Databases in Microsoft Access

David M. Marcovitz, Ph.D.

Introduction

Schools have been using integrated programs, such as Microsoft Works and Claris/AppleWorks, for many years to fulfill word processing, spreadsheet, and database needs. Some schools are still using these programs, but many are switching to Microsoft Office products. For word processing and spreadsheets, the transition is fairly easy: Microsoft Word and Excel work just like the word processors and spreadsheets of the integrated packages; i.e., the functionality is similar, in which menu each function is located is all that changes.

Unfortunately, databases are a different story. Microsoft Access behaves nothing like Claris/AppleWorks. While many database functions can be done in a spreadsheet (especially with some of the newer features of Microsoft Excel), some people might still prefer to use a database to perform searches, sorts, and reports.

This article introduces the basic features of Access 2000 with an example that you might be able to use in your classroom. If you are using a different version of Access, don't worry: Access 95, 97, 2000, and 2002 all work in a similar way. You should be able to follow everything; just keep in mind that the screen pictures in this article might be a little different from the screen pictures on your computer. If you are using a Macintosh, you are out of luck. Microsoft has not made a version of Access for the Mac.

What Is a Database?

A database is a collection of related information. The information is stored as a series of records. Each record is all the information stored about one item (such as a person, an item in a store's inventory, a student, an employee, a patient, an animal in the rainforest, etc.) A database program helps to organize this information and lets you manipulate it to answer questions.

The kinds of information stored in a database are known as fields. For example think about the kind of information you might want to know about your students: name, address, phone number, ID number, math group, reading group, grade on test 1, grade on test 2, etc. Each of these kinds of information could be a field in the database. It is often helpful to picture a database as a table with each row representing a record and each column representing a field.

The information in a record is tied together because it is all about one item (a student in this case). If a record gets deleted, all the information for that record gets deleted. If a record gets moved to another location, all the information for that record gets moved.

The information in a field is also tied together because it is all the same kind of information. Everything in the birthday field is a date, for example. Fields can be of different types (exactly

which types are available varies based on which database program you are using). In Access, some of the common types of fields you might use are: text, number, date/time, and currency.

Fields in a database can be formatted to look nice, and you can base calculations on fields. For example, you might want to base a final grade on the grades of assignments and tests. While this is a powerful function of databases, the same thing can be done just as easily with a spreadsheet.

The two primary things you will do with records in a database are sorting and selecting. You can sort your database in any order based on any fields (e.g., alphabetical order, highest grade first, etc.), and you can select particular records based on any criteria (e.g., choose only those students with an average higher than 90%). While some newer spreadsheets now offer features that can do some of this, databases are made for sorting and selecting. The advantage of sorting and selecting in a database is that these operations do not affect the underlying data. A database keeps all your underlying data in a table, and you can choose to look at in a number of different ways. In a spreadsheet, the spreadsheet is the table. When you sort, you have changed the order of the spreadsheet. If you accidentally sort a column of your spreadsheet (instead of the entire table), that column will be out of order from the rest of your data. Imagine sorting the grades for your final exam without sorting the rest of the information along with them. The students who appear first in your table would receive the highest grades regardless of how they really did This cannot happen in a database.

Once you get more advanced, you might choose to use forms to enter data and reports to output data in an organized way. These features are not available in spreadsheets. Forms allow you to create a screen layout with boxes that get filled in (just like filling out a form on paper). This is particularly useful if others are going to be entering the data. For example, students registering for a class might come to a computer and enter the registration information.

Reports allow you to print out information in any format you want. Sometimes tables are nice, but reports allow for data to be organized in other ways. For example, you might choose to give your students a mid-term update on their grades with grades on all their assignments listed, the current average in big bold letters, and a place for parents to sign. While these advanced features are not covered in this article, they are not too difficult, and they are beyond the power of spreadsheets.

Relational Databases

Microsoft Access is a very powerful relational database tool. Hopefully, this document will guide you through some of the basics so you will be able to perform many of the simple tasks you might want to do with a database. With this document and a little practice, you should be able to understand how to do many basic things. That will be 5% (at best) of what Access can do. If you want to be able to take full advantage of all the power of Access, you might spend some time with a good Access book, such as *Sams' Teach Yourself Microsoft Access 2000 in 21 Days* by Paul Cassel.

Access is a professional-level tool used by people who maintain large, complex databases. Your school district, for example, probably uses a tool comparable to Access to keep track of student

and personnel records. However, Access is also good at maintaining small databases, such as your class gradebook or your students' collection of rocks and minerals.

In your school, you could use a relational database to keep track of scheduling for all students. You would have a table that lists information about each student. You would have a table that lists information about each class. You would have a table that lists information about each teacher. You would have a table that lists information about each section of a class being scheduled. For example, a student would have a name, ID number, address, other personal information, and a list of all classes taken. A class would have a number, a title, a number of credits, a description, and a list of prerequisites. When you want to schedule a particular section, you would include a teacher, a time, and a room number, and eventually, you would add a list of students taking the class.

This is much more powerful than having separate databases for each table. In a simple database, a student would look at the list of classes and add that to his/her schedule. Then, the roster for the class would have to be updated to reflect that the student is in that class. The students' database has no way of checking if the selected class even exists. In a relational database, the student selects from classes that are in the available course selections. A selection of a class, automatically updates the information for the student and for the class. When it comes time to print a student schedule, all the information is automatically there and correct; there is no need to cross-reference the schedule with lists of teachers (to print the teachers' names next to each class) or room numbers or course titles. All the different kinds of information are tied together.

This is much more power than the average classroom teacher needs, so the rest of this article will stick to simple databases that are easy enough for any teacher to master and powerful enough that you'll never think about manipulating data in the same way again.

Starting a New Database

When you first open Access, you should tell Access you want to create a new database. It will ask you to name the database. Be sure to save the database in a reasonable place, possibly on your own disk by paying attention to the Save In box at the top of the screen. Access databases end with the extension .mdb, so you can include that extension when you name your file, or you can let access automatically add that to your file name for you. Since the database in the example in this article will be a sample food nutrition file, you can name your database nutrition.mdb.

Tables

Access stores data in tables. A simple database program will only allow you to have one table for all your information in the database. Access will allow you to create several tables and relate them to one another. This can be very powerful, but it is an advanced feature. In this article, we will limit ourselves to one table.

When you open Access, you will see the following screen:

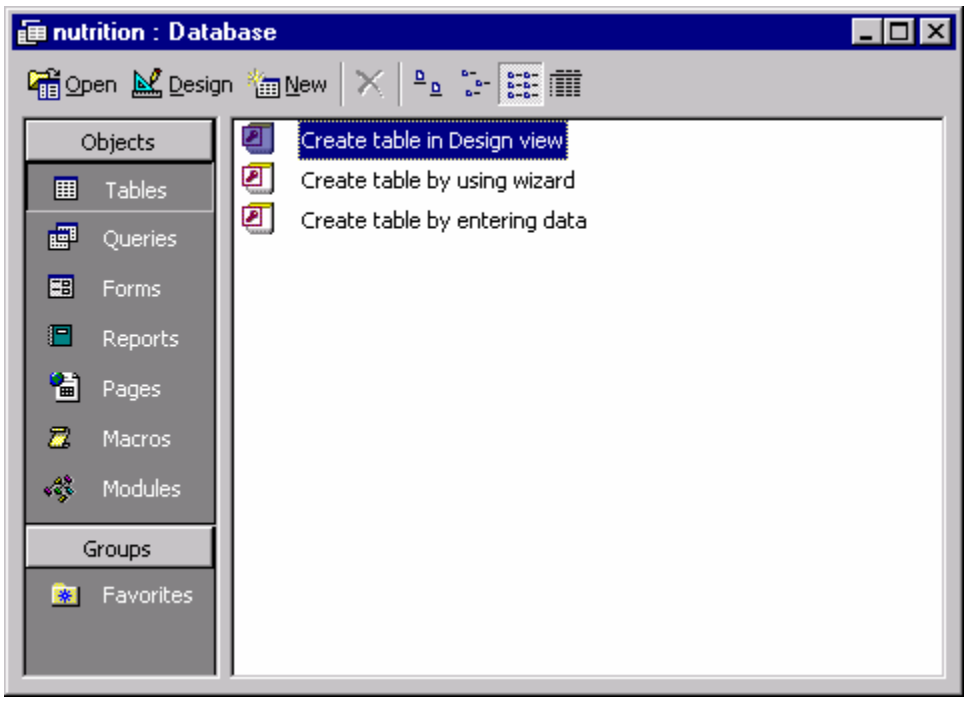


Figure 1. Main View of the Empty Database

The choices on the left allow you to look at any of the tables, queries, forms, reports, macros, and modules that you have created for this database. Notice that Table is selected on the left, and in the main window, there are three icons for "Create table…." Note that in earlier versions of Access, the main window is blank. Other than the three options for creating a table, nothing is in the window because we haven't created a table yet. Click on the New button, choose Design View, and click OK. You should be presented with the following window:

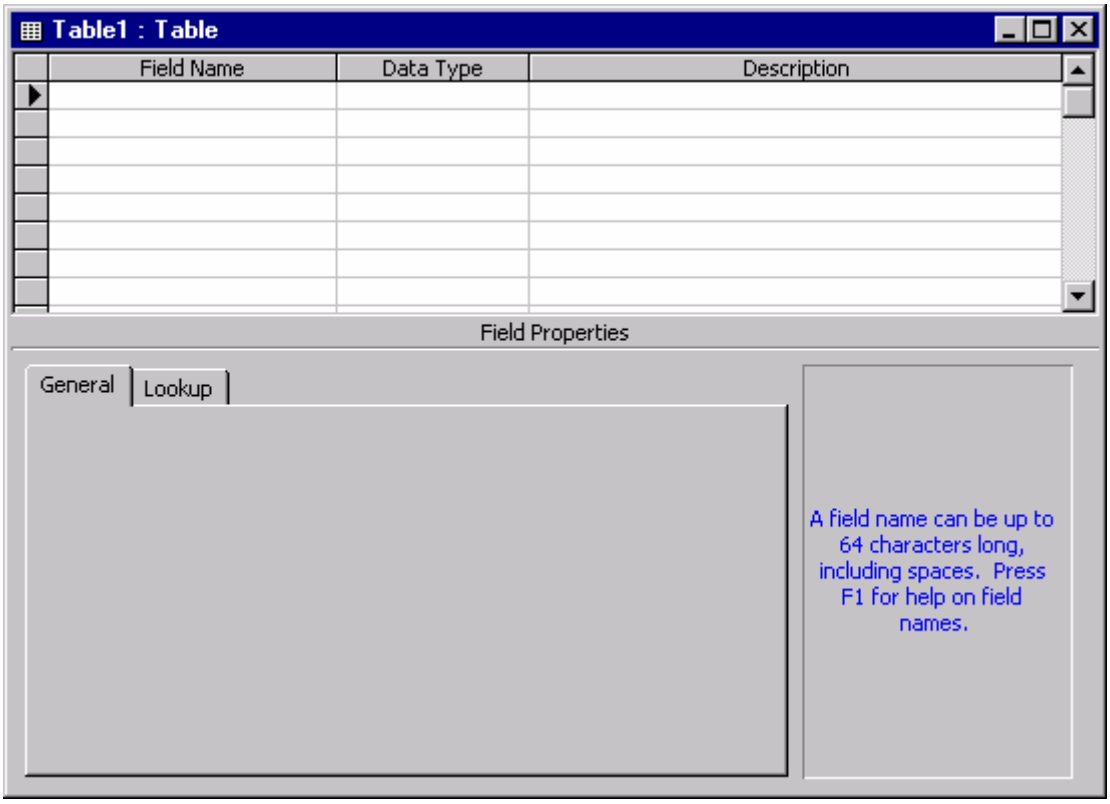


Figure 2. Setting Up Fields in a Table

This is the window where you define your fields. Fields are the kinds of information you want to store. For example, if your database is students, you might want to store each student's name, birthday, and grades, so you would want to create the following fields: First Name, Last Name, Birthday, Grade1, Grade2, and Grade3. We are going to make a database of foods and their nutritional values, so we will want to store: Food Name, Weight, Common Measure, Fat, Calories, Protein, Calcium, and Vitamin C. Of course, this is just an example, and you can find much more nutritional information for your database from the United States Department of Agriculture nutrient database Web site (http://www.nal.usda.gov/fnic/foodcomp/Data/). Use discretion when having students search this database; it is rather extensive and includes data on a wide variety of foods and beverages, including alcoholic beverages.

Next you have to decide what the data type will be for each of these fields. The most common field types are: Text, Number, and Date/Time The Food Name and Common Measure fields will be of type Text, and the rest of the fields will be of type Number.

In the window above, there is one column for the Field Names, one column for the Data Type, and one column for a description. The description is optional, but you might want to describe each nutrient, especially what units are being used. If you click in the first row under Field Name, you will be able to type the first field name: Food Name. You can hit Enter or Tab or click in the box next to it to set the Data Type. When the Data Type area is active, you will see an arrow indicating that you can use the pull-down menu to select the Data Type. Click once on

the arrow and select Text from the list. Hit Enter or Tab or click in the next box, and your screen should look like this:

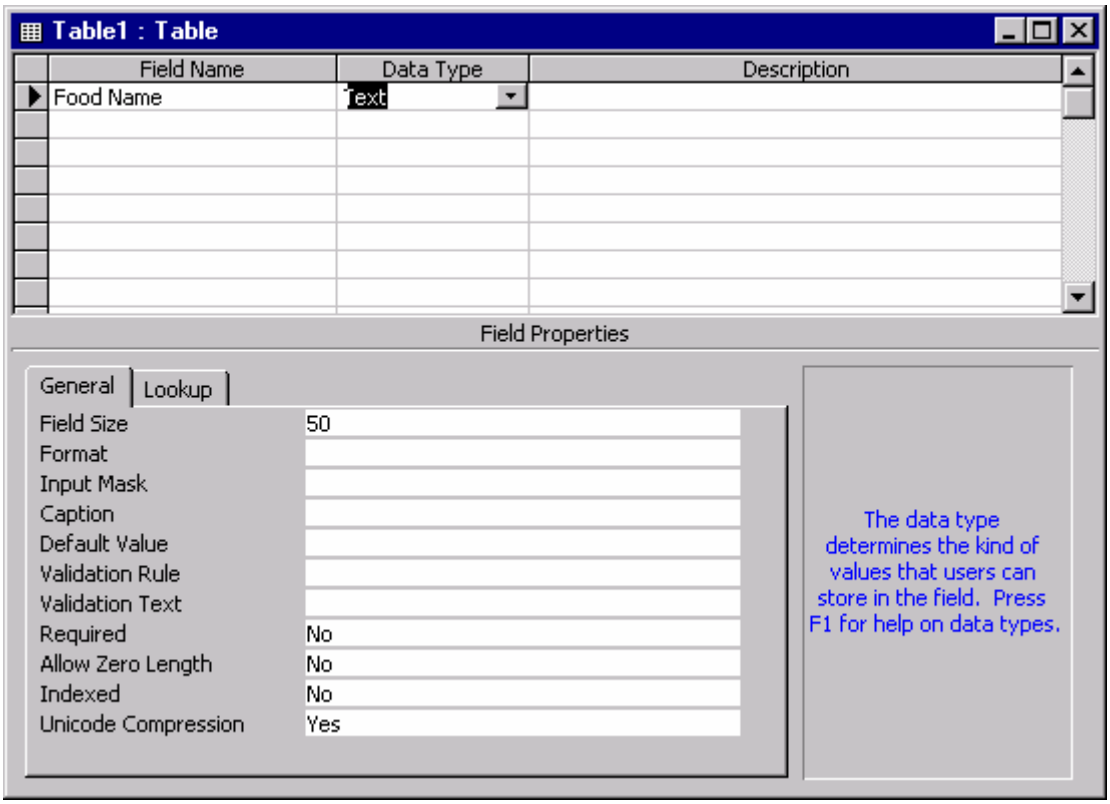


Figure 3. Setting a Field's Data Type to Text

The bottom half of the window will change to indicate Field Properties that are appropriate for a text field. For the most part, you can ignore these; they are for advanced users. For Text fields, the only property that interests us is the Field Size property. This tells us how many characters (letters, numbers, spaces, punctuation marks), the field can hold. You want to be sure that this number is big enough to hold the largest thing you might put in that field. The field size is probably set to 50, which should be plenty of space to hold long food names, so leave it alone.

For numbers, we need to pay attention to the properties. We all know from math class that there are different kinds of numbers: integers, whole numbers, real numbers, rational numbers, etc. Access also has different kinds of numbers. Unfortunately, they are not the same as our math-class numbers. To keep things simple, we will use two types of numbers: Long Integer and Double. Ignore everything else. Long Integer will be used for any integer (any number that doesn't have anything after the decimal point). Double will be used for real numbers (any number that has something after the decimal point). You must set the number type properly before entering any data. If you leave numbers as Long Integers, Access will round the numbers, losing some precision. To change what type of number, we need to change the Field Size property. Since all our number fields will be real numbers, we want to change them all to Double. Click on any field name at the top of the window (or do this right after you enter the data type of Number), and click on the Field Size property at the bottom of the window. Either

type Double or choose Double from the list (you get the list by clicking on the triangle that shows up when you click in the box next to Field Size).

You can type a description if you want to and then get your cursor (by hitting Enter, Tab, or clicking) into the second row under Field Name. You are now ready to enter the rest of the field names and types. When you are done, your screen should look like this:

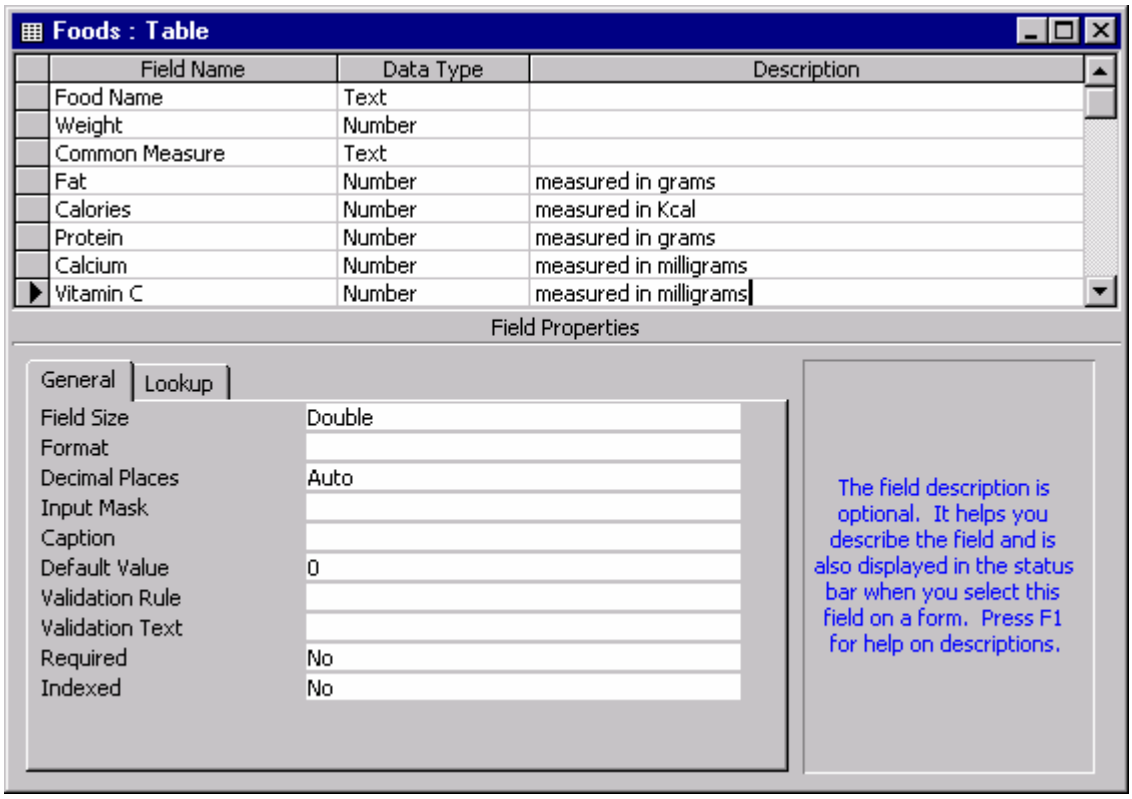


Figure 4. All Fields for the Foods Table

If you made a mistake in typing a name or setting a Data Type, simply click in the box where you made the mistake and fix your mistake.

There are two views for tables. That is, you can look at tables in two different ways. You can look at tables in Design View, which is the way we have been looking at our table. This view, as the name suggests, is for designing the table; i.e., setting up the fields we want in the table. The second view is Datasheet View. This is the view you can use to look at and enter the data for the table. We will switch to Datasheet View to enter our data. To switch to Datasheet View, select Datasheet View from the View menu.

When you do this, Access will ask you if you want to save the table. Click Yes to save the table and give it a name that makes sense to you, such as Foods. In Access, each element of the database has its own name, so every time you create a table or a query or a report, you will have to name it. Always pick a name that makes sense to you, so you will be able to refer back to it later.

Next, Access is going to complain that you haven't assigned a "Primary Key." A primary key is important in large databases with multiple tables. The primary key is a field that contains something unique for each record in the table. For small, single-table databases, a primary key is not necessary. Click on No to tell Access that you do not want a primary key. Note that if you accidentally clicked Yes, don't worry. Access will just create an extra field, which you can ignore.

Now, you should be in Datasheet View. This is a table with rows for your fields and columns for your records. It is the easiest place for the Access beginner to enter data (when you get to be an advanced Access user, you will probably create Forms for entering your data). Your screen should now look like this:

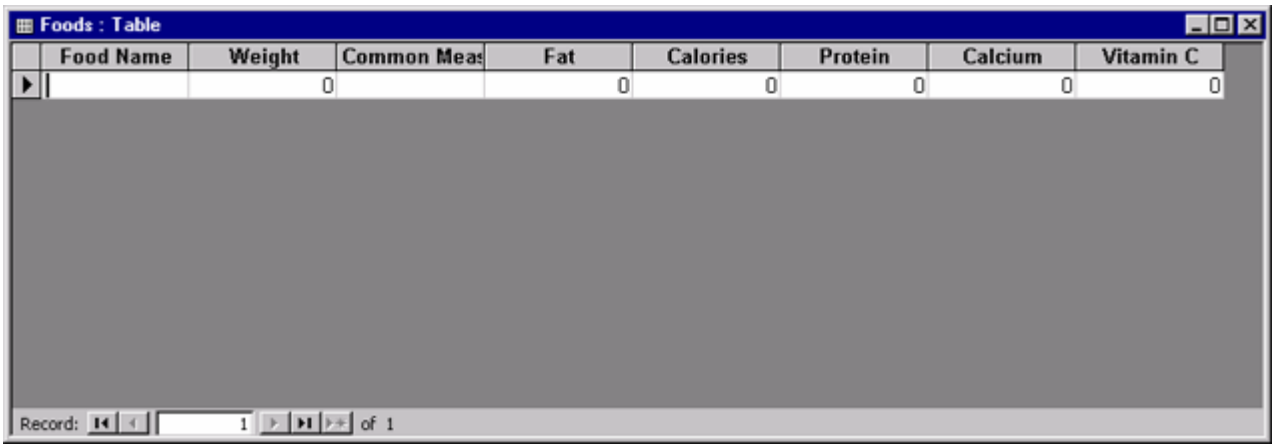


Figure 5. Ready to Type Data in Datasheet View

Notice that all of your field names are across the top of the window as headers for each column. Notice that at the bottom of the screen is a small control panel labeled "Record." This control panel allows you to move from record to record and tells you, in the box, which record you are currently editing. Finally, notice that the fields Weight, Fat, Calories, Protein, Calcium, and Vitamin C already have values of 0. That is because Access automatically sets a default value of 0 for numbers; the default value is the number that will show up when no number has been entered. This can be convenient or it can be a problem.

In some databases, such as a gradebook, you might not want a value of 0 for grades that you have not entered yet. If you want to eliminate the default value, go back to Design View (by choosing it from the View menu), click on a field name, and delete the 0 from the Default Value property (in the property list at the bottom of the window). Repeat for any numbers for which you don't want a default value, and switch back to Datasheet View.

You can now start entering your data. Click in the first row under Food Name and type egg. Hit Enter or Tab or click in the next box and type 33.4. Hit Enter or Tab or click in the next box and type 1 large egg. Type 0, 16.7, 3.514, 2.004, and 0 for Fat, Calories, Protein, Calcium, and

Vitamin C. When you hit Enter or Tab at the end of the first record, Access will automatically take you to the second record. Keep typing the data until your table looks like Figure 6.

Foods : Table

Food Name	Weight	Common Measure	Fat	Calories	Protein	Calcium	Vitamin C
egg	33.4	1 large egg	0.000	16.70	3.514	2.004	0.000
orange	131.0	1 orange	0.157	61.57	1.231	52.400	69.692
orange juice	248.0	1 cup	0.496	111.60	1.736	27.280	124.000
hamburger (fast food)	106.0	1 sandwich	9.773	272.42	12.317	126.140	2.226
fried chicken (fast food, white meat)	163.0	2 pieces	29.519	493.89	35.713	60.310	0.000
milk (whole)	244.0	1 cup	8.150	148.84	8.028	290.360	2.196
milk (2%)	244.0	1 cup	4.685	122.00	8.125	297.680	2.440
apple	138.0	1 apple	0.497	81.42	0.262	9.660	7.866
broccoli	88.0	1 cup	0.308	24.64	2.622	42.240	82.016
chocolate chip cookie	10.0	1 medium cookie	2.260	48.10	0.540	2.500	0.000
	0.0		0.000	0.00	0.000	0.000	0.000

Record: 1 of 10

Figure 6: Complete Foods Table

If you made a mistake, you can always click in the box where you made the mistake and correct it.

You might notice that as you type, some of your columns are not wide enough to fit everything while others take up too much space. You can fix this by moving the mouse cursor after the name of the field you want to make wider or narrower. When the cursor is right between the field name and the next field name, it will change to a vertical bar with arrows pointing to the right and the left. When the cursor takes this shape, hold down the left mouse button and drag the line to the left to make the field narrower or to the right to make it wider. Alternatively, you can double-click the left mouse button and Access will make it just the right width to fit all your data.

You also might notice that Access puts as many decimal places as needed on your numbers. This is fine, but it will probably be easier to read if you set your numbers to fixed decimal places. To do this, switch back to Design View. Click on a number field, such as Weight. Since the weights we have entered all have one or zero decimal places, we will set that field to one decimal place. Under properties, change Format to "Fixed" and Decimal Places to "1" as in Figure 7. Change the other numbers to "Fixed" with 2 or 3 Decimal Places.

Unlike the Field Size (which was Long Integer or Double), the Format and Decimal Places can be changed later. Field Size tells the computer how the number is *stored*, so if you don't tell it to store the field in a way that can hold real numbers, the computer won't remember what is after the decimal place. Format and Decimal Places tell the computer how the number is *displayed*, so if you display one decimal place now, you can always display more or fewer later.

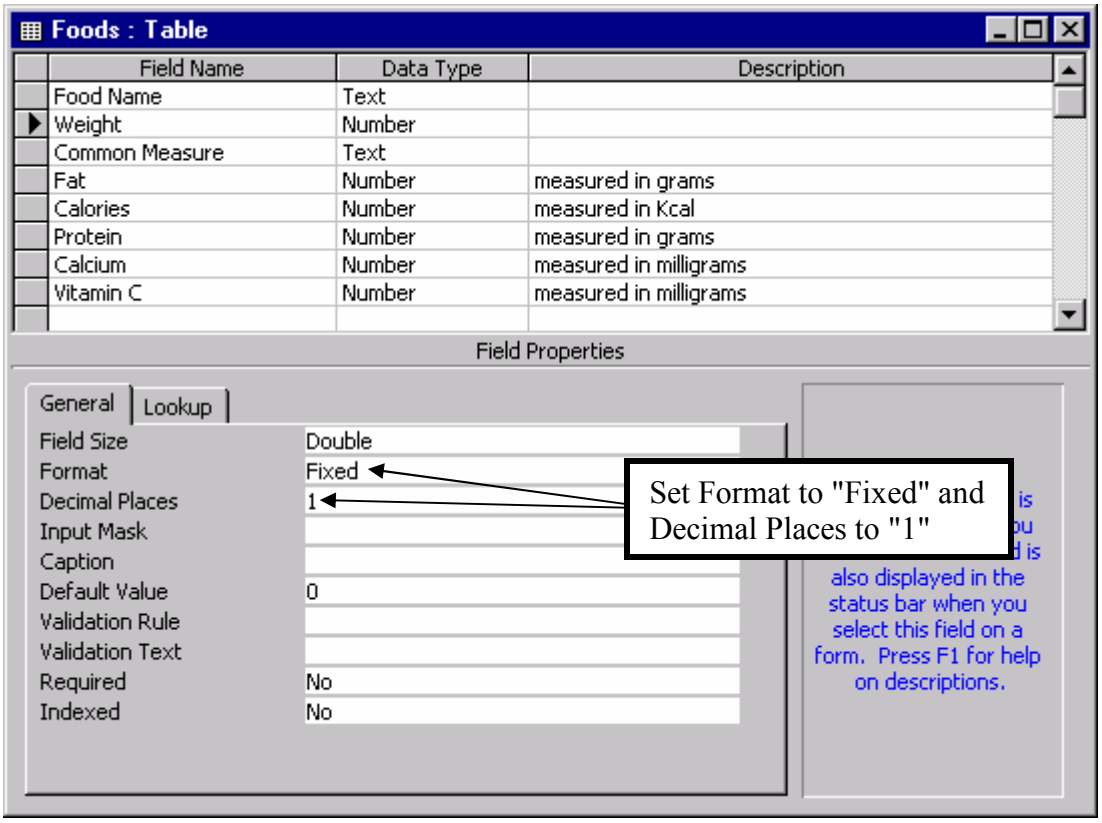


Figure 7. Fixed Decimal Places

You have now entered your data and made it look pretty. You can always come back to add more data or to change data. For now, we will leave the table. When you are done with a table, in Datasheet view or Design view, you can close it by clicking on the close box in the upper right hand corner of the window (that's the X). You will probably be asked if you want to Save changes; click Yes to save changes. Access saves the changes to the data as you type, so when you save, you are saving the format of the table.

You should now be back in the main view of your database, where you can choose to look at the Tables, Forms, Queries, etc. (see Figure 8).

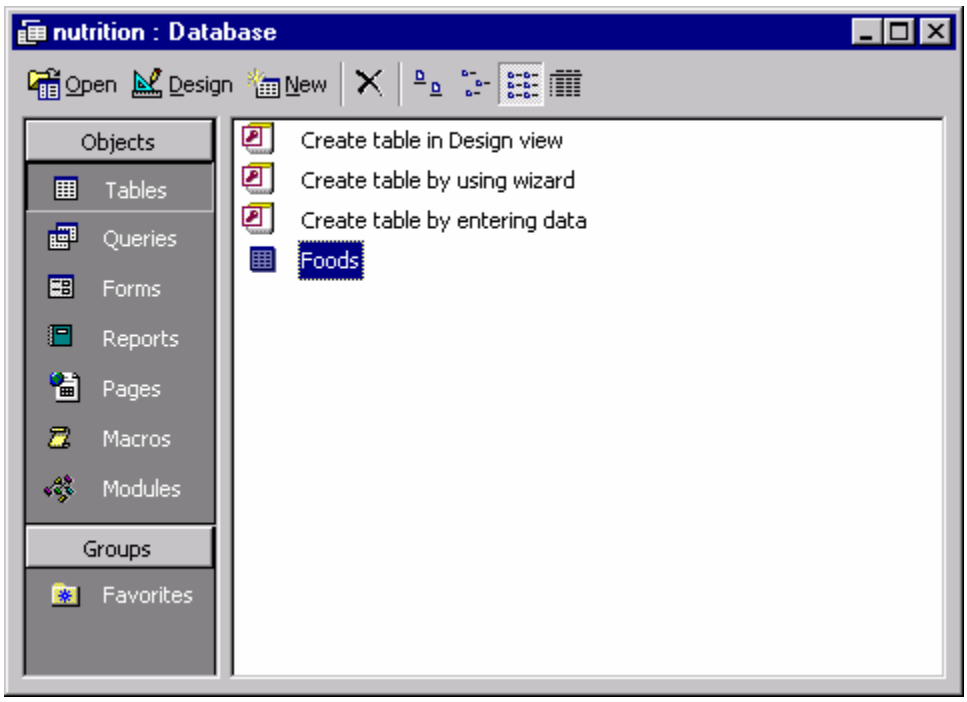


Figure 8. Main View With Foods Table

Notice that if you click the Tables button, you will see the Foods table that we just created. If you ever want to make changes to the Foods table (including adding more foods, changing the information for some foods, or changing some of the fields), you can do that by clicking on the Foods table and clicking on the Open button. Remember that if you want to change the data, you need to be in Datasheet View, and if you want to change the field names, types, or properties, you need to be in Design View.

Queries: Ask Some Interesting Questions

Now, you have entered some data into the database, but you haven't done anything that wouldn't have been just as easy in a word processor. The power of the database comes when you start using it to ask questions (queries). The first kind of question we want to ask is about each record in our database. We can do anything we want with our data, but we'll start with something simple. If we want to know pound for pound (or in this case, gram for gram) which food has the most protein, we will need to know how much protein per gram each food has. We will need to divide our Protein field by our Weight field.

If you are in Design View or Datasheet View for the table, click on the close box to get back to the screen in Figure 8. Click on the Query button. You should see a window with, at most, a couple of options for creating queries but no actual queries. Click on New to create a new query. Click on Design View and OK to begin creating your query in Design View. You will notice that queries have Design View and Datasheet View, just like tables, and Datasheet View looks almost identical for queries and tables.

Next, we need to tell Access that we want to get our data from the Foods table that we just created. A Show Table window pops up and lists the Foods table. Click on the Foods table to select it and then click the Add button. Close the Show Table window by clicking on the Close box. Your screen should look like Figure 9.

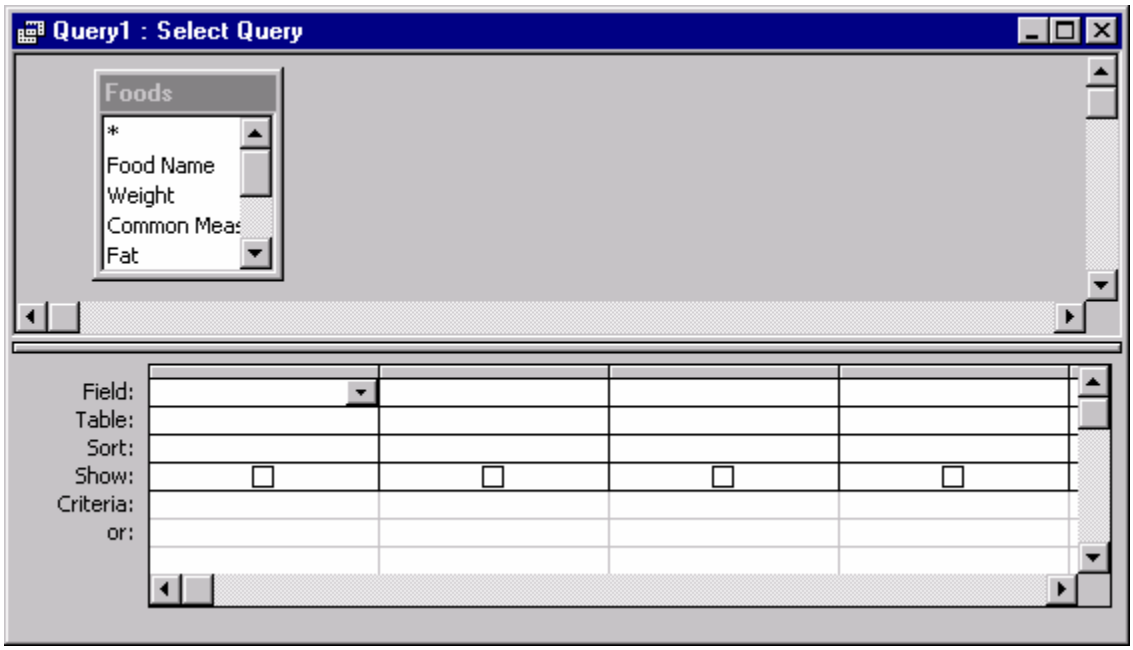


Figure 9: Setting Up a Query

At the top of the window you see a scroll box with all the fields from the Foods table listed. At the bottom of the screen you see a table. Drag each of the fields from the list of fields at the top to a separate column in the table at the bottom. You can use the scroll bar at the bottom to scroll over and see more columns when you use up all the columns that you can see. When you are finished, your screen should look like Figure 10.

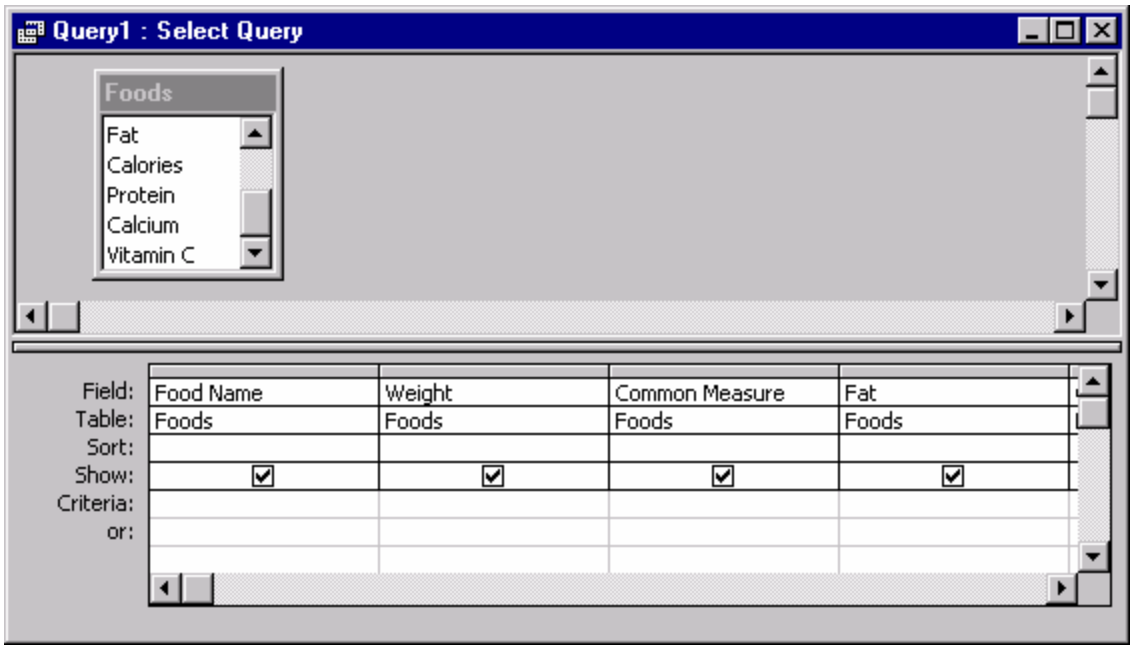


Figure 10. Adding Fields to the Query

Each column has the field name and the table from which that field comes. In our case, all the fields come from the Foods table.

You have now set up your first query. In the View menu, choose Datasheet View. Your query should look just like the Datasheet View of the table. That's because the question you asked was simply to show all the fields in the Foods table. Select Design View from the View menu to take you back to the query so we can ask a more interesting question.

Calculations

Access has several ways to perform calculations. When you get more advanced, you might want to perform some calculations in Reports, but we are going to start by performing calculations in a Query. The advantage of performing calculations in a Query is that you can easily base other Queries on your calculation, and if you want fancy output from a Report, you can base a Report on the Query as well.

In Design View of the query, scroll to the right until you can see the first empty column. Click the right mouse button in the Field box and select Zoom from the menu. This will bring up a larger window to make it easier to type our long expression. In the larger window, type the following

Protein Per Gram: [Protein]/[Weight]

Protein Per Gram is going to be the name of the column (it acts like a field name). What comes after the colon is a formula to calculate the protein per gram. This formula could be any

mathematical expression, including ones that use some powerful mathematical functions that come with Access. Your screen should look like Figure 11.

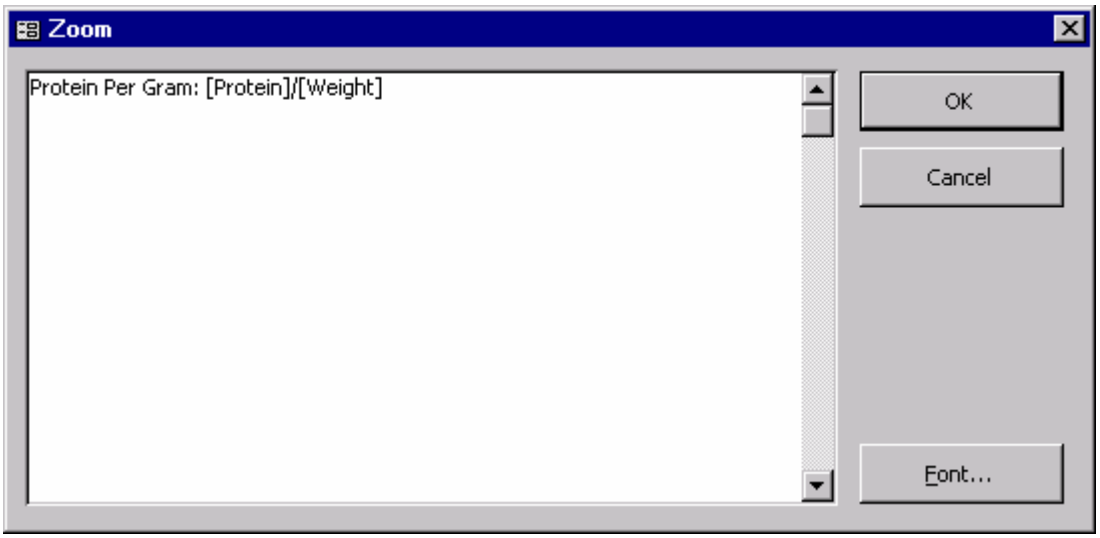


Figure 11: Zoom for Typing Formulas

Make sure you type your field names (Protein and Weight) exactly as you typed them when you created the fields. Access is picky about misspellings. Also, be sure your field names are in square brackets because that tells Access that they are field names. If you got the formula right, click OK. Now run your query again by picking Datasheet View from the View menu.

Your query should look the same as the first time you ran it, except that you have an extra column. The column is titled Protein Per Gram, and it contains the results of our calculation for each of our foods. Your query should look like Figure 12.

Protein Calculation : Select Query

Food Name	Weight	Common Measure	Fat	Calories	Protein	Calcium	Vitamin C	Protein Per Gram
egg	33.4	1 large egg	0.000	16.70	3.514	2.004	0.000	0.105209580838323
orange	131.0	1 orange	0.157	61.57	1.231	52.400	69.692	0.009396946564886
orange juice	248.0	1 cup	0.496	111.60	1.736	27.280	124.000	0.007
hamburger (fast food)	106.0	1 sandwich	9.773	272.42	12.317	126.140	2.226	0.116198113207547
fried chicken (fast food, white meat)	163.0	2 pieces	29.519	493.89	35.713	60.310	0.000	0.219098159509202
milk (whole)	244.0	1 cup	8.150	148.84	8.028	290.360	2.196	0.032901639344262
milk (2%)	244.0	1 cup	4.685	122.00	8.125	297.680	2.440	0.033299180327869
apple	138.0	1 apple	0.497	81.42	0.262	9.660	7.866	0.001896550724638
broccoli	88.0	1 cup	0.308	24.64	2.622	42.240	82.016	0.029795454545455
chocolate chip cookie	10.0	1 medium cookie	2.260	48.10	0.540	2.500	0.000	0.054
	0.0		0.000	0.00	0.000	0.000	0.000	

Record: 1 of 10

Figure 12. Query Results From Calculation

There is one more thing that we want to do with this query. We want to limit the number of decimal places that are shown in the Protein Per Gram field. This is similar to what we did to fix decimal places in the table, but it is slightly different for a query. Go back into Design View (by choosing Design View from the View menu). Click the right mouse button on the Protein Per

Gram column and choose Properties from the menu that pops up. Click in the boxes for Format and Decimal Places and change them to Fixed and 2 so your window looks like Figure 13.

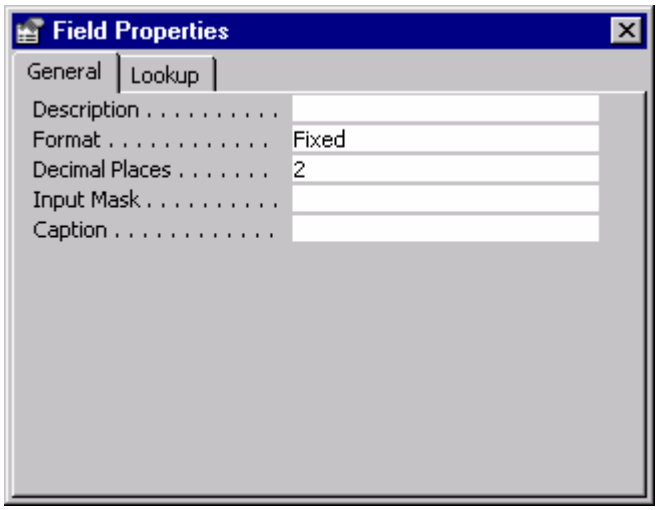


Figure 13. Field Properties for a Calculated Field

Click on the close box of the window and switch back to Datasheet view. It should look exactly the same as the last time except that all the Protein Per Gram entries are showing two digits after the decimal point.

Click in the close box of the window and click Yes when asked if you want to save the query. Give the query a name that makes sense, such as "Protein Calculation." You should now see "Protein Calculation" listed as a Query in the database. If you want to check to see that your table is still there, click on the Table button, and you will see Foods listed there.

Something to Try

Now that you have added one calculated field, here is something to try. You can create a new query or add this to the query you just made. Create a nutrition index by adding together Vitamin C, Calcium, and Protein, and subtracting Calories, and Fat. The higher number would be the healthier food. Of course, this would not be a realistic nutrition index, so you could adjust these numbers by multiplying Fat by 10 and dividing Calcium by 5 or by multiplying other numbers by some factor. Note also, that if you really use this in your class, you will probably include many other nutritional elements. Have your class decide which are the healthiest foods by creating a formula to represent all the nutritional elements in your database.

Sorting

When you think about the records in your database, you should not think of them as existing in any particular order. Right now, they are in the order in which you typed them, but that might not

be the most useful order. Sorting allows you to put your records in any order you choose. For a database of ten foods, this might not be particularly useful, but imagine you entered hundreds or thousands of foods. Suddenly, automatic sorting becomes very useful.

Access provides many ways to sort a document. The simplest way to sort is in Datasheet View. Simply click on the field on which you want to base the sort and go to the Records menu, choose Sort, and choose either Sort Ascending or Sort Descending. That is, if you want your records in alphabetical order, go to Datasheet View of your table or query, click on the Food Name field and choose your sort from the menu. Sort Ascending will sort from lowest to highest (or A to Z, which is regular alphabetical order), and Sort Descending will sort from highest to lowest (or Z to A, which is reverse alphabetical order).

To answer our earlier question—pound for pound, which food contains the most protein—we can simply sort based on our Protein Per Gram field. While looking at the query in Datasheet View, right click on Protein Per Gram and choose Sort Descending (see Figure 14) to find out that fried chicken has the most protein per gram of the foods in our database.

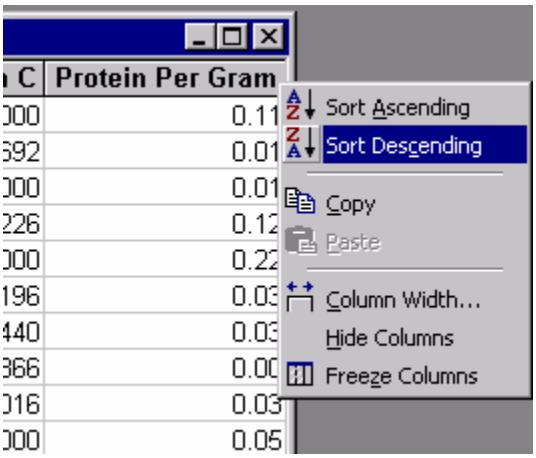


Figure 14. Sort Descending

This works very well for a simple sort, based on one field. This does not work as well if you need a more complex sort based on more than one field. There are many times when your first choice for a sort might result in a tie. If you sorted names, for example, you might have several people with the same last name, so you might want to break the tie by sorting on the first name (so all the Smiths are together and John Smith would come before Kimberly Smith).

To do this, we will need another Query. Go to your main database window, click on the Query tab and click on the New button to get a new Query. Select Design View and click OK. Now, you will be asked on which tables or queries to base this new query. Since we want to see all of our data, including our calculated Protein Per Gram, we can base this query on the Protein Calculation Query that we created earlier. In the Show Table dialog, click on the Queries tab, click on Protein Calculation, and click Add. Now close the Show Table dialog. So far, this is the same as what we did to create the Protein Calculation Query, except that we are using the query instead of the Foods Table.

Just as we did to create the Protein Calculation Query, you want to drag each of the fields listed in the top half of the screen to a column in the bottom half of the screen. Note that you don't need to include all of the fields. If you just wanted to show a few fields, you could just drag those fields to the table at the bottom. When you switch to Datasheet View, only the fields you add to the table will show up for this Query.

Notice that the third row of the table at the bottom is labeled Sort. If you click in the sort row for any field, you will get a pull down menu. From that menu, you can choose Ascending, Descending, or not sorted:

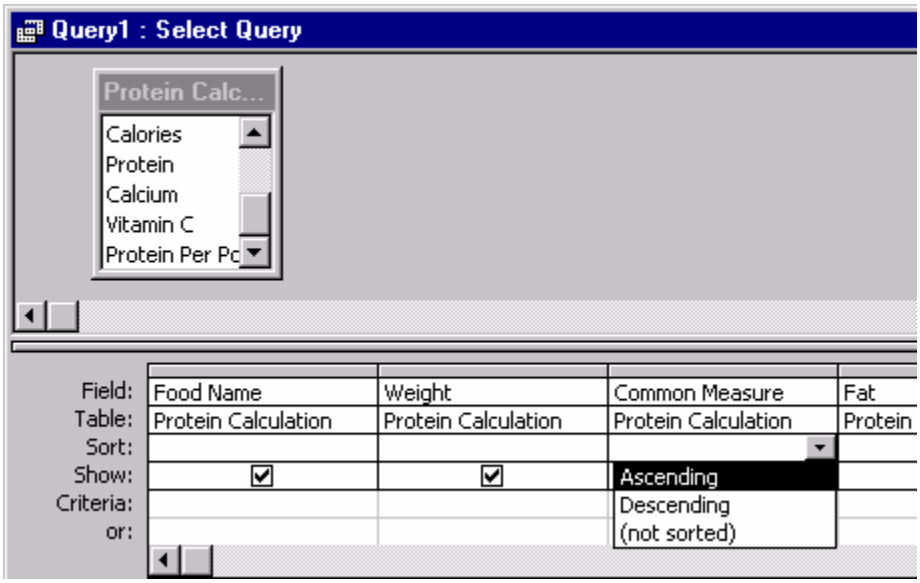


Figure 15. Sorting With a Query

Select Ascending for the Common Measure field. Switch back to Datasheet view by choosing Datasheet View from the View menu. You should now see all the records sorted by common measure:

Query1 : Select Query

Food Name	Weight	Common Measure	Fat	Calories	Protein	Calcium	Vitamin C	Protein Per Gram
apple	138.0	1 apple	0.497	81.42	0.262	9.660	7.866	0.00
broccoli	88.0	1 cup	0.308	24.64	2.622	42.240	82.016	0.03
milk (2%)	244.0	1 cup	4.685	122.00	8.125	297.680	2.440	0.03
milk (whole)	244.0	1 cup	8.150	148.84	8.028	290.360	2.196	0.03
orange juice	248.0	1 cup	0.496	111.60	1.736	27.280	124.000	0.01
egg	33.4	1 large egg	0.000	16.70	3.514	2.004	0.000	0.11
chocolate chip cookie	10.0	1 medium cookie	2.260	48.10	0.540	2.500	0.000	0.05
orange	131.0	1 orange	0.157	61.57	1.231	52.400	69.692	0.01
hamburger (fast food)	106.0	1 sandwich	9.773	272.42	12.317	126.140	2.226	0.12
fried chicken (fast food, white meat)	163.0	2 pieces	29.519	493.89	35.713	60.310	0.000	0.22
	0.0		0.000	0.00	0.000	0.000	0.000	

Record: 1 of 10

Figure 16. Results of Simple Sort

Unfortunately, we still have a problem with broccoli, whole milk and 2% milk. They all have the same common measure. If the order is important, we can add a second sort criterion. We will add Weight as the second sort criteria. Return to Design View (select Design View from the View menu) and click in the Sort row for Weight. Set that to Ascending and switch back to Design View.

Uh oh! Access has sorted based on the weight, but we only wanted weight to break ties in common measure. That is because whatever appears first in Design View is the first criterion to use in the sort. What we need to do is go back to Design View and put the Common Measure field before the weight field. To do this, select Design View from the View menu and click on the Common Measure column to highlight it. Drag this column to the left of the Weight column and let go. Common Measure should now appear before Weight in the table. Switch back to Design View. Now, all your records should be in the correct order.

We still have a tie with the two kinds of milk. That can be broken by a third criterion if it is important what order the records appear.

If you really want Common Measure after Weight, but you still want Common Measure to be your primary sort criterion, you can drag your fields around in Datasheet View. The order in Design View determines the order of the sort, and the order in Datasheet View determines the order in which you see the fields.

Click on the close box of the query, and click Yes when asked if you want to save changes. Give this query a reasonable name, such as Common Measure Sort. You should now see two queries in your main database window: Common Measure Sort and Protein Calculation. Both of these queries have the same data, but Common Measure Sort has the data sorted by common measure.

You can define several different queries for several different sorts because sometimes you will want to look at your records in one order, and sometimes you will want to look at them in another order.

For example you might want to see which are the highest fat foods, and you might want to see which foods have the most Vitamin C.

Something to Try

Now that you have done a simple sort, try some of the following exercises or make up your own:

- Sort to determine which are the highest fat foods.
- Sort to determine which foods have the most Vitamin C.
- If you completed the exercise at the end of the Calculation section, sort to determine which foods have the highest nutrition index.

Selecting

Sometimes you will only want to see certain records. For example, you might want to pick out the foods in which one serving has less than 1 gram of fat. With only 10 foods, this is easy without a database. But with a large number of foods, the power of the database becomes evident.

In Access, selections are also done with a Query. Get a new Query in Design View and use all the fields in the Protein Calculation Query (just like what we did originally to set up our Sort). Notice that under each field, there is the Sort row (which we used to set up sort criteria). Under the Sort row is the Show row (ignore that). Under the Show row is the Criteria row. This is where we set up our selection criteria.

In the selection row, you can enter formulas to limit which records the Query will show. These formulas can be fairly simple or very complex. You can try entering a few expressions in the criteria for different fields. Here are some examples to try (when you try one selection, be sure to delete the criteria you set for previous selections):

- Enter =broccoli in the Food Name field. This will give you any records with the Food Name of broccoli.
- Enter =milk in the Food Name field. This will give you nothing because = is looking for an exact match.
- Enter milk* in the Food Name field. This will give you any foods whose names start with milk. Note that the * is a wildcard, so milk* means milk plus anything else.
- Enter >100 in the Vitamin C field. This will give you any foods with more than 100 mg of Vitamin C.
- Enter <=2.5 in the Calcium field. This will give you any foods with less than or equal to 2.5mg of calcium.
- Enter *a* in the Food Name field. This will give you any foods whose names contain the letter a.

Note that after you enter your criteria, Access might alter them slightly. That is because Access is very picky, but it is also fairly smart. It wants your selection criteria to be in an exact format (with quotes and spaces and special characters in exactly the right places), but it can usually figure out what you mean, so after you enter an expression, it alters it to what it really wants.

Let's walk through setting up one simple selection. If you tried some of the examples above, delete your selection criteria by dragging the mouse across the criteria and hitting the Delete key. We are going to select only those foods with more than .1 grams of protein per gram of food (based on our calculated Protein Per Gram field). If you are not in Design View, select Design View from the View menu. If your window is not wide enough, you might need to scroll to the right to see the Protein Per Gram field. In the Criteria row for the Protein Per Gram field, type >.1 as shown in Figure 17.

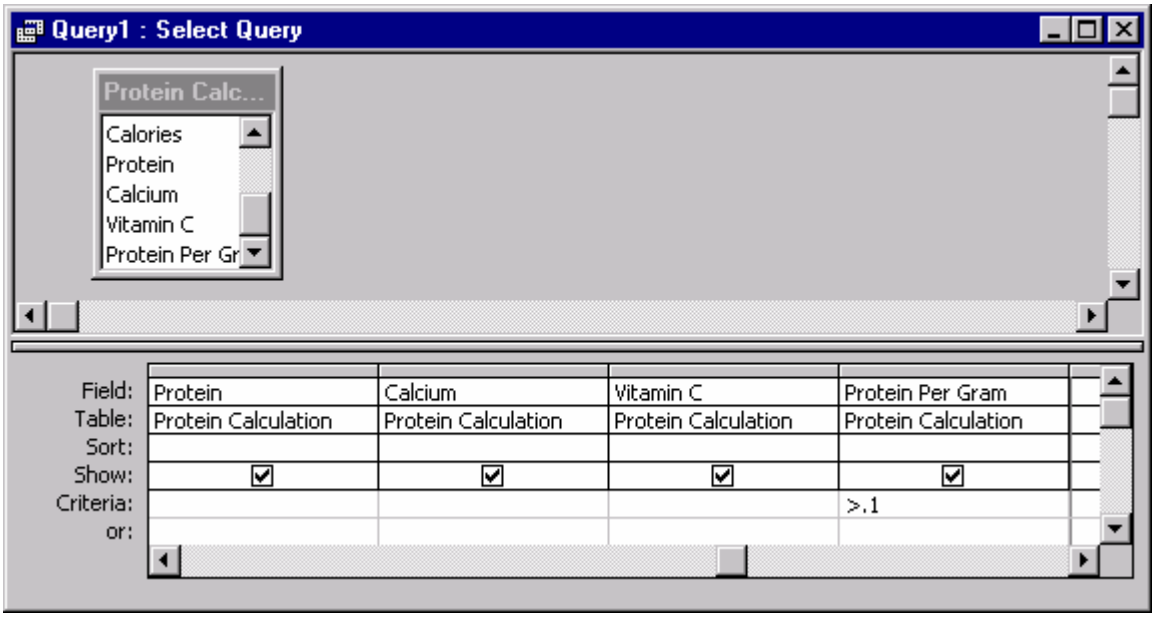
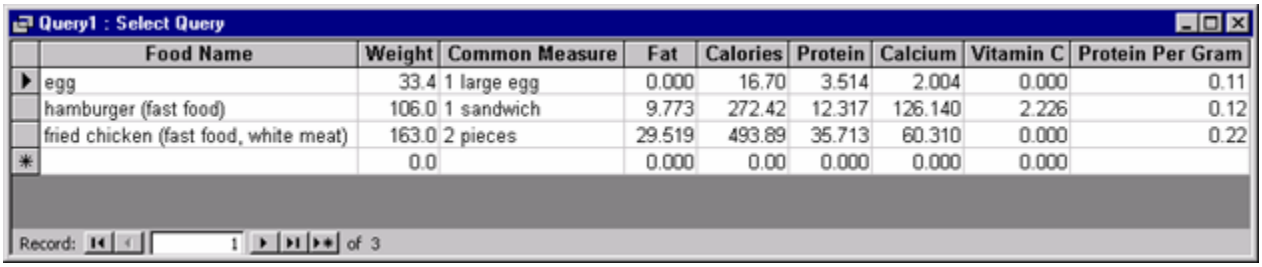


Figure 17. Selection Query

Switch to Datasheet View, and you should see three records: egg, hamburger, and fried chicken (see Figure 18). These are the only foods in our database with more than .1 grams of protein per gram of food.



Food Name	Weight	Common Measure	Fat	Calories	Protein	Calcium	Vitamin C	Protein Per Gram
egg	33.4	1 large egg	0.000	16.70	3.514	2.004	0.000	0.11
hamburger (fast food)	106.0	1 sandwich	9.773	272.42	12.317	126.140	2.226	0.12
fried chicken (fast food, white meat)	163.0	2 pieces	29.519	493.89	35.713	60.310	0.000	0.22
	0.0		0.000	0.00	0.000	0.000	0.000	

Figure 18. Results of Selection

Click on the close box for your Query, say Yes to saving it, and give it a name that makes sense, such as High Protein Per Gram. You should now see three queries in your main database window: Common Measure Sort, High Protein Per Gram, and Protein Calculation.

Using queries to make selections can be very powerful because you can pick out records based on any criteria you choose. For simple selections, you could perform a sort and look at the appropriate place on the list, but for some selections (especially complex selections with two or more criteria), that will not work.

Finally, note that you can include criteria for several fields at the same time, and you can add sort criteria as well to make a very complex query. You could, for example find high-protein, low-fat

foods by searching for foods with more than 2 grams of protein and less than 5 grams of fat, and you could sort these foods in alphabetical order.

Something to Try

Now that you have made some selections, try some of the following exercises or make up your own:

- Select those foods that are fast food (note that the name contains "fast food").
- Select those foods that are have a lot (more than 40mg) of Calcium and a lot (more than 40 mg) of Vitamin C.

Conclusion

Databases are very powerful tools (and we have just scratched the surface of what they can do) for organizing lots of information. The example used in this article shows you how to make a simple database of foods, but you can use databases for any information. You could easily create a database to replace your gradebook. If your students are collecting information about insects or rocks, they can put that information into a database and easily find out which insects have wings or which ones have more than 6 legs. For organizing large quantities of information, nothing beats a database.