



# Using Microsoft Access

<b>USING MICROSOFT ACCESS</b>	<b>1</b>
<b>Interfaces</b>	<b>2</b>
Basic Macros	2
Exercise 1. Creating a Test Macro	2
Exercise 2. Creating a Macro with Multiple Steps	3
Exercise 3. Using Sub Macros	5
<b>Expressions</b>	<b>6</b>
Exercise 4. Creating an Expression in a SubForm	7
Exercise 5. Creating a Linked Expression	10
<b>Additional Form Formatting</b>	<b>12</b>
Exercise 6. Formatting a Text Box	12
Exercise 7. Changing Form and Subform Properties	13
Exercise 8. Adding Form Navigation Tools	14
Exercise 9. Adding Control Buttons to a Form	17
<b>Database Maintenance</b>	<b>20</b>
Exercise 10. Running Compact & Repair on a Database	20
<b>Creating an Interface</b>	<b>21</b>
Exercise 11. Creating a Switchboard	21
Exercise 12. Setting Database Startup Options	27
Exercise 13. Adding the Switchboard icon to the Ribbon Bar	29

## Interfaces

In the following exercises we will use a file called *Music Collection 3.mdb*<sup>1</sup>. This database is similar to the one you worked on in the previous section except it has a lot of additional queries, forms and reports. Before you begin, you may want to open that database and familiarise yourself with the additional objects that have been created.

### Basic Macros

Macros provide a way to automate tasks that would usually be complex or involve many steps. Macros can be created easily in Access by choosing from a number of common procedures and combining them in whatever way you need. These macros can then be activated by various events, such as clicking a button or opening a form.

#### Exercise 1. Creating a Test Macro

1. Make sure the *Music Collection 3.mdb* database is open.
2. Click the **Create** tab on the **Ribbon**.



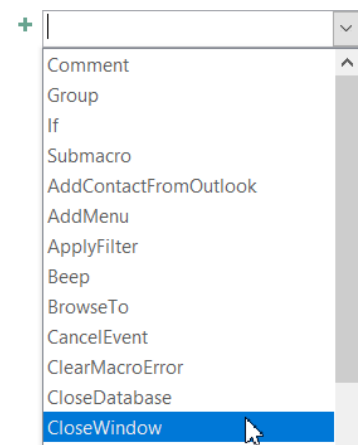
3. Click the **Macro** button.



A new Macro will be created and an **Add New Action** combo box will appear. To the right of the screen is the **Action Catalog** which allows you to browse through all of the action types available (including actions from macros you have already created).


4. Open the **Add New Action** list.
5. Select the **CloseWindow** action from the list. You can also type the first few letters of an action to select it, though in this case you will get the **CloseDatabase** action before you type enough letters to get the one we want.

Some additional options will appear for the action.

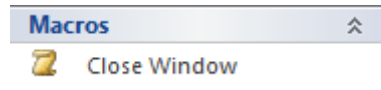


6. The first options allow you to specify that you want the action to close a specific type of object or an object with a particular name. Since we want it to be able to be used with any type of object, we will leave those options blank. The last option allows you to determine if the any changes should be saved before closing the window. Leave that option as Prompt so that the user will be asked if they want to save changes or not.


<sup>1</sup> You can get it from <http://oneil.com.au/pc/access2010.html>

7. Click the **Save** icon  on the toolbar to save the Macro so we can test it.
8. Enter *Close Window* for the macro name.

The Macro will appear at the bottom of the **Navigation Pane**.



**Tip** You can show or hide the **Navigation Pane** by pressing **F11**.

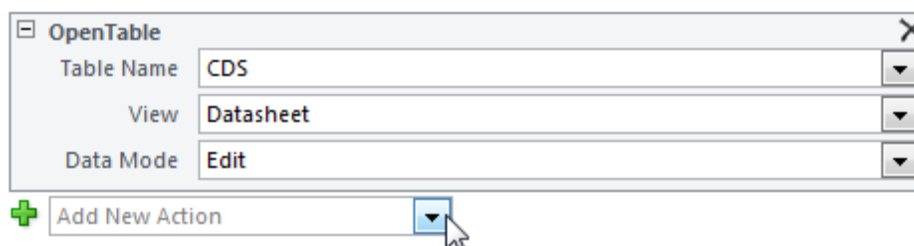
9. Click the **Run** icon  on the **Ribbon** to test the macro.

When the macro runs, it will close the current window (in this case, the macro design window). Later we will see how you can run a macro from other locations in the database.

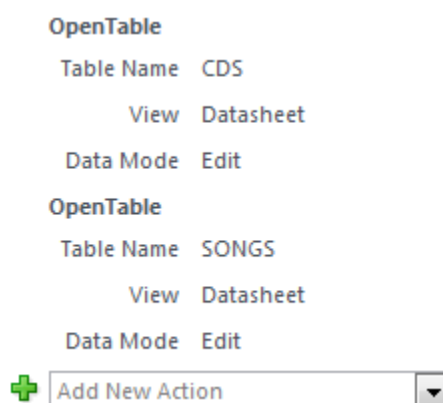
## Exercise 2. Creating a Macro with Multiple Steps

Now we'll create a macro that does more than one thing in a sequence.

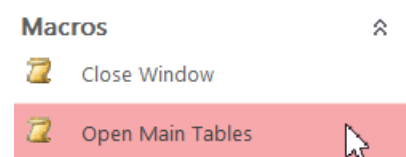
1. Select the **Create** tab on the **Ribbon** and then click the **Macro** icon.
2. From the list of actions select **OpenTable**.
3. Change the **Table Name** to *CDS* as shown.



4. Select the next **Add New Action** list and choose **OpenTable** again.
5. For the second action, change the table name to *SONGS*.



6. Save the Macro as *Open Main Tables* and close it.
7. Double click the macro from the Objects list on the left to run it. The macro will open both of the tables specified.

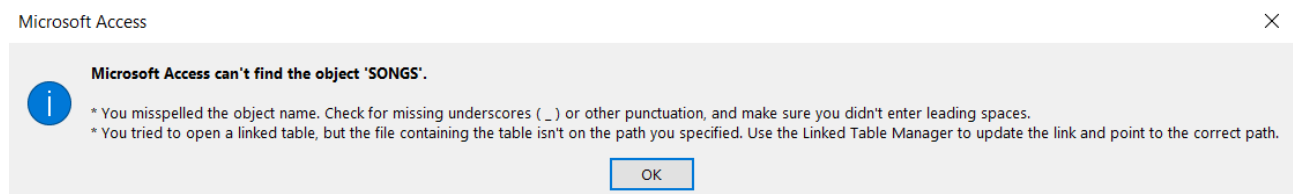


A tab will be shown for each of the open objects (two tables). You can use these tabs or press **Ctrl F6** to switch to an open object.

Open Main Tables CDS SONGS			
Song #	Song Title	Artist	Genre
1	Daydreamer	Adele	Pop

8. Close both tables.

**Note** Sometimes there may be a problem with your macro. This may be because it refers to an object that can't be found, often because it has been renamed or deleted. When a Macro can't run correctly, you may get messages like the following.



Macro Single Step

?

×

Macro Name:

Open Main Tables

Step

Stop All Macros

Continue

Condition:

Action Name:

OpenTable

Error Number:

3021

Arguments:

SONGS, Datasheet, Edit

### Exercise 3. Using Sub Macros

Sub macros allow you to have several related macros grouped together.

1. Select the **Create** tab on the **Ribbon** and then click the **Macro** icon.
2. From the list of actions select **Submacro**.
3. Enter *Open CD Types* for the **Submacro** name.
4. For the **Macro Action** select *OpenForm*.
5. In form name enter (or select from the list) *CD Types Form*.
6. Make sure **View** is set to *Form* and **Change Window Mode** to *Dialog*. The Dialog option will open the form as a small window instead of one that fills the screen.
7. Add a second Submacro for opening the Genres form as shown below.

The screenshot shows the Microsoft Access Macro Editor. At the top, a tab labeled 'Dialogs' is selected. Below it, two submacros are listed:

- Submacro: Open CD Types**
  - OpenForm**
    - Form Name: CD Types Form
    - View: Form
    - Filter Name:
    - Where Condition:
    - Data Mode:
    - Window Mode: Dialog
  - End Submacro**
- Submacro: Open Genres**
  - OpenForm**
    - Form Name: Genres Form
    - View: Form
    - Filter Name:
    - Where Condition: =
    - Data Mode:
    - Window Mode: Dialog
  - End Submacro**

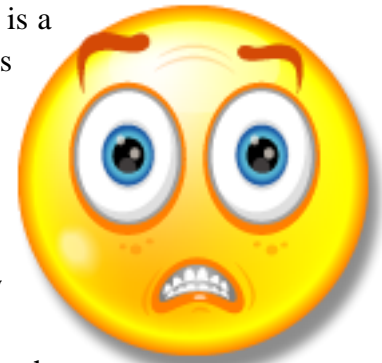
Below each submacro, there is a button with a plus sign and the text 'Add New Action'.

8. Save the macro as *Dialogs* and close it.

*Open CD Types* and *Open Genres* can both be used as separate macros, but they are kept together in the Navigation Pane. If you have a lot of macros, grouping similar ones together as sub macros keeps things neater. We will make use of these macros later on.

## Expressions

Anyone familiar with Excel will find expressions easy. An expression is a statement (usually a calculation) that produces a certain result. It is very similar to an excel formula and is written in much the same way. Like a formula in Excel, expressions in Access use the following rules:



- An expression must begin with a = sign.
- An expression can use built-in functions (such as Sum) to simplify certain types of expressions.
- Any calculations in expressions follow the **order of operations**, or the BIMDAS rule where different parts of the calculation are performed in the following order.
  - 1) **B** - Anything in brackets is calculated first.
  - 2) **I** - Any indices are calculated next (E.g.  $=2^5$  or 2 to the power of 5).
  - 3) **MD** - Any multiplication or division goes next.
  - 4) **AS** - Addition and subtraction are calculated last.
- Instead of cell references as in Excel, an expression will usually refer to objects such as text boxes. Any object names in an expression must be enclosed in square brackets.

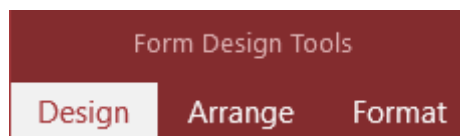
You will notice that these rules are very similar to the rules explained in the advanced queries section for calculated fields. In some of the previous sections you already created some simple expressions to calculate totals on reports. Now it's time for some more involved expressions.

Because so many expressions rely on references to objects in your forms and reports, it is a good idea to provide meaningful names for the objects in your forms and reports. Since all of the forms and reports we have created use text boxes that are based on fields in your tables, most of them will have the same name as the field they are based on. A common practice among developers and programmers is to specify the type of object in the name of the object. For example, a text box that has been created to display a sales total might be named txtSalesTotal. This name would make it clear that the object is a textbox and would also clarify the purpose of the textbox. In a moment you will see how object names are used.

## Exercise 4. Creating an Expression in a SubForm

1. Open the *CD Entry Subform* form in **Design View**.

2. Make sure the **Design** tab is selected from the **Form Design Tools Ribbon** group.



3. Click the **Textbox** icon in the **Ribbon**.
4. Click in the **Form Footer** area to place the textbox there. The precise location you click on won't matter since this text box in the footer won't be visible when you view the *CDs* form.

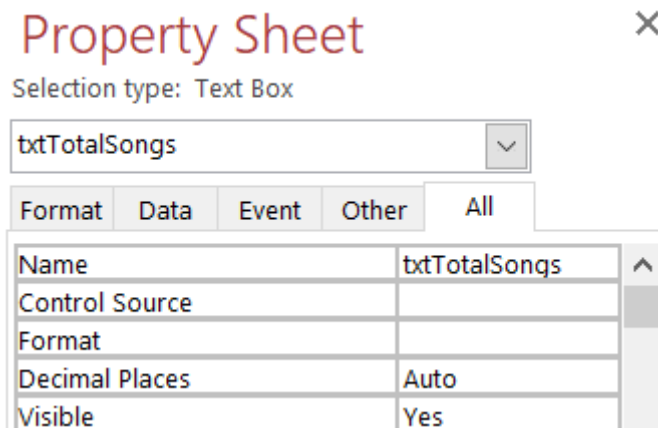


This textbox will be used to add up the number of songs in the subform. The result won't be visible in the main form but later we will create a text box on the main form that will refer to this one. Since the total on this subform won't be visible there is no need for the label to the left of the text box. You can delete it if you like. Make sure the new text box is still selected before proceeding. Previously we have entered expressions by typing them in to the text box directly. This time we will do it using the **Property Sheet** since we will change other properties as well.

5. If the **Property Sheet** is not already showing, click the icon on the **ribbon** to display it. If the text box is still selected, the properties window should appear as below. You can resize the property sheet if needed by dragging its left edge.




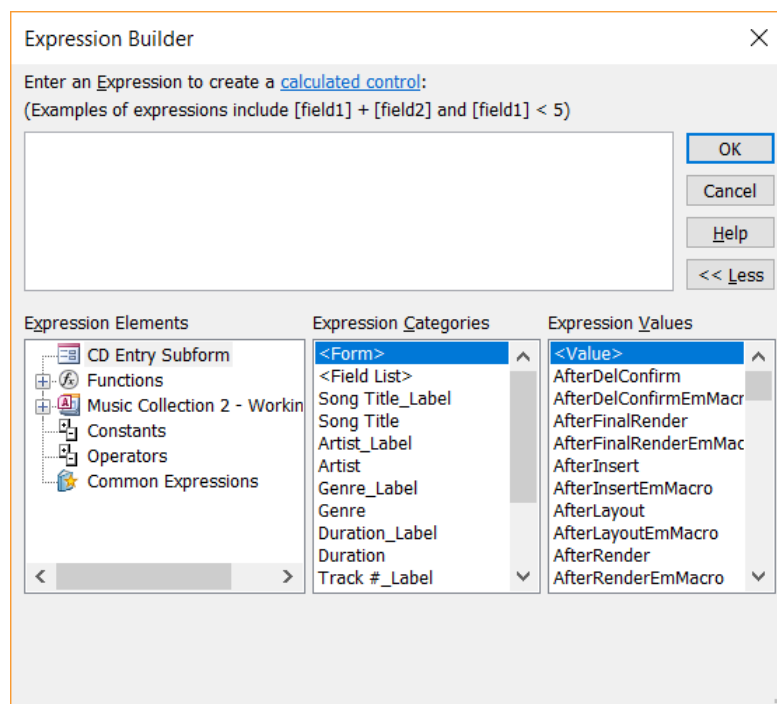
**Tip** You can also double-click many of a form's objects to quickly open the property sheet if it is not already showing.



- In the **Name** property enter *txtTotalSongs*. This name will be used later when we refer to this object.

**Note** What you call the textbox is not important. What is important is that when you refer to it as we will do later, the name has to be typed exactly the same including spacing (though it's not case sensitive).

- Click in the **Control Source** property. This is the property which determines what will appear in the textbox so it is where our expression will go. You can use the right-click and **zoom** option shown in the queries section to get more room for editing but an even better way is to use the **Expression Builder**. This handy tool gives you quick access to common functions as well as making it easy to refer to database objects in an expression.
- You will see an icon  to the right of the **Control Source** property. Click on this icon to open the **Expression Builder**.

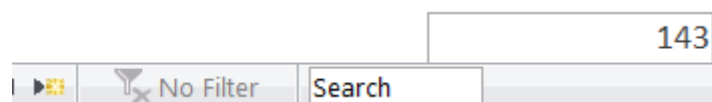
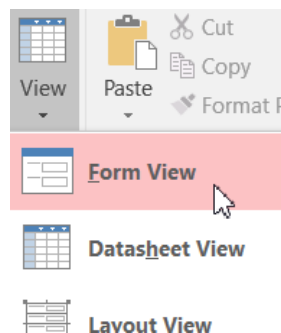




9. Start your expression by entering `=Count(` in the top section.
10. The middle of the three lists in the expression builder lists all of the objects in the form.  
**Double-click** on *Song Title* (not *Song Title\_Label*) to add the name of that object to the expression.
11. Finish the expression by adding a closing bracket. The completed expression should look like the one below.

Enter an Expression to create a [calculated control](#):  
(Examples of expressions include `[field1] + [field2]` and `[field1] < 5`)  
`=Count( [Song Title] )`

12. Click **OK** to complete the expression and return to the **Property Sheet**.
13. Previewing the won't show the total since this form is set to display in datasheet view by default. To view the form normally, click on the arrow under the **View** icon and then select **Form View**. The total number of songs will appear at the bottom of the form.



14. Close the form and save the changes.


### Exercise 5. Creating a Linked Expression

Now that our subform has an expression which adds up all of the songs, we will place an expression on the main form which refers to that subform total. Because of the relationships between our tables, the main form will only show the total of the songs showing at the time (i.e. the songs for each CD).

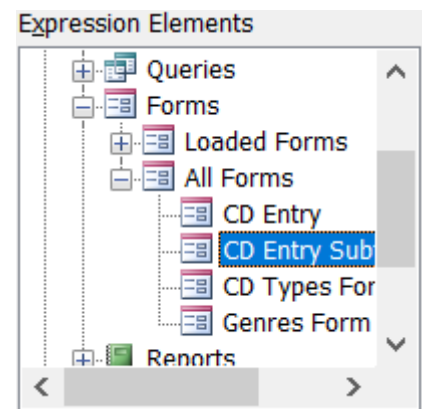
The tricky part here is that the expression on the main form needs to refer to the subform as well as the object within that form. Most mistakes in this kind of expression are made when the object name you are trying to refer to is typed differently when you refer to it.

1. Open the *CD Entry form* in **Design View**.

2. Scroll down if needed until you can see the bottom of the form. We will place a text box for the total in the form footer for the main form (not to be confused with the footer for the songs form which already has the text box we just made).
3. Click the **Text Box** icon in the **Ribbon**.
4. Click in the Footer area approximately 6cm from the left of the form to place a textbox. The ruler at the top might be showing units other than centimetres depending on your settings. You can use the gridlines on the design view as a guide as they will be lined up with the major markings (cm) on the ruler.

5. Make sure the textbox is still selected and open the **Property Sheet** if it is not already open.
6. In the **Name** property enter *txtTotalCDSongs*.
7. Click in the **Control Source** property and then click the  button to open the **Expression Builder**.
8. Start the expression by typing an equals (=) sign.

The **Expression Elements** list in the expression builder shows all of the objects in the database that can be used in expressions.



9. Double-click on **Music Collection 3.accdb** in the list to expand the selection.
10. Double-click on **Forms** in the list to expand the selection.
11. Double-click on **All Forms** in the list to expand the selection.
12. Double-click on **CD Entry Subform** in the list. The middle list will now display all of the objects from that subform.
13. In the **Expression Categories** list (middle list), **double-click** on *txtTotalSongs*. This will put a reference to that text box in your expression.
14. Remove the *Forms!* section from your expression. The completed expression should appear similar to the one below.

Enter an Expression to create a [calculated control](#):  
 (Examples of expressions include [field1] + [field2] and [field1] < 5)  
 =[CD Entry Subform]!txtTotalSongs

15. Click the **OK** button to close the expression builder.
16. Click the **View** button to preview the form. Your CD total should appear below the **subform**.

View some different CDs to see the total change with each one.

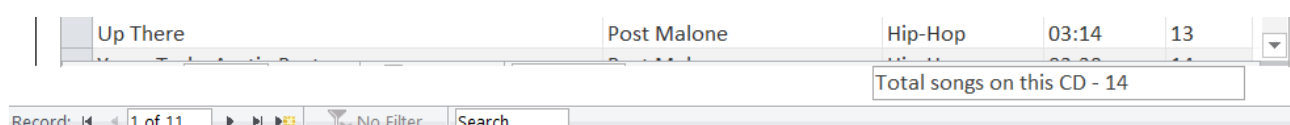
17. Return to **Design View**. Now we'll get fancy by adding some text to the expression.
18. Edit the expression so that it appears as follows.

=*"Total songs on this CD - "* & [CD Entry Subform]!txtTotalSongs

=*"Total songs on this CD - "* & [CD Entry Subform]!txtTotalSongs OK

The **&** symbol can be used to join text to a number in an expression.





19. Since we will have some text in the expression itself, there will be no need for the label. Click on the label next to the textbox and press the **[Delete]** key to remove it.
20. Resize the textbox so that it is approximately 6cm wide (you can use the properties for the textbox to make it exact by simply entering 6 in the **Width** property).
21. Move the textbox toward the right side of the form.
22. Save and preview the form to see the results of the expression and return to Design view once you are sure it is working properly.



## Additional Form Formatting

### Exercise 6. Formatting a Text Box

Now that our expression is working properly we can work on formatting. At the moment it looks like just another text box on the form. Since text boxes are often used for data entry, this may be confusing for users. We can change to properties to that it looks more like a message that is integrated in to the form.

1. In **design view**, click on the *txtTotalCDSongs* text box.
2. Select the **Home** tab on the ribbon.
3. Click the **Center** align icon on the ribbon (under the **Home** or **Format** tab). 
4. Click on the arrow next to the **Fill/Back Color** icon. 
5. When the colour palette appears, click on the *Transparent* option.
6. Click on the arrow next to the **Font/Fore Color** icon. 
7. Choose red for the font colour.
8. Select the **Format** tab from the Ribbon.
9. Click on the arrow next to the **Shape Outline** icon.  Shape Outline ▾
10. When the colour palette appears, click on the *Transparent* option.
11. Click the Bold formatting icon. **B**
12. Make sure the properties window is still displayed and change the following properties for that text box. These property changes will make the text box act less like a text box and more like a message.

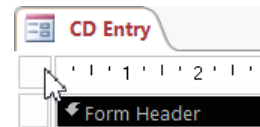
Property	Change to:	Explanation
Enabled	No	User can't interact with the textbox.
Locked	Yes	Contents of the textbox can't be edited by the user.
Tab Stop	No	When the user is pressing the [Tab] key to move through the text boxes in the form it will skip this object.

**Tip** When a property has two or more options you can select from (such as yes or no) you can change the option by double-clicking inside the property box. For example if a property is set to no, you can double-click it to quickly change it to yes.

13. Preview the form to see the changes. Click on the total songs textbox. Because of the property changes, you won't be able to select or change it.
14. Return to **Design View**.

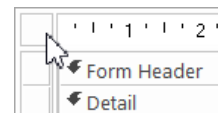
### Exercise 7. Changing Form and Subform Properties

1. Double-click the top-left corner of the form (as shown to the right). The properties for the form should be displayed.
2. Change the properties for the form as shown in the table below.



Property	Change to:	Explanation
Caption	Add / Edit CDs	Changes the text which appears at the top of the form.
Scroll Bars	Neither	Any scrollbars on the form will be removed.
Record Selectors	No	The Bar on the left of the form that can be used for selecting a record will be removed since the form will only show one record at a time.
Dividing Lines	No	No visible lines between the form header/footer and detail.

3. Double-click the top left corner of the subform to view the properties for the subform.
4. Change the properties for the subform as shown in the table below.



Property	Change to:	Explanation
Default View	Datasheet	This property will already be set to datasheet but this is the reason why the subform displays as a datasheet table rather than a regular form.
Scroll Bars	Vertical Only	If you resized the columns in the subform so that they all fit there will be no need for a horizontal scrollbar.
Navigation Buttons	No	This removes the navigation buttons (next/previous record) from the subform. They are not really needed for the subform since it is easy enough to scroll through the records. Also, having navigation buttons for both the form and subform can be confusing for users.

5. Preview the in **Form View**.

6. Save and close the form when done.

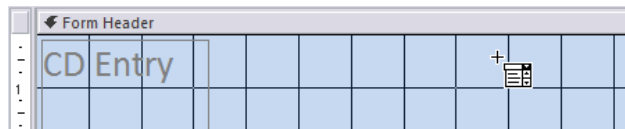
## Exercise 8. Adding Form Navigation Tools

1. Open the *CD Entry* form in design view.

We will place a navigation tool in the Form Header to the right of the title.



2. Click the **Combo Box** icon from the **Design** tab on the **Ribbon**.
3. Click in the Form Header area to the right of the Title to place a Combo Box there. A Combo Box Control Wizard will begin.



4. In the first step of the wizard, choose the third option; **Find a record on my form based on the value I selected in my combo box.**
5. Click **Next** to continue.

This wizard creates a combo box, which displays a list of values you can choose from. How do you want your combo box to get its values?

☐ I want the combo box to get the values from another table or query.

☐ I will type in the values that I want.

☒ Find a record on my form based on the value I selected in my combo box.

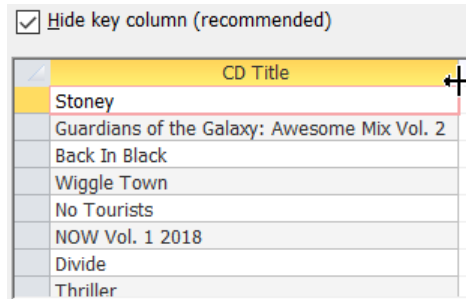
6. In the next step, select the CD# and CD Title fields as shown.

CD Type  
Label  
Released  
Purchased  
Cost

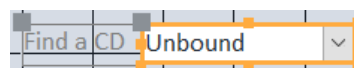
CD #  
CD Title

7. Click **Next** to continue.

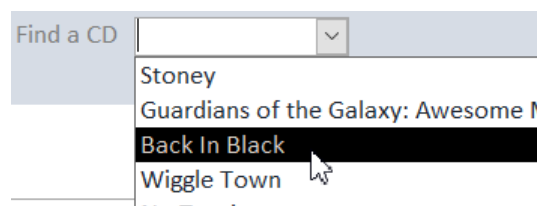
8. In the next step, make sure the Hide key column option is selected. This will mean that the list will be used to select CDs by CD# but only the CD Titles will show in the list to make it easier on the user.
9. Resize the list as shown. This will determined the width of the list when our combo box is complete.




10. Click **Next** to continue.
11. In the last step, enter *Find a CD* for the label and then click **Finish**.
12. Resize and move the combo box and label so that they are closer together as shown.



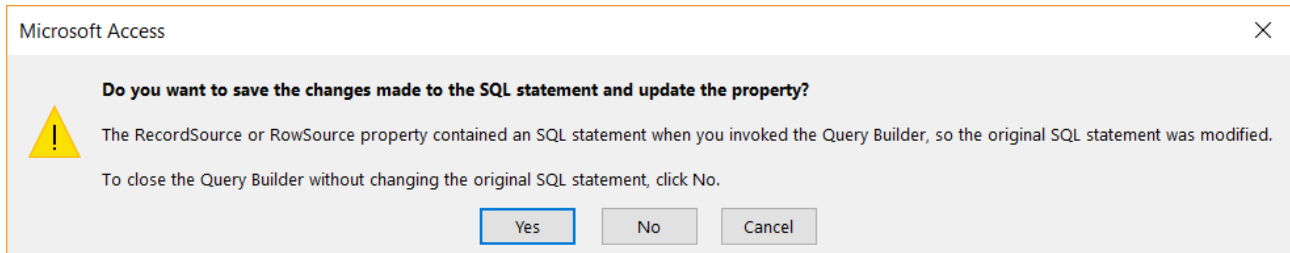
13. Preview the form in **Form View**.



14. Test the new combo box at the top of the form. When you select a CD title from the list, the details for that CD will display in the form.
15. Return to **Design View**. Now we'll sort the list in alphabetical order. This will make it easier for the user because they will be able to quickly move through the list by typing the first letters of the CD title they want.
16. Select the combo box and make sure the properties window is displayed.
17. Change the **Name** property to *cboSearchCD*.
18. Click in the Row Source property (not control source). You may remember earlier it was stated that Queries are based on SQL (Structured Query Language). This property is currently showing an example of an SQL statement. This SQL statement can be modified in a query design window just as your queries can.
19. Click the Build icon  to the right of the Row Source property. This will display a query design window, which allows you to easily edit information about what appears in the combo box's list.
20. When the Query Design window appears, change the sort order of the *CD Title* field to Ascending as shown on the next page.

Field:	CD #	CD Title
Table:	CDS	CDS
Sort:		Ascending
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		
or:		


21. Close the Query Design window.

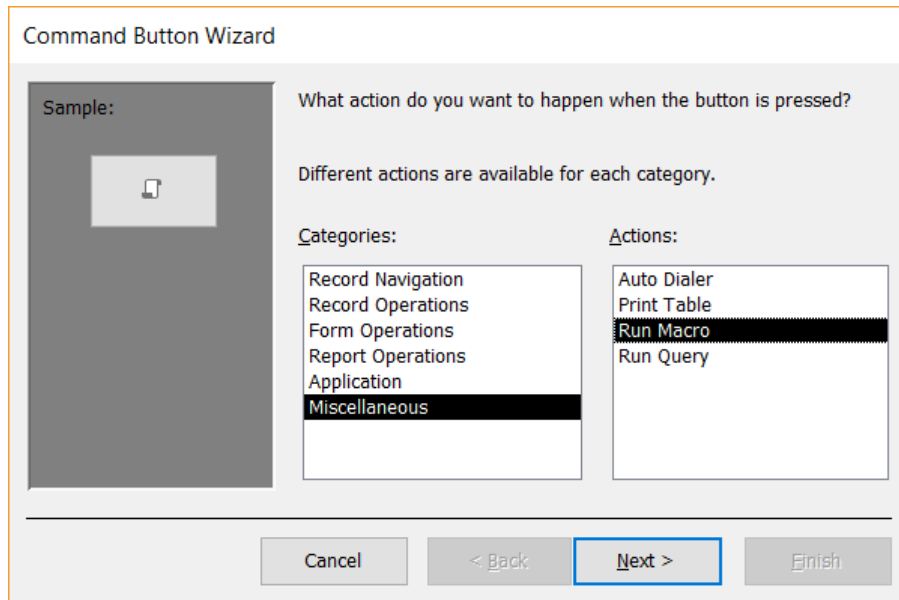


22. When prompted to save changes to the SQL statement click **Yes**. You will be returned to the form and the properties window.
23. Preview the form and test the combo box again. The CD titles in the list will now be sorted in alphabetical order. This will make it easy to select a cd title by typing its first letters.
24. Close the form, saving changes.



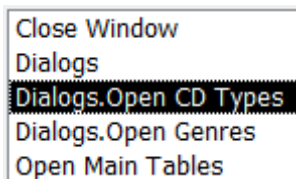
## Exercise 9. Adding Control Buttons to a Form

1. Open the *CD Entry* form in **Design View**.
2. Click the **Button** icon in the **Design** tab on the **Ribbon**. 
3. Click in the form header just to the right of the combo box. A Command Button Wizard will begin.

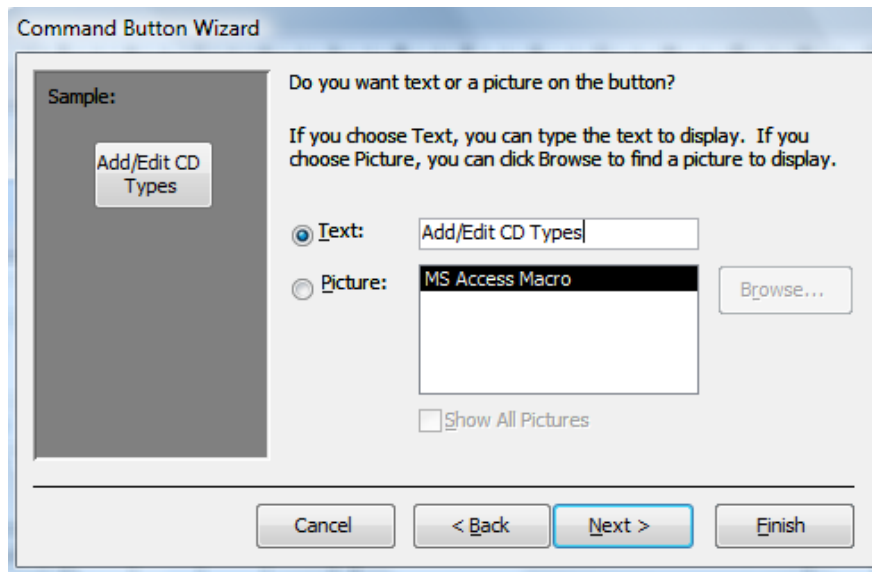


For the first step of the wizard, we choose what type of action to take when the button is clicked. We want this button to open a form. We could do it by selecting **Form Operations** and then select the form we want to open. Since we created sub macros earlier that not only open forms but specify some details about how they we will open, we will use them instead.

4. For the first step of the wizard, choose **Miscellaneous** from the left list and then **Run Macro** from the right list as shown above.
5. Click **Next** to continue.



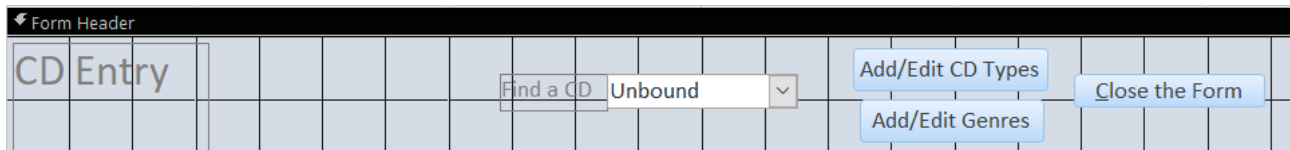
6. When the list of the macros in your database appears, select the *Dialogs.Open CD Types* Sub macro and click **Next** to continue.




7. For the next step we can choose whether to put a picture or a text caption on the form. Select the Text option and then enter *Add/Edit CD Types* for the text.
8. Click **Next** to continue.
9. For the last step in the wizard, enter *cmdEditCDTypes* for the button name and then click **Finish**.
10. Preview the form.
11. Click the new **Add/Edit CD Types** button. The *CD Types* form will open.
12. Close the *CD Types* form and then return to design view in the main form.
13. Follow the steps above to add a second button next to the one you just made. This one needs to open the Genres Form. The name property of the button should be *cmdEditGenres*.
14. Place a new command button next to the other two. Move / resize them to make room first if needed. Click **Cancel** to close the wizard when it begins.
15. Make sure the new button is selected and that the properties window is displayed.
16. Change the following properties for the new button.

Property	Change to:	Explanation
Name	cmdClose	
Caption	&Close the Form	The “&” in a caption means that the following letter will be underlined in the button. That will allow the user to activate that button by holding down the <b>[Alt]</b> key and pressing the underlined letter which in this case would be <b>[Alt] [C]</b> .
On Click	Close Window	When the button is clicked, it will activate the Close Window macro we created earlier and the form will close.
ControlTip Text	Close This Form	Provides a pop-up caption when the mouse is over the button. Similar to the tips on Ribbon icons – You may also like to add ControlTip Text for your other 2 buttons.

17. Resize and arrange the buttons so they appear as in the example below.



**Tip** You can use **Height** and **Width** properties to ensure the buttons have consistent sizes.

18. Click on the Save icon  to save changes to the form.
19. Preview the form.
20. Click on the **Add/Edit Genres** button. If it is working correctly, the *Genres Form* will open.
21. Close the *Genres Form* to return to the main form.
22. Click on the new **Close the Form** button. The form will close.

**Note** When you use either of the additional forms to add a new Genre or CD Type, the lists in your main form might not update with the changes until the next time you open that form.


## Database Maintenance

Windows includes a number of utilities to keep your computer running smoothly. Examples of these utilities are **Scandisk** or **Check Disk** (depending on the version of Windows) - which can check disks for errors and **Defrag** – which re-arranges files more efficiently on a disk. Access also includes similar utilities. Older versions of Access included a **Compact Database** utility and a **Repair Database** utility. In newer versions of Access they are combined as the **Compact and Repair** option but they will be explained here as separate features.

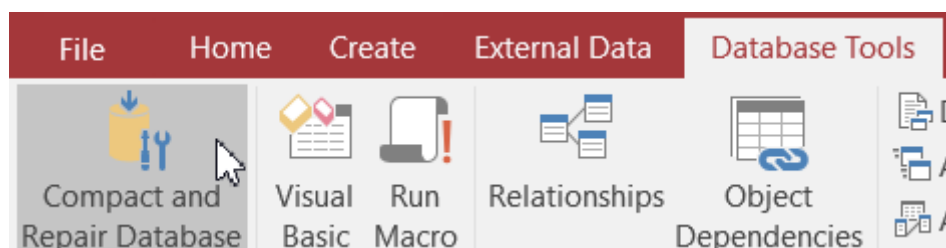
The **Repair Database** utility was designed to fix errors in the database file and is comparable to the Windows Scandisk utility. Such errors could occur if the database has been closed incorrectly (such as when the database is open during a power failure or the computer crashes) and can also arise from general database use. These errors can result in the database randomly crashing or running slowly. The Repair Database utility can fix such problems reliably. It is a good idea to run it on a regular basis as this will often detect and fix errors that have not yet begun to cause problems.

The **Compact Database** utility will reorganise all of the data in a database so that it is stored more efficiently. It is similar to the Windows Defrag utility. Consider the example of packing a suit case. If you throw things in carelessly, you won't be able to fit much in and it will take longer to find things in there. If you pack things neatly, everything can fit in less space and will take less time to find. Likewise, using the Compact database utility on your database file makes the database store data more efficiently so that the database file will be smaller and will run faster. Since general database use can clutter things up, it is a good idea to use this utility on a regular basis. It can often more than halve the size of the database file which could be the difference between fitting on a disk and not fitting.

### Exercise 10. Running Compact & Repair on a Database

Before commencing this exercise, you might like to open Windows Explorer (you can use the  E shortcut) to check the current file size of your database. Note that if the database is stored on a network, you might not be able to use these tools if other people are also using the database.

1. From the **Ribbon** menu, select **Database Tools** and then **Compact and Repair Database**. If you don't already have the database open, you will be prompted for the location of the file.



2. You may want to check the file size of your database afterward to see how much difference it has made to the file size.

**Note** The bigger the database, the longer it will take to run the **Compact and Repair** utility.

## Creating an Interface

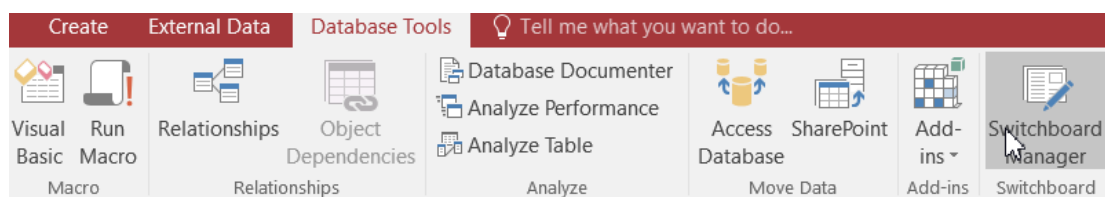
Now it's time to put the final touches on the database by creating a user friendly interface. We could create a front end by creating a blank form and manually adding command buttons to access the various objects in the database. We will do it instead by using the simple **Switchboard Manager**. We will also use the Access **Startup Options** to change what happens when the database is opened.

### Exercise 11. Creating a Switchboard

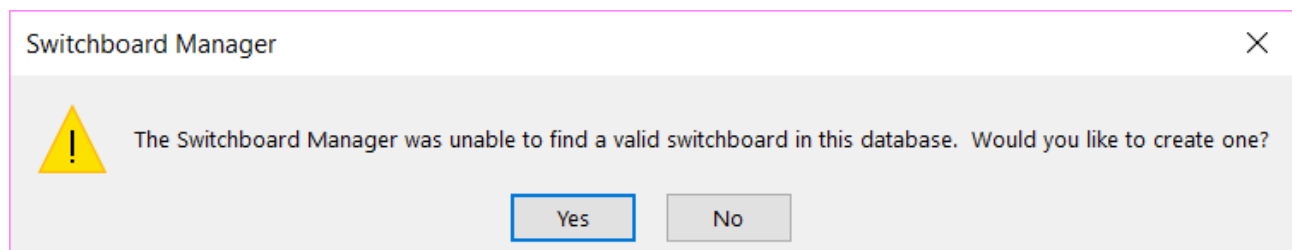
A switch board is a database menu that consists of two parts. A form and a table containing the menu options that will appear on the form. The two components of a switchboard can be edited directly, such as formatting the form, though it is generally best to limit your changes to the **Switchboard Manager**.

**Note** In access 2016, the switchboard manager will not always appear on your Ribbon bar. If that is the case, then follow the steps on the [last page](#) of this exercise to add the switchboard manager icon to your ribbon bar.

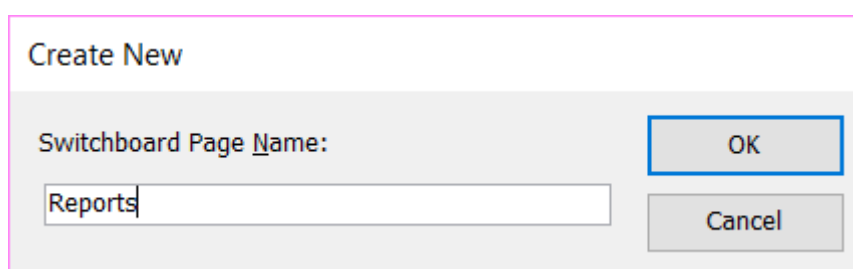
- 1) From the **Ribbon** menu, select **Database Tools** and then click **Switchboard Manager**.



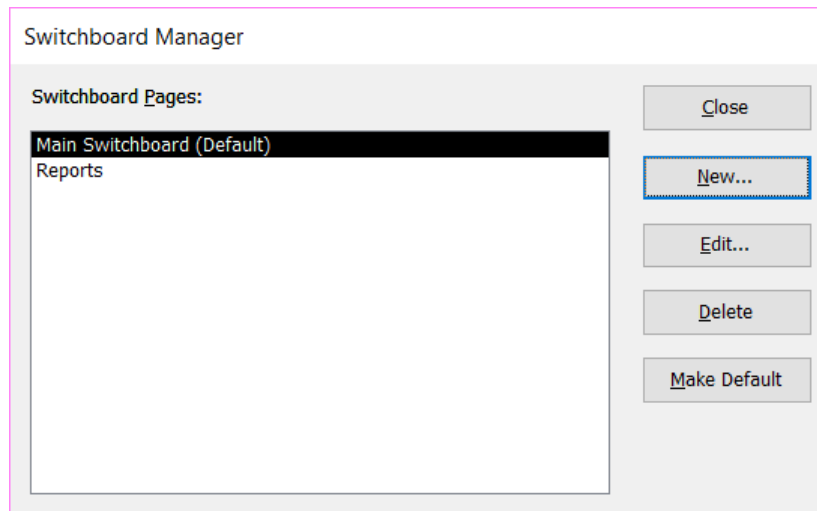
- 2) Because there is no existing switchboard, Access asks you if you want to create one. Click Yes. A new main switchboard will be created and we will use this as our main menu.



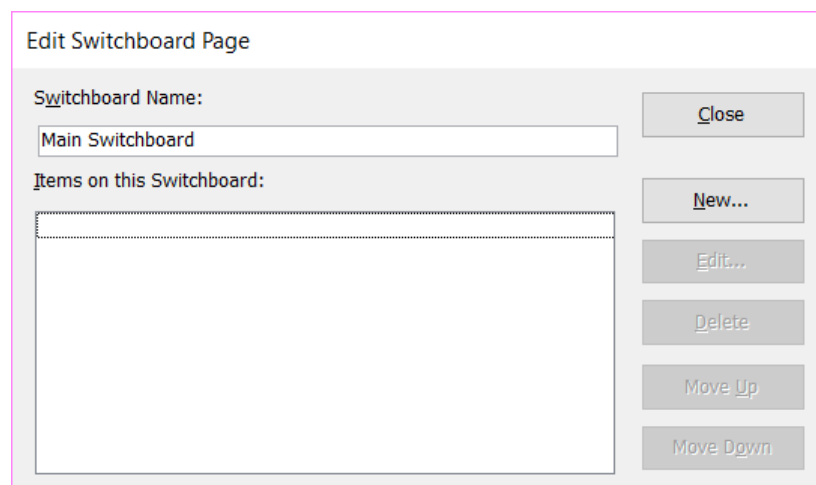
- 3) We also want to create another switchboard to be used as a Reports sub menu so click **New** to create a second switchboard.



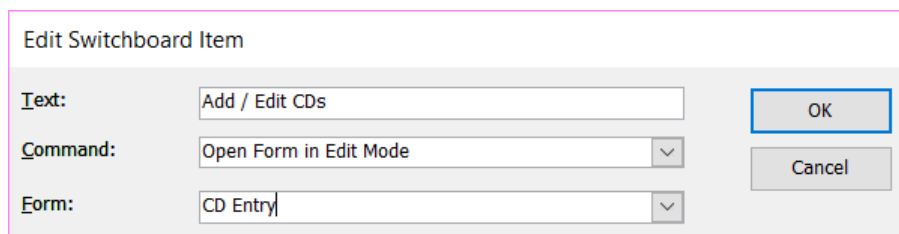
- 4) Enter *Reports* for the name of the new Switchboard and click **OK**. The Switchboard Manager will display both the main switch board menu which it created to begin with, and the Reports menu.



3. Make sure the *Main Switchboard* is selected and click **Edit**. Now we can begin to add menu items to this switchboard.



- 5) We'll start by adding a menu option which will open our main editing form. Click **New**.



- 6) Complete the **Switchboard Items** dialog as shown above and then click **OK**.  
 7) Add additional **Switchboard Items** as shown below.

**Edit Switchboard Item**

<b>T</b> ext:	Database Reports	<b>OK</b>
<b>C</b> ommand:	Go to Switchboard	<b>Cancel</b>
<b>S</b> witchboard:	Reports	

**Edit Switchboard Item**

<b>T</b> ext:	Add / Edit CD Types	<b>OK</b>
<b>C</b> ommand:	Open Form in Edit Mode	<b>Cancel</b>
<b>F</b> orm:	CD Types Form	

**Edit Switchboard Item**

<b>T</b> ext:	Add / Edit Genres	<b>OK</b>
<b>C</b> ommand:	Open Form in Edit Mode	<b>Cancel</b>
<b>F</b> orm:	Genres Form	

**Edit Switchboard Item**

<b>T</b> ext:	Exit the Database	<b>OK</b>
<b>C</b> ommand:	Exit Application	<b>Cancel</b>

**Edit Switchboard Page**

<b>S</b> witchboard Name:	Main Switchboard	<b>C</b> lose
<b>I</b> tems on this Switchboard:	<ul style="list-style-type: none"><li>Add / Edit CDs</li><li>Database Reports</li><li>Add / Edit CD Types</li><li>Add / Edit Genres</li><li>&amp;Exit the Database</li></ul>	<b>N</b> ew... <b>E</b> dit... <b>D</b> elete <b>M</b> ove <b>U</b> p <b>M</b> ove <b>D</b> own

8) Click **Close** to return to the **Switchboard Manager**.

- 9) Now to create the items on the *Database Reports* Sub menu. Select *Reports* from the main **Switchboard Manager** and click **Edit**.
- 10) Add the following Menu Items.

Edit Switchboard Item

Text:	Complete CD Listing	OK
Command:	Open Report	Cancel
Report:	Complete CD Listing	

Edit Switchboard Item

Text:	Number of Songs in Each Genre	OK
Command:	Open Report	Cancel
Report:	Number of songs in each Genre	

Edit Switchboard Item

Text:	Song Search by Artist	OK
Command:	Open Report	Cancel
Report:	Song search by artist	

Edit Switchboard Item

Text:	Song Search by Genre	OK
Command:	Open Report	Cancel
Report:	Songs by Genre	

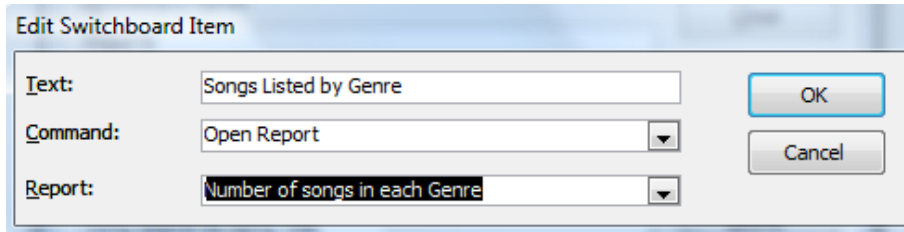
Edit Switchboard Item

Text:	Song Search by Song Title	OK
Command:	Open Report	Cancel
Report:	Song search by song title	

Edit Switchboard Item

Text:	Song Search by Year	OK
Command:	Open Report	Cancel
Report:	Song Search By Year	





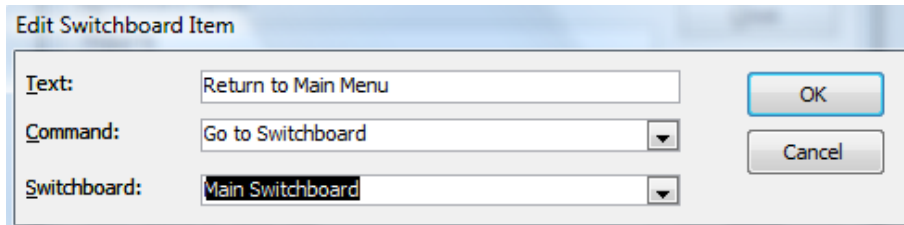
**Edit Switchboard Item**

**Text:** Songs Listed by Genre

**Command:** Open Report

**Report:** Number of songs in each Genre

OK Cancel




**Edit Switchboard Item**

**Text:** Return to Main Menu

**Command:** Go to Switchboard

**Switchboard:** Main Switchboard

OK Cancel

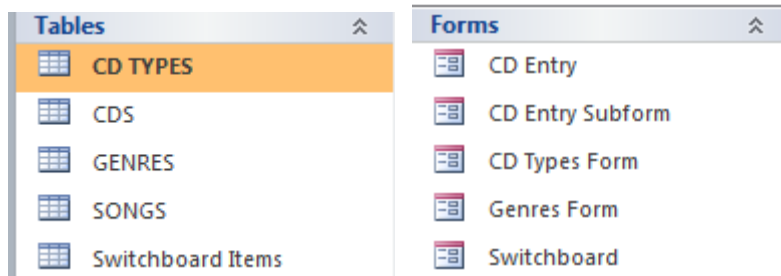


**Items on this Switchboard:**

- Complete CD Listing
- Number of Songs in Each Genre
- Song Search by Artist
- Song Search by Genre
- Song Search by Song Title
- Song Search by Year
- Songs Listed by Genre
- Return to Main Menu

New... Edit... Delete Move Up Move Down

- 11) Click **Close** twice to exit the **Switchboard Manager**.



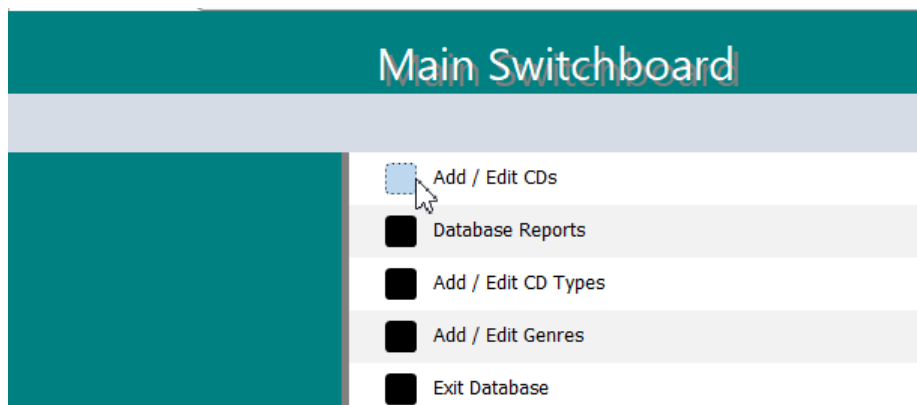
**Tables**

- CD TYPES
- CDS
- GENRES
- SONGS
- Switchboard Items

**Forms**


- CD Entry
- CD Entry Subform
- CD Types Form
- Genres Form
- Switchboard

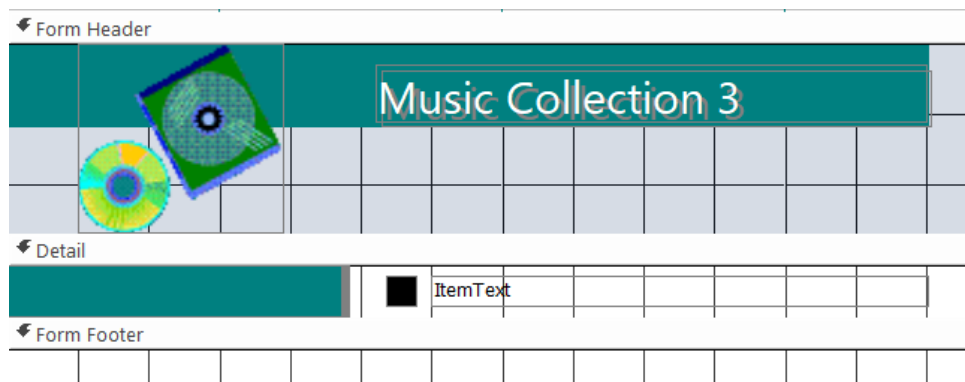
- 12) A new form has been created for your switchboard and a new table has been created to store the switchboard menu items.
- 13) Open the *Switchboard* form to test it.



**Main Switchboard**

- Add / Edit CDs
- Database Reports
- Add / Edit CD Types
- Add / Edit Genres
- Exit Database

- 14) Click on the **Add/Edit CDs** option. Your main editing form will open.
- 15) Click the **Close the Form** button to return to the switchboard.
- 16) Click on the **Database Reports** option to change to the sub-menu.
- 17) Click on some of the different options to preview the reports in the database and close each report after previewing it.
- 18) Click the **Return to the Main Menu** option.
- 19) Close the switchboard.
- 20) Open the *Switchboard* form in **Design View**.
- 21) Click the **Insert Image** icon on the **Ribbon**. 
- 22) Click the **Browse** option under the icon to select an image file.
- 23) When the **Insert Picture** dialog appears, select the *cds.gif* file (or another suitable picture) and click **OK**.
- 24) Click in the blank area to the left of the switchboard's menu to place a picture.
- 25) Position and resize the picture to the left of the title as shown.



- 26) Save and close the form.

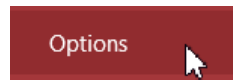
## Exercise 12. Setting Database Startup Options

The **Startup** options are used to set what happens when the database opens.

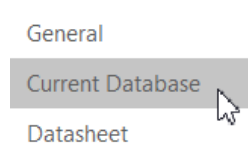
1. From the **Ribbon** choose the **File** tab.




2. From the **File** menu select **Options**.



3. From the list of categories on the left select **Current Database**.

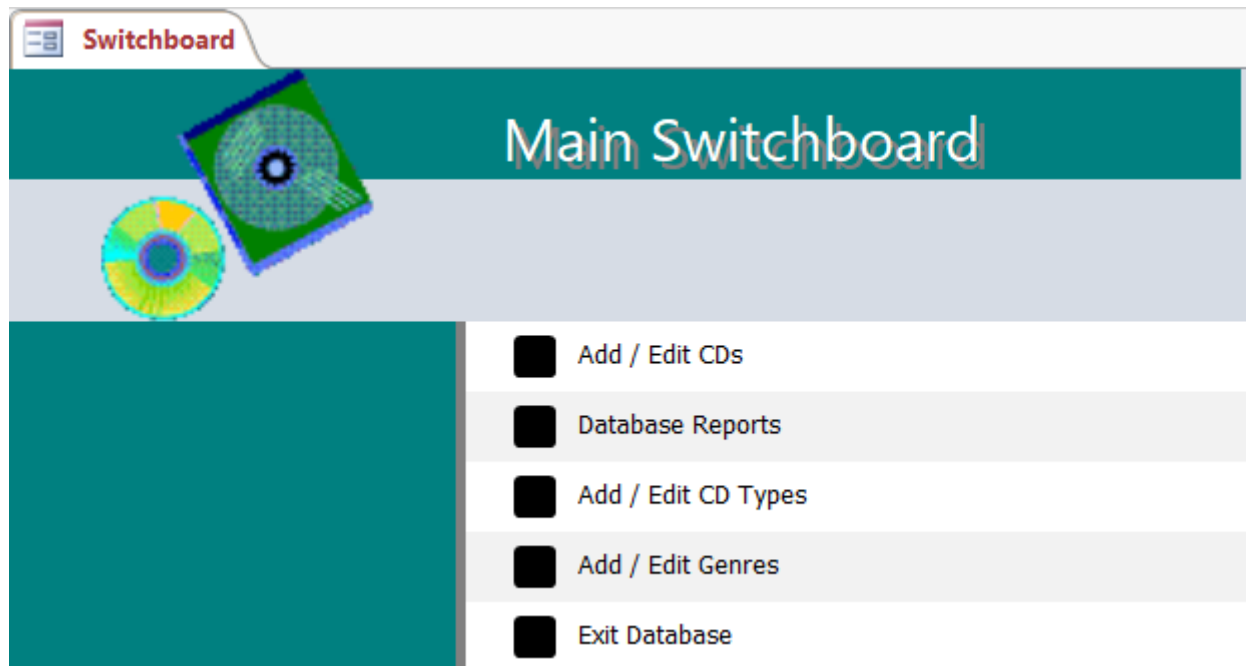


4. Make the following changes in your Startup options.

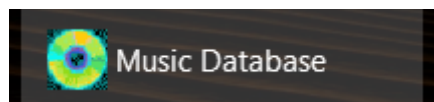
Property	Change to:	Explanation
Application Title	Music Database	Displays the title at the top of the window instead of the name of the program.
Display Form/Page:	Switchboard	The selected form will open as soon as the database opens. In this case we want our menu to be the first thing to appear.
Application Icon	cds_icon.bmp	You can browse for a custom icon to be used at the top of the screen when the database is open instead of the usual access icon.  You might need to change the file type to bmp when you are opening the file.
Display Navigation Pane	Uncheck this option	The Navigation Pane will not display when the database opens. This avoids the problem of inexperienced users accidentally removing tables or other crucial objects. Remember you can always display it when you need it by pressing <b>[F11]</b> .

5. Click **OK** to confirm these settings.

6. Close the database and then re-open it. Your Switchboard will open and the Database Window will not appear.



Notice also that the application name and icon both appear at the top of the window and in the taskbar.



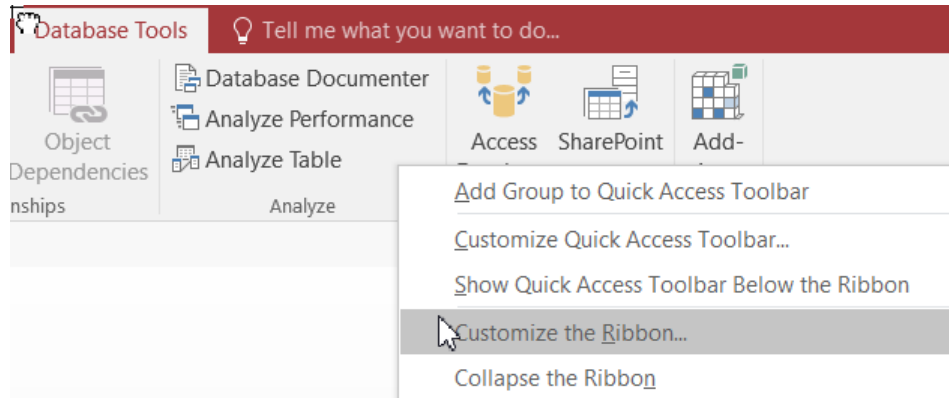
7. Test some of the menu options and then click **Exit Database**.

**Tip** You can also specify additional actions to take place when the database is opened by placing them in a macro called *Autoexec*. For example, you could create a macro that opens several forms at startup.

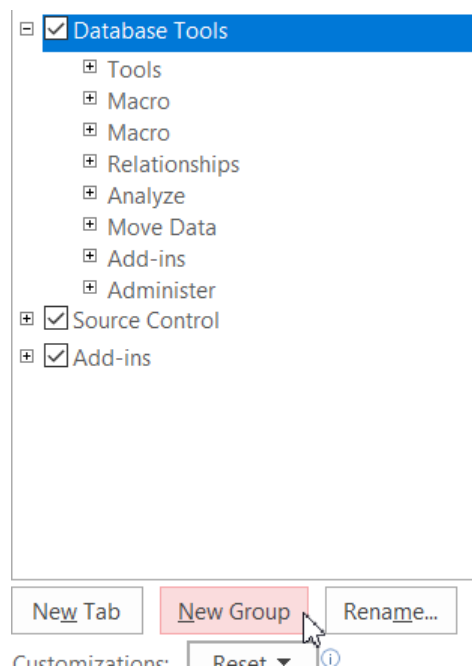
### Exercise 13. Adding the Switchboard icon to the Ribbon Bar

In recent versions of Access, the switchboard manager icon will often appear on the Ribbon when you are working on an old access database file, but not when you are working on one in the new database format. If that is the case, then you can add the icon to a custom ribbon group.

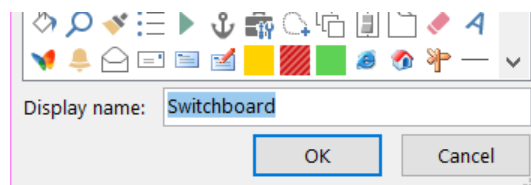
1. **Right-click** anywhere on the ribbon and select **Customise the Ribbon**.



2. In the right column, select **Database Tools** and then click **New Group**.



3. Click **New Group** and Click **Rename** to name the group **Switchboard**.



4. Click **OK** to return to the Options window.
5. In the **Choose commands from:** combo box at the top of the window, select **All Commands**.
6. Scroll down and highlight **Switchboard Manager**.
7. Click **Add** to add the **Switchboard Manager** to the new group you created under the **Database Tools** tab.

8. Click **Ok** to close the Access Options.
9. You should now see the **Switchboard Manager** under the **Database Tools** tab.

