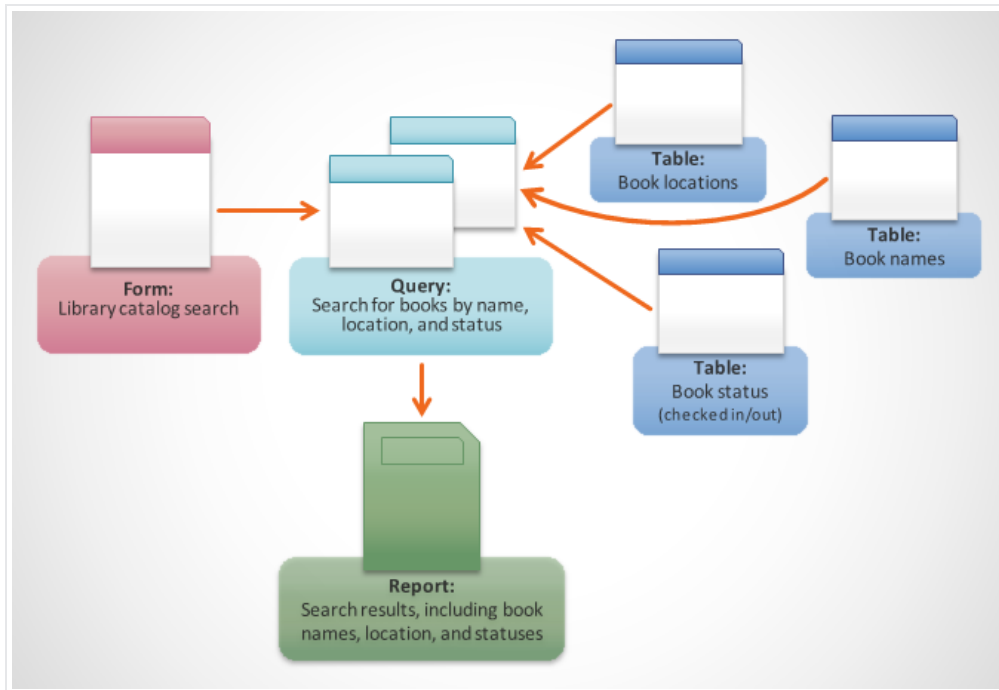


Forms allow you to both **add** data to tables and **view** data that already exists. Reports **present** data from tables and also from queries which, in turn, **search for** and **analyze** data within those same tables.

These relationships sound complicated, but in fact they work together so well and so naturally that we often don't even notice when we're using connected database objects. Have you ever used an electronic card catalog to search for a book at the library? Chances are, you entered your search in something that looks like this:

The screenshot shows a library catalog search interface. At the top, it says "Search the library catalog. Fill in at least one field. The more words you search for, the smaller and more refined your results list will be." Below this is a "Search by:" section with four dropdown menus: "Title Keyword", "Author Keyword", "General Keyword", and "Subject Keyword", each followed by a text input field and an "AND" dropdown. To the right is a "Refine your search (optional):" section. It includes a "Limits" section with two dropdown menus: "Library Branch" (listing Athens Drive Community Library, Cameron Village Regional Library, Cary Public Library, Dureleigh Road Library, and East Regional Library) and "Type of Book" (listing All Audio Books, Downloadable Audio Books, Adult Audio Fiction, Adult Audio Non Fiction, and Children's Audio). Below the limits is a "Sorting" section with a dropdown menu labeled "Select...".

When you performed that search, you were entering your search terms into a **form** that, in turn, created and ran a **query** based on your request. When the query finished searching the database's **tables** for records that matched your search, you were shown a **report** that drew information from the query and the related tables—in this case, a list of books matching your search terms. You could represent the connections between the objects like this:



Imagine that instead of using these tools you had to search within a giant table containing every book in the library system. The relevant records would likely be spread out across many tables: a table for book titles and descriptions, a table containing information about which books are checked in or out, and a table with each branch of the library, just to name a few.

You'd have to search at least three tables just to find a book, learn its location, and see whether it's checked in! It's easy to imagine how difficult it could become to find the right book. If you weren't careful, you might even mess something up by accidentally deleting or editing a record. It's easy to see how the database objects make this search much more manageable.

In **Lesson 1**, we discussed the concept of a **relational database**, which is a database that is able to understand how different sets of data **relate** to one another. Situations like the example above are exactly why people find relational databases so useful. Without a relational database, what should be a simple task—searching for a book and seeing if it's checked in and where—becomes incredibly complicated and time consuming. Knowing how to use the four Access objects can make even complicated tasks fairly user friendly.