



Create the following subform based on tbl_Pets as a datasheet form. Save the form as frm_Final_sub.

Field Name	Field Value
PetID	PetID
CustomerID	CustomerID
PetName	PetName
TypeOfAnimal	TypeOfAnimal
DoB	DoB
Gender	Gender

Property	Value
Selection type	Form
Form	Form
Caption	
Default View	Datasheet
Allow Form View	No
Allow Datasheet View	Yes
Allow PivotTable View	No
Allow PivotChart View	No
Allow Layout View	No
Picture	(none)
Picture Tiling	No
Picture Alignment	Center
Picture Type	Embedded
Picture Size Mode	Clip
Width	4.0313"
Auto Center	Yes
Auto Resize	Yes
Fit to Screen	Yes
Border Style	Sizable
Record Selectors	No
Navigation Buttons	No
Navigation Caption	
Dividing Lines	No
Scroll Bars	Vertical Only

Create the following main form based on tbl_Customer and save it as frm_Final:

Field Name	Field Value
CustomerID	CustomerID
Type of Customer	TypeofCustomer
CustomerName	CustomerName
Street/Apt	Street/Apt
City	City
State	State
ZipCode	ZipCode
PhoneNumber	PhoneNumber

Property	Value
Selection type	Form
Form	Form
Caption	
Default View	Single Form
Allow Form View	Yes
Allow Datasheet View	No
Allow PivotTable View	No
Allow PivotChart View	No
Allow Layout View	No
Picture	(none)
Picture Tiling	No
Picture Alignment	Center
Picture Type	Embedded
Picture Size Mode	Clip
Width	8.75"
Auto Center	No
Auto Resize	Yes
Fit to Screen	Yes
Border Style	Sizable
Record Selectors	No
Navigation Buttons	Yes
Navigation Caption	
Dividing Lines	No
Scroll Bars	Neither



On the main form, create a subform control and use frm_Final_sub as the subform. Use the wizard to build the subform control. Test your form at this point that it shows correctly for each customer their pets.

Here is the actual problem. The staff at the Mountain Animal Hospital would like to enter a new customer and at the same time create the number of pet records needed for this new customer.

Add the following features to the form:

New Customer button:

Create a new button on the form as shown above and do not use the command button wizard. Name the button cmdNewCustomer. Set its caption to New Customer.

Create Pet Records button:

Create an unbound text box and name it txtNoPets. Name the associated label lblNoPets. Add a command button, do not use the command button wizard (simply cancel the wizard). Name the button cmdCreatePets. Select the label, textbox and the command button for creating new pet records, and the subform control and set their Visible property to False. That means, when the form opens these controls will not be shown initially.



Specific VBA Tasks:

1. Event procedure for cmdNewCustomer:

Once the user clicks this button they are ready to add a new customer. Use the following command to add a new record: `DoCmd.GoToRecord , , acNewRec` . Set the `txtNoPets` to Null in case there was a value from a previous operation. Also hide the subform by setting the subform's `Visible` property to `False`, again in case if from a previous operation the subform was visible.

Unhide the label and textbox control of `NoPets` as well as the command button for creating new pet records (`cmdCreatePets`). Lastly, move the focus to the `CustomerID` control by using the following method: `CustomerID.SetFocus`.

Make sure to save and then test the functionality.

2. Event procedure for cmdCreatePets:

The basic idea is to create loop to create pet records. The loop starts at 1 and ends at the value entered in the `NoPets` textbox on the form. You only need to populate the following fields:

CustomerID: The same value as entered on the form

PetID: The customerID as entered on the form, then a hyphen, and a two-digit counter (leading 0, for example 05)

PetName: Just populate `Pet` as a name and concatenate the counter number, such `Pet1`, `Pet2`, `Pet3`, etc.

You may use the `Docmd.RunSQL` method to create Insert statements (`Insert into tbl_Pets(CustomerID, PetID, PetName) VALUES(your values)`) or use `DAO (rs.AddNew)`.

Also include some validation rules:

If the number of pets entered is 0 or null, issue a message box statement alerting the user and not run the procedure.

The maximum number of pets to be added per customer is 5, in case it is 6 or greater issue a message box statement alerting the use and not run the procedure.

If the user enters a valid number, you need to first save the entered customer record on the form. Use the `Docmd.RunCommand acCmdSaveRecord` statement. Then create a loop to execute the Inserts into `tbl_Pets`. Create a loop integer variable `intI` to keep track of the loop status. Also use this variable to build the two-digit counter for the `PetID` and the counter for the `PetName`.

After the pet records are inserted, issue a requery on the subform: `frm_Final_sub.Requery`. And finally show the subform control by setting its `Visible` property to `True`.



A sample view of adding a new customer and 4 pet records is shown below:

The screenshot shows a Microsoft Access form titled 'frm_Final'. It contains several text boxes for customer information and a table for pet records.

Customer Information:

- CustomerID: ZZ100
- Type of Customer: 1
- CustomerName: New Customer
- Street/Apt: (empty)
- City: (empty)
- State: (empty)
- ZipCode: (empty)
- PhoneNumber: (empty)

A 'New Customer' button is located to the right of the CustomerID field.

Pet Records Table:

PetID	CustomerID	PetName	TypeOfAnimal	DoB	Gender
ZZ100-01	ZZ100	Pet1			
ZZ100-02	ZZ100	Pet2			
ZZ100-03	ZZ100	Pet3			
ZZ100-04	ZZ100	Pet4			

At the bottom of the form, there is a status bar showing 'Record: 57 of 57', 'No Filter', and a 'Search' button.