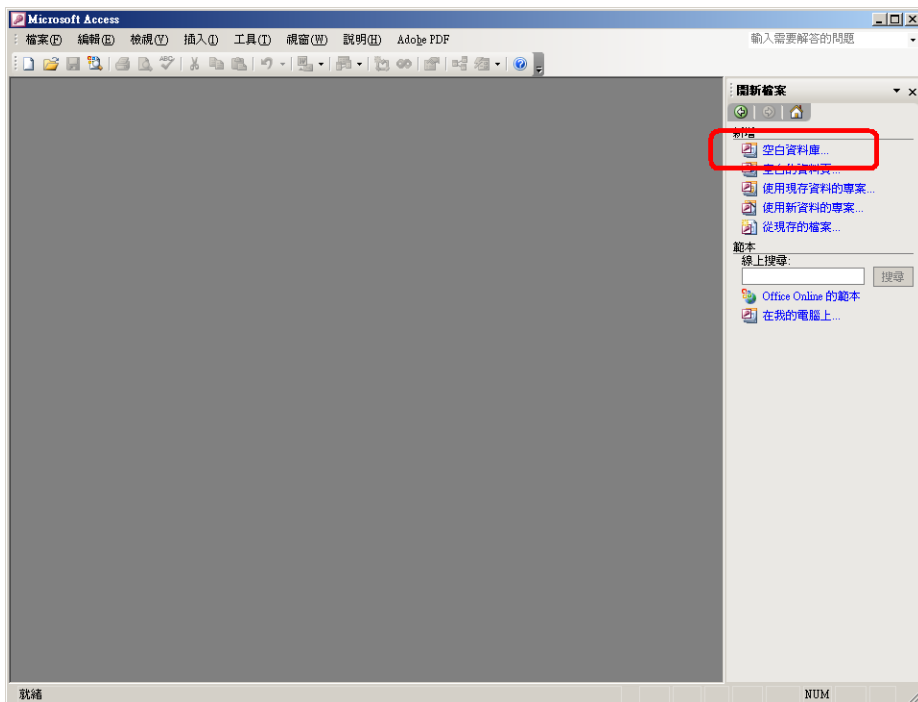
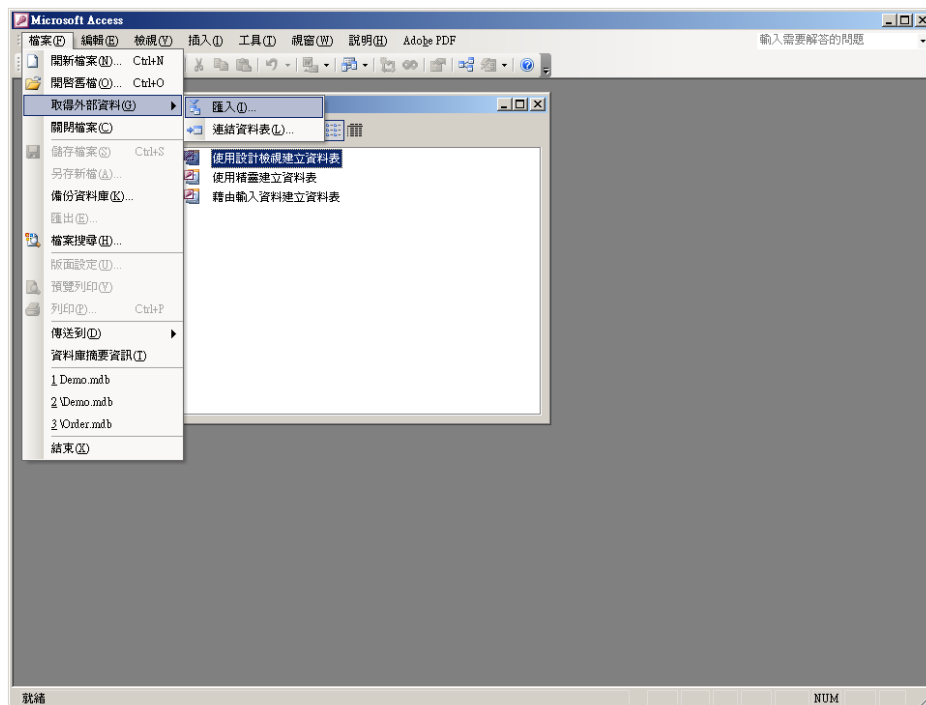


1. Import Excel to Microsoft Access

1. Download the lab exercise file from <http://www.peter-lo.com/Teaching/I154-1-A/Source08.zip>, and then start Microsoft Access and select **Access Blank Database** to create the new database “Demo.mdb”.



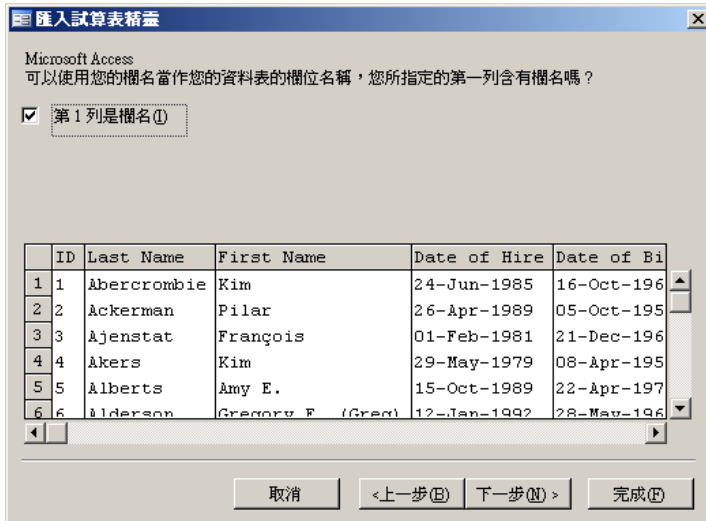
2. Select **File → Import External Data → Import**



3. Select the worksheet to be import. In this example, the only worksheet is “Original”



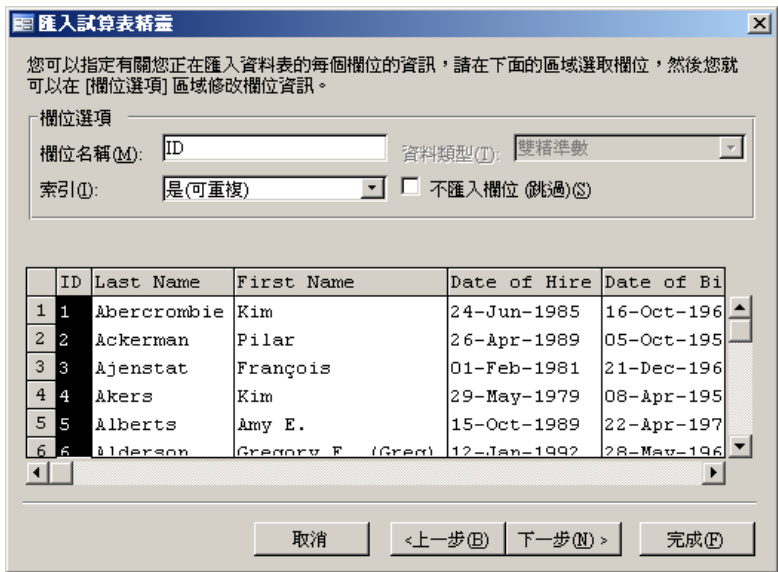
4. Select the worksheet to be import. In this example, the only worksheet is “Original”



5. You can import the data to the existing database or a new one.



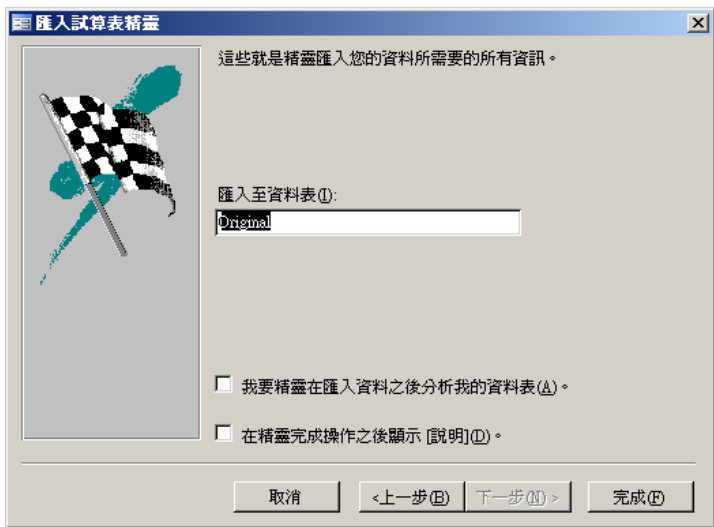
6. Define which column will be imported to the database.



7. Define the column “ID” as the index.



8. Define the database to be imported. We will use “Original” in this example



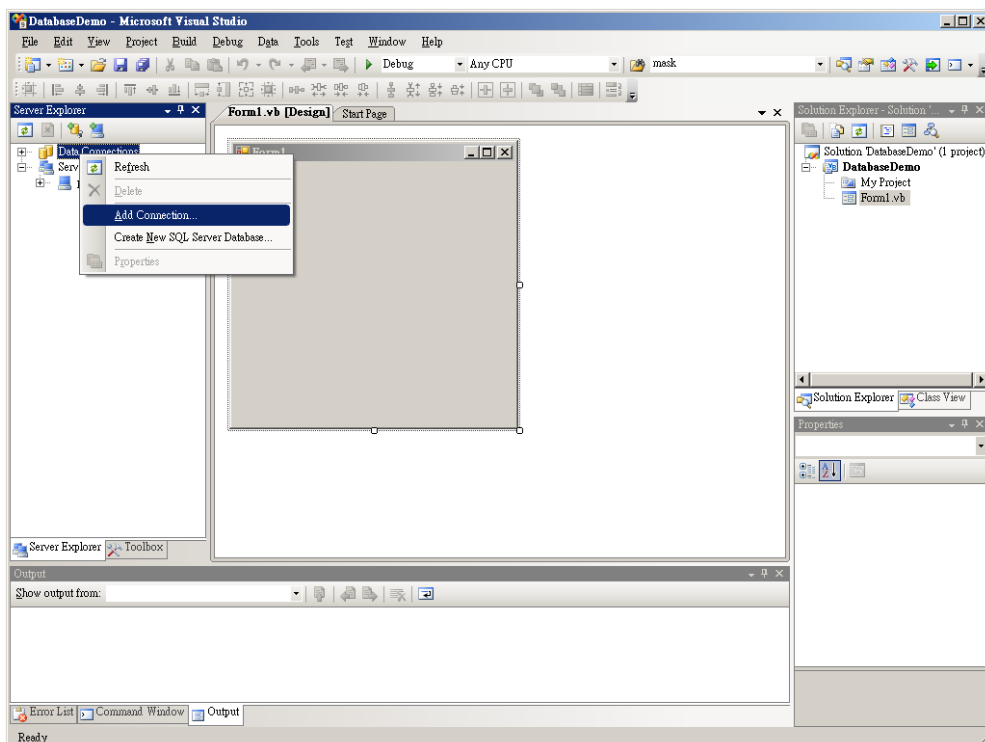
9. You can browse the table “Original” to display the imported data.

ID	Last Name	First Name	Date of Hire	Date of Birth	Dept	Salary	Age
1	Abercrombie	Kim	24/6/1985	16/10/1962	D	HK\$91,000.00	46
2	Ackerman	Pilar	26/4/1989	5/10/1950	E	HK\$31,000.00	58
3	Ajenstat	François	1/2/1981	21/12/1964	C	HK\$48,000.00	43
4	Akers	Kim	29/5/1979	8/4/1958	C	HK\$47,000.00	50
5	Alberts	Amy E.	15/10/1989	22/4/1970	D	HK\$60,000.00	38
6	Alderson	Gregory F. (Greg)	12/1/1992	28/5/1964	D	HK\$100,000.00	44
7	Alexander	Sean P	6/7/2000	16/10/1962	E	HK\$29,000.00	46
8	Anderson	Nancy	6/6/1986	20/6/1976	C	HK\$68,000.00	32
9	Bacon Jr.	Dan K.	6/4/1988	14/1/1961	B	HK\$85,000.00	47
10	Bankert	Julie	21/4/1983	24/1/1977	C	HK\$100,000.00	30
11	Barbariol	Angela	16/2/1996	18/4/1964	A	HK\$92,000.00	44
12	Barnhill	Josh	26/7/1977	12/10/1954	E	HK\$83,000.00	54
13	Barr	Adam	9/7/1993	8/4/1959	C	HK\$66,000.00	49
14	Bashary	Shay	14/3/1995	30/1/1972	B	HK\$35,000.00	36
15	Beck	Bradley	20/12/1998	5/2/1958	C	HK\$70,000.00	50
16	Ben-Sachar	Ido	21/9/1993	12/7/1956	A	HK\$57,000.00	52
17	Benson	Max	22/11/1977	2/9/1975	B	HK\$97,000.00	33
18	Berge	Karen	1/6/1997	27/2/1965	B	HK\$35,000.00	43
19	Berglund	Andreas	19/3/1988	21/10/1954	E	HK\$93,000.00	54
20	Berndt	Matthias	17/3/1991	28/1/1978	B	HK\$48,000.00	30
21	Berry	Jo	20/9/2000	7/1/1966	D	HK\$76,000.00	42
22	Bolender	Corinna	20/8/1998	18/11/1951	D	HK\$43,000.00	56
23	Bonifaz	Luis	10/6/1996	4/4/1968	D	HK\$77,000.00	40
24	Boseman	Randall	18/10/1976	2/4/1971	C	HK\$74,000.00	37
25	Boseman	Randall	3/6/1986	13/1/1963	B	HK\$55,000.00	44
26	Bourne	Stephanie	18/1/1986	10/9/1953	C	HK\$82,000.00	55
27	Bourne	Stephanie	30/3/1977	5/6/1951	B	HK\$33,000.00	57
28	Bradley	David M.	3/2/1986	3/1/1972	A	HK\$65,000.00	36
29	Bradley	David M.	28/9/1997	24/2/1956	C	HK\$50,000.00	52
30	Bradley	David M	13/7/1984	28/2/1954	A	HK\$33,000.00	54
31	Bready	Richard	22/12/1984	11/6/1956	C	HK\$27,000.00	52
32	Bremer	Ted	4/7/1989	26/3/1959	D	HK\$99,000.00	49
33	Brown	Jo	29/1/1982	16/10/1968	E	HK\$35,000.00	40
34	Brown	Jo	11/12/1989	6/3/1977	C	HK\$34,000.00	31
35	Browne	Kevin F	15/7/1998	18/11/1956	D	HK\$83,000.00	51
36	Brundage	Michael	25/9/1988	7/10/1973	B	HK\$26,000.00	35
37	Burke	Brian	30/6/1983	23/4/1977	B	HK\$83,000.00	31

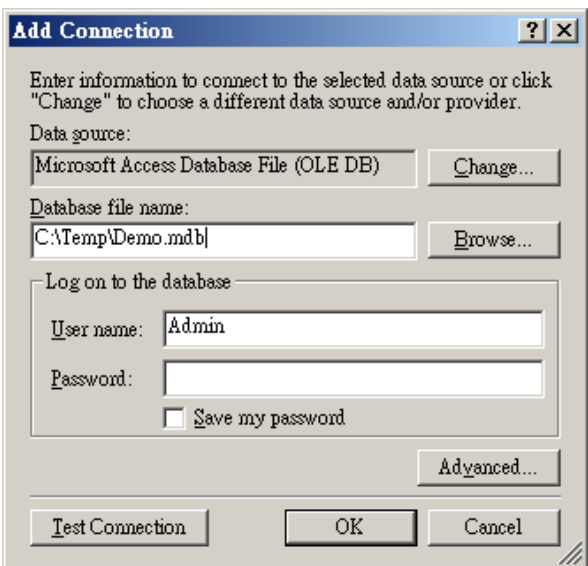
2. Create Database Connection

1. Open the Microsoft Visual Studio and start a new Visual Basic Project **DatabaseDemo** and customize the properties. Switch to the **Solution Explorer**, then right click on the **Data Connection** and select **Add Connections**.

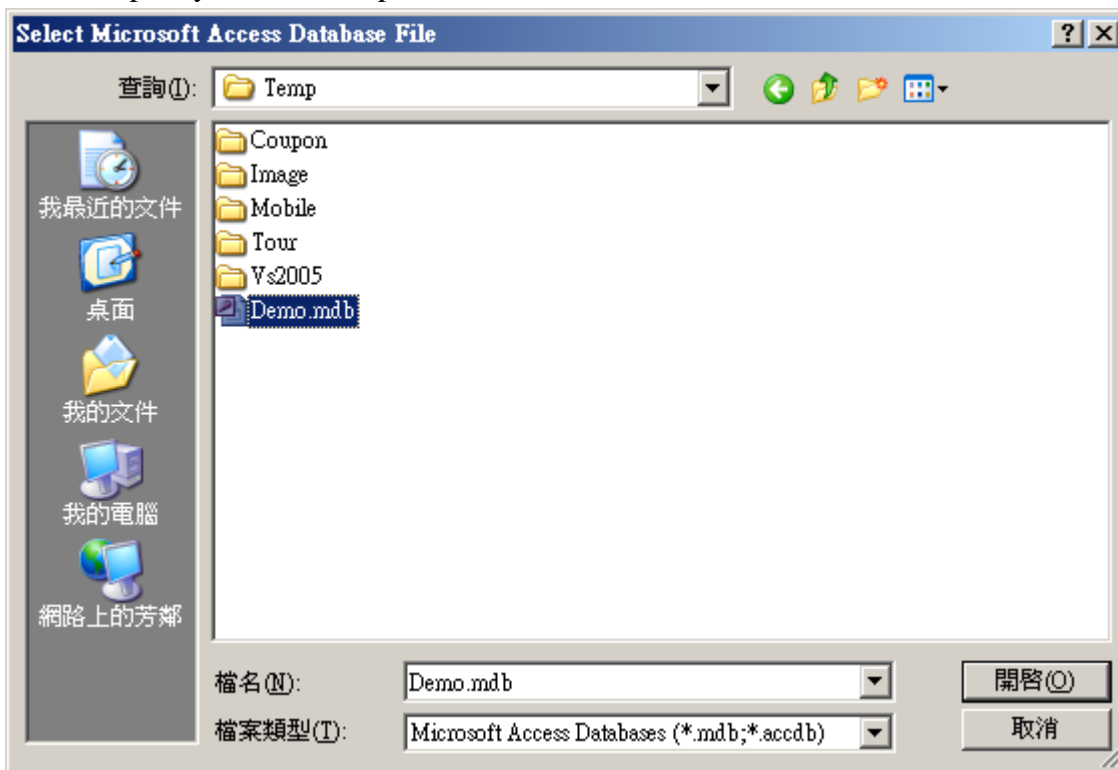
Object	Name	Property	Property Value
Form	frmMain	Text	Database Demo



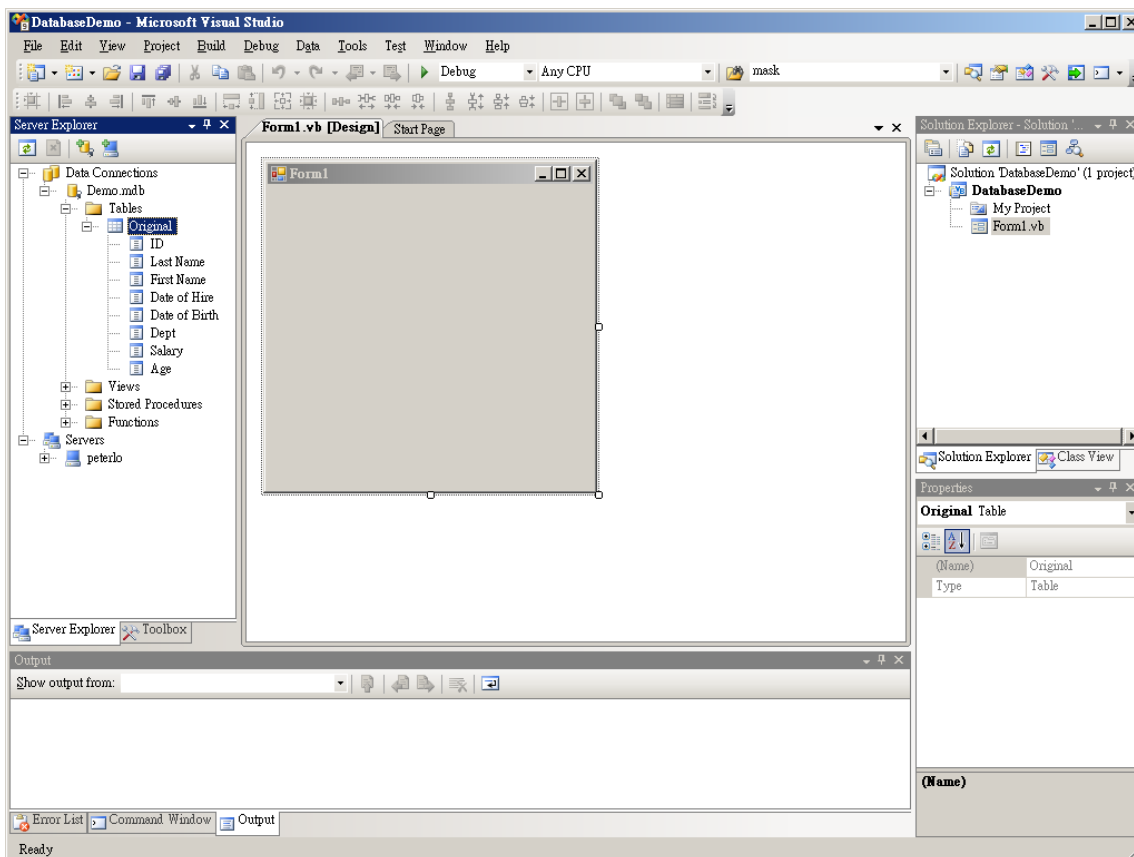
2. In the **Add Connection** dialog box, select the **Microsoft Access Database File (OLE DB)** as the **Data source**,



- Press the **[Browse]** button in **Database file name**. Select the access database file **order.mdb** in the **Select Database** dialog box, and then press **[OK]** button to access the database file. You can also specify the network path for remote connection.

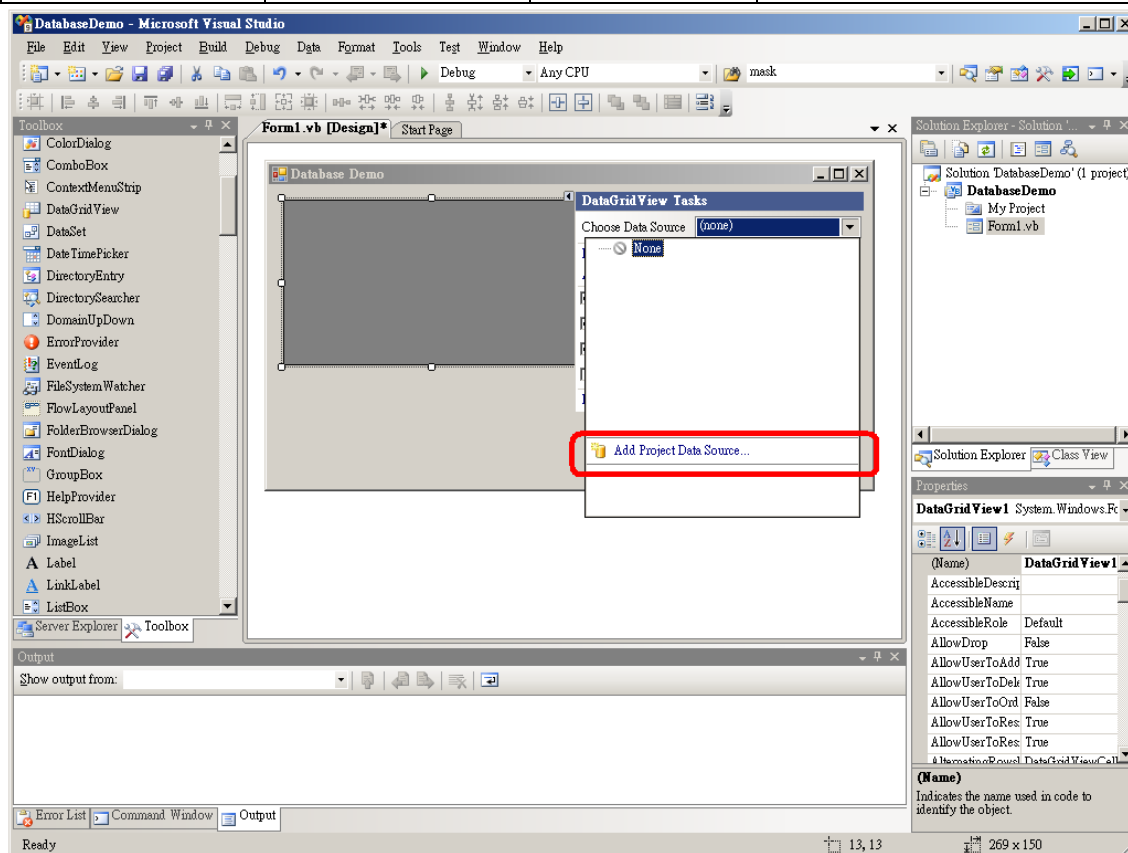


- After the connection created, you can display the database structure.

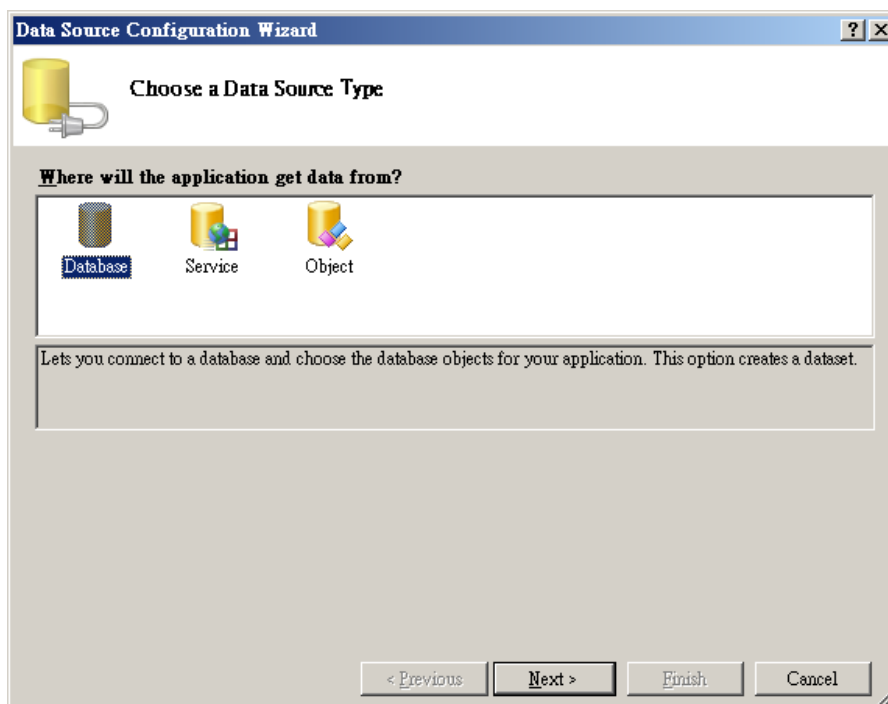


- From the Toolbox, drag a **DataGridView** controls onto the form and customize the properties. Then press the **Add Project Data Source** in the **DataGrid View Tasks**

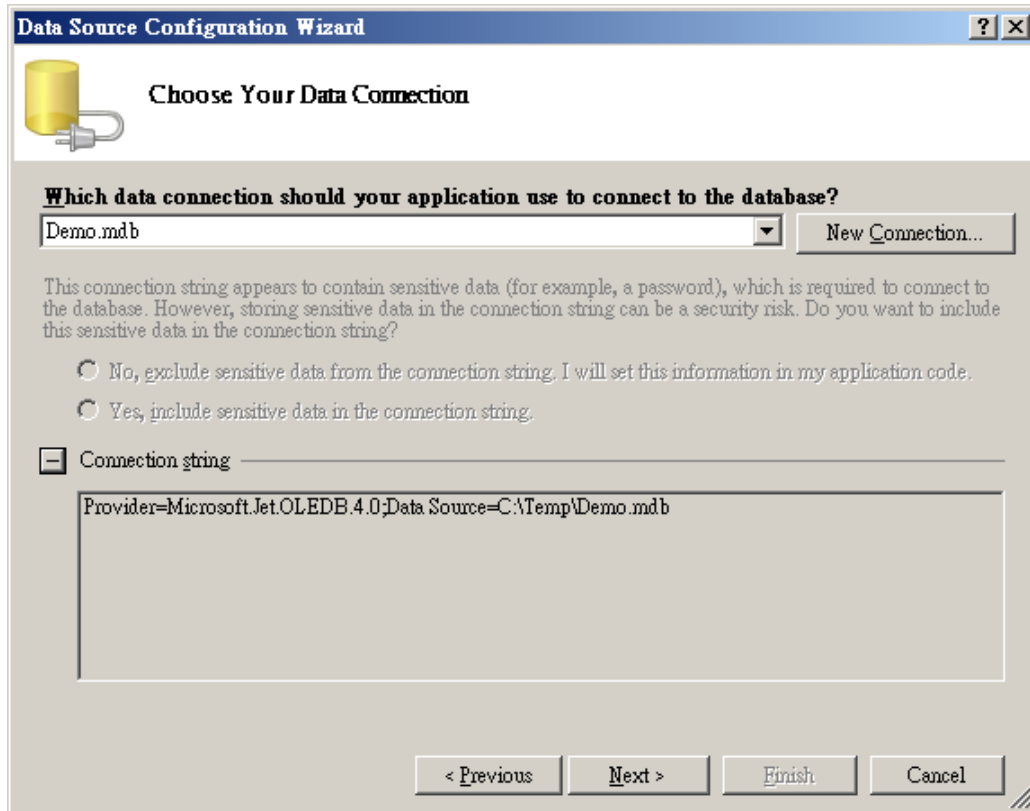
Object	Name	Property	Property Value
DataGridView	DataGridView1	Dock	Fill



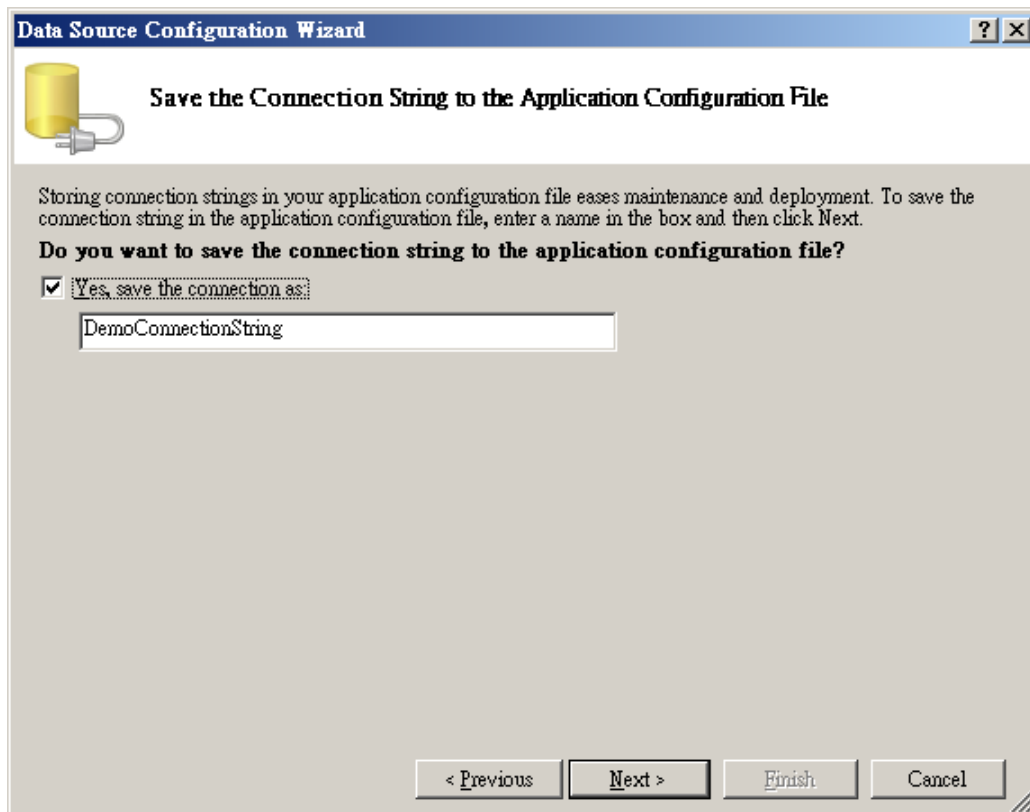
- Select **Database** in the **Data Source Configuration Wizard** dialog, and then press **[Next>]** button.



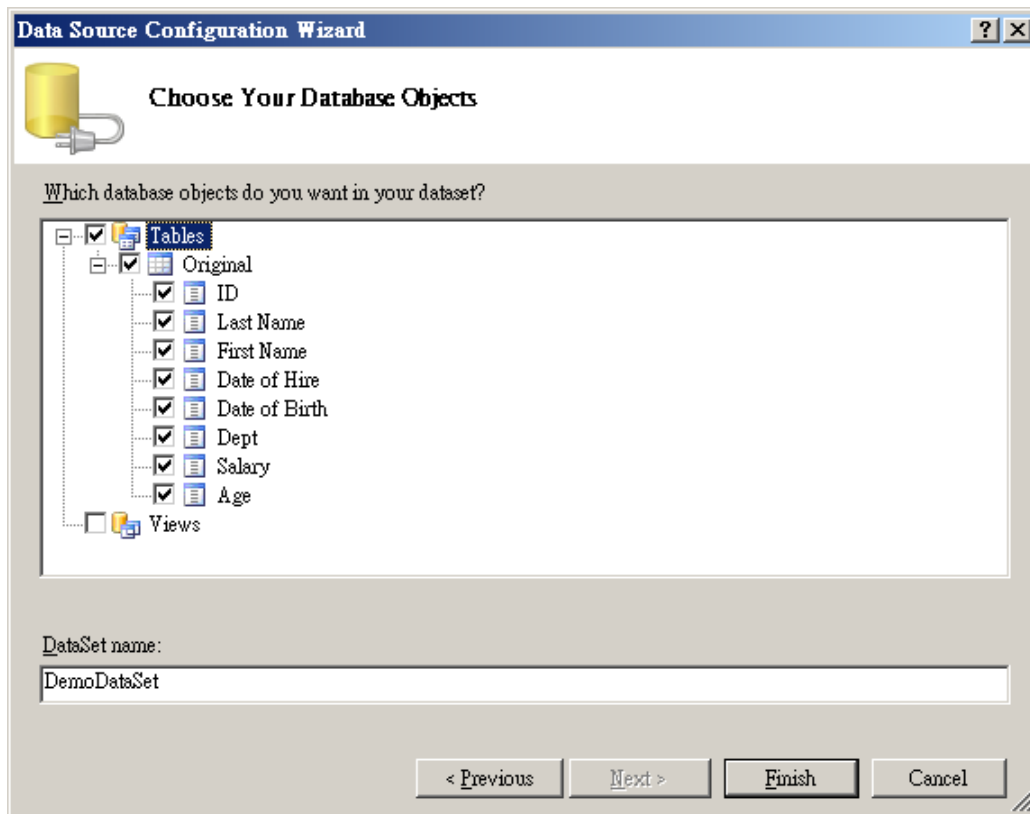
7. Select "Demo.mdb" in the **Choose Your Connection** section, and then press [Next >] button.



8. Save the Connection String to the Application Configuration file **DemoConnectionString**.



9. Select the table field wish to be displayed.



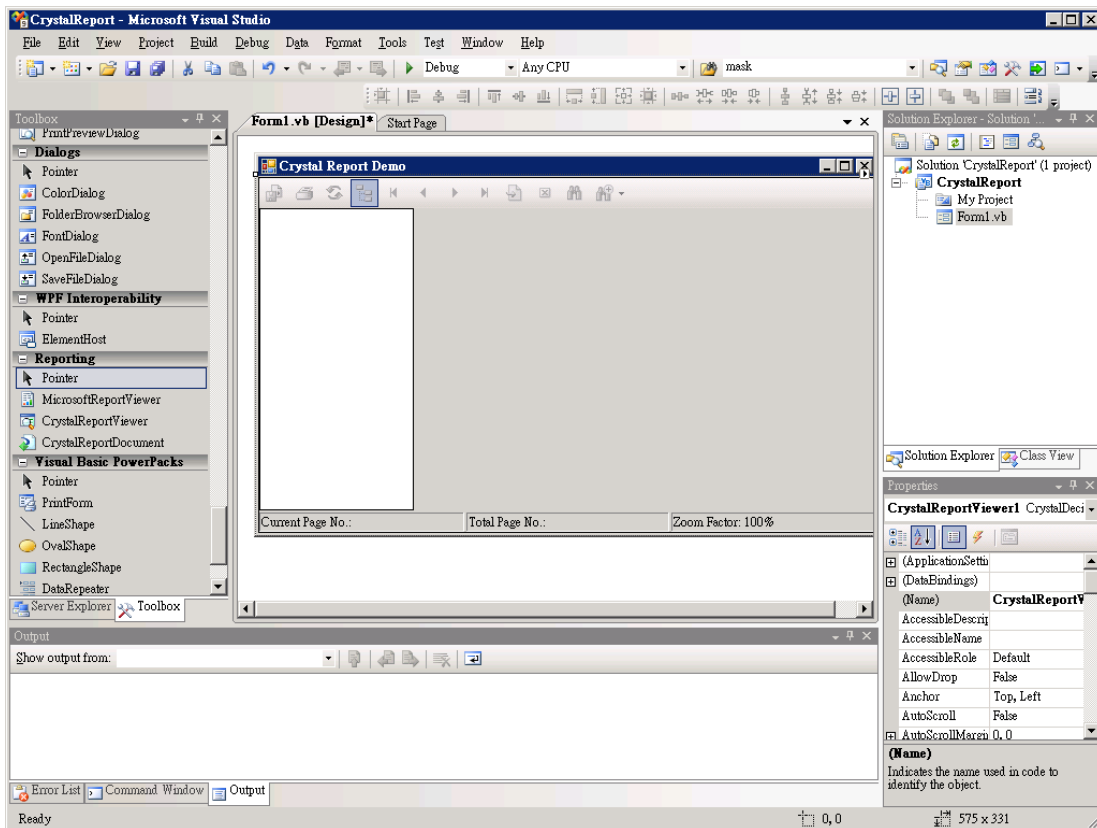
10. Build and execute the program, can you see the database record?

ID	Last Name	First Name	Date of Hire	Date of Birth	Dept	Salary	Age
1	Abercrombie	Kim	24/6/1985	16/10/1962	D	91000	46
2	Ackerman	Pilar	26/4/1989	5/10/1950	E	31000	58
3	Ajenstat	François	1/2/1981	21/12/1964	C	48000	43
4	Akers	Kim	29/5/1979	8/4/1958	C	47000	50
5	Alberts	Amy E.	15/10/1989	22/4/1970	D	60000	38
6	Alderson	Gregory F. (Greg)	12/1/1992	28/5/1964	D	100000	44
7	Alexander	Sean P	6/7/2000	16/10/1962	E	29000	46
8	Anderson	Nancy	6/6/1986	20/6/1976	C	68000	32
9	Bacon Jr.	Dan K.	6/4/1988	14/1/1961	B	85000	47
10	Bankert	Julie	21/4/1983	24/11/1977	C	100000	30
11	Barbariol	Angela	16/2/1996	18/4/1964	A	92000	44
12	Barnhill	Josh	26/7/1977	12/10/1954	E	83000	54
13	Barr	Adam	9/7/1993	8/4/1959	C	66000	49
14	Bashary	Shay	14/3/1995	30/1/1972	B	35000	36
15	Beck	Bradley	20/12/1998	5/2/1958	C	70000	50
16	Ben-Sachar	Ido	21/9/1993	12/7/1956	A	57000	52
17	Benson	Max	22/11/1977	2/9/1975	B	97000	33
18	Berge	Karen	1/6/1997	27/2/1965	B	35000	43
19	Berglund	Andreas	19/3/1988	21/10/1954	E	93000	54
20	Berndt	Matthias	17/3/1991	28/1/1978	B	48000	30
21	Berry	Jo	20/9/2000	7/1/1966	D	76000	42
22	Bolender	Corinna	20/8/1998	18/11/1951	D	43000	56
23	Bonifaz	Luis	10/6/1996	4/4/1968	D	77000	40

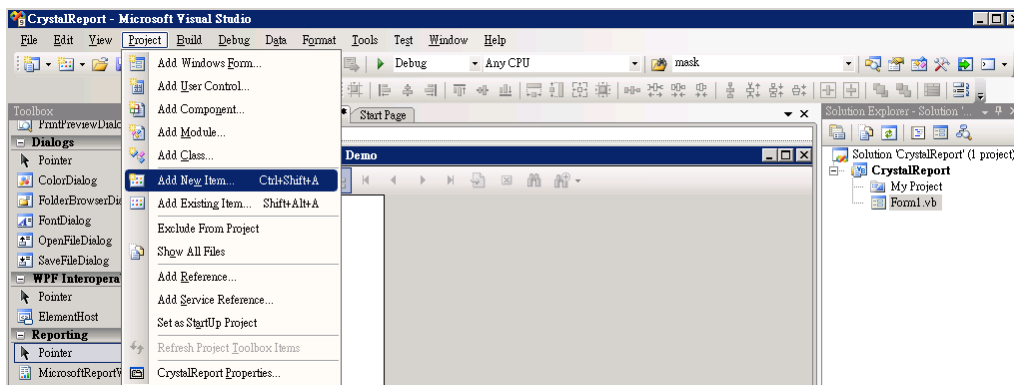
3. Crystal Report

1. Open the Microsoft Visual Studio and start a new Visual Basic Project **CrystalReport**. From the Toolbox and drag a **CrystalReportViewer** control onto your form and customize the properties

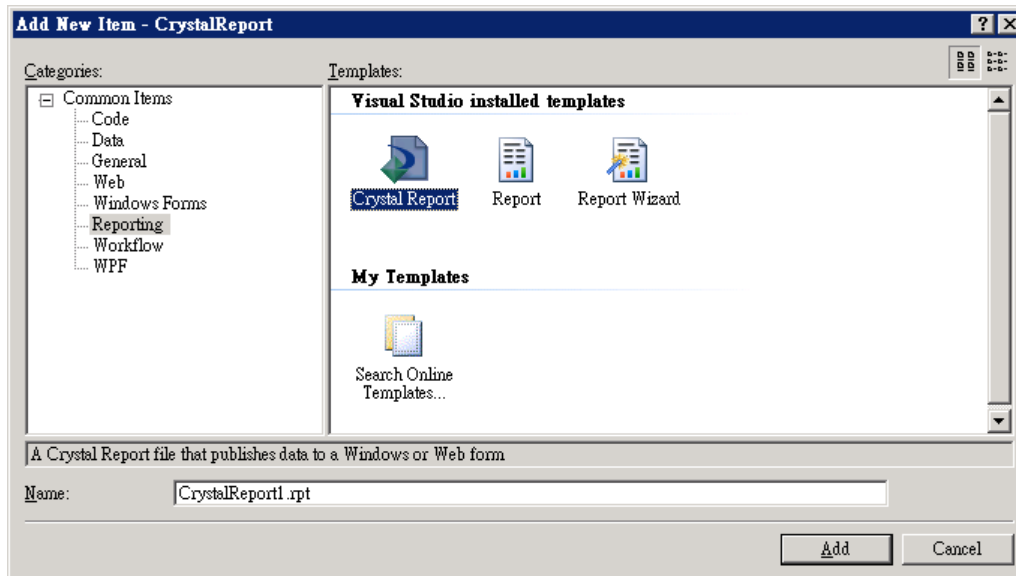
Object	Name	Property	Property Value
Form	frmMain	Text	Crystal Report Demo
Crystal Report Viewer	CrystalReportViewer1	Dock	Fill



2. Select **Add New Item** from the **Project** menu in the Visual Studio **Standard Toolbar**.

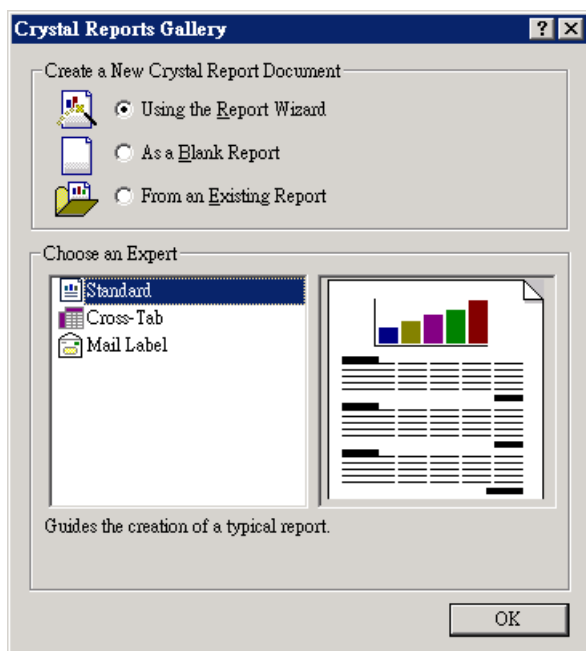


3. In the **Add New Item** dialog box, select **Crystal Report** from the **Templates** pane. Then enter a report name with an **.rpt** extension in the **Name** field. Or, accept the default report file name, **CrystalReport1.rpt**, if it exists. Click **[Add]** to invoke the **Crystal Report Designer** that helps you create and design the new report.

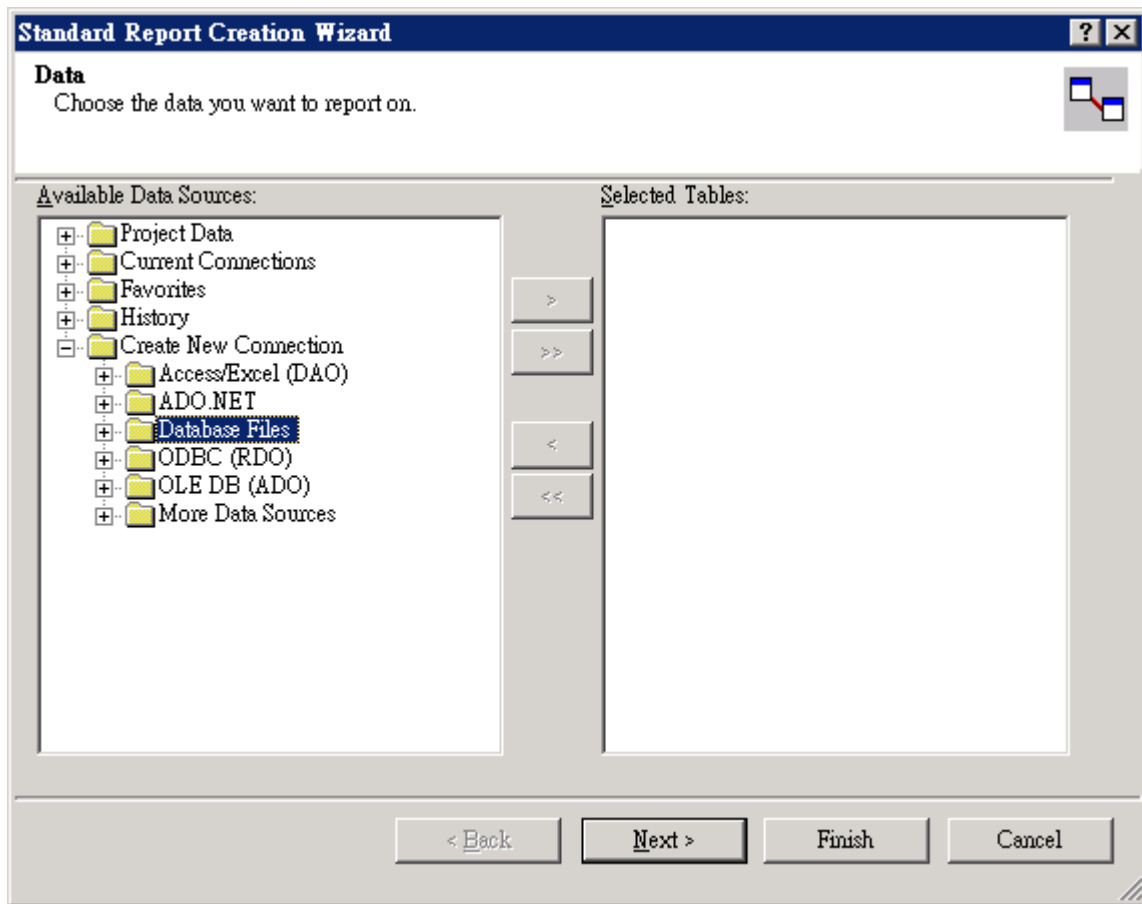


4. In the Crystal Report Gallery, select one of the following options and click **[OK]**.
- **Using the Report Expert** – Guides you through the report creation process and adds your choices to the Crystal Report Designer.
 - **As a Blank Report** – Opens the Crystal Report Designer.
 - **From an Existing Report** – Creates a new report with the same design as another report you specify.

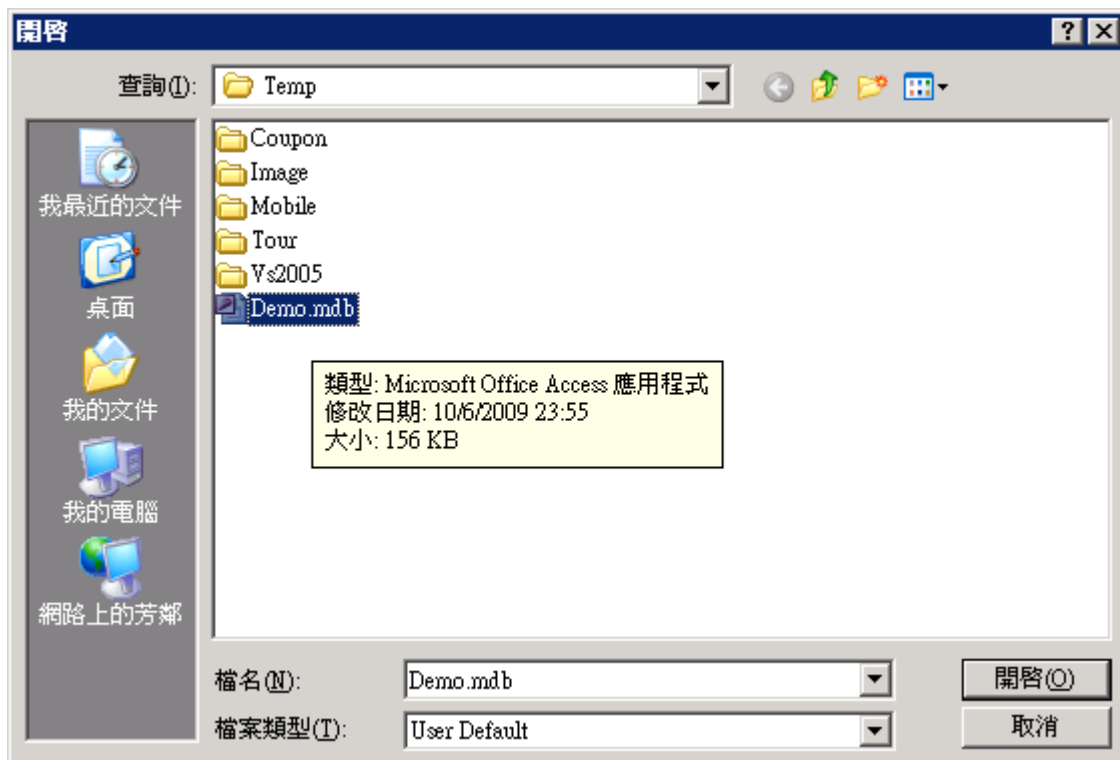
If you chose to use the Report Expert, the Report Expert dialog box appears, along with the Data Explorer. Select the data required for each folder, work through the Report Expert's tabbed interface, and click Finish to access the Crystal Report Designer.



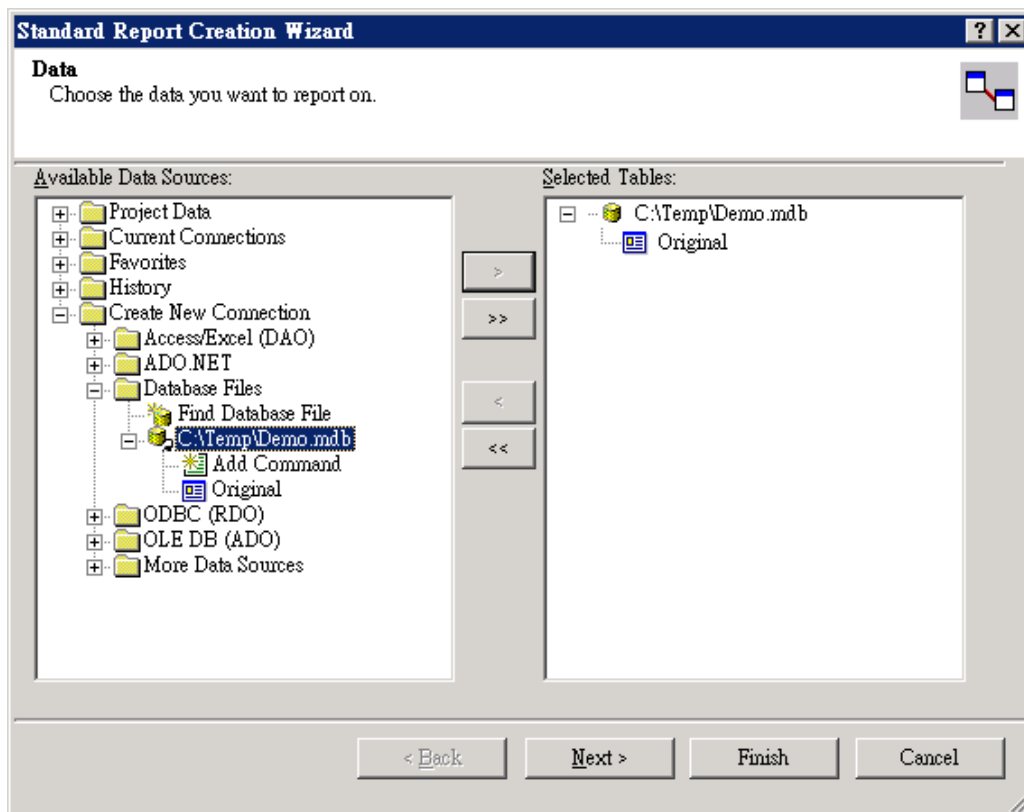
5. Double click the **Create New Connection → Database Files** folder in the **Available Data Source** panel of the **Standard Report Creation Wizard** dialog.



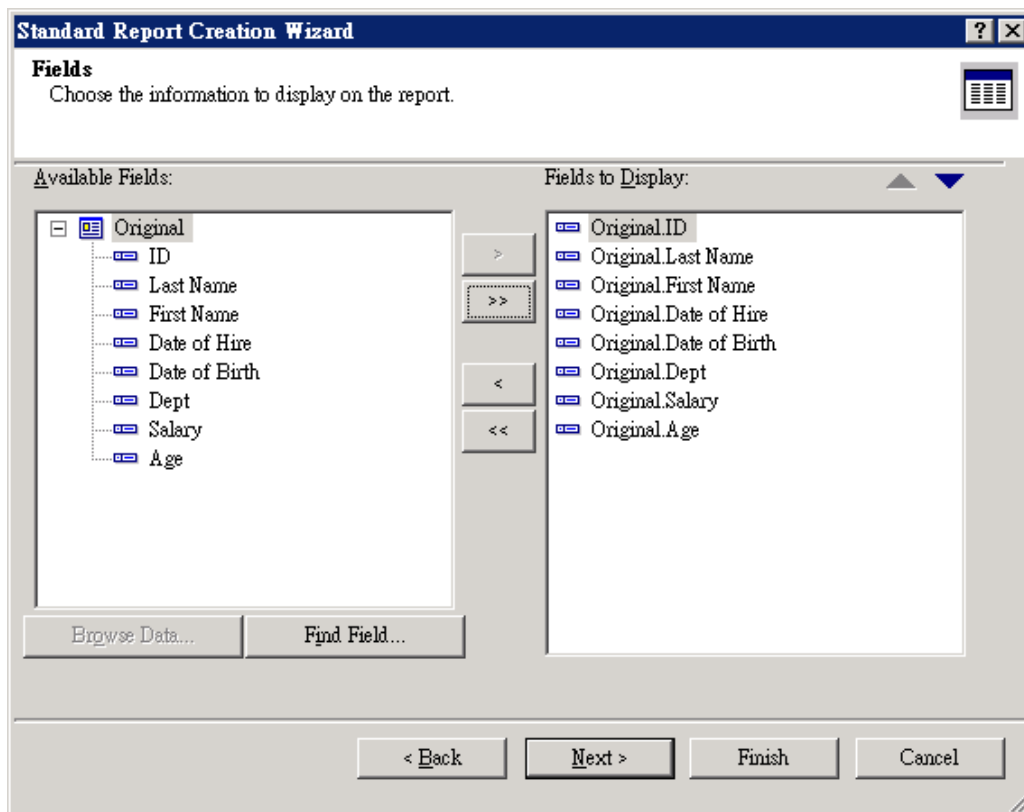
6. Use the **Open File** dialog to select the corresponding database file.



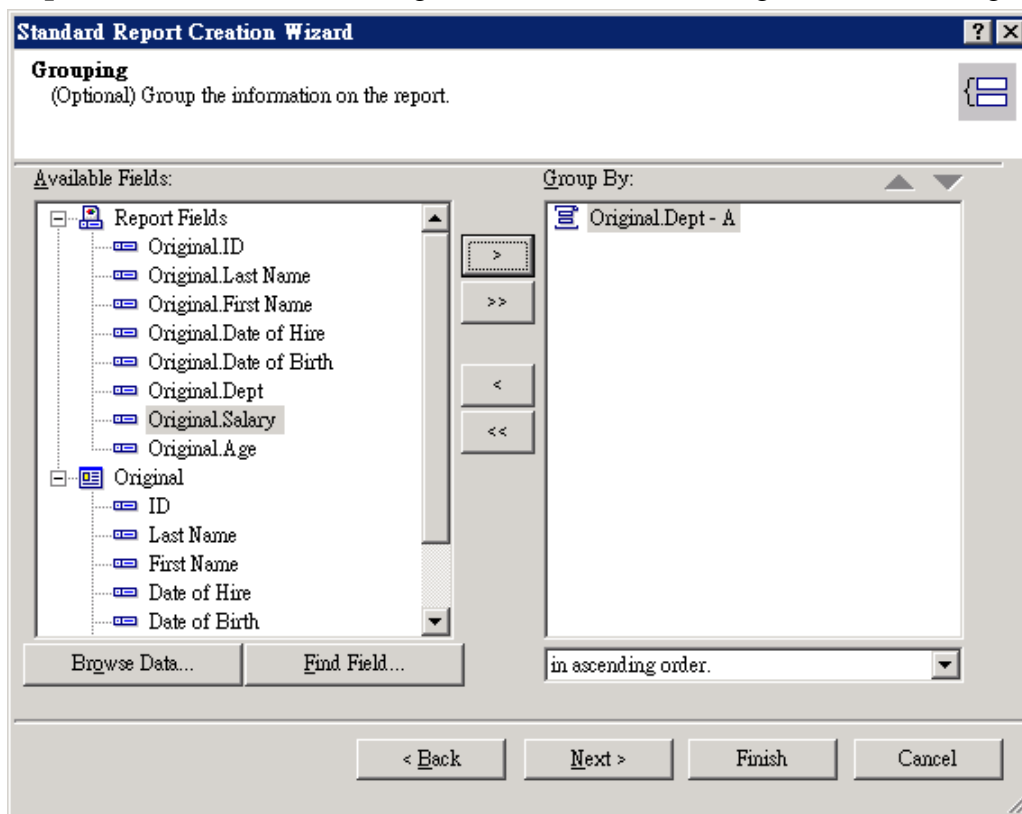
7. Use the [>] button to insert the tables that will be used for the report.



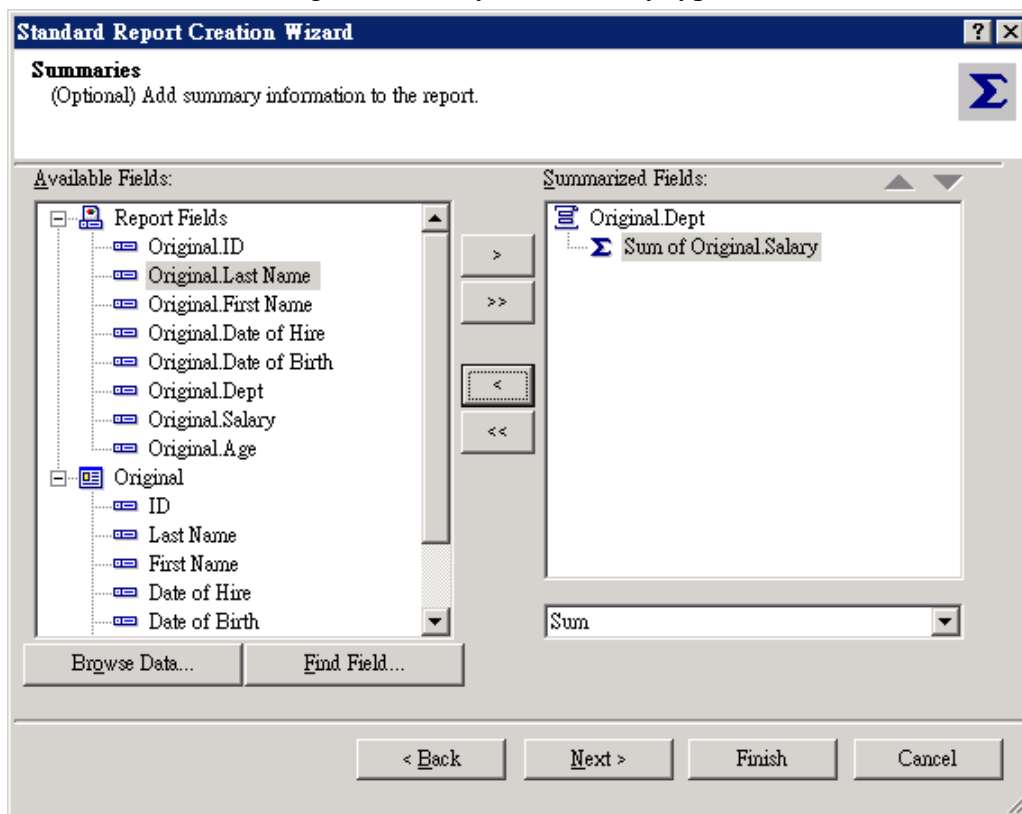
8. Use the [>] button to select the output fields in the **Fields** session of the **Standard Report Creation Wizard** dialog.



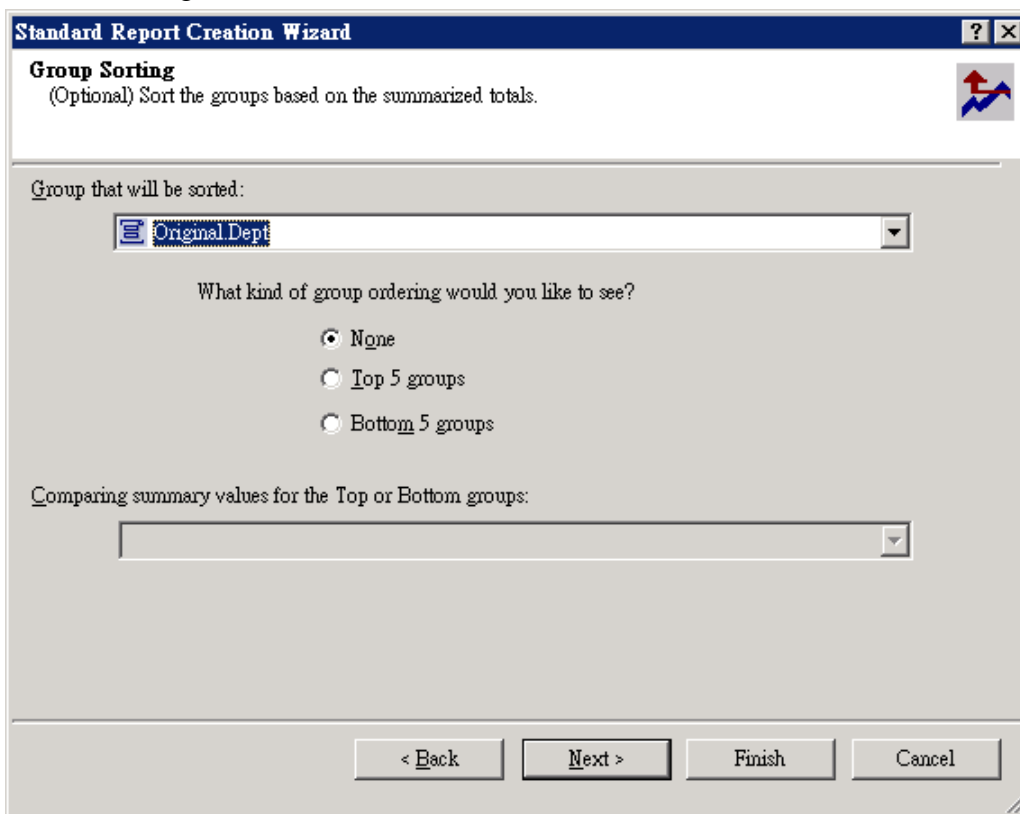
- Use the [>] button to select the group by method in the **Grouping** session of the **Standard Report Creation Wizard** dialog and then select the sorting order (Ascending/Descending).



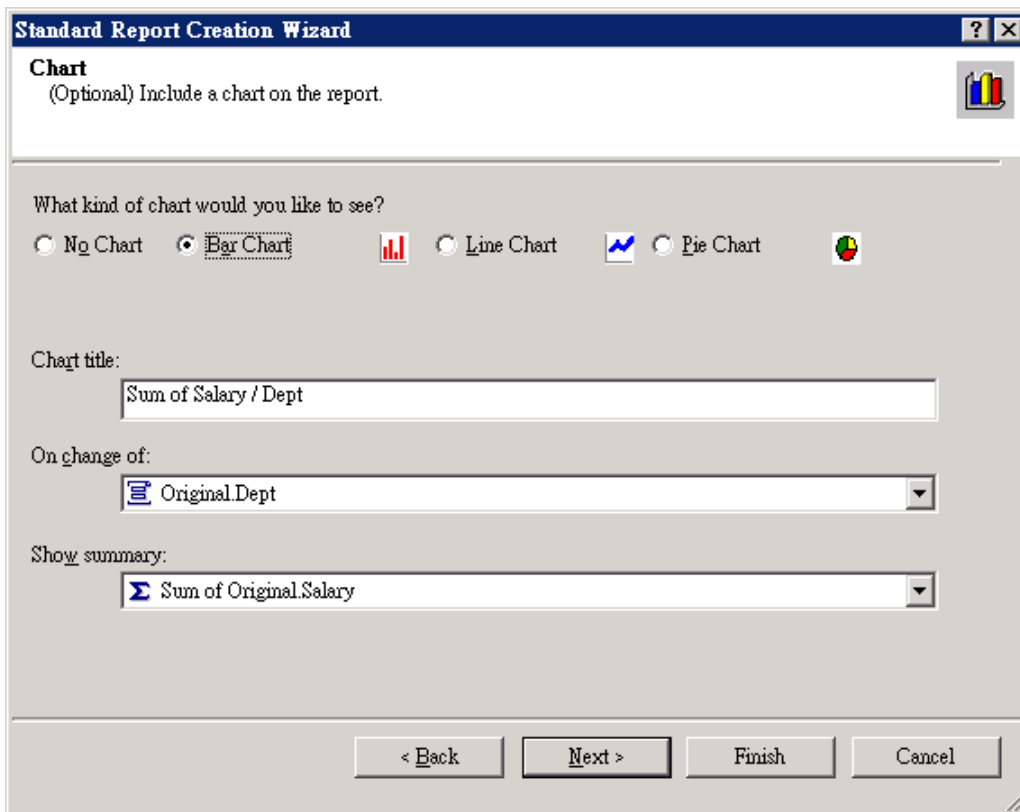
- Use the [>] button to select the summation field in the **Total** session of the **Standard Report Creation Wizard** dialog and define your summary type.



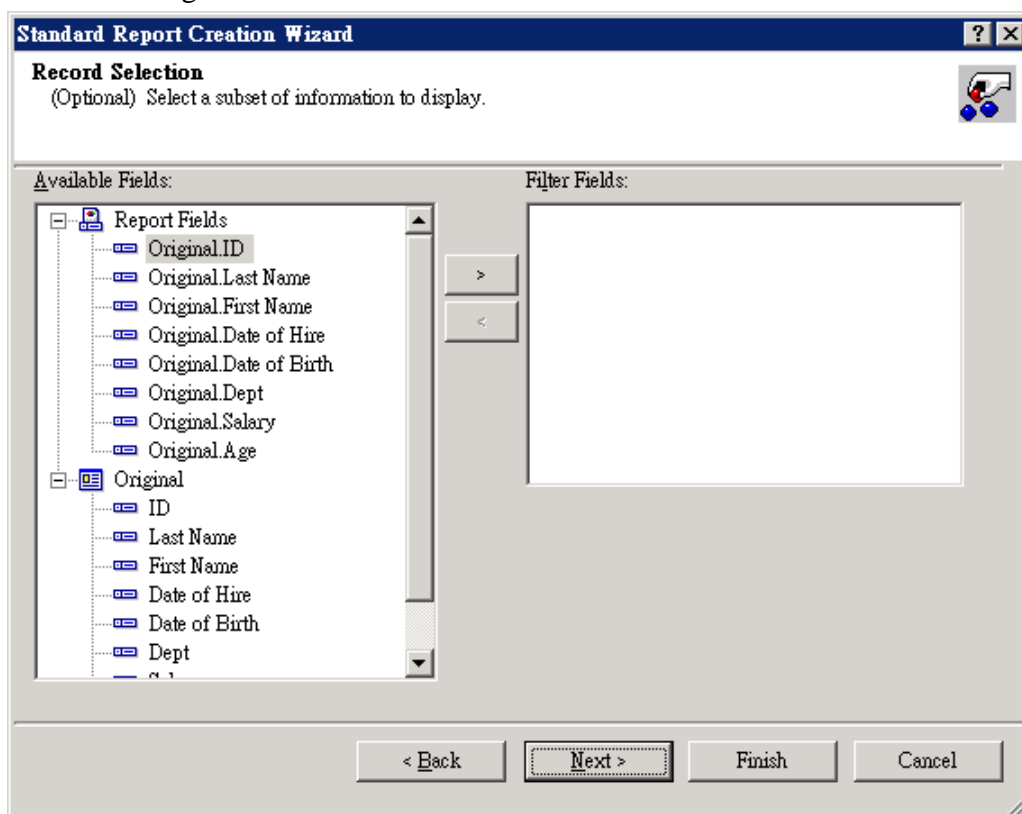
11. Select the Top N method in the **Group Sorting** session of the **Standard Report Creation Wizard** dialog.



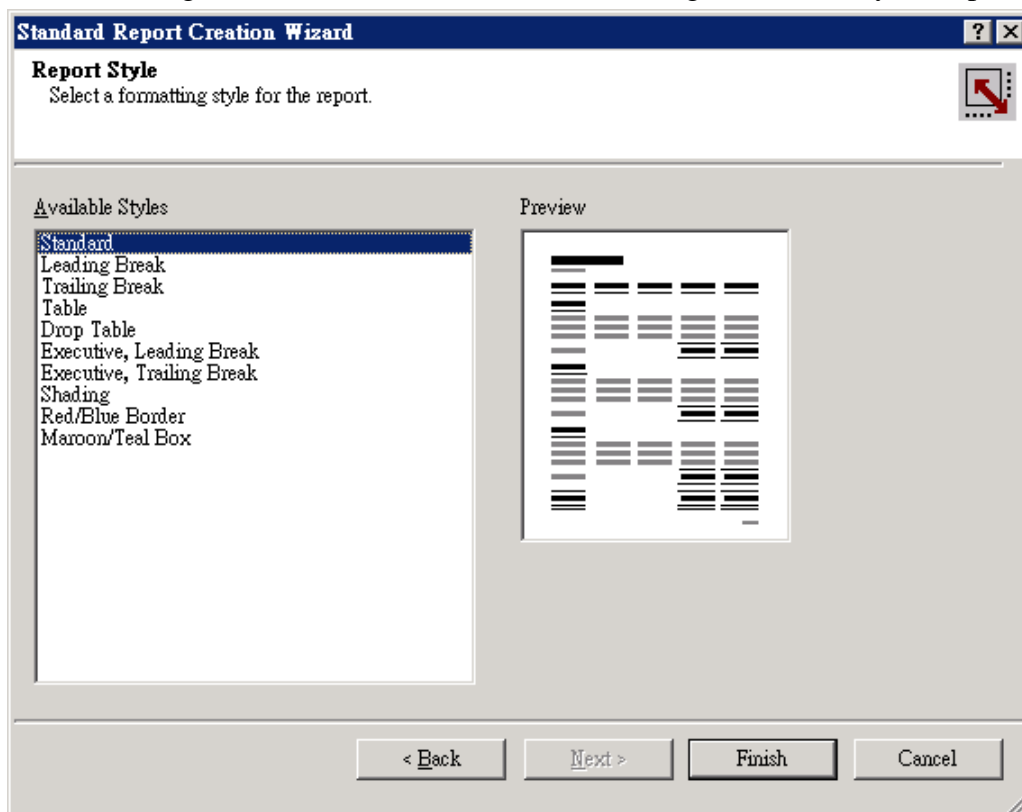
12. Select the output chart type (Bar, Line, etc) in the **Chart** session of the in the **Standard Report Creation Wizard** dialog.



13. Create the Filters fields in the **Record Selection** session in the **Standard Report Creation Wizard** dialog.

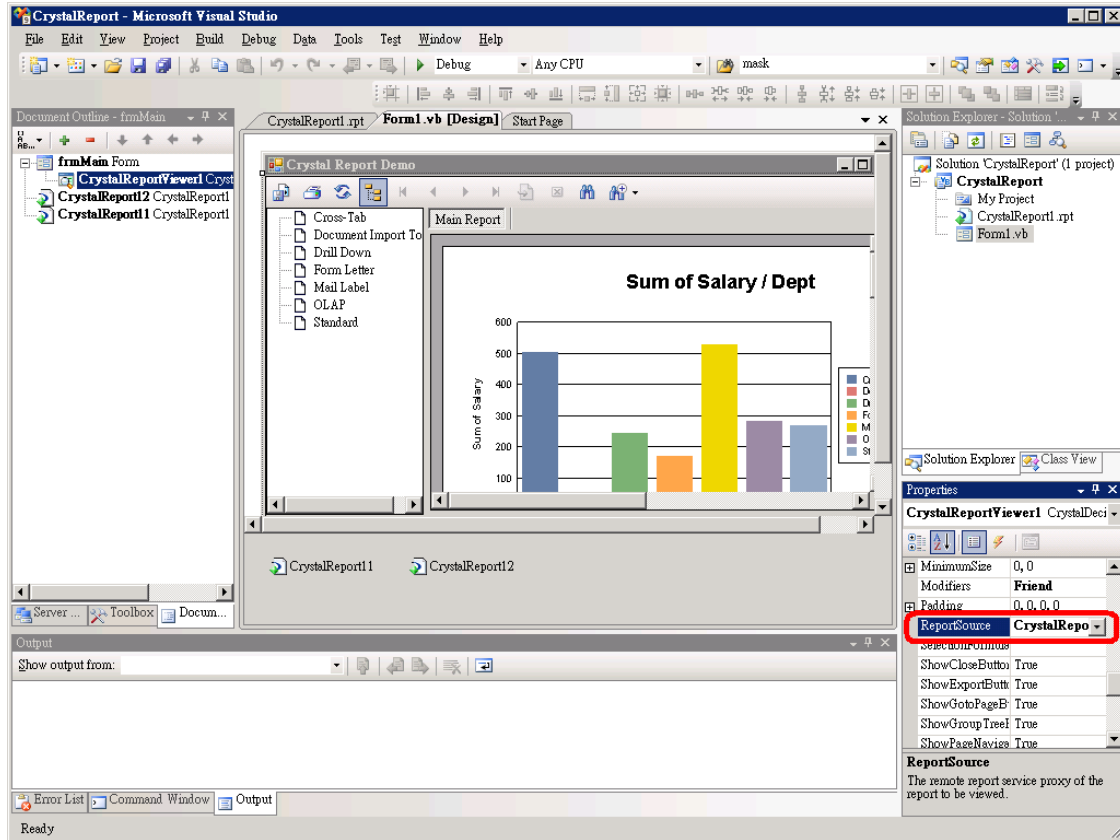


14. Select the title and style in the **Report Style** session of the **Standard Report Creation Wizard** dialog, and then click the [Finish] button to generate the crystal report.

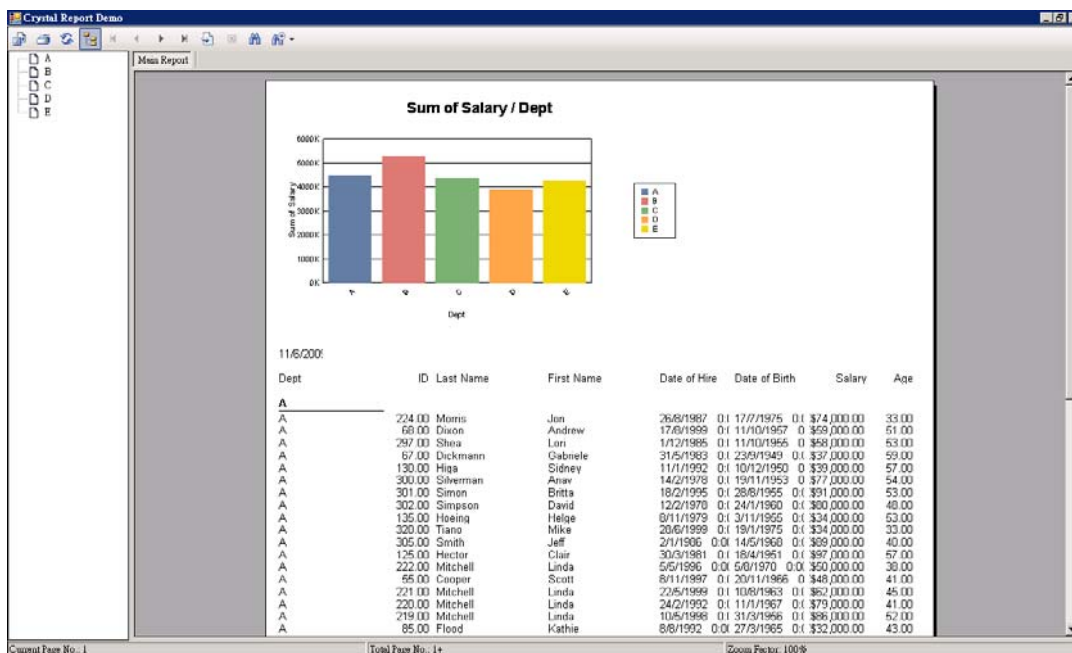


- Define the Crystal report in the **ReportSource** of the Crystal Report Viewer properties. Besides, you can copy the report to the "Bin" folder of the project, then add the following code in the **Load** event procedure of the Crystal Report Viewer control (**CrystalReportViewer1**),

```
CrystalReportViewer1.ReportSource = _
    AppDomain.CurrentDomain.BaseDirectory & "CrystalReport1.rpt"
```

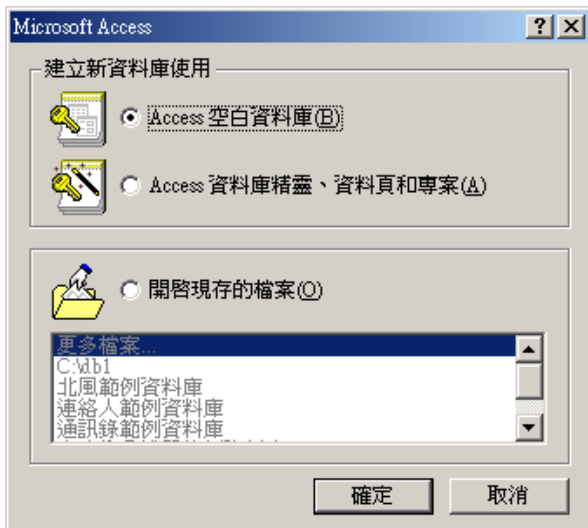


- Build and execute your project, you should be able to obtain the following report..

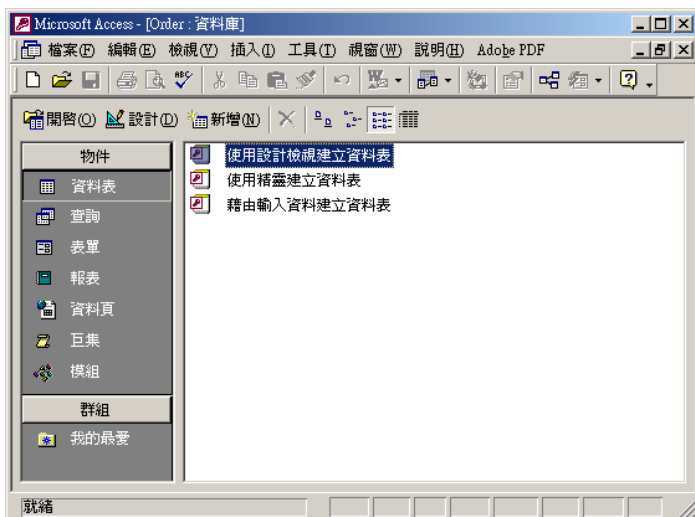
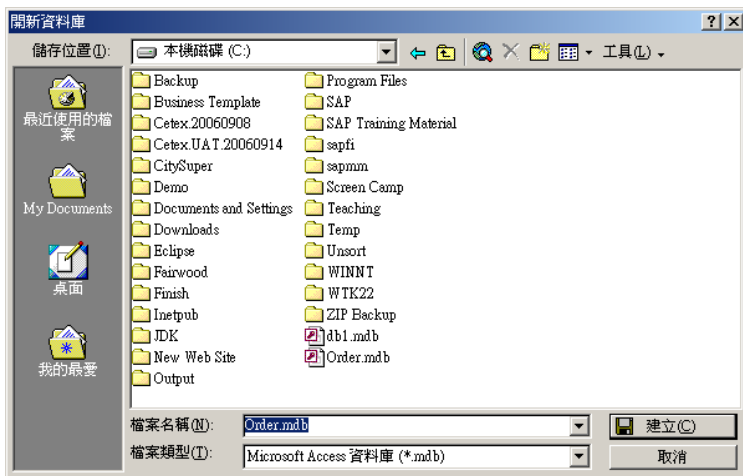


4. Create Database using Microsoft Access

1. Start “Microsoft Access” and select “Access Blank Database”



2. Specify the name and the path to create a new database file. We will create a database **ORDER.MDB** in the folder **C:\Temp** in this demo.



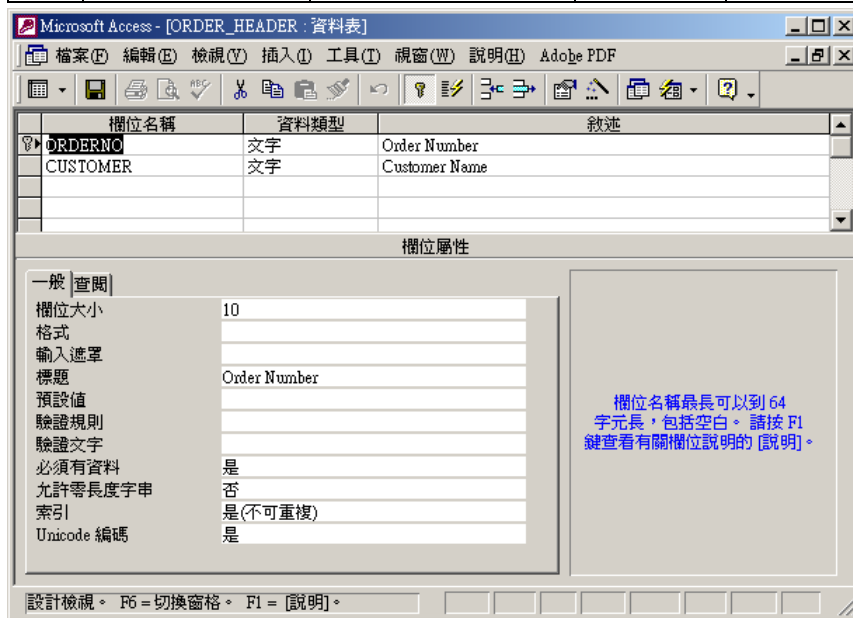
- Design the structure for two tables **ORDER_HEADER** & **ORDER_DETAIL** by using the **Create Table in Design View** wizard and press the [Save] button when finish.

ORDER_HEADER

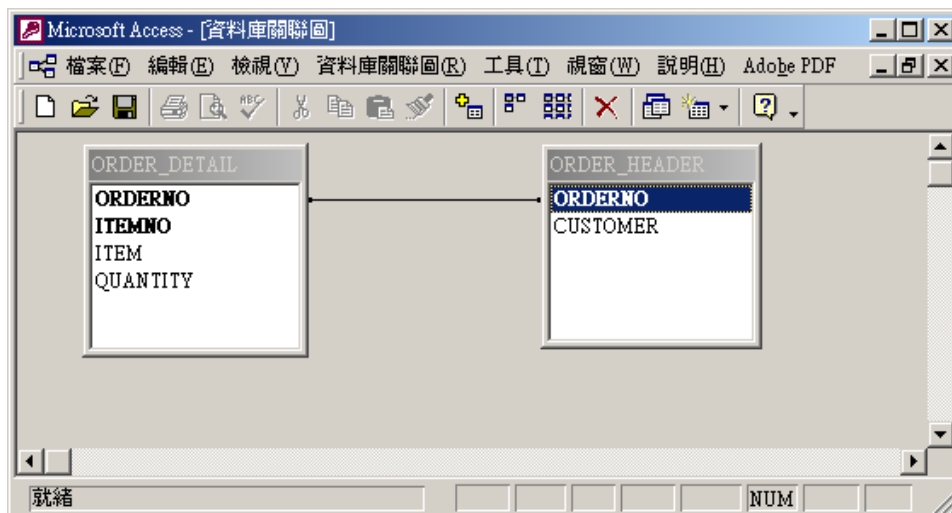
Key	Field	Date Structure	Length	Description
*	ORDERNO	CHARACTER	10	Order Number
	CUSTOMER	CHARACTER	50	Customer Name

ORDER_DETAIL

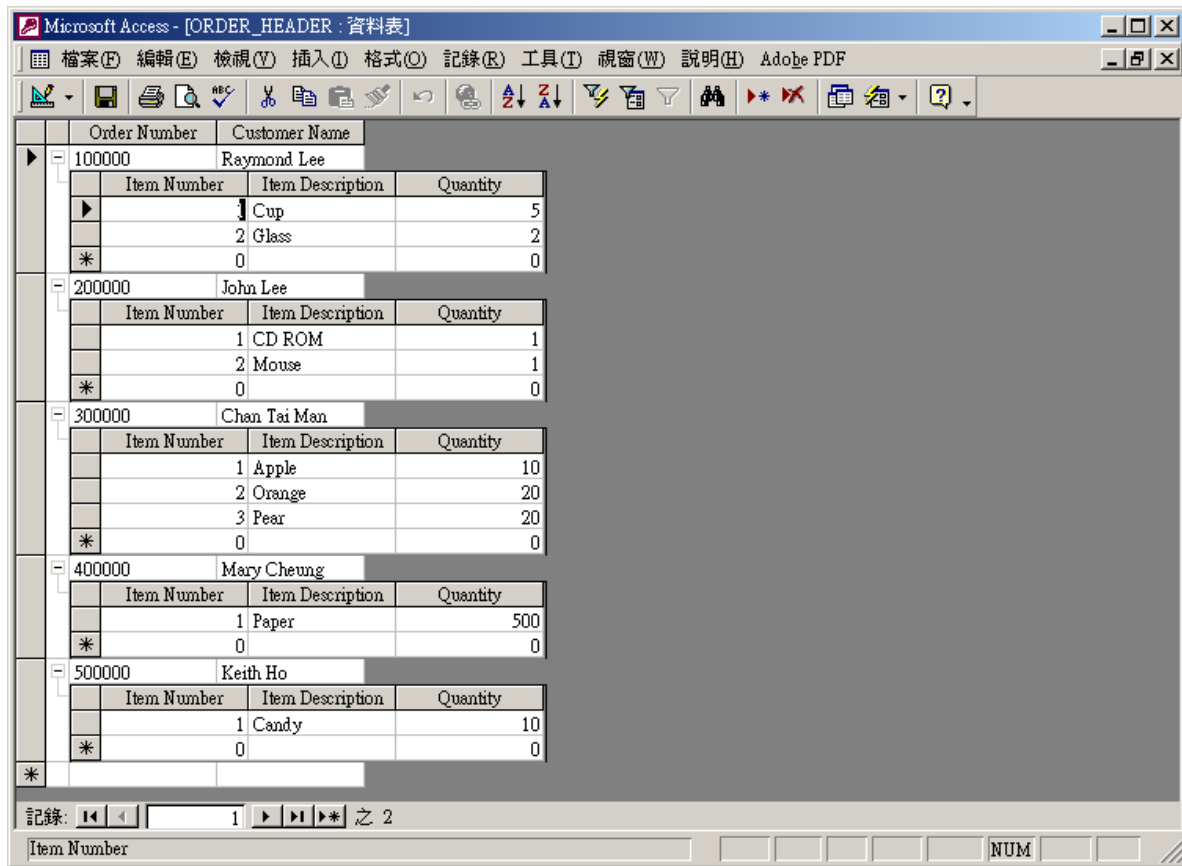
Key	Field	Date Structure	Length	Description
*	ORDERNO	CHARACTER	10	Order Number
*	ITEMNO	CHARACTER	5	Item Number
	ITEM	CHARACTER	50	Item Description
	QUANTITY	INTEGER		Quantity



- Apply the following relationship to the tables and save the relation table.



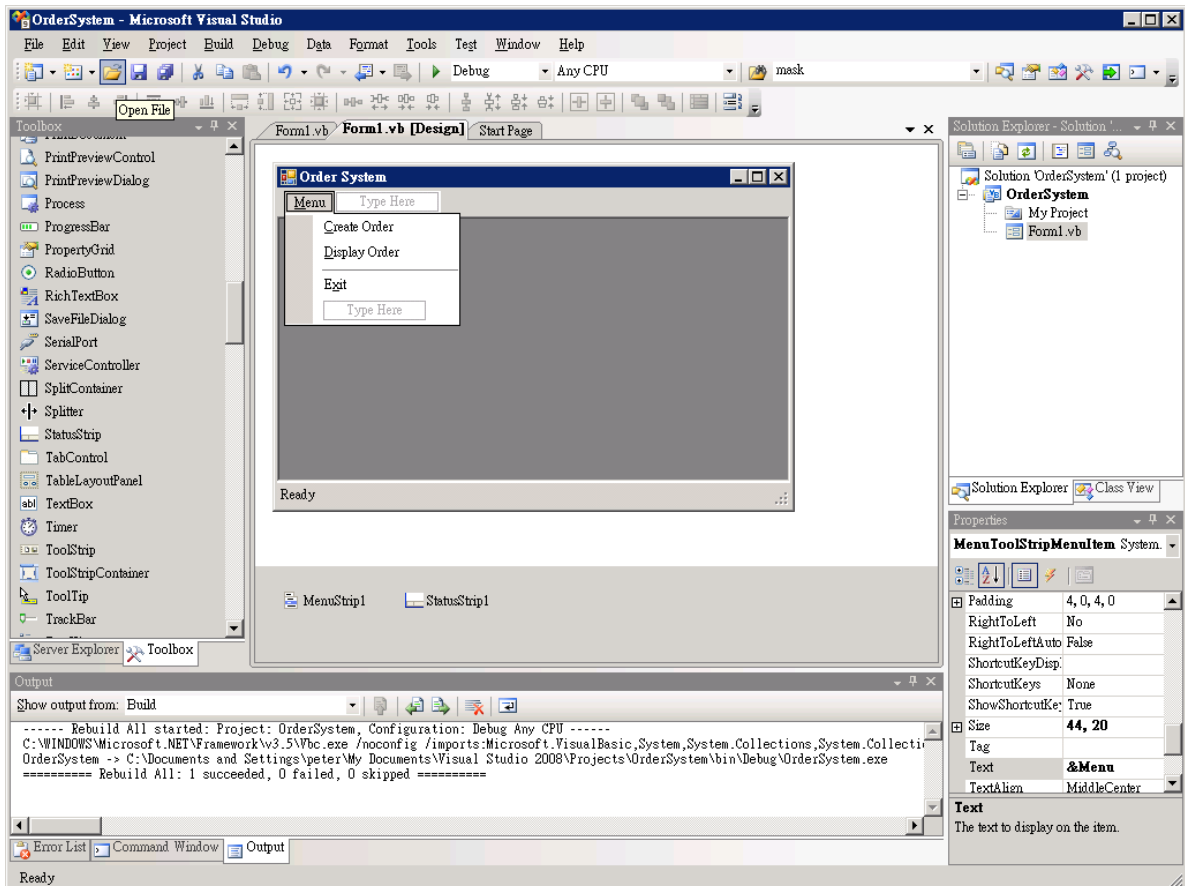
5. Input some data to the table, and then exit the Microsoft Access.



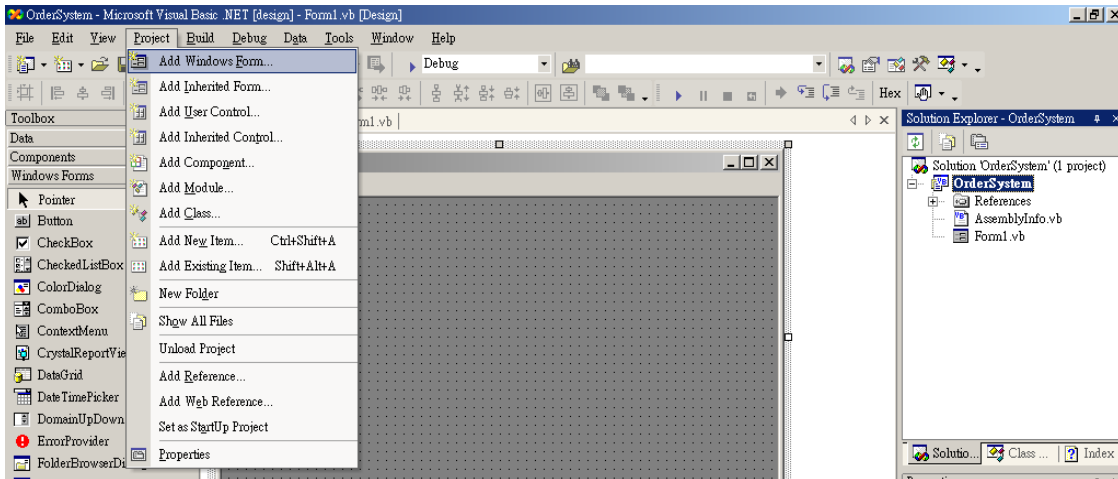
5. Simple Order System – Displaying Order

17. Open the Microsoft Visual Studio and start a new Visual Basic Project **OrderSystem**. From the Toolbox, drag a **MenuStrip** control and a **StatusStrip** control onto the form and customize the properties.

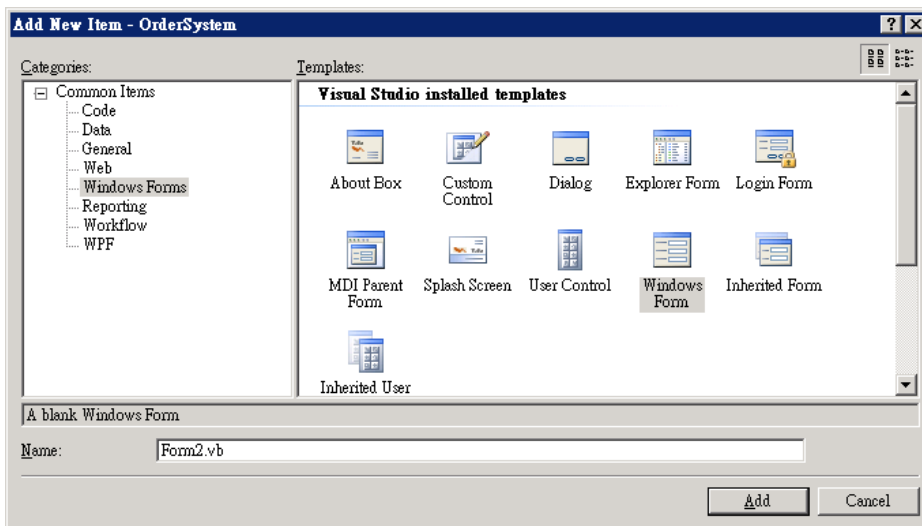
Object	Name	Property	Property Value
Form	frmMain	Text	Order System
		IsMdiContainer	True
		MainMenuStrip	MenuStrip1
		WindowState	Maximized
MenuStrip	MenuStrip1	N/A	N/A
ToolStripMenuItem	MenuToolStripMenuItem	Text	&Menu
	CreateOrderToolStripMenuItem	Text	&Create Order
	DisplayOrderToolStripMenuItem	Text	&Display Order
	ToolStripSeparator1	N/A	N/A
	ExitToolStripMenuItem	Text	E&xit
StatusStrip	StatusStrip1	N/A	N/A
ToolStripStatusLabel	ToolStripStatusLabel1	Text	Ready



18. Select **Project** → **Add Windows Form...** menu to create a new Form **frmList** for Order summary display.

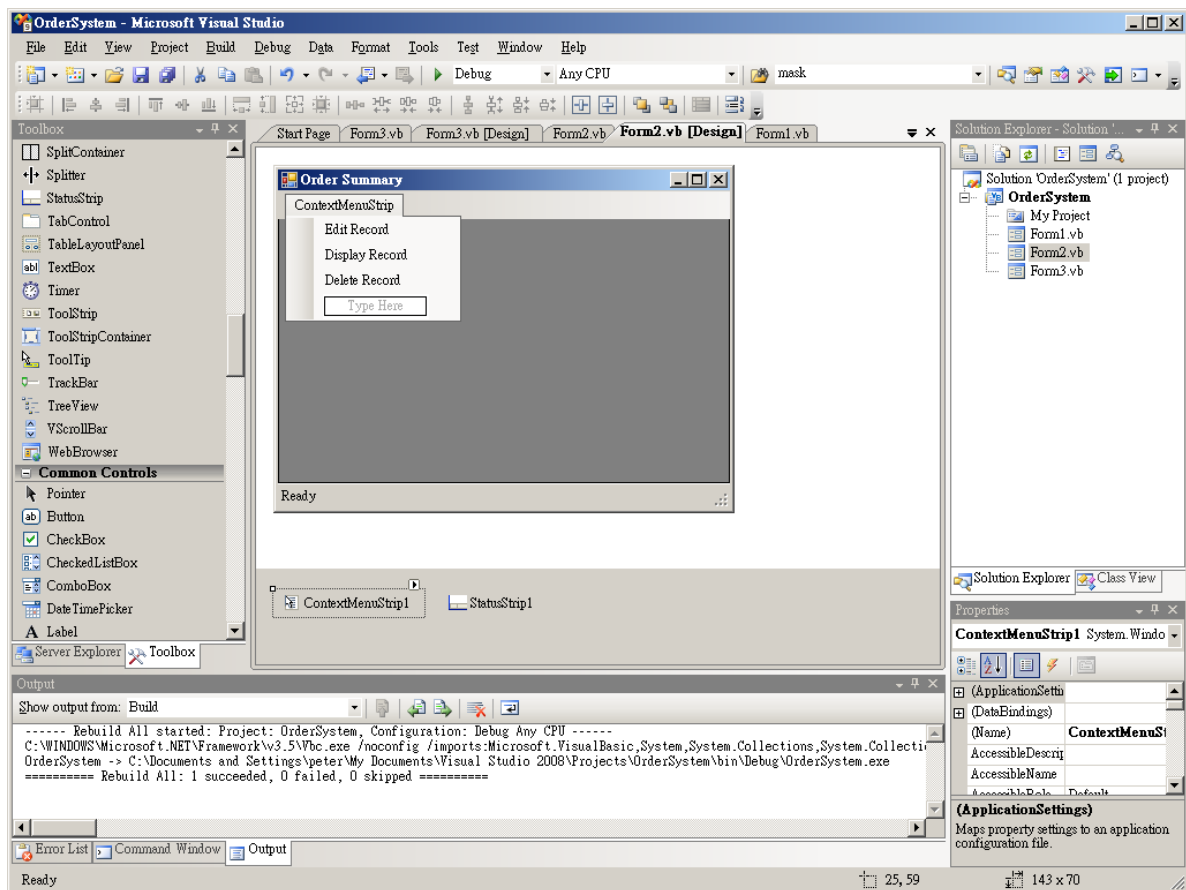


19. Select [**Windows Forms**] from the template, and press [**Add**] to continue.



20. From the Toolbox, drag a **Context Menu** control, a **Data Grid** control, and a **Status Bar** control onto the form 2 and customize the properties.

Object	Name	Property	Property Value
Form	frmList	Text	Order Summary
ContextMenuStrip	ContextMenuStrip1	N/A	N/A
	EditRecordToolStripMenuItem	Text	Edit Record
	DisplayRecordToolStripMenuItem	Text	Display Record
	DeleteRecordToolStripMenuItem	Text	Delete Record
DataGridView	DataGridView1	ContextMenu	ContextMenu1
		MultiSelect	False
		Dock	Fill
StatusStrip	StatusStrip1	N/A	N/A
	ToolStripStatusLabel1	Text	Ready



21. Add an imports statement at the top of the code to specify the namespace.

```
Imports System.Data
Imports System.Data.OleDb
```

22. Declare the constant for the database connection string inside the class

```
' Declare the Access Database from "C:\Temp\order.mdb"
Const ConnString As String = _
    "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=" & "C:\Temp\ORDER.mdb;"
```

23. Create a procedure **FillSummary** to retrieve order header from database.

```
Public Sub FillSummary()
    ' Declare the Database connection
    Dim OleDbConn As OleDbConnection = _
        New OleDbConnection(ConnString)

    ' Open the Database Connection
    OleDbConn.Open()
```

```
' Declare the Data Adapter
Dim MyOleDbDataAdapter As OleDbDataAdapter = New OleDbDataAdapter

' Specify the SQL Command for the Data Adapter
MyOleDbDataAdapter.SelectCommand = _
    New OleDbCommand("SELECT * FROM order_header", OleDbConn)

' Declare the Data Set
Dim MyDataSet As DataSet = New DataSet

' Use the Data Adapter to fill the Data Set
MyOleDbDataAdapter.Fill(MyDataSet)

' Specify the Data Source based on the Data Set
Me.DataGridView1.DataSource = MyDataSet.Tables(0)

' Destroy the Data Adapter
MyOleDbDataAdapter.Dispose()

' Destroy the Data Set
MyDataSet.Dispose()

' Close the Database connection
OleDbConn.Close()

' Destroy the Database connection
OleDbConn.Dispose()
End Sub
```

24. Create a **Load** event handler for the **Form**.

```
Call FillSummary()
```

25. Go back to form **frmMain**, create a **Click** event handler for the Exit menu item (**ExitToolStripMenuItem**).

```
End
```

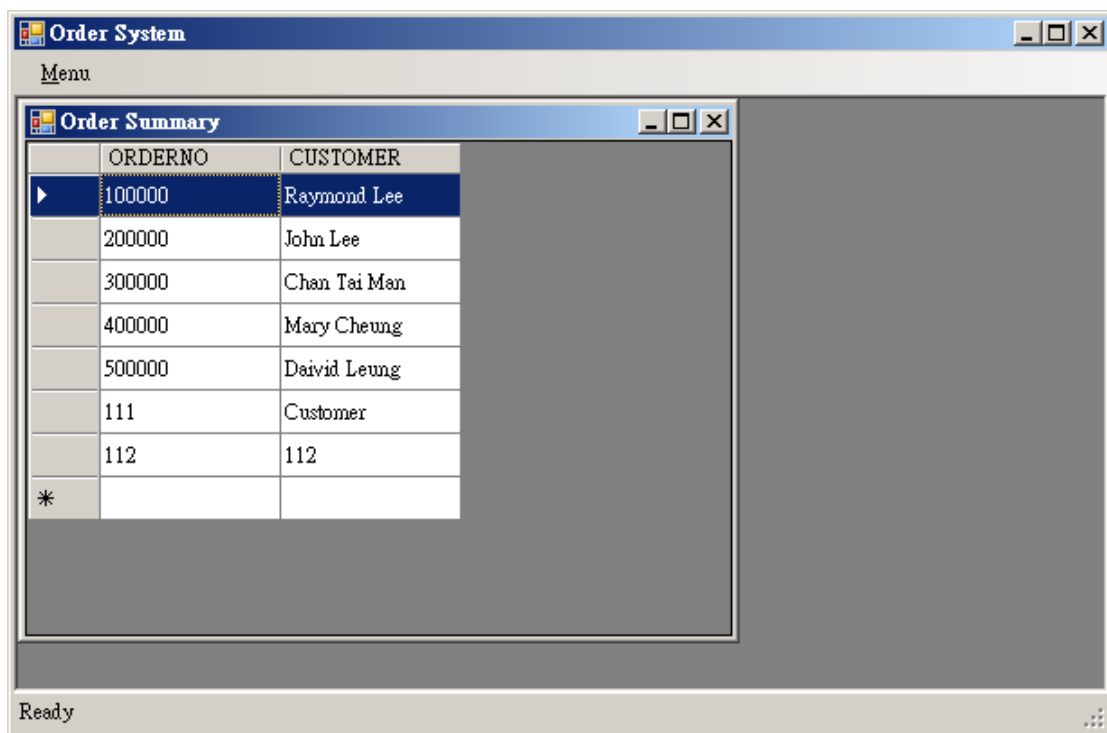

26. Create a **Click** event handler for the Display menu item (**DisplayOrderToolStripMenuItem**) in form **frmMain**.

```
Dim NewMDIChild As New frmList

'Set the Parent Form of the Child window.
NewMDIChild.MdiParent = Me

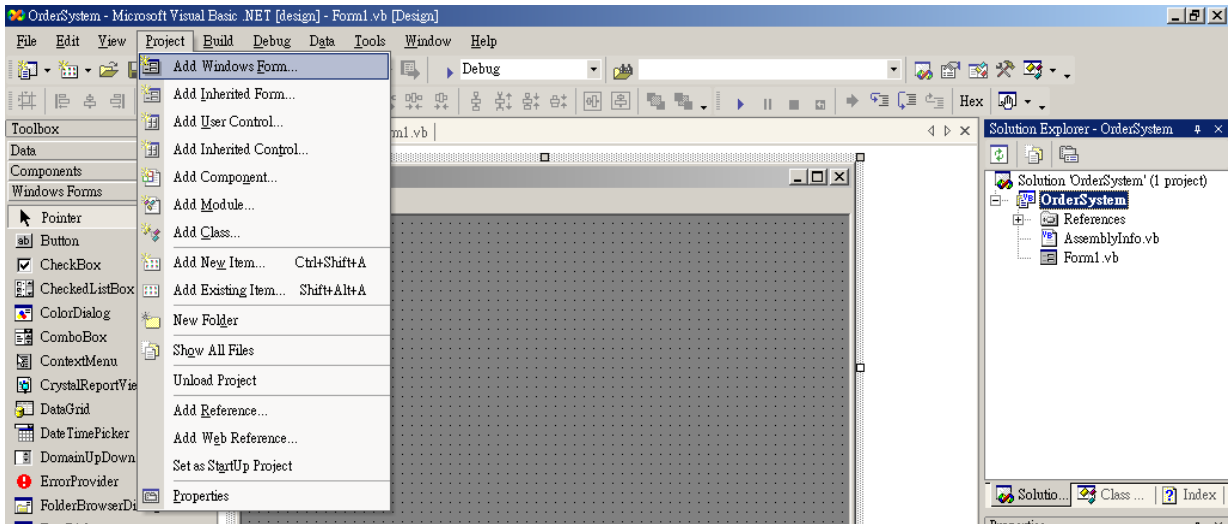
'Display the new form.
NewMDIChild.Show()
```

27. Build and execute the project, you can press **Menu** → **Display Record** to view the Order Summary.

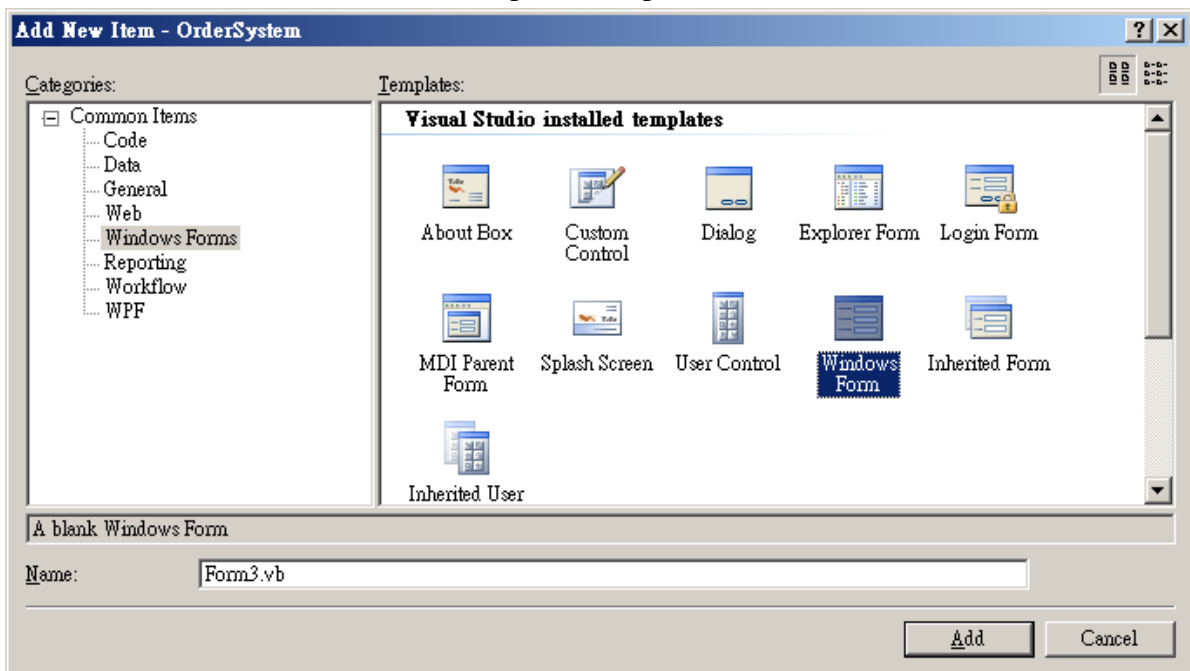


6. Simple Order System – Create Order

1. Start the Microsoft Visual Studio and open the previous project **OrderSystem**. Select **Project** → **Add Windows Form...** to create a new Form **frmOrder** for Order Detail entry and display.

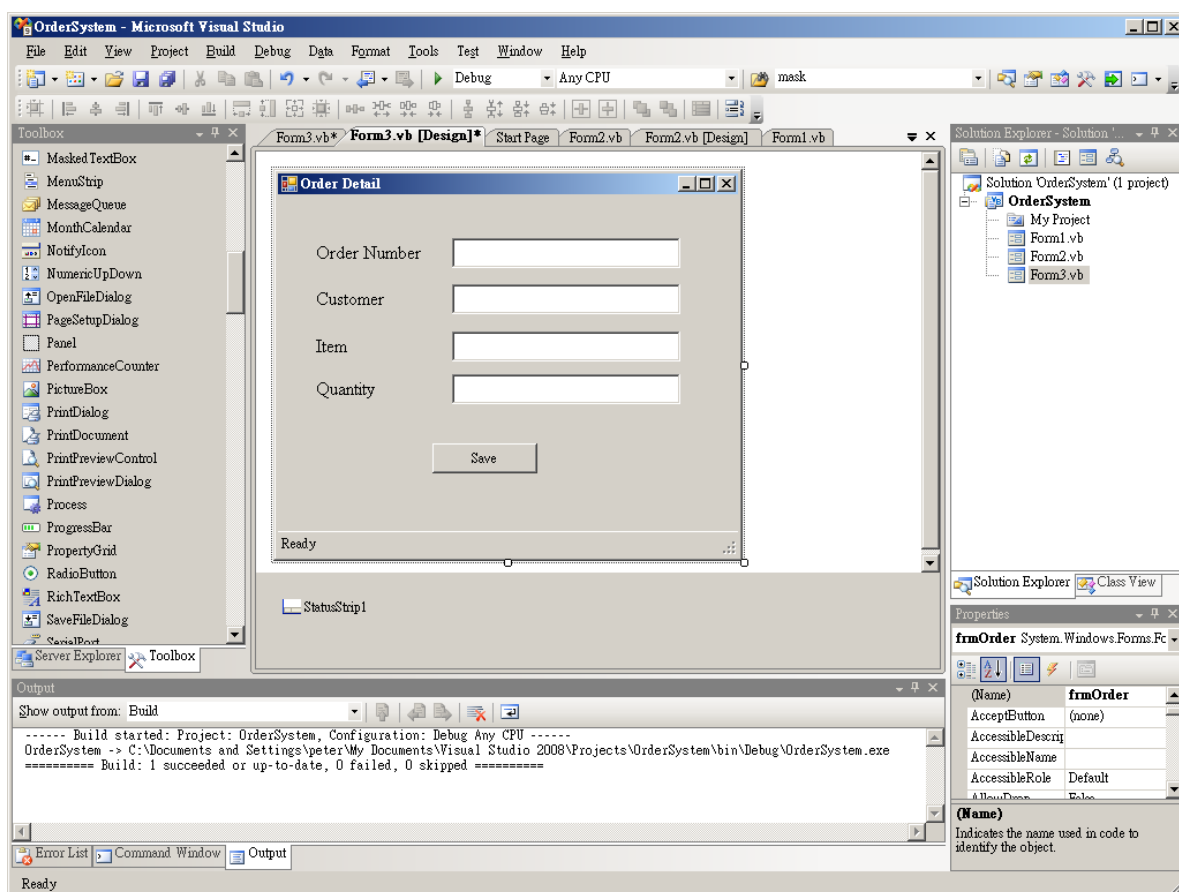


2. Select [**Windows Forms**] from the template, and press [**Add**] to continue.



- From the **Toolbox**, drag four **Label** control, four **Textbox** controls, a **Button** control, and a **Status Bar** control onto the form and customize the properties.

Object	Name	Property	Property Value
Form	frmOrder	Text	Order Detail
Label	Label1	Text	Order Number
	Label2	Text	Customer
	Label3	Text	Item
	Label4	Text	Quantity
TextBox	txtOrder	Text	(Blank)
	txtCustomer	Text	(Blank)
	txtItem	Text	(Blank)
	txtQuantity	Text	(Blank)
Button	btnSave	Text	Save
StatusStrip	StatusStrip1	N/A	N/A
	ToolStripStatusLabel1	Text	Ready



- Add an imports statement at the top of the code to specify the namespace.

```
Imports System.Data
Imports System.Data.OleDb
```

5. Create a **Click** event handler for the **Save** Button.

```

Const ConnString As String = _
    "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=" & "C:\Temp\ORDER.mdb;"

Dim SqlStr1, SqlStr2 As String

If txtOrder.Enabled = True Then
    ' SQL for Insert a new record
    SqlStr1 = "INSERT INTO order_header " _
        & "(orderno,customer) " _
        & "Values (" _
        & "'" & txtOrder.Text & "', " _
        & "'" & txtCustomer.Text & "')"
    SqlStr2 = "INSERT INTO order_detail " _
        & "(orderno,itemno,item,quantity) " _
        & "Values (" _
        & "'" & txtOrder.Text & "', " _
        & "1, " _
        & "'" & txtItem.Text & "', " _
        & txtQuantity.Text & ")"
Else
    ' Update an existing record
    SqlStr1 = "UPDATE order_header SET " _
        & "customer =" & txtCustomer.Text _
        & "' WHERE orderno = '" & txtOrder.Text & "'"
    SqlStr2 = "UPDATE order_detail SET " _
        & "item =" & txtItem.Text & "', " _
        & "quantity =" & txtQuantity.Text _
        & " WHERE orderno = '" & txtOrder.Text & "'" _
        & " AND itemno = 1"
End If

' Declare the Database connection
Dim OleDbConn As OleDbConnection = _
    New OleDbConnection(ConnString)

' Open the Database Connection
OleDbConn.Open()

```

```

' Declare the Database Command
Dim MyOleDbCommand As OleDbCommand = New OleDbCommand

' Declare the Connection for the Database Command
MyOleDbCommand.Connection = (OleDbConn)

' Declare the SQL for the Database Command
MyOleDbCommand.CommandText = SqlStr1

' Insert/Update the data from Order Header table
MyOleDbCommand.ExecuteNonQuery()

' Declare the SQL for the Database Command
MyOleDbCommand.CommandText = SqlStr2

' Insert/Update the data from Order Item table
MyOleDbCommand.ExecuteNonQuery()

' Close the Database connection
OleDbConn.Close()

If txtOrder.Enabled = True Then
    ' Display the Insert message
    ToolStripStatusLabel1.Text = _
        "Order " & txtOrder.Text & " created."
Else
    ' Display the Update message
    ToolStripStatusLabel1.Text = _
        "Order " & txtOrder.Text & " updated."
End If

```

6. Go back to from **frmMain**, create a **Click** event handler for the Create menu item (**CreateOrderToolStripMenuItem**).

```

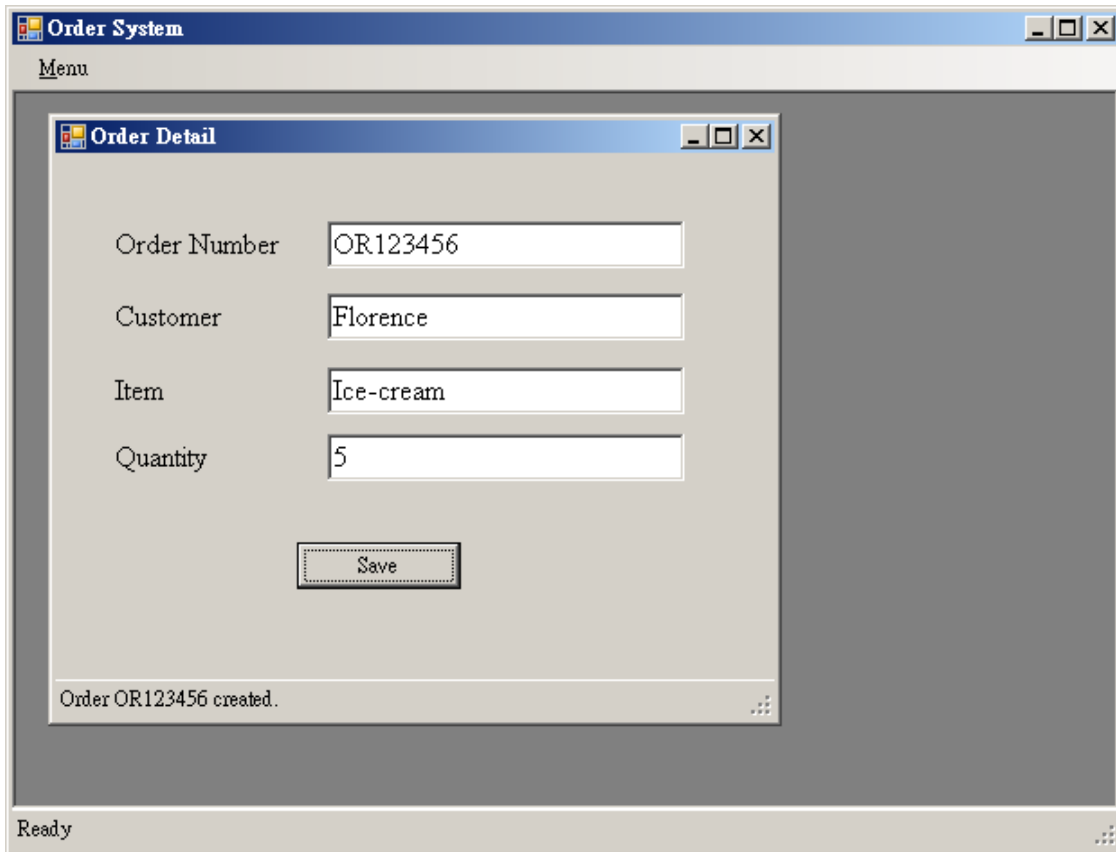
Dim NewMDIChild As New frmOrder

' Set the Parent Form of the Child window.
NewMDIChild.MdiParent = Me

' Display the new form.
NewMDIChild.Show()

```

7. Build and execute the project, you can press **Menu** → **Create Record** to create new order



7. Simple Order System – Order Manipulation

1. Start the Microsoft Visual Studio and open the previous project **OrderSystem**. In the form **frmList**, create a procedure **Get_Order_Detail** to retrieve order item detail from database, the **OutputMode** parameter is used to control the edit mode for the textbox.

```
Public Sub Get_Order_Detail(ByVal OutputMode As Boolean)
    ' Declare the MDI Child for the form
    Dim NewMDIChild As New frmOrder

    ' Declare the SQL string
    Dim SqlStr As String = _
        "SELECT * FROM order_detail WHERE orderno = '" & _
        & DataGridView1.CurrentRow.Cells(0).Value & "' " & _
        & "AND itemno = 1"

    ' Declare the Database connection
    Dim OleDbConn As OleDbConnection = _
        New OleDbConnection(ConnString)

    ' Open the Database Connection
    OleDbConn.Open()

    ' Declare the Database Command
    Dim MyOleDbCommand As OleDbCommand = New OleDbCommand

    ' Declare the Connection for the Database Command
    MyOleDbCommand.Connection = (OleDbConn)

    ' Declare the SQL for the Database Command
    MyOleDbCommand.CommandText = SqlStr

    ' Declare the Data Reader
    Dim MyDataReader As OleDbDataReader

    ' Execute the SQL Command and put the result in the Data Reader
    MyDataReader = MyOleDbCommand.ExecuteReader
```

```
Try
  Do While MyDataReader.Read
    ' Read Column 3 in Item Table
    NewMDIChild.txtItem.Text = (MyDataReader.Item(2))

    ' Read Column 4 in Item Table
    NewMDIChild.txtQuantity.Text = (MyDataReader.Item(3))
  Loop
Catch err As System.Exception
  ' Close the Data Reader
  MyDataReader.Close()

  ' Close the Database connection
  OleDbConn.Close()

  ' Destroy the Database connection
  OleDbConn.Dispose()
End Try

' Transfer the result of the highlight data grid to next form
NewMDIChild.txtOrder.Text = _
  DataGridView1.CurrentRow.Cells(0).Value
NewMDIChild.txtCustomer.Text = _
  DataGridView1.CurrentRow.Cells(1).Value

' Enable/Disable the Edit property for the control
NewMDIChild.txtOrder.Enabled = False
NewMDIChild.txtCustomer.Enabled = OutputMode
NewMDIChild.txtItem.Enabled = OutputMode
NewMDIChild.txtQuantity.Enabled = OutputMode
NewMDIChild.btnSave.Enabled = OutputMode
' Set the Parent Form of the Child window.
NewMDIChild.MdiParent = Me.ParentForm

' Display the new form.
NewMDIChild.Show()
End Sub
```


2. Create a **Click** event handler for the Display menu item **DisplayRecordToolStripMenuItem**.

```
Call Get_Order_Detail(False)
```

3. Create a **Click** event handler for the Edit menu item **EditRecordToolStripMenuItem**.

```
Call Get_Order_Detail(True)
```

4. Create a **Click** event handler for the Delete menu item **DeleteRecordToolStripMenuItem**.

```
' Declare the SQL string for deletion
Dim SqlStr1 As String = _
    "DELETE FROM order_detail WHERE orderno = '" _
    & DataGridView1.CurrentRow.Cells(0).Value & "' " _
    & "AND itemno = 1"
Dim SqlStr2 As String = _
    "DELETE FROM order_header WHERE orderno = '" _
    & DataGridView1.CurrentRow.Cells(0).Value & "' "

' Declare the Database connection
Dim OleDbConn As OleDbConnection = New OleDbConnection(ConnString)

' Open the Database Connection
OleDbConn.Open()

' Declare the Database Command
Dim MyOleDbCommand As OleDbCommand = New OleDbCommand

' Declare the Connection for the Database Command
MyOleDbCommand.Connection = (OleDbConn)

' Declare the SQL for the Database Command
MyOleDbCommand.CommandText = SqlStr1

' Delete the data from Order Item table
MyOleDbCommand.ExecuteNonQuery()

' Declare the SQL for the Database Command
MyOleDbCommand.CommandText = SqlStr2

' Delete the data from Order Header table
MyOleDbCommand.ExecuteNonQuery()
```

```

' Close the Database connection
OleDbConn.Close()

' Show the result to the Status Bar
ToolStripStatusLabel1.Text = "Order " & _
    DataGridView1.CurrentRow.Cells(0).Value & " deleted."

' Reload the data to DataGrid
FillSummary()

```

5. Save the project and build the solution, and then execute it.

