Public Class frmSortMethod
    Inherits System.Windows.Forms.Form

    Dim numArray(4) As Integer

    Windows Form Designer generated code

    Private Sub btnEnter_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnEnter.Click
        Dim i As Integer
        For i = 0 To 4
            numArray(i) = InputBox("Enter Number", "Sort")
            lstEnter.Items.Add(numArray(i))
        Next
    End Sub

    Private Sub btnSort_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSort.Click
        Dim i As Integer
        Array.Sort(numArray)
        For i = 0 To 4
            lstSort.Items.Add(numArray(i))
        Next
    End Sub

    End Class
Public Class frmTestRedim
    Inherits System.Windows.Forms.Form

    Dim accuArray() As Integer
    Dim ct As Integer
    Dim lstOriginal As List(Of String)
    Dim vMore As Integer

    Private Sub btnStart_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnStart.Click
        ReDim accuArray(vMore - 1)
        lstOriginal.Add("Entries:")
        ct = 0
        btnGatherAmount.Enabled = True
        btnClear.Enabled = True
        btnStartGather.Enabled = False
        txtAmount.Focus()
    End Sub

    Private Sub btnGatherAmount_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnGatherAmount.Click
        accuArray(ct) = CInt(txtAmount.Text)
        lstOriginal.Add(accuArray(ct))
        ct = ct + 1
        btnGatherAmount.Enabled = False
    End Sub

    Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClear.Click
        lstOriginal.Clear()
        If ct < accuArray.Length Then
            btnGatherAmount.Enabled = True
        End If
        txtAmount.Focus()
    End Sub
If ct < accuArray.Length Then
    btnContGather.Enabled = True
    txtAmountFocus()
Else
    btnClear.Enabled = False
    btnContinue.Enabled = True
End If

Private Sub btnContinue_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim vkMore = InputBox("How many more do you want?")
    ReDim Preserve accuArray(vkHowMany - 1)
    ReDim accuArray(vkHowMany - 1 + vkMore)
    lstSecond.Items.Add("Entries: " & CStr(accuArray(vkHowMany - 1)) & vbCrLf)
    btnContGather.Enabled = True
    btnClearSecond.Enabled = True
End Sub

Private Sub btnContGather_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    accuArray(ct) = CInt(txtAmountSecond.Text)
    lstSecond.Items.Add(accuArray(ct))
    ct = ct + 1
    btnContGather.Enabled = False
End Sub

Private Sub btnClearSecond_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    If ct < accuArray.Length Then
        btnContGather.Enabled = True
        txtAmountFocus()
    Else
        btnShowContents.Enabled = True
    End If
End Sub
Private Sub mnuMultiply_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles mnuMultiply.Click
    Dim intLastResult As Integer
    intLastResult = CInt(txtFirst.Text) * CInt(txtSecond.Text)
    txtAns.Text = intLastResult
End Sub

Private Sub mnuDivide_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles mnuDivide.Click
    Dim intLastResult As Integer
    intLastResult = CInt(txtFirst.Text) / CInt(txtSecond.Text)
    txtAns.Text = intLastResult
End Sub

Private Sub mnuArsRed_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles mnuArsRed.Click
    Dim txtAsColor As String
    txtAsColor = Color.Red
    txtAns.ForeColor = txtAsColor
End Sub

Private Sub mnuPopupAdd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles mnuPopupAdd.Click
    Dim intLastResult As Integer
    intLastResult = CInt(txtFirst.Text) + CInt(txtSecond.Text)
    txtAns.Text = intLastResult
End Sub

Private Sub mnuPopupSubtract_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles mnuPopupSubtract.Click
    Dim intLastResult As Integer
    intLastResult = CInt(txtFirst.Text) - CInt(txtSecond.Text)
    txtAns.Text = intLastResult
End Sub

Private Sub mnuPopupMultiply_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles mnuPopupMultiply.Click
    Dim intLastResult As Integer
    intLastResult = CInt(txtFirst.Text) * CInt(txtSecond.Text)
    txtAns.Text = intLastResult
End Sub

Private Sub mnuPopupDivide_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles mnuPopupDivide.Click
    Dim intLastResult As Integer
    intLastResult = CInt(txtFirst.Text) / CInt(txtSecond.Text)
    txtAns.Text = intLastResult
End Sub
End Class
```vbnet
Private Sub btnOnOne_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnOnOne.Click
    Dim newVarTwo As New frmTwo
    MsgBox("Display before going to Form #2")
    newVarTwo.Show()
    MsgBox("Display from form #1 after form #2 is open")
    MsgBox("Another post form #2 open")
End Sub

Private Sub btnShow_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnShow.Click
    Dim newVarTwo As New frmTwo
    newVarTwo.Show()
End Sub

Private Sub frmOne_Closing(ByVal sender As Object, ByVal e As System.ComponentModel.CancelEventArgs) Handles frmOne.FormClosing
    If MessageBox.Show("Click YES to confirm closing", "Confirm", MessageBoxButtons.YesNo) = DialogResult.Yes Then
        e.Cancel = True
    Else
        e.Cancel = False
    End If
End Sub
End Class
```
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>Text</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Text</td>
<td></td>
</tr>
<tr>
<td>Price</td>
<td>Number</td>
<td></td>
</tr>
<tr>
<td>Cost</td>
<td>Number</td>
<td></td>
</tr>
<tr>
<td>On Hand</td>
<td>Number</td>
<td></td>
</tr>
<tr>
<td>Order</td>
<td>Number</td>
<td></td>
</tr>
</tbody>
</table>

The field description is optional. It helps you describe the field and is also displayed in the status bar when you select this field on a form. Press F1 for help on descriptions.
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Price</th>
<th>Cost</th>
<th>Unprofit</th>
<th>Oorder</th>
</tr>
</thead>
<tbody>
<tr>
<td>11111</td>
<td>mouse</td>
<td>29.99</td>
<td>20</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>22222</td>
<td>console</td>
<td>500</td>
<td>300</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>11111</td>
<td>thumb drive</td>
<td>19.99</td>
<td>10</td>
<td>50</td>
<td>25</td>
</tr>
</tbody>
</table>

The changes you requested to the table were not successful because they would create duplicate values in the index, primary key, or relationship. Change the data in the field or fields that contain duplicate data, remove the index, or redefine the index to permit duplicate entries and try again.
<table>
<thead>
<tr>
<th>item</th>
<th>name</th>
<th>price</th>
<th>cost</th>
<th>onhand</th>
<th>onorder</th>
</tr>
</thead>
<tbody>
<tr>
<td>11111</td>
<td>mouse</td>
<td>29.99</td>
<td>30</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>22222</td>
<td>console</td>
<td>500</td>
<td>300</td>
<td>7</td>
<td>0</td>
</tr>
</tbody>
</table>

Records: [ 1 of 2 ]
SELECT idno, namez, price, cost, onhand, onorder
FROM inventory
WHERE price > 25;
Query1: Select Query

SELECT name, price, cost, onhand, onorder
FROM Inventory
WHERE price > 25 and onhand > 5;
SELECT dia, names, price, cost, onhand, order
FROM Inventory
WHERE price > 25 and (onhand > 5 or order > 5);

either
always
SELECT name, price, cost, onhand, order
FROM Inventory
WHERE price > 25 and onhand > 5 or order > 5

GROUP BY stand alone
<table>
<thead>
<tr>
<th>idno</th>
<th>name</th>
<th>price</th>
<th>cost</th>
<th>onhand</th>
<th>onorder</th>
<th>vendorno</th>
</tr>
</thead>
<tbody>
<tr>
<td>11111</td>
<td>mouse</td>
<td>29.99</td>
<td>20</td>
<td>8</td>
<td>9</td>
<td>ABC</td>
</tr>
<tr>
<td>22222</td>
<td>console</td>
<td>500</td>
<td>300</td>
<td>7</td>
<td>0</td>
<td>DEF</td>
</tr>
<tr>
<td>33333</td>
<td>thumb drive</td>
<td>19.99</td>
<td>10</td>
<td>50</td>
<td>25</td>
<td>ABC</td>
</tr>
<tr>
<td>44444</td>
<td>whatever</td>
<td>25</td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>DEF</td>
</tr>
</tbody>
</table>
Primary key

Foreign key
<table>
<thead>
<tr>
<th>ID</th>
<th>name</th>
<th>inventory</th>
<th>vendor</th>
<th>name</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>mouse</td>
<td>ABC</td>
<td>ABC</td>
<td>Whatever</td>
<td></td>
</tr>
<tr>
<td>2222</td>
<td>console</td>
<td>DEF</td>
<td>DEF</td>
<td>Another</td>
<td></td>
</tr>
<tr>
<td>3333</td>
<td>thumb drive</td>
<td>ABC</td>
<td>ABC</td>
<td>Whatever</td>
<td></td>
</tr>
<tr>
<td>4444</td>
<td>whatever</td>
<td>DEF</td>
<td>DEF</td>
<td>Another</td>
<td></td>
</tr>
</tbody>
</table>

1 + 0 = many
SELECT inventory.idno, inventory.namez, inventory.vendorno, vendor.vendorno, vendor.vendorname
FROM inventory INNER JOIN vendor ON inventory.vendorno = vendor.vendorno;
SELECT idno, namez, inventory.vendorno, vendor.vendorno, vendorname
FROM inventory, vendor
WHERE inventory.vendorno = vendor.vendorno;
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Vendor</th>
<th>Vendor Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>11111</td>
<td>mouse</td>
<td>ABC</td>
<td>Whatever</td>
</tr>
<tr>
<td>22222</td>
<td>console</td>
<td>DEF</td>
<td>Another</td>
</tr>
<tr>
<td>33333</td>
<td>thumb drive</td>
<td>ABC</td>
<td>Whatever</td>
</tr>
<tr>
<td>44444</td>
<td>whatever</td>
<td>DEF</td>
<td>Another</td>
</tr>
</tbody>
</table>
SELECT idno, names, inventory.vendorno, vendor.vendorno, vendname
FROM inventory, vendor
WHERE inventory.vendorno = vendor.vendorno;