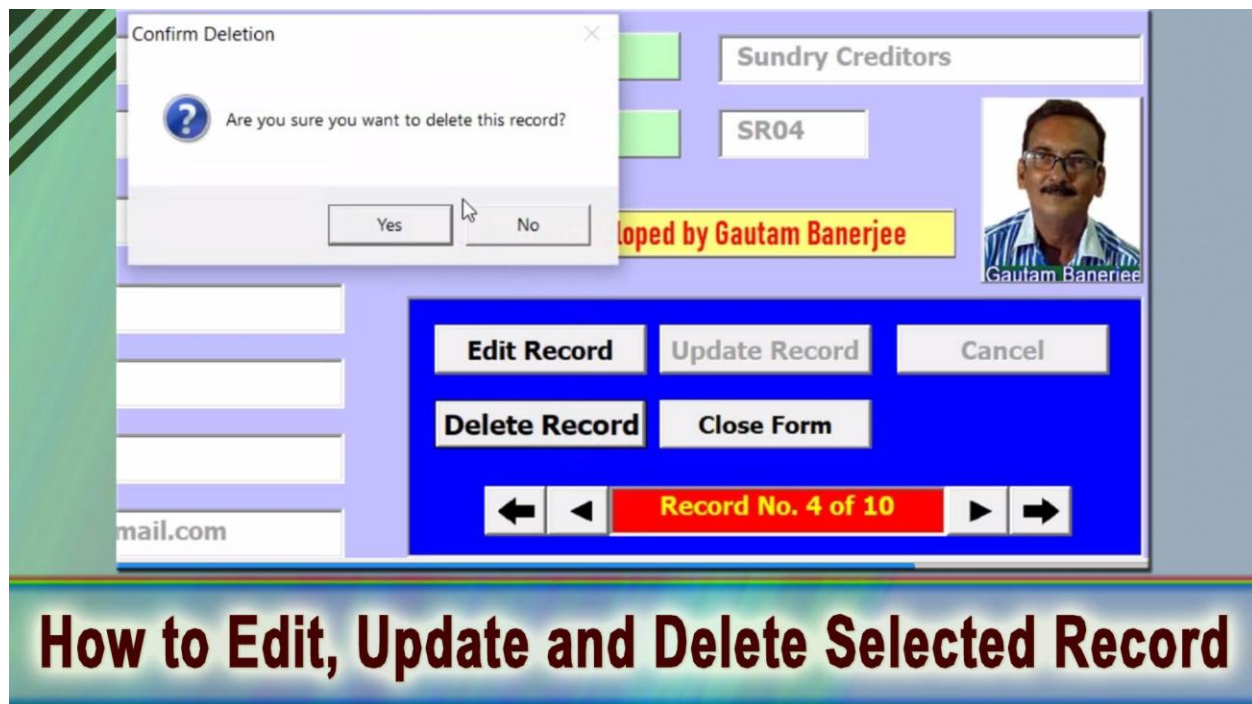




Excel VBA Tutorial: Edit, Update, and Delete Records with User-Friendly Interface

Learn how to effortlessly manage your data using Excel VBA! In this tutorial, we'll walk you through the process of editing, updating, and deleting selected records in your Excel spreadsheet. From user-friendly forms to streamlined actions, discover the power of automation and data manipulation. Get ready to enhance your Excel skills and make data management a breeze.

Edit Button Click :

```
Private Sub CmdEdit_Click()
```

```
    ' Enable editing of fields
```

```
    EnableEditing True
```

' Disable the Edit button

CmdEdit.Enabled = False

' Enable the Update and Cancel buttons

CmdUpdate.Enabled = True

CmdViewCancel.Enabled = True

BtnFirst.Enabled = False

BtnPrev.Enabled = False

BtnNext.Enabled = False

BtnLast.Enabled = False

CmdDelete.Enabled = False

CmdViewClose.Enabled = False

End Sub

When the "Edit Record" button is clicked, this subroutine is executed. It is responsible for preparing the user form for data editing. Here's what each part of the code does:

EnableEditing True: Calls the EnableEditing subroutine with the argument True, allowing the user to edit the fields on the form.

CmdEdit.Enabled = False: Disables the "Edit Record" button to prevent multiple edits while an edit is in progress.

CmdUpdate.Enabled = True: Enables the "Update Record" button to allow the user to save the changes after editing.

CmdViewCancel.Enabled = True: Enables the "Cancel" button, allowing the user to cancel the edit operation.

BtnFirst.Enabled = False, BtnPrev.Enabled = False, BtnNext.Enabled = False, BtnLast.Enabled = False: Disables the navigation buttons (First, Previous, Next, Last) to prevent navigating while editing.

CmdDelete.Enabled = False: Disables the "Delete Record" button to prevent deletion while editing.

CmdViewClose.Enabled = False: Disables the "Close" button to prevent closing the view while editing.

In summary, this code creates a smooth user experience by enabling edit mode, disabling certain buttons that shouldn't be used during editing, and enabling buttons relevant to the editing process. This helps users easily update records without any confusion.

Private Sub EnableEditing(enable As Boolean)

 ' Enable or disable editing of form fields based on the 'enable' parameter

 TxtViewAdd.Enabled = enable

 TxtViewCity.Enabled = enable

 TxtViewPIN.Enabled = enable

 TxtViewState.Enabled = enable

 TxtViewPAN.Enabled = enable

 TXtViewGSTIN.Enabled = enable

 ' Enable or disable other fields as needed

End Sub

This subroutine is designed to enable or disable the editing mode of various form fields on the user interface. The parameter enable determines whether the fields should be enabled for editing (True) or disabled (False). Here's a breakdown of the code's functionality:

`TxtViewAdd.Enabled = enable`: Enables or disables the editing capability of the "Address" field based on the enable parameter.

`TxtViewCity.Enabled = enable`: Similarly, this line enables or disables the "City" field.

`TxtViewPIN.Enabled = enable`: This line does the same for the "PIN" field.

`TxtViewState.Enabled = enable`: This line controls the "State" field's editability.

`TxtViewPAN.Enabled = enable`: Enables or disables the "PAN" field.

`TxtViewGSTIN.Enabled = enable`: Lastly, this line determines whether the "GSTIN" field can be edited or not.

You can adapt this code to include other fields in a similar fashion. This subroutine is helpful for managing the editing capability of multiple fields in a more organized manner.

Update Button Click:

```
Private Sub CmdUpdate_Click()
```

```
    Dim recordIndex As Long
```

```
    recordIndex = currentRow ' Update currentRow with the selected record index
```

```
    CmdDelete.Enabled = False
```

' Update the corresponding row in the worksheet

Dim ws As Worksheet

Set ws = ThisWorkbook.Worksheets("Customer_Master")

ws.Cells(currentRow + 1, 3).Value = TxtViewAdd.Value

ws.Cells(currentRow + 1, 4).Value = TxtViewCity.Value

ws.Cells(currentRow + 1, 5).Value = TxtViewPIN.Value

ws.Cells(currentRow + 1, 6).Value = TxtViewState.Value

ws.Cells(currentRow + 1, 9).Value = TxtViewPAN.Value

ws.Cells(currentRow + 1, 11).Value = TXtViewGSTIN.Value

MsgBox "Record Successfully Updated", vbExclamation, "Success... Gautam Banerjee"

' ... Update other fields as needed

BtnFirst.Enabled = True

BtnPrev.Enabled = True

BtnNext.Enabled = True

BtnLast.Enabled = True

' Disable editing of fields

EnableEditing False

' Enable the Edit button

CmdEdit.Enabled = True

```
' Disable the Update and Cancel buttons  
CmdUpdate.Enabled = False  
CmdViewCancel.Enabled = False  
CmdViewClose.Enabled = True  
CmdDelete.Enabled = True  
End Sub
```

This subroutine is executed when the "Update Record" button is clicked after editing. Here's a breakdown of what the code does:

recordIndex: Stores the index of the currently selected record for later use.

CmdDelete.Enabled = False: Disables the "Delete Record" button during the update process.

Updates the corresponding fields in the worksheet based on the edited values entered by the user.

Displays a message box notifying the user that the record has been successfully updated.

Re-enables navigation buttons (First, Previous, Next, Last) for moving through records.

EnableEditing False: Disables the editing mode, making the fields non-editable again.

Re-enables the "Edit Record" button to allow further edits.

Disables the "Update Record" and "Cancel" buttons, and enables the "Close" and "Delete Record" buttons for navigation and further actions.

This code ensures that the edited data is saved to the worksheet and provides feedback to the user upon successful update. It also manages button states and enables proper navigation and actions after the update.

Cancel Button :

```
Private Sub CmdViewCancel_Click()  
    ' Disable editing of fields  
    EnableEditing False  
  
    ' Enable the Edit button  
    CmdEdit.Enabled = True  
  
    ' Disable the Update and Cancel buttons  
    CmdUpdate.Enabled = False  
    CmdViewCancel.Enabled = False  
  
    BtnFirst.Enabled = True  
    BtnPrev.Enabled = True  
    BtnNext.Enabled = True  
    BtnLast.Enabled = True  
End Sub
```

This subroutine is executed when the "Cancel" button is clicked while viewing or editing a record. Here's an explanation of the code's functionality:

EnableEditing False: Disables the editing mode, making the fields non-editable.

CmdEdit.Enabled = True: Enables the "Edit Record" button to allow further edits.

Disables the "Update Record" and "Cancel" buttons, as these buttons are relevant only during editing.

Re-enables navigation buttons (First, Previous, Next, Last) to allow movement through records.

This code ensures that when the user cancels their editing or viewing action, the fields are set to non-editable mode, and the buttons are adjusted to their appropriate states for navigation and actions.

Delete Button :

```
Private Sub CmdDelete_Click()
```

```
    Dim ws As Worksheet
```

```
    Set ws = ThisWorkbook.Worksheets("Customer_Master")
```

```
    Dim deletedWs As Worksheet
```

```
    Set deletedWs = ThisWorkbook.Worksheets("Customer_Deleted")
```

```
    Dim selectedName As String
```

```
    selectedName = TxtViewName.Value
```

```
    Dim rng As Range
```

```
    Set rng = ws.Columns("B:B").Find(What:=selectedName, LookIn:=xlValues,  
LookAt:=xlWhole)
```

```
    If Not rng Is Nothing Then
```

```
        Dim deleteRow As Long
```

```

deleteRow = rng.Row

' Confirm deletion with a message box
Dim response As VbMsgBoxResult
response = MsgBox("Are you sure you want to delete this record?", vbYesNo +
vbQuestion, "Confirm Deletion")

If response = vbYes Then
    ' Transfer data to "Customer_Deleted" sheet
    Dim lastDeletedRow As Long
    lastDeletedRow      =      deletedWs.Cells(deletedWs.Rows.Count,
"A").End(xlUp).Row
    ws.Rows(deleteRow).Copy deletedWs.Rows(lastDeletedRow + 1)

    ' Delete the row from "Customer_Master" sheet
    ws.Rows(deleteRow).Delete

    ' Refresh list box and show next record
    PopulateListBox
    ShowRecord currentRow
End If
End If
End Sub

```

The subroutine starts by setting up references to the "Customer_Master" and "Customer_Deleted" worksheets.

It gets the selected customer's name from the TxtViewName field on the form.

Using the customer's name, it searches for the corresponding row in the "Customer_Master" sheet.

If the customer is found, a confirmation message box is displayed to ensure the user intends to delete the record.

If the user confirms deletion, the data from the deleted row is copied to the "Customer_Deleted" sheet, and the row is then deleted from the "Customer_Master" sheet.

After deletion, the list box is refreshed using the PopulateListBox subroutine, and the next record is shown using the ShowRecord subroutine.



Gautam Banerjee

“Helping beginners learn something new is a great way to share your knowledge and make a positive impact”.

Email: gincom1@yahoo.com

If you have any queries, please visit our "Contact Us" page.

Thank You. See you again!

PLEASE

Gautam Banerjee

Age: 63

Design Form, Name to Text Box, Command Button etc. and Copy Below Code and Paste in VBA Code Window.

```
Dim currentRecord As Long
```

```
Dim currentRow As Long
```

```
Private Sub BtnFirst_Click()
```

```
    ShowRecord 1
```

```
    CmdEdit.Enabled = True
```

```
End Sub
```

```
Private Sub BtnLast_Click()
```

```
    Dim ws As Worksheet
```

```
    Set ws = ThisWorkbook.Worksheets("Customer_Master")
```

```
    Dim lastRow As Long
```

```
    lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row
```

```
    ShowRecord lastRow - 1
```

```
    LblRecordNo.Caption = "Record No. " & lastRow - 1 & " of " & lastRow - 1
```

```
CmdEdit.Enabled = True
```

```
End Sub
```

```
Private Sub BtnNext_Click()
```

```
    If currentRow > lastRow - 1 Then
```

```
        currentRow = currentRow + 1
```

```
        ShowRecord (currentRow)
```

```
    End If
```

```
    CmdEdit.Enabled = True
```

```
End Sub
```

```
Private Sub BtnPrev_Click()
```

```
    If currentRow > 0 Then
```

```
        currentRow = currentRow - 1
```

```
        ShowRecord (currentRow)
```

```
    End If
```

```
    CmdEdit.Enabled = True
```

```
End Sub
```

```

Private Sub CmdDelete_Click()

    Dim ws As Worksheet

    Set ws = ThisWorkbook.Worksheets("Customer_Master")


    Dim deletedWs As Worksheet
    Set deletedWs = ThisWorkbook.Worksheets("Customer_Deleted")


    Dim selectedName As String
    selectedName = TxtViewName.Value


    Dim rng As Range

    Set rng = ws.Columns("B:B").Find(What:=selectedName, LookIn:=xlValues,
LookAt:=xlWhole)


    If Not rng Is Nothing Then

        Dim deleteRow As Long
        deleteRow = rng.Row


        ' Confirm deletion with a message box

        Dim response As VbMsgBoxResult

        response = MsgBox("Are you sure you want to delete this record?", vbYesNo +
vbQuestion, "Confirm Deletion")

```

```

If response = vbYes Then
    ' Transfer data to "Customer_Deleted" sheet
    Dim lastDeletedRow As Long
    lastDeletedRow = deletedWs.Cells(deletedWs.Rows.Count,
"A").End(xlUp).Row
    ws.Rows(deleteRow).Copy deletedWs.Rows(lastDeletedRow + 1)

    ' Delete the row from "Customer_Master" sheet
    ws.Rows(deleteRow).Delete

    ' Refresh list box and show next record
    PopulateListBox
    ShowRecord currentRow
End If
End If
End Sub

```

```

Private Sub CmdEdit_Click()
    ' Enable editing of fields
    EnableEditing True

    ' Disable the Edit button
    CmdEdit.Enabled = False

```

' Enable the Update and Cancel buttons

CmdUpdate.Enabled = True

CmdViewCancel.Enabled = True

BtnFirst.Enabled = False

BtnPrev.Enabled = False

BtnNext.Enabled = False

BtnLast.Enabled = False

CmdDelete.Enabled = False

CmdViewClose.Enabled = False

End Sub

Private Sub EnableEditing(enable As Boolean)

' Enable or disable editing of form fields based on the 'enable' parameter

TxtViewAdd.Enabled = enable

TxtViewCity.Enabled = enable

TxtViewPIN.Enabled = enable

TxtViewState.Enabled = enable

TxtViewPAN.Enabled = enable

TXtViewGSTIN.Enabled = enable

' Enable or disable other fields as needed

End Sub

Private Sub DisableTextBox()

TxtViewID.Enabled = False

TxtViewName.Enabled = False

TxtViewAdd.Enabled = False

TxtViewCity.Enabled = False

TxtViewPIN.Enabled = False

TxtViewState.Enabled = False

TxtViewPhone.Enabled = False

TxtViewEmail.Enabled = False

TxtViewPAN.Enabled = False

TxtViewType.Enabled = False

'If ws.Cells(recordIndex, 12).Value = "SD" Then

 ' TxtViewType.Value = "Sundry Debtors"

' Else

 ' TxtViewType.Value = "Sundry Creditors"

' End If

TXtViewGSTIN.Enabled = False

TxtViewSRID.Enabled = False

CmdUpdate.Enabled = False

CmdViewCancel.Enabled = False

End Sub

Private Sub CmdUpdate_Click()

Dim recordIndex As Long

recordIndex = currentRow ' Update currentRow with the selected record index

CmdDelete.Enabled = False

' Update the corresponding row in the worksheet

Dim ws As Worksheet

Set ws = ThisWorkbook.Worksheets("Customer_Master")

ws.Cells(currentRow + 1, 3).Value = TxtViewAdd.Value

ws.Cells(currentRow + 1, 4).Value = TxtViewCity.Value

ws.Cells(currentRow + 1, 5).Value = TxtViewPIN.Value

ws.Cells(currentRow + 1, 6).Value = TxtViewState.Value

ws.Cells(currentRow + 1, 9).Value = TxtViewPAN.Value

ws.Cells(currentRow + 1, 11).Value = TXtViewGSTIN.Value

MsgBox "Record Successfully Updated", vbExclamation, "Success... Gautam Banerjee"

' ... Update other fields as needed

BtnFirst.Enabled = True

BtnPrev.Enabled = True

BtnNext.Enabled = True

BtnLast.Enabled = True

' Disable editing of fields

EnableEditing False

' Enable the Edit button

CmdEdit.Enabled = True

' Disable the Update and Cancel buttons

CmdUpdate.Enabled = False

CmdViewCancel.Enabled = False

CmdViewClose.Enabled = True

CmdDelete.Enabled = True

End Sub

Private Sub CmdViewCancel_Click()

' Disable editing of fields

EnableEditing False

' Enable the Edit button

```
CmdEdit.Enabled = True
```

```
' Disable the Update and Cancel buttons
```

```
CmdUpdate.Enabled = False
```

```
CmdViewCancel.Enabled = False
```

```
BtnFirst.Enabled = True
```

```
BtnPrev.Enabled = True
```

```
BtnNext.Enabled = True
```

```
BtnLast.Enabled = True
```

```
End Sub
```

```
Private Sub CmdViewClose_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub UserForm_Initialize()
```

```
' Initialize the user form
```

```
PopulateListBox
```

```
DisableTextBox
```

```
'TxtViewLength
```

```
ShowRecord 1 ' Display the first record initially
```

End Sub

Private Sub PopulateListBox()

 ' Populate the list box with customer names

 Dim ws As Worksheet

 Set ws = ThisWorkbook.Worksheets("Customer_Master")

 Dim lastRow As Long

 lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row

 Dim rng As Range

 Set rng = ws.Range("B2:B" & lastRow)

 TxtNameViewList.Clear

 TxtNameViewList.List = rng.Value

End Sub

Private Sub ShowRecord(recordIndex As Long)

 Dim ws As Worksheet

 Set ws = ThisWorkbook.Worksheets("Customer_Master")

Dim lastRow As Long

lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row

If recordIndex < 1 Then recordIndex = 1

If recordIndex > lastRow - 1 Then recordIndex = lastRow - 1

currentRow = recordIndex

LblRecordNo.Caption = "Record No. " & currentRow & " of " & lastRow - 1

TxtViewID.Value = ws.Cells(recordIndex + 1, 1).Value

TxtViewName.Value = ws.Cells(recordIndex + 1, 2).Value

TxtViewAdd.Value = ws.Cells(recordIndex + 1, 3).Value

TxtViewCity.Value = ws.Cells(recordIndex + 1, 4).Value

TxtViewPIN.Value = ws.Cells(recordIndex + 1, 5).Value

TxtViewState.Value = ws.Cells(recordIndex + 1, 6).Value

TxtViewPhone.Value = ws.Cells(recordIndex + 1, 7).Value

TxtViewEmail.Value = ws.Cells(recordIndex + 1, 8).Value

TxtViewPAN.Value = ws.Cells(recordIndex + 1, 9).Value

If ws.Cells(recordIndex, 12).Value = "SD" Then

 TxtViewType.Value = "Sundry Debtors"

Else

 TxtViewType.Value = "Sundry Creditors"

End If

TxtViewGSTIN.Value = ws.Cells(recordIndex + 1, 11).Value

TxtViewSRID.Value = ws.Cells(recordIndex + 1, 13).Value

CmdUpdate.Enabled = False

CmdViewCancel.Enabled = False

End Sub

Private Sub TxtNameViewList_Click()

Dim selectedName As String

selectedName = TxtNameViewList.Value

Dim ws As Worksheet

Set ws = ThisWorkbook.Worksheets("Customer_Master")

Dim rng As Range

Set rng = ws.Columns("B:B").Find(What:=selectedName, LookIn:=xlValues,
LookAt:=xlWhole)

If Not rng Is Nothing Then

Dim recordIndex As Long

recordIndex = rng.Row - 1 ' Adjust for header row

ShowRecord recordIndex

End If

End Sub

Sub TxtViewLength()

TxtViewID.MaxLength = 3

TxtViewName.MaxLength = 25

TxtViewAdd.MaxLength = 30

TxtViewCity.MaxLength = 25

TxtViewPIN.MaxLength = 6

TxtViewPhone.MaxLength = 10

TxtViewEmail.MaxLength = 40

TxtViewPAN.MaxLength = 10

TXtViewGSTIN.MaxLength = 15

End Sub

If any queries please e-mail : gincom1@yahoo.com

Video Link: 1. <https://youtu.be/d600PTI14t4> 2. <https://youtu.be/Q9tqbxaBNww>



Gautam Banerjee

“Helping beginners learn something new is a great way to share your knowledge and make a positive impact”.

Email: gincom1@yahoo.com

If you have any queries, please visit our "Contact Us" page.

Thank You. See you again!



Gautam Banerjee

Age: 63

Pay by UPI

9748327614