Option Compare Database

Private Sub cmbAreaFilter\_AfterUpdate()

Dim selectedArea As Variant

Dim strSQL As String

selectedArea = Me.cmbAreaFilter.Value ' Retrieve the selected value from cmbAreaFilter

If Not IsNull(selectedArea) Then

If selectedArea = "All" Then

Me.Filter = ""

Me.FilterOn = False

Else

Me.Filter = "[Area] = '" & selectedArea & "'"

Me.FilterOn = True

End If

Else

Me.Filter = ""

Me.FilterOn = False

End If

If selectedArea = "All" Then

' Use the original SQL query to show all names

strSQL = "SELECT [ID], [Student Search Name] FROM qry\_PhDStudentSearchName ORDER BY [Student Search Name]"

Else

' Modify the SQL query to filter names based on selectedArea

strSQL = "SELECT q.[ID], q.[Student Search Name] " & \_

"FROM qry\_PhDStudentSearchName AS q " & \_

"INNER JOIN tbl\_PhDTotal AS t ON q.ID = t.ID " & \_

"WHERE t.[Area] = '" & selectedArea & "' " & \_

"ORDER BY q.[Student Search Name]"

End If

' Set the RowSource of Combo35 and requery

Me.Combo35.RowSource = strSQL

Me.Combo35.Requery

End Sub

Private Sub cmbGradYear\_AfterUpdate()

Dim selectedValue As Variant

Dim strSQL As String

selectedValue = Me.cmbGradYear.Value

If Not IsNull(selectedValue) Then

If selectedValue = "All" Then

Me.Filter = ""

Me.FilterOn = False

Else

Me.Filter = "[Graduation Year] = " & selectedValue

Me.FilterOn = True

End If

Else

Me.Filter = ""

Me.FilterOn = False

End If

If selectedValue = "All" Then

' Use the original SQL query to show all names

strSQL = "SELECT [ID], [Student Search Name] FROM qry\_PhDStudentSearchName ORDER BY [Student Search Name]"

Else

' Modify the SQL query to filter names based on selectedValue

strSQL = "SELECT q.[ID], q.[Student Search Name] " & \_

"FROM qry\_PhDStudentSearchName AS q " & \_

"INNER JOIN tbl\_PhDTotal AS t ON q.ID = t.ID " & \_

"WHERE t.[Graduation Year] = " & selectedValue & " " & \_

"ORDER BY q.[Student Search Name]"

End If

' Set the RowSource of Combo35 and requery

Me.Combo35.RowSource = strSQL

Me.Combo35.Requery

End Sub

Private Sub CmbCompanyFilter\_AfterUpdate()

Dim selectedValue As Variant

Dim strSQL As String

selectedValue = Me.CmbCompanyFilter.Value

If Not IsNull(selectedValue) Then

If selectedValue = "All" Then

Me.Filter = ""

Me.FilterOn = False

Else

Me.Filter = "[Company] = '" & selectedValue & "'"

Me.FilterOn = True

End If

Else

Me.Filter = ""

Me.FilterOn = False

End If

If selectedValue = "All" Then

' Use the original SQL query to show all names

strSQL = "SELECT [ID], [Student Search Name] FROM qry\_PhDStudentSearchName ORDER BY [Student Search Name]"

Else

' Modify the SQL query to filter names based on selectedValue

strSQL = "SELECT q.[ID], q.[Student Search Name] " & \_

"FROM qry\_PhDStudentSearchName AS q " & \_

"INNER JOIN tbl\_PhDTotal AS t ON q.ID = t.ID " & \_

"WHERE t.[Company] = '" & selectedValue & "' " & \_

"ORDER BY q.[Student Search Name]"

End If

' Set the RowSource of Combo35 and requery

Me.Combo35.RowSource = strSQL

Me.Combo35.Requery

End Sub

Private Sub cmbYearFilter\_AfterUpdate()

Dim selectedValue As Variant

Dim strSQL As String

selectedValue = Me.cmbYearFilter.Value

If Not IsNull(selectedValue) Then

If selectedValue = "All" Then

Me.Filter = ""

Me.FilterOn = False

Else

Me.Filter = "[Matric Year] = " & selectedValue

Me.FilterOn = True

End If

Else

Me.Filter = ""

Me.FilterOn = False

End If

If selectedValue = "All" Then

' Use the original SQL query to show all names

strSQL = "SELECT [ID], [Student Search Name] FROM qry\_PhDStudentSearchName ORDER BY [Student Search Name]"

Else

' Modify the SQL query to filter names based on selectedValue

strSQL = "SELECT q.[ID], q.[Student Search Name] " & \_

"FROM qry\_PhDStudentSearchName AS q " & \_

"INNER JOIN tbl\_PhDTotal AS t ON q.ID = t.ID " & \_

"WHERE t.[Matric Year] = " & selectedValue & " " & \_

"ORDER BY q.[Student Search Name]"

End If

' Set the RowSource of Combo35 and requery

Me.Combo35.RowSource = strSQL

Me.Combo35.Requery

End Sub

Private Sub Combo35\_AfterUpdate()

' Find the record that matches the control.

Dim rsListBox As Object

Dim IDValue As Variant

IDValue = Me.[Combo35].Value

If Not IsNull(IDValue) Then

Set rsListBox = Me.Recordset.Clone

rsListBox.FindFirst "[ID] = " & IDValue

If Not rsListBox.EOF Then

Me.Bookmark = rsListBox.Bookmark

End If

Set rsListBox = Nothing

End If

End Sub

Private Sub Command32\_Click()

On Error GoTo ErrorHandler

' Check if the form is dirty (changes have been made)

If Me.Dirty Then

' Save the form

DoCmd.RunCommand acCmdSaveRecord

' Store the current record bookmark

Dim currentBookmark As Variant

currentBookmark = Me.Bookmark

' Display a message box

MsgBox "Form saved."

' Reselect the current record

If Not IsNull(currentBookmark) Then

Me.Bookmark = currentBookmark

End If

End If

Exit Sub

ErrorHandler:

' Display error message if necessary

MsgBox "An error occurred while saving the form: " & Err.Description, vbExclamation

' Restore the focus to the form

Me.SetFocus

End Sub

Private Sub Command33\_Click()

Dim db As DAO.Database

Dim rsCompany As DAO.Recordset

Dim rsTitle As DAO.Recordset

Dim fieldValue As Variant

Dim fieldValue2 As Variant

Dim fieldValue3 As Variant

' Display a confirmation message box

Dim response As VbMsgBoxResult

response = MsgBox("Are you sure you want to continue? This will clear data and add old data to a table", vbQuestion + vbYesNo, "Confirmation")

' Get the source field value from the form

fieldValue = Me.Company.Value

fieldValue2 = Me.Title.Value

fieldValue3 = Me.ID.Value

' Check the user's response

If response = vbYes Then

' Continue with the code execution

' Add your code here

' Copy the value to the destination table

Set db = CurrentDb

Set rsCompany = db.OpenRecordset("tbl\_PreviousEmployersPhD", dbOpenDynaset)

rsCompany.AddNew

rsCompany("fkID") = fieldValue3

rsCompany("Previous Employer") = fieldValue

rsCompany("Previous Title") = fieldValue2

rsCompany.Update

rsCompany.Close

Set rsCompany = Nothing

Set db = Nothing

' Clear the source field on the form

Me.Company.Value = Null

Me.Title.Value = Null

' Example: Display a message after the code execution

MsgBox "Code execution completed.", vbInformation, "Confirmation"

Else

' Cancel the code execution

Exit Sub

End If

End Sub

Private Sub Command34\_Click()

' Declare variables

Dim strSQL As String

Dim rs As DAO.Recordset

Dim newID As Long

' Get the highest existing record ID

strSQL = "SELECT MAX(ID) AS MaxID FROM tbl\_PhDTotal"

Set rs = CurrentDb.OpenRecordset(strSQL)

' Generate the new record ID

If rs.RecordCount > 0 And Not IsNull(rs!MaxID) Then

newID = rs!MaxID + 1

Else

newID = 9990001 ' If no existing records, start with ID 1

End If

rs.Close

' Set the form to New Record mode

DoCmd.GoToRecord acDataForm, Me.Name, acNewRec

Me![ID] = newID

Me![Area] = IIf(Nz(Me![Area]) = "", "No Information", Me![Area])

Me![Work Location ID] = Nz(Me![Work Location ID], 4)

End Sub

Private Sub Form\_BeforeInsert(Cancel As Integer)

' Check if any required fields are blank

If IsNull(Me![ID]) Then

MsgBox "Please fill in all required fields.", vbExclamation, "Incomplete Data"

Cancel = True ' Cancel the record creation

End If

End Sub

Private Sub Form\_Current()

Combo35 = ID

End Sub

Private Sub Form\_Load()

Me.OrderBy = "Last Name" ' Replace "YourFieldName" with the field name you want to sort by

Me.OrderByOn = True

' Set the default value of the combo boxes to "All"

Me.cmbGradYear.Value = "All"

Me.CmbCompanyFilter.Value = "All"

Me.cmbYearFilter.Value = "All"

Me.cmbAreaFilter.Value = "All"

' Trigger the AfterUpdate events of the combo boxes to apply the "All" filter

Call cmbGradYear\_AfterUpdate

Call CmbCompanyFilter\_AfterUpdate

Call cmbYearFilter\_AfterUpdate

Call cmbAreaFilter\_AfterUpdate

End Sub

Private Sub ApplyFilters()

Dim selectedArea As Variant

Dim selectedGradYear As Variant

Dim selectedCompany As Variant

Dim selectedMatricYear As Variant

Dim strSQL As String

selectedArea = Me.cmbAreaFilter.Value

selectedGradYear = Me.cmbGradYear.Value

selectedCompany = Me.CmbCompanyFilter.Value

selectedMatricYear = Me.cmbYearFilter.Value

strSQL = "SELECT q.[ID], q.[Student Search Name] " & \_

"FROM qry\_PhDStudentSearchName AS q " & \_

"INNER JOIN tbl\_PhDTotal AS t ON q.ID = t.ID " & \_

"WHERE 1=1"

If Not IsNull(selectedArea) And selectedArea <> "All" Then

strSQL = strSQL & " AND t.[Area] = '" & selectedArea & "'"

End If

If Not IsNull(selectedGradYear) And selectedGradYear <> "All" Then

strSQL = strSQL & " AND t.[Graduation Year] = " & selectedGradYear

End If

If Not IsNull(selectedCompany) And selectedCompany <> "All" Then

strSQL = strSQL & " AND t.[Company] = '" & selectedCompany & "'"

End If

If Not IsNull(selectedMatricYear) And selectedMatricYear <> "All" Then

strSQL = strSQL & " AND t.[Matric Year] = " & selectedMatricYear

End If

strSQL = strSQL & " ORDER BY q.[Student Search Name]"

' Set the RowSource of Combo35 and requery

Me.Combo35.RowSource = strSQL

Me.Combo35.Requery

End Sub

Masters

Option Compare Database

Private Sub cmbMAreaFilter\_AfterUpdate()

Dim selectedArea As Variant

Dim strSQL As String

selectedArea = Me.cmbMAreaFilter.Value ' Retrieve the selected value from cmbAreaFilter

If Not IsNull(selectedArea) Then

If selectedArea = "All" Then

Me.Filter = ""

Me.FilterOn = False

Else

Me.Filter = "[Area] = '" & selectedArea & "'"

Me.FilterOn = True

End If

Else

Me.Filter = ""

Me.FilterOn = False

End If

If selectedArea = "All" Then

' Use the original SQL query to show all names

strSQL = "SELECT [ID], [Student Search Name] FROM qry\_MastersSearchName ORDER BY [Student Search Name]"

Else

' Modify the SQL query to filter names based on selectedArea

strSQL = "SELECT q.[ID], q.[Student Search Name] " & \_

"FROM qry\_MastersSearchName AS q " & \_

"INNER JOIN tbl\_MastersTotal AS t ON q.ID = t.ID " & \_

"WHERE t.[Area] = '" & selectedArea & "' " & \_

"ORDER BY q.[Student Search Name]"

End If

' Set the RowSource of Combo35 and requery

Me.Combo53.RowSource = strSQL

Me.Combo53.Requery

End Sub

Private Sub cmbMGradYear\_AfterUpdate()

Dim selectedValue As Variant

Dim strSQL As String

selectedValue = Me.cmbMGradYear.Value

If Not IsNull(selectedValue) Then

If selectedValue = "All" Then

Me.Filter = ""

Me.FilterOn = False

Else

Me.Filter = "[Graduation Year] = " & selectedValue

Me.FilterOn = True

End If

Else

Me.Filter = ""

Me.FilterOn = False

End If

If selectedValue = "All" Then

' Use the original SQL query to show all names

strSQL = "SELECT [ID], [Student Search Name] FROM qry\_MastersSearchName ORDER BY [Student Search Name]"

Else

' Modify the SQL query to filter names based on selectedValue

strSQL = "SELECT q.[ID], q.[Student Search Name] " & \_

"FROM qry\_MastersSearchName AS q " & \_

"INNER JOIN tbl\_MastersTotal AS t ON q.ID = t.ID " & \_

"WHERE t.[Graduation Year] = " & selectedValue & " " & \_

"ORDER BY q.[Student Search Name]"

End If

' Set the RowSource of Combo35 and requery

Me.Combo53.RowSource = strSQL

Me.Combo53.Requery

End Sub

Private Sub cmbMCompanyFilter\_AfterUpdate()

Dim selectedValue As Variant

Dim strSQL As String

selectedValue = Me.cmbMCompanyFilter.Value

If Not IsNull(selectedValue) Then

If selectedValue = "All" Then

Me.Filter = ""

Me.FilterOn = False

Else

Me.Filter = "[Company] = '" & selectedValue & "'"

Me.FilterOn = True

End If

Else

Me.Filter = ""

Me.FilterOn = False

End If

If selectedValue = "All" Then

' Use the original SQL query to show all names

strSQL = "SELECT [ID], [Student Search Name] FROM qry\_MastersSearchName ORDER BY [Student Search Name]"

Else

' Modify the SQL query to filter names based on selectedValue

strSQL = "SELECT q.[ID], q.[Student Search Name] " & \_

"FROM qry\_MastersSearchName AS q " & \_

"INNER JOIN tbl\_MastersTotal AS t ON q.ID = t.ID " & \_

"WHERE t.[Company] = '" & selectedValue & "' " & \_

"ORDER BY q.[Student Search Name]"

End If

' Set the RowSource of Combo35 and requery

Me.Combo53.RowSource = strSQL

Me.Combo53.Requery

End Sub

Private Sub cmbMYearFilter\_AfterUpdate()

Dim selectedValue As Variant

Dim strSQL As String

selectedValue = Me.cmbMYearFilter.Value

If Not IsNull(selectedValue) Then

If selectedValue = "All" Then

Me.Filter = ""

Me.FilterOn = False

Else

Me.Filter = "[Matric Year] = " & selectedValue

Me.FilterOn = True

End If

Else

Me.Filter = ""

Me.FilterOn = False

End If

If selectedValue = "All" Then

' Use the original SQL query to show all names

strSQL = "SELECT [ID], [Student Search Name] FROM qry\_MastersSearchName ORDER BY [Student Search Name]"

Else

' Modify the SQL query to filter names based on selectedValue

strSQL = "SELECT q.[ID], q.[Student Search Name] " & \_

"FROM qry\_MastersSearchName AS q " & \_

"INNER JOIN tbl\_MastersTotal AS t ON q.ID = t.ID " & \_

"WHERE t.[Matric Year] = " & selectedValue & " " & \_

"ORDER BY q.[Student Search Name]"

End If

' Set the RowSource of Combo35 and requery

Me.Combo53.RowSource = strSQL

Me.Combo53.Requery

End Sub

Private Sub Combo53\_AfterUpdate()

' Find the record that matches the control.

Dim rsListBox As Object

Dim IDValue As Variant

IDValue = Me.Combo53.Value

If Not IsNull(IDValue) Then

Set rsListBox = Me.Recordset.Clone

rsListBox.FindFirst "[ID] = " & IDValue

If Not rsListBox.EOF Then

Me.Bookmark = rsListBox.Bookmark

End If

Set rsListBox = Nothing

End If

End Sub

Private Sub Command35\_Click()

On Error GoTo ErrorHandler

' Check if the form is dirty (changes have been made)

If Me.Dirty Then

' Save the form

DoCmd.RunCommand acCmdSaveRecord

' Store the current record bookmark

Dim currentBookmark As Variant

currentBookmark = Me.Bookmark

' Display a message box

MsgBox "Form saved."

' Reselect the current record

If Not IsNull(currentBookmark) Then

Me.Bookmark = currentBookmark

End If

End If

Exit Sub

ErrorHandler:

' Display error message if necessary

MsgBox "An error occurred while saving the form: " & Err.Description, vbExclamation

' Restore the focus to the form

Me.SetFocus

End Sub

Private Sub Command36\_Click()

Dim db As DAO.Database

Dim rsCompany As DAO.Recordset

Dim rsTitle As DAO.Recordset

Dim fieldValue As Variant

Dim fieldValue2 As Variant

Dim fieldValue3 As Variant

' Display a confirmation message box

Dim response As VbMsgBoxResult

response = MsgBox("Are you sure you want to continue? This will clear data and add old data to a table", vbQuestion + vbYesNo, "Confirmation")

' Get the source field value from the form

fieldValue = Me.Company.Value

fieldValue2 = Me.Title.Value

fieldValue3 = Me.ID.Value

' Check the user's response

If response = vbYes Then

' Continue with the code execution

' Add your code here

' Copy the value to the destination table

Set db = CurrentDb

Set rsCompany = db.OpenRecordset("tbl\_PreviousEmployersMasters", dbOpenDynaset)

rsCompany.AddNew

rsCompany("fkID") = fieldValue3

rsCompany("Company") = fieldValue

rsCompany("Title") = fieldValue2

rsCompany.Update

rsCompany.Close

Set rsCompany = Nothing

Set db = Nothing

' Clear the source field on the form

Me.Company.Value = Null

Me.Title.Value = Null

' Example: Display a message after the code execution

MsgBox "Code execution completed.", vbInformation, "Confirmation"

Else

' Cancel the code execution

Exit Sub

End If

End Sub

Private Sub Command38\_Click()

' Set the form to New Record mode

' Declare variables

Dim strSQL As String

Dim rsNewMasters As DAO.Recordset

Dim newID As Long

' Get the highest existing record ID

strSQL = "SELECT MAX(ID) AS MaxID FROM tbl\_MastersTotal"

Set rsNewMasters = CurrentDb.OpenRecordset(strSQL)

' Generate the new record ID

If rsNewMasters.RecordCount > 0 And Not IsNull(rsNewMasters!MaxID) Then

newID = rsNewMasters!MaxID + 1

Else

newID = 1 ' If no existing records, start with ID 1

End If

rsNewMasters.Close

' Set the form to New Record mode

DoCmd.GoToRecord acDataForm, Me.Name, acNewRec

Me![ID] = newID

Me![Area Code] = Nz(Me![Area Code], 59)

Me![Work Location ID] = Nz(Me![Work Location ID], 4)

End Sub

Private Sub Form\_Current()

Combo53 = [ID]

End Sub

Private Sub Form\_Load()

Me.OrderBy = "Last Name" ' Replace "YourFieldName" with the field name you want to sort by

Me.OrderByOn = True

' Set the default value of the combo boxes to "All"

Me.cmbMGradYear.Value = "All"

Me.cmbMCompanyFilter.Value = "All"

Me.cmbMYearFilter.Value = "All"

Me.cmbMAreaFilter.Value = "All"

' Trigger the AfterUpdate events of the combo boxes to apply the "All" filter

Call cmbMGradYear\_AfterUpdate

Call cmbMCompanyFilter\_AfterUpdate

Call cmbMYearFilter\_AfterUpdate

Call cmbMAreaFilter\_AfterUpdate

End Sub