

Access Level 2

Step-By-Step

PowerPoint Presentation

Create Database

1. Create a new database and save it in My Documents. Name it “Purchase Order Database.”
2. Import vendors.xls from the wdata folder to create a table. This table contains information about companies that order stuff from my company.
3. (Talk about DB design and why we would add more tables. If you are going to keep everything in one table it would be simpler just to use Excel. However, if you do that you loose much functionality.) So, what is this database going to do? Every company has vendors from whom they purchase stuff. This database will be used to manage PO’s that are issued by my companies to the vendors that we use. We have vendor information in a table, but what other information are we going to need to make this a useful db? We are going to need at least two more tables. We are going to need a table that will store orders (a record of all purchase orders that I have issued for purchase of supplies, etc) and a table that will store information about all the products that can be ordered from each vendor. So let’s create an orders table and a products table.
4. Create a new table with the following fields and properties. It is imperative that you do this exactly according to instructions.

Products Table

Field Name	Data Type	Properties
ProductID	AutoNumber	Primary Key
ItemNo	Text	
ItemName	Text	
UOM	Text	
Unit Price	Currency	
Product Code	Text	

5. Save the table as Products Table
6. Select File – Get External Data – Import and import the Products.xls file from the wdata folder

7. Create a new table with the following properties. It is imperative that you do this exactly according to instructions.

Orders Table

Field Name	Data Type	Properties
OrderNumber	AutoNumber	Primary Key
PurchaseOrder	Number	
VendorNo	Text	
Quantity	Number	
ShippingChrg	Currency	
OrderDtd	Date/Time	Short Date
ShipDtd	Date/Time	Short Date
ProductID	Number	

8. Save the table as Orders Table
9. Import Orders.xls from the wdata folder into this table.
10. (Discuss how these tables are going to be related and why.) How should these tables be related to one another? What type of relationship do we need between these three tables?
 - First of all consider the relationship between the Vendors and the Purchase Orders. What kind of relationship exists there? Is it possible that I will every issue more than one PO to any given Vendor? The vendors certainly hope that I will be a repeat customer. So the relationship between the vendor table and the Purchase Order table is going to be a one-to-many relationship with the Vendor table being on the “one” side of the relationship. That way, for every one vendor in the database I can have many purchase orders and that will make them happy.
 - What about the relationship between the Orders table and the products table. The Products table contains all the products that I purchase from all the different vendors. These are the things that I buy. Will a product ever be ordered on more than one purchase order? Yes. So what is the relationship between the Orders table and the Products table going to be? Again, it is going to be a one-to-many with the Products table being on the one side. That way, for every product I can have many purchase orders. When I run out of toilet paper (a product) I can order more (purchase order).
11. Set up the relationships in the relationships window (Set properties to Enforce Referential Integrity and Cascade both updates and deletes.)

Creating the Queries

Now we are going to create several different types of queries, some of which are useful for data maintenance, some of which will perform some important calculations, and some of which we will use in a report that we are going to create later.

Find Duplicates Query

It is quite easy during the course of using a database for duplicate entries to be made. For example, you might get a new vendor today and you enter the data. Then some time later you want to order from the same vendor again. But you don't remember that you have already entered that vendor into the database so you enter them again. Duplicates are not good because they make the job of analysis and reporting difficult and inaccurate. So you need to have a way to check for duplicates, and you do. It is called a "Find Duplicates" query.

1. Click the queries tab and click New
2. Select "Find Duplicates Query Wizard" and click OK
3. Select the Vendors Table and click Next
4. Now select Name, Address, City, State (Why not the other fields? If you entered a company twice you would probably give the second one a different id, the contact person could have changed, and the phone number could have easily changed.)
5. Add the other fields to the Additional Query Fields list
6. Run the query. As you see, there is, indeed a duplicate entry in the database. So, what are we going to do about that?
7. In the query results set simply select the record that you want to delete and press the delete key on your keyboard. However, some people (me included) are a little hesitant to delete data out of an important database so before we do that, let's make a backup copy of the data that we are about to delete. Close the Find Duplicates Query. It will be automatically saved for later use.

Make Table Query

The Make Table Query allows you to take a set of data and create a new table out of it. Let's see how it will work in this situation.

1. Run the Find Duplicates query
2. Change to design view and click on the Query Type button on the toolbar
3. Select the Make Table Query and name the new table that you are going to create "Duplicate Entries Table"
4. Run the query
5. Close the query and **don't save the changes**. (If you do it will no longer be a find duplicates query)
6. Select the Tables tab and notice the new table. Open it and you will see the duplicate records in the table.

7. Now run the find duplicates query again and delete one of the records with complete confidence that if you make a mistake the data can be recovered.

Delete Query

Another important type of query is the delete query. There will be times that you need to delete information from a table. For example, you might want to delete records for canceled PO's or for vendors that are no longer in business. It is a good idea to run a select query to display the records that you are about to delete and examine them closely before you delete them. In fact, many people will run the select query, convert it to a make table query, and then convert it to a delete query. Here is how the process works.

1. Open a new query in design view and select the products table. It will be a select query by default.
2. Add the Products table to the design grid.
3. You can add some or all the fields in the table or you can just add the field or fields that you are going to base your query on. For example, in this query we are going to delete all the records that do not have a P for product code. Add the product code field to the first column.
4. In the criteria row of the design grid type "Not p" and press the tab key.
5. Run the query and verify the results.
6. Return to design view and change the query to a delete query. (If I was concerned about accidentally deleting data that I shouldn't delete, this is where I would do a make table query and save a backup copy of this data.)
7. Run the delete query and then close it. Save the query as Delete Query.
8. Open the products table and verify that the data is gone.

Update Query

Suppose that you have a database with addresses that you use for mailing labels and shipping labels and envelopes and all of the sudden you have a zip code change. If your database is small that is really no big deal. But what if you have 25,000 records in your database and 10,000 of them just had a zip code change. Now that could be some major work and take a lot of man hours to update.

Update queries are designed just for this purpose. Using an update query you can change the values of data in a table. Such queries are useful when you must update field values for many records with a common expression. For example, you might need to increase or decrease the unit prices of all products or products within a particular category by a fixed percentage. Or you might need to increase the shipping charge for all items or for a subgroup of all items in your database. That is what we are going to do.

The boss has just called you and told you that all three shipping companies that we use have just announced an increase in their charges and we need to increase our shipping charges so that we don't lose money on all our shipments. Here is how you would go about doing that.

1. Open the Orders table and make a note of the shipping charges

2. Close the Orders table
3. Open a new query in design view and add the Orders Table
4. Change the query type to Update Query (Again, you might want to consider making a new table, but with this one to undo something you just did won't be too difficult.)
5. Select the shippingChrg field
6. In the **Update To** row type [shippingChrg]+10 or invoke the Expression builder and select the shippingChrg field from the Orders table, click the + sign, and type 10. (describe what this doing)
7. Run the query, close it and save it as **Update Shipping Charge Query**.
8. Open the table to verify that the shipping charges have been updated
9. Now, if you wanted to you could simply update the table again and change the shippingChrg field to [shippingChrg]-10.

Calculation Queries

One of the real powers of a database is the power to perform calculations on your data. For example, if your database has a field for unit price, quantity ordered, and shipping charge, you can use a query to calculate the charge for the order and amount to bill the customer for, which would be the total charge for all products ordered plus shipping. Or, if you were doing a database to record student progress and grades, you could use a calculation in a query to calculate test averages for all your students, daily averages, and final grade. The possibilities are really endless, limited only by your ability to dream up ways to use them. Since we have an Orders database we are going to create a query that will calculate the amount to charge our customers for any product, any amount, on any given order and then add the shipping charge to that amount.

1. Open a new query and add the Orders Table, the Vendors Table, and Products table
2. Add fields to the query in this order...
 - VendorNo
 - Name
 - PurchaseOrder
 - ShippingChrg
 - OrderDtd
 - ShipDtd
3. Run the query, take a note of the results, and return to design view
4. Click in the next field and type Amount:. This will become the name of this query field. Now we are going to use the Expression Builder to create a calculation that will calculate the amount of the order.

5. Right click on the field and select Build
6. This calculation is going to multiply Unit Price times Quantity Ordered. Find Unit Price by clicking on the expansion button on the left side of the Tables icon
7. Click on the Products Table to show all the fields in the table
8. Double click on UnitPrice
9. You will probably have to manually delete the <<express>> that is added before the field name
10. Click after the UnitPrice field and click on the * Sign
11. Find the Quantity field (it is in the Orders Table)
12. Double click on it and click OK
13. Press the tab key to make sure that Access is understanding what you want to do
14. Run the query and check out the results
15. Notice that the Amount column is formatted as currency. In earlier versions of access you would have to manually format it for currency. Starting in Access 2000 you don't have to. If you find that you have to format it manually, here are the steps.
 - Return to the design view and fix that
 - Right click on the Amount column and select Properties
 - On the General Tab click in the Format Row and select Currency
 - Run the query again and note the difference
16. Save the query as OrdersCalculationQuery

Now we know how much the bill will be for the products ordered, but we don't know how much the total bill will be, including the shipping charge. Let's create another calculation field to calculate that Grand Total amount.

1. Click in the next column and type Grand Total:
2. Invoke the Expression Builder
3. Where will we find the amount of the product order? (In this query but only if you saved it.) Expand the Queries icon and select this query
4. Double click on Amount
5. What do we want to add to that? (Shipping) And where do we find it? (Again, in this query) Add the shippingChrg to the Amount field
6. Click OK and tab away from the field
7. Run the query and check out the results
8. Notice that the results are already formatted as currency. The program is smart enough to know that if you are adding two fields that are currency that you probably want the results to be currency.

9. Save it and close it

If Then Query

Access has many very powerful built in functions that you can use for more things than we could name for the rest of the afternoon. We are going to look at one of those functions today and see how you can use it in a query. The Iif function, which is identical in every way to the If function in Excel, is a function that returns one value if a condition you specify evaluates to TRUE and another value if it evaluates to FALSE. The Syntax is IIF(logical_test,value_if_true,value_if_false). Let's see how that will work in our query.

We want to create a query that says if the amount of the order is over a certain value, we will not pay the shipping. If it is not, then we will pay the shipping. And, we want to query to go ahead and calculate the total amount of order, including shipping, if there is any.

1. Open a new query in design view
2. Add the Orders Table and the Product Table
3. Add the Purchase Order, Vendor Number, Shipping Charge, and Quantity to the design grid
4. Save the Query as the IfThenQuery
5. Use the Expression Builder for the next column and create a Total Amount calculation for the cost of the products (Unit Price * Quantity) and set the format property to currency if necessary
6. Save the query
7. In the next column type Payment Due: and invoke the Expression Builder
8. Double click Functions – Built-in Functions
9. In the far right column find and double click the IIF function
10. Delete the first <<Expr>> and the empty space
11. Click on the <<expr>> inside the parenthesis
12. Expand the Queries icon and click on the IfThenQuery. (Since this query is open you will also find it at the top of the list in the far left window of the Expression Builder)
13. Double click on Total Amount
14. Click on the > symbol
15. Click on the <<expr>> that is added after the > symbol and type 2000. (We have told Access “If the value in the Total Amount field is greater than 2000, I want you to do something.” The something comes next.)
16. Click on <<truepart>> and select Total Amount. (If the value in Total Amount is greater than 2000, then this field should have the value in Total Amount.)
17. Click on <<falsepart>> and select Total Amount
18. Click on the plus sign and then click on ShippingChrg (You may have to get rid of some more of those pesky little <<expr>> things)
19. Click on OK and tab away
20. Run the query and take a look at the results
21. Be sure that you save the query

Now we need to create one more query using some of the things that we have learned so far. This will be the query that we will use in the next part of the workshop, creating reports.

1. Open a new query in design view and add the Vendor Table, the Orders Table, and the Products Table
2. Save the query as PurchaseOrderQuery
3. Add the following fields to the design grid
 - PurchaseOrder
 - VendorNo
 - Name
 - Address
 - City
 - State
 - Zip
 - Contact
 - Phone
 - ProductID
 - ItemNo
 - ItemName
 - UOM
 - Quantity
 - UnitPrice
 - Product Code
 - ShippingChrg
 - Save the Query
 - A Subtotal: field – [Quantity]*[UNIT PRICE]. Format it as currency
 - Save the query
 - A Grand Total: field –
IIF([subtotal]>2000,[subtotal],[subtotal]+[shippingchrg])
 - Save and close the query.

Reports

Now that we have designed all the tables and queries that we need (at least to begin with) it is time to start doing something with all our data. Many times when you need information from your database for someone (like your boss) you are going to want it in a printed format. In Access that means creating a Report. So, we are going to create a Report using some of these cool things that we have been working on.

1. Open a new report in design view
2. You will notice that the blank report is divided up into three sections. It is important that you understand what each section does.

- The page header appears at the top of every page in the report. You use it to display items such as column headings.
- The detail section contains the main body of a report's data. This section is repeated for each record in the report's underlying record source.
- The page footer appears at the bottom of every page in the report. You use it to display items such as page numbers.

There are also some other sections that you don't see by default but that you will find useful.

- The report header appears once at the beginning of a report. You can use it for items such as a logo, report title, or print date. The report header is printed before the page header on the first page of the report. (View Report Header/Footer)
 - The report footer appears once at the end of the report. You use it to display items such as report totals. The report footer is the last section in the report design but appears before the page footer on the last page of the printed report.
 - A group header appears at the beginning of a new group of records. You use it to display information that applies to the group as a whole, such as a group name. We will add this section a little later.
 - A group footer appears at the end of a group of records. You use it to display items such as group totals. We will add this section also.
3. Stretch the work area (the gridded area) and stop at 6 ½”.
 4. Make the Page Header Section a little larger
 5. If your toolbox is not showing, click on the Toolbox shortcut on the toolbar
 6. Use the label tool to create a label with the text “**King Head, Inc.**”
 7. Set the properties to size 14, bold and centered. Center the text box in the page header
 8. Add another one with the same properties that says **Purchase Order**
 9. Select Format – Align – Left to align the two text boxes
 10. Draw a line between the second text box and the detail section
 11. Change the Border Width property to 5 points (right click on the line and select properties)
 12. Now we need to attach this form to a data source, the **PurchaseOrderQuery**. Double click on the square that is at the top of the left side ruler and the left of the top ruler. That should open the property sheet for the report
 13. Select the **Data** tab and change the **Record source** property to **PurchaseOrderQuery**.

14. The field list from that query should open. If it does not, click on the Field List button just to the left of the Toolbox button on the toolbar
15. We need to add group header. To do that, select **View** on the menu bar and click on **Sorting and Grouping**
16. In the Field/Expression list select **Purchase Order**
17. Set the properties as follows...
 - Group Header – Yes
 - Group Footer – Yes
 - Group On – Each Value
 - Group Interval – 1
 - Keep Together – Whole Group
18. You should see the difference in the layout of your report. This will enable us to create a single report with a separate page for each PO (the group).
19. Click and drag the **Purchase Order** field from the field list to the **PurchaseOrder header**.
20. Delete the label, change the font to Vernada, size 12, bold and Red
21. Place the text box on the left side of the report and adjust the size of the header sections
22. View your work in Print Preview mode. You may notice that the PO number isn't where you expected it to be. Go back to design view and click on the text box. Click on the left align button on the toolbar and take another look at the report in print preview mode.
23. Use the page navigation tools at the bottom to the page to view more than one page. You will find that there are a number of pages, and each one has more than one purchase order on it. That is not a good thing. We want this report to show all of the important information about a single purchase order. Let's fix that.
24. Go back to design view and double click on the **Purchase Order Header** bar to open the properties sheet for this section.
25. Select the format tab
26. Set the "**Force New Page**" property to "**Before Section.**"
27. Now take a look at the report and you will see that each page has a single PO number on it and that is what we are after.
28. In the **Details** section put a label that says **Item Information**
29. Make it size 12, bold, italic, and blue
30. Find **ItemNo** in the field list and drag it under the **Item Information** label
31. Position the label directly on top of the field box.

32. Do the same for **Item Name, UOM, Product Code, unit price, and quantity**
33. Arrange the items on the report in a like manner with the sample
34. View the report and notice that unit price and quantity don't line up like they should. (By default, a number field is going to be right aligned)
35. Change the alignment of those two fields to left aligned and view the report again.
36. If necessary, change the format of the number to currency (Right click on the text box, select properties, click on the format tab, and select Currency in the Format row)
37. Place another Label on the report, this time with the text **Vendor Information**. Format like the Item Information label
38. Place the **Name** field on the report, delete the label, change the size to 10 and make it bold. Position it under the Vendor Information label
39. Add the address field, again with no label, under the name field.
40. Place a text box on the form, right click and select properties, on the Data tab right click on the control source line and select Zoom
41. Type the following information... (make sure the names here match up with the name of the table.)
=[Vendors.city] & ", " & [Vendors.state] & " " & [Vendors.zip]
42. View the report in print preview mode.
43. Add the **Contact field** under the Vendor Name information.
44. Place the Contact label on top of the Contact field and make the label bold
45. Add the **Phone field**
46. Delete the Phone label and position the field directly under the Contact field
47. Use the alignment submenu to align all the labels and fields on the left side of the form
48. Copy and paste the **Vendor Information label**. You will notice that when you paste an object it will be placed in the upper left hand corner of the report section into which you paste it.
49. Move the pasted label to the right of the Vendor Information label and change the text to **Item Charges**
50. Line up the two labels (Format – Align – Top (or Bottom))
51. Place the **Subtotal, Grand Total, and ShippingChrg** fields on the form under the Item Charges label
52. Move each label on top of it field and make them bold
53. View the report in Print Preview mode
54. Change the alignment of the three Item Charges fields to left aligned

55. Set the Format property of each of the three fields to Currency
56. View the report
57. In the **Purchase Order Footer**, place a label for signature. Type signature and use the underline key to create the line. Make it bold.
58. Place a text box in the Purchase Order Footer
59. Change the label to **Total For All Orders** and make it larger and bold
60. For the text box, use the Expression Builder (Right click on text box, select Properties, select Data tab, click on the ellipse button) to create the calculation that calculates the sum of the **Grand Total** fields for each purchase order.
=Sum([Grand Total])
61. Make the text box larger and bold and format it as Currency
62. Select View – Report Header/Footer to add a report footer to the report
63. In the Report Footer add a text box
64. Change the label to **Grand Total of All Orders on All PO's**
65. For the text box, use the Expression Builder (Right click on text box, select Properties, select Data tab, click on the ellipse button) to create the calculation that calculates the sum of the total amount fields for each purchase order.
=Sum([Grand Total])
66. Save the report and view it. Be sure to look at the last page
67. Right click on the **Report Footer header bar** and select Properties
68. Set the “**Force New Page**” property on the Format tab to “**Before Section**”
69. Save and view the report, including the last page
70. In the Page Footer section of the report place a text box and delete the label
71. Use the Expression Builder to create an expression that will insert the current date. =date() (This function is located in the expression builder under Functions – Built-in Functions)
72. Add another for current time. =time() (This function is located in the expression builder under Functions – Built-in Functions)
73. Add another for page number. =”Page ”&”[Page] (You will find this option in the expression builder under Common Expressions.)