Protected Sub Button3\_Click(sender As Object, e As EventArgs) Handles Button3.Click

'Here we DELETE a customer record. IF they have any sales, those sales are archived first then deleted from the sales table. 'When the tables are connected using a foreign key | primary key relationship you have to delete the related transaction records before you can delete a row from a dimension table (here customers).

Dim intSales As Integer 'used to show how many sales the customer had

'Here we will intercept the delete request and check if the customer has any sales, by counting the number of sales.

'If it is discovered that there are zero sales for the customer, then sure delete the customer. Otherwise,

'if it is discovered that the customer has sales then a process of moving sales to archives is performed

'when next line is run it can count the number of rows in the table that meet the criteria specified (a selected customer)

Dim cmdCountSalesForOneCustomer As New SqlCommand("SELECT COUNT(\*) From Sales2020 WHERE CustomerID = @p1", con)

'Archiving data means copying the rows of data from one database table to another. Often transaction records more than two years old are moved to archives in an effort to manage the size of the operational database table. This is an excellent SQL statement that copies (inserts) rows that will be deleted FROM the Sales table INTO the Sales Archives table.

Dim cmdSaveSalesToArchives As New SqlCommand("INSERT INTO Sales2020Archives SELECT \* FROM Sales2020 WHERE CustomerID = @p1", con)

'the program user will want to see the archived sales records. This SQL SELECT statement does that.

Dim DaShowArchives As New SqlDataAdapter("SELECT \* FROM Sales2020Archives WHERE CustomerID = @p1", con)

'After the sales have been copied to the archives table, we use this SQLcommand to delete the sales rows but ONLY for the customer that will be deleted.

Dim cmdDeleteSalesFirstB4DeleteCustomer As New SqlCommand("DELETE From Sales2020 WHERE CustomerID = @p1", con)

'Now that the customer’s sales rows have been archived and deleted, we use this SQLcommand delete the customer record.

Dim cmdDeleteCustomer As New SqlCommand("DELETE From Customers2020 WHERE CustomerID = @p1", con)

'here the parameters are set to identify the specific customer, and pass their CustomerID number to the SQL queries

With cmdSaveSalesToArchives.Parameters

.Clear()

.AddWithValue("@p1", DropDownList2.SelectedValue)

End With

With DaShowArchives.SelectCommand.Parameters

.Clear()

.AddWithValue("@p1", DropDownList2.SelectedValue)

End With

With cmdCountSalesForOneCustomer.Parameters

.Clear()

.AddWithValue("@p1", DropDownList2.SelectedValue)

End With

With cmdDeleteCustomer.Parameters

.Clear()

.AddWithValue("@p1", DropDownList2.SelectedValue)

End With

With cmdDeleteSalesFirstB4DeleteCustomer.Parameters

.Clear()

.AddWithValue("@p1", DropDownList2.SelectedValue)

End With

'All the code above was set-up. Here the data management occurs. When the customer is selected for deletion, we count the sales for the customer selected. If they have sales records then we have to archive them (save to another table) and delete the sales before we delete the customer record. In database terms when you delete a row in the parent table (customers) you want to make sure you do not orphan any of the records in the related child table (sales) Can you archive the customer?

Try

If con.State = ConnectionState.Closed Then con.Open()

‘Executescalar is used when you are fetching one number (calculated or not) from a database (rather than a set of rows). In math terms a scalar is a number. Here we calculate a count based on a criteria, and assign the result to a local variable

intSales = cmdCountSalesForOneCustomer.ExecuteScalar

If intSales >= 1 Then

'if the customer had sales, save the sales to archives, which is a second table with the same schema as Sales.

cmdSaveSalesToArchives.ExecuteNonQuery()

'Next display the records that are moved to archives, so clear and refill the global table, and display them.

If gArchivesTable.Rows.Count > 0 Then gArchivesTable.Rows.Clear()

DaShowArchives.Fill(gArchivesTable)

gvArchives.DataSource = gArchivesTable

gvArchives.DataBind()

'after the sales are archived then delete the sales for the selected customer, and then delete the customer record

cmdDeleteSalesFirstB4DeleteCustomer.ExecuteNonQuery()

cmdDeleteCustomer.ExecuteNonQuery()

txtOutput.Text = DropDownList2.SelectedItem.Text & " was removed, and their " & intSales   
 & " sales in the connected sales table was archived."

Else

'This section of code only runs if there are zero sales records, therefore no sales to archive.

If con.State = ConnectionState.Closed Then con.Open()

cmdDeleteCustomer.ExecuteNonQuery() 'now we delete the customer record.

txtOutput.Text = "Records for " & DropDownList2.SelectedItem.Text & " removed. They had "   
 & gSalesTable.Rows.Count & " sales."  
End If

'After deleting a customer we need to refresh the dropdown lists to show that customer is no longer on the list

UpdateDDL()

'Also after deleting a customer you need to refresh the customer list in the gridview

Call getCustomers()

Catch ex As Exception

Response.Write(ex.Message)

Finally

con.Close()

End Try End Sub