# Unit 7 - Exercises

## ADO.NET

To complete this exercise, download the Visual Studio Starter Solution from [here](https://thethirstycoder.gitlab.io/adev-2005-learn/unit_7/downloads/adev-2005_unit_7_exercise_files.zip).

This solution serves as the starting point for the exercise. The solution contains a single Windows Form Application project called ContactsApp. The ContactsApp project has one form class called ContactsForm, which is the starting point of the application. The form contains a menu strip and a data grid view.

The database the application will interface with is in the ContactsApp\bin\Debug directory and is called Contacts.mdb. It is recommended to connect to the database in the Server Explorer panel of Visual Studio prior to beginning the exercise.

**EX1**. Declare Form Class Variables (Fields)

Add the following fields in the ContactForm class.

* \_dataAdapter - Reference to a data adapter object.
* \_dataset - Reference to a data set object.
* \_bindingSource - Reference to a binding source object.

**EX2**. Query the Database

In the constructor of the ContactForm class, query the database for all contacts in the Contact table.

1. Construct an instance of a connection object. Set the connection object’s connection string to the connection string for the provided database.
2. Open a connection to the database.
3. Run the application.
   * If an exception occurs, you’ve done a previous step incorrectly. -If an exception does not occur, you are on the right track.
4. Construct an instance of a command object. Set the command object’s command text to the SQL required to query all rows and columns from the Contacts table. The command object will also need a reference to the connection object.
5. Construct an instance of a data adapter object. Set the data adapter object’s select command to the command object constructed in step 4.
6. Construct an instance of a data set object.
7. Using the data adapter object, fill the data set. Give the table a name of “Contacts”.
8. Run the application.
   * If an exception occurs, you’ve done a previous step incorrectly.
   * If an exception does not occur, you are on the right track.

**EX3**. Populate DataGridView

Continue coding in the constructor of the Form class.

1. Construct an instance of a binding source object.
2. Handle the Load event of the Form.

In the handler method, code the following:

1. Set the binding source’s data source to the “Contacts” table in the data set.
2. Set the data grid view’s data source to the binding source object.
3. Run the application.
   * Two records will appear in the data grid view along with an empty row for adding new records.
   * If no records appear, check your SQL query or the data binding.
   * If an exception occurs, you’ve done a previous step incorrectly.
4. Set the “ID” column in the data grid view such that it is hidden.
5. Run the application and confirm the “ID” field is not visible.
6. Handle the Save main menu item’s Click event.

**EX4**.Updating the Database

In the Save button Click event handler, code the following:

1. Using the data adapter, update the DataSet. You will need to specify that the table “Contacts” will be updated.
2. Run the application:
3. Edit any cell in either row. Press the Tab key or click another cell to end the edit.
4. Click the Save main menu item or Ctrl+S.
5. Exit the application.

Run the application again:

### Questions

1. The edit you made was not saved to the database. Why?

In the constructor for the form class, add the following code:

1. Construct an instance of a command builder object. Set the command builder object’s data adapter to the instance of the data adapter.
2. Use the command builder to set the other three data adapter commands.
3. Run the application:
4. Edit any cell in either row. Press the Tab key or click another cell to end the edit.
5. Click the Save main menu item or Ctrl+S.
6. Exit the application.

Run the application again:

### Questions

1. The edit you made was not saved to the database. Why?

In the Save button Click event handler, code the following:

1. End the edit of the binding source object before the data adapter update.
2. Run the application:
3. Edit any cell in either row. Press the Tab key or click another cell to end the edit.
4. Click the Save main menu item or Ctrl+S.
5. Exit the application.

Run the application again:

### Questions

1. The edit you made is now saved to the database…finally. Why does it work now?

Test deleting a row:

1. Run the application.
2. Click the header cell of an existing row. This will select the entire row, rather than a single cell.
3. Press the delete button.
4. Save your changes.
5. Exit the application and start it again to verify the deletion was saved to the database.

Test inserting a new row.

1. Run the application
2. Click the first cell in the bottom row.
3. Fill in all the fields with information about yourself.
4. Save your changes.
5. Exit the application and start it again to verify the insertion was saved to the database.

**EX4**. Detecting Changes to the Data

In the Load event handler of the Form, code the following:

1. Handle the click event of the Has Changes menu item.

In the Has Changes menu item’s Click event handler method, code the following:

1. Show a dialog (message) box that shows if changes were made to the data within the data set. The dialog box will display “Has Changes: {true/false}”, where the true/false is a result of one of the data set’s methods.

Run the application.

1. Click the Has Changes menu item. The result is false.
2. Edit any cell in the data grid view. Press the tab key or click another cell to end the edit.
3. Click the Has Changes menu item.

### Questions

1. The result is still false. Why?

In the Has Changes menu item’s Click event handler method, code the following:

1. Add a statement at the beginning of the method to end the edit of the binding source.

Run the application.

1. Click the Has Changes menu item. The result is false.
2. Edit any cell in the data grid view. Press the tab key or click another cell to end the edit.
3. Click the Has Changes menu item.

### Questions

1. The result is now true. Why?

**EX5**. Analyze RowState

In the Load event handler method of the Form class, code the following:

1. Handle the Click event of the RowState main menu item.

In the RowState main menu item’s Click event handler method, code the following:

1. End the edit of the binding source.
2. Construct an instance of the TableInformationForm class. Pass the data set as an argument. Show this form as a modal window.

Run the application the application and make changes to the data. After a making change, click the RowState menu item to verify how the RowState property of the DataRow within the DataTable is changed.