

## Field Observation Screen Shots and VBA/SQL Code

This was a customized Access database application I created using VBA and SQL. The purpose of the application was to ensure that work facilities at Southern California Edison Company were following appropriate safety procedures. In order to do this, I created a Field Observation System which was a series of safety related questions. Answers for each observation question were saved to the database. Safety specialists could view the Field Observation answer details to determine if a work facility was following safety procedures. The safety related questions were different for every business group and the application was programmed to accommodate this requirement.

This application was created by me when I was an Edison employee. Then when I started my own business, Edison used me as a consultant to make further database design and programming changes for the purpose of upgrading the functionality of the Field Observation System.

Questions	NA/OK?	Comments/Corrective Actions
1.01. Discuss purpose and critical steps of job	-	
1.02. Each employee understands what they must do	-	
1.03. Each employee understands what other employees must do	-	
1.04. Each employee understands supervisor's methods for job	-	
1.05. Each employee understands hazards, actions to overcome and PPE needed	-	
1.06. Each employee notified of conditions or clearance on lines and equipment	-	
1.07. Changing conditions recognized and new tailboard conducted	-	
2.01. Appropriate clothing is worn at all times	-	
2.02. Head protection is used when needed	-	
2.03. Gloves are worn when needed (rubber and/or leather)	-	
2.04. Reflective vests are worn when needed	-	
2.05. Eye protection and face shields are worn when needed	-	
2.06. Hearing protection worn when needed	-	
2.07. Respiratory protection worn when needed	-	
2.08. Fall protection gear worn when needed	-	
3.01. Documented control procedure used	-	
3.02. All appropriate equipment is identified for control	-	
3.03. All appropriate equipment is blocked, locked and/or tagged	-	
3.04. Safe work distances maintained	-	
3.05. Protective covers used appropriately	-	
4.01. Approved signs, barriers, guards, flags and/or lights are used and maintained	-	
4.02. Warning equipment appropriately placed to provide adequate notice to public	-	
4.03. Flaggers provided when needed	-	
5.01. Conduct station/equipment inspection	-	

Figure 1: Partial screen shot of the save field observation record form.

Below in blue font is the VBA and SQL written to save a field observation record to the database. When a user clicked on the button "Save Field Observation" in Figure 1 above the following code was executed:

```
Private Sub cmdSaveFieldObservation_Click()  
    'calls on sub procedure to execute the save operation  
    Me.ProcSaveQuestions  
End Sub
```

```

*save new record to the database
Public Sub ProcSaveQuestions()
On Error GoTo errHand
    Dim ctl As Control
    Dim str As String
    Dim msgstr As String
    Dim sql As String, rs As New ADODB.Recordset
    Dim observationID As Single, res As Integer
    Dim textBoxComments As TextBox
    Dim comboBoxAnswer As ComboBox
    Dim textBoxControlName As String
    Dim comboBoxControlName As String
    Dim questNum As Double
    Dim questID As Integer
    Dim sql2 As String, rs2 As New ADODB.Recordset
    Dim counter As Integer
    counter = 0

'error handling performed
If IsNull(Me.cboObservationCatID.Value) = True Then
    MsgBox "Please pick an observation category", vbInformation, "Required Field"
    Me.cboObservationCatID.SetFocus
    Exit Sub
End If

'code updated May 2009
'code added May 2009
If IsNull(Me.cboEmployeeID.Value) = True Then
    MsgBox "Please pick an observer", vbInformation, "Required Field"
    Me.cboEmployeeID.SetFocus
    Exit Sub
End If

If IsNull(Me.cboObservationDate.Value) = True Then
    MsgBox "Please pick an observation date", vbInformation, "Required Field"
    Me.cboObservationDate.SetFocus
    Exit Sub
End If

If IsNull(Me.cboShiftID.Value) = True Then
    MsgBox "Please pick Shift", vbInformation, "Required Field"
    Me.cboShiftID.SetFocus
    Exit Sub
End If

If IsNull(Me.cboLocation.Value) = True Then
    MsgBox "Please pick a location", vbInformation, "Required Field"
    Me.cboLocation.SetFocus
    Exit Sub
End If

```

```

If IsNull(Me.cboTimeSpentID.Value) = True Then
    MsgBox "Please pick the amount of time spent on observation", vbInformation, "Required Field"
    Me.cboTimeSpentID.SetFocus
    Exit Sub
End If

```

```

If IsNull(Me.cboWorkTypeID.Value) = True Then
    MsgBox "Please pick a work type", vbInformation, "Required Field"
    Me.cboWorkTypeID.SetFocus
    Exit Sub
End If

```

```

If IsNull(Me.cboCrewType.Value) = True Then
    MsgBox "Please enter a crew type", vbInformation, "Required Field"
    Me.cboCrewType.SetFocus
    Exit Sub
End If

```

```

'code added 4/30/2008
'check to ensure that all the fields are answered. specifically the combo box fields
sql = "Select Questions.questID, Questions.questNum, Questions.textBoxControlName,
Questions.comboBoxControlName from "
sql = sql & "Questions inner join ObservationCategoryQuestions on Questions.questID =
ObservationCategoryQuestions.questID "
sql = sql & "where ObservationCategoryQuestions.observationCatID = " & Me.cboObservationCatID.Value
sql = sql & " and questions.questID not in (83,84,85) "
sql = sql & " order by Questions.questID asc"

```

```

rs.Open sql, CurrentProject.Connection, adOpenDynamic, adLockOptimistic
Do Until rs.EOF

```

```

    questID = rs("questID").Value
    questNum = rs("questNum").Value
    textBoxControlName = rs("textBoxControlName").Value
    comboBoxControlName = rs("comboBoxControlName").Value
    .....
```

```

    For Each ctl In Me.Controls
        If TypeOf ctl Is ComboBox Then
            Set comboBoxAnswer = ctl
            If comboBoxAnswer.Name = comboBoxControlName Then
                counter = 1
                'check for null values. code updated May 2009
                'not checking for null values anymore because this is a change made by Lisa Bloch
                'and Martin Reta not to check for null values anymore
                'If IsNull(comboBoxAnswer.Value) = True Then
                    'comboBoxAnswer.SetFocus
                    'MsgBox "Please enter an answer for question number " & questNum, vbInformation, "Data Entry
Error Message"
                    'Exit Sub
                'End If
                'code added May 2009. This is a requirement added by Lisha Bloch
                'and Martin Reta. Now if "?" is the answer choice then the equivalent text

```

```

'box needs to have a comment. very important code
If IsNull(comboBoxAnswer.Value) = False Then
    If comboBoxAnswer.Value = "?" Then
'then make sure there is a comment for answers marked as "?"
        Set textBoxComments = Me.Form.Controls.Item(textBoxControlName)
        If IsNull(textBoxComments.Value) = True Then
            textBoxComments.SetFocus
            MsgBox "Please enter a comment for question number " & questNum, vbInformation, "Data
Entry Error Message"
                Exit Sub
            End If
        End If
    End If

'check to make sure that each comment box has less than equal to 250 characters
Set textBoxComments = Me.Form.Controls.Item(textBoxControlName)
If IsNull(textBoxComments.Value) = False Then
    charLength = Len(textBoxComments.Value)
    If charLength > 250 Then
        textBoxComments.SetFocus
        MsgBox "Comment for question number " & questNum & " has " & charLength & " characters.
Maximum character limit is 250 characters.", vbInformation, "Data Entry Error Message"
            Exit Sub
        End If
    End If
'check to make sure that each comment box has less than equal to 250 characters

    End If
End If
.....
If counter = 1 Then
    counter = 0
    Exit For 'exit the looping of the controls on the form
End If
Next ctl
rs.MoveNext
Loop
rs.Close
'check to ensure all the fields are answered above
res = MsgBox("Are you sure you want to SAVE this field observation record?", vbYesNoCancel, "Save field
observation confirmation")
If res = vbNo Or res = vbCancel Then
    MsgBox "Field observation record will NOT be saved to the database.", vbOKOnly, "Field Observation record
NOT saved"
    Exit Sub
End If

'save header observation information into the observation table
sql = "Select * from Observation"
rs.Open sql, CurrentProject.Connection, adOpenDynamic, adLockOptimistic

```

```

rs.AddNew
rs("observationCatID").Value = Me.cboObservationCatID.Value
rs("employeeID").Value = Me.cboEmployeeID.Value

rs("observationDate").Value = FormatDateTime(Me.cboObservationDate.Value, vbShortDate)
rs("dateFirstSaved").Value = FormatDateTime(Date, vbShortDate)
rs("weekNumber").Value = functions.GetWeekNumber(Me.cboObservationDate.Value)
rs("shiftID").Value = Me.cboShiftID.Value
rs("timeSpentID").Value = Me.cboTimeSpentID.Value
rs("workTypeID").Value = Me.cboWorkTypeID.Value
rs("crewTypeID").Value = Me.cboCrewType.Value
rs("locationID").Value = Me.cboLocation.Value

observationID = rs("observationID").Value
rs.Update
rs.Close

'save the questions and answers for the observation
counter = 0
sql = "Select Questions.questID, Questions.questNum, Questions.textBoxControlName,
Questions.comboBoxControlName from "
sql = sql & "Questions inner join ObservationCategoryQuestions on Questions.questID =
ObservationCategoryQuestions.questID "
sql = sql & "where ObservationCategoryQuestions.observationCatID = " & Me.cboObservationCatID.Value
sql = sql & " order by Questions.questID asc"

rs.Open sql, CurrentProject.Connection, adOpenDynamic, adLockOptimistic
Do Until rs.EOF
questID = rs("questID").Value
questNum = rs("questNum").Value
textBoxControlName = rs("textBoxControlName").Value
comboBoxControlName = rs("comboBoxControlName").Value
'.....

sql2 = "Select * from Answer"
rs2.Open sql2, CurrentProject.Connection, adOpenDynamic, adLockOptimistic
rs2.AddNew
rs2("observationID").Value = observationID
rs2("questID").Value = questID
For Each ctl In Me.Controls
If TypeOf ctl Is ComboBox Then
Set comboBoxAnswer = ctl
If comboBoxAnswer.Name = comboBoxControlName Then
'save to answer table:
rs2("answer").Value = TranslateAnswerValueToDB(comboBoxAnswer.Value)
counter = counter + 1
End If
End If
'.....

If TypeOf ctl Is TextBox Then
'Dim i As Integer
Set textBoxComments = ctl

```

```

If ctl.Name = textBoxControlName Then
    'save comments to the table:
    rs2("comments").Value = textBoxComments.Value
    counter = counter + 1
End If
End If
If counter = 2 Then
    rs2.Update
    rs2.Close
    'this means that comments and answer for a specific questID have been saved to the db.
    'reset counter to 0
    counter = 0
    Exit For 'exit the looping of the controls on the form
End If
Next ctl
'tmp error handling checker
'If counter = 0 Then
    'msgstr = msgstr & "both controls is not found counter =" & questNum & ". Save question procedure
cancelled." & vbCrLf
    'Exit Sub
'End If
'If counter = 1 Then
    'msgstr = msgstr & "one control is not found counter =" & questNum & ". Save question procedure
cancelled." & vbCrLf
    'Exit Sub
'End If
rs.MoveNext
Loop
rs.Close
Set rs = Nothing
Set rs2 = Nothing
MsgBox "Field Observation record has been SAVED successfully with Log ID of " & observationID,
vbInformation
Me.ProcClearForm
Exit Sub
errHand:
MsgBox "Error Number: " & Err.Number & vbCrLf & "Error Description: " & Err.Description
Exit Sub
End Sub

```

Field Observation System DEVELOPMENT - [Field Observation Update Existing Record Form]

Tahoma

File Edit View Insert Format Records Tools Window Help

**SOUTHERN CALIFORNIA EDISON**  
An EDISON INTERNATIONAL Company

**Update Field Observation Record** Main Menu

Search Observations using search criteria:

Category: [ ] Observer: [ ] Type of Crew: [ ]

Observed Location: [ ] From Date: 6/2/2009 To Date: 6/2/2009

Reset Search

Search Results. Double click on a record to view detail:

ID	Category	Observer Name	Job Title	Crew Type	Date	Location Name	Work Type
739	Power Delivery Grid Operation	Director Person	GMCM	Line Crew	6/2/2009	Antelope Valley Service C	OH
738	Power Delivery Grid Operation	Gridman GMCM	GMCM	Patrol Crew	5/31/2009	Antelope Valley Service C	OH
737	Power Delivery Grid Operation	Director Person	GMCM	Patrol Crew	6/1/2009	Alhambra Facility	OH
736	Power Delivery DC&M Field Ob	Jane Doe Doe	SGS	E-Crew	5/20/2009	29 Palms Service Center	OH

Category (Read Only): [ ] Observer: [ ] Date: 6/2/2009 Shift: [ ]

Observed Location: [ ] Time Spent: [ ] Work Type: [ ] Crew Type: [ ]

Questions	NA/OK?	Comments/Corrective Actions
1.01. Discuss purpose and critical steps of job	-	
1.02. Each employee understands what they must do	-	
1.03. Each employee understands what other employees must do	-	
1.04. Each employee understands supervisor's methods for job	-	
1.05. Each employee understands hazards, actions to overcome and PPE needed	-	
1.06. Each employee notified of conditions or clearance on lines and equipment	-	
1.07. Changing conditions recognized and new tailboard conducted	-	
2.01. Appropriate clothing is worn at all times	-	
2.02. Head protection is used when needed	-	
2.03. Gloves are worn when needed (rubber and/or leather)	-	
2.04. Reflective vests are worn when needed	-	
2.05. Eye protection and face shields are worn when needed	-	
2.06. Hearing protection worn when needed	-	
2.07. Respiratory protection worn when needed	-	
2.08. Fall protection gear worn when needed	-	
3.01. Documented control procedure used	-	
3.02. All appropriate equipment is identified for control	-	

DELETE Record | Print Preview Selected Observation Record | Clear Form | Update Field Observation

Figure 2: Partial screen shot of the update field observation record form.

Below is the VBA and SQL code that will automatically read the questions from the questions table and write the question descriptions onto the form in Figure 2 above. The questions are located under the "Questions" header in black font with a dark gray background in Figure 2 above. The code below is executed when the form is first loaded or opened by a user:

```
Private Sub Form_Load()
    'call on procedures to clear the form
    Me.ProcClearForm
    Me.ProcClearSearchSection
    'call on procedure to write questions from the database onto the form
    Me.procLoadQuestions
    Me.ProcLoadLstObservationAll
End Sub
```

'this procedure loads the questions from the questions table and writes the question descriptions onto the form.

```
Public Sub procLoadQuestions()
    Dim sql As String, rs As New ADODB.Recordset, labelControlName As String
    sql = "SELECT Questions.labelControlName, Questions.quest, Questions.questNum "
    sql = sql & " FROM Questions"
    sql = sql & " ORDER BY Questions.questNum;"
```

```
Dim i As Integer, obj As Object, controlId As Integer, lbl As Label
rs.Open sql, CurrentProject.Connection, adOpenDynamic, adLockOptimistic
Do Until rs.EOF
    labelControlName = rs("labelControlName").Value
    For i = 0 To Me.Controls.Count - 1
        Set obj = Me.Controls(i)
        controlId = obj.ControlType
        If controlId = 100 Then '100 is a label control
            Set lbl = obj
            If lbl.Caption = "" Or IsNull(lbl.Caption) = True Then
                If lbl.Name = labelControlName Then
                    lbl.Caption = rs("questNum").Value & ". " & rs("quest").Value
                End If
            End If
        End If
    Next
    rs.MoveNext
Loop
rs.Close
Set rs = Nothing
End Sub
```

Figure 3: Partial screen shot of the field observation reporting form.

The VBA code below is for the report named “Field Observations Per Month” shown in the upper right hand corner of Figure 3 above. This report is dynamic because it depends on the search criteria entered by the user which is group name, from date, and to date. Also, all the form controls for dates utilize date pickers so that user’s don’t have to type in the dates:

This sub procedure will open the report “Field Observations Per Month” when user clicks on the “View Report” button. The code below has to analyze the various search criteria on the form so that the proper dynamic report will be generated

```
Private Sub cmdRptObsPerMonth_Click()
    '10/2/2009 Observation Enhancement ProjectID_10
    Dim fromDate As Date, toDate As Date
    Dim fromMonthNum As Integer, fromYearNum As Integer
    Dim toMonthNum As Integer, toYearNum As Integer
    Dim group As String
    Dim whereQry As String
    whereQry = ""

    If IsNull(Me.cboGroupRptObsPerMonth.Value) = True Then
        MsgBox "Please pick a group", vbInformation, "Required Field"
    End If
End Sub
```

```

Me.cboGroupRptObsPerMonth.SetFocus
Exit Sub
End If
If IsNull(Me.txtFromDateRptObsPerMonth.Value) = False Then
    fromDate = FormatDateTime(Me.txtFromDateRptObsPerMonth.Value, vbShortDate)
Else
    MsgBox "You must pick a date range. Please pick a date from.", vbInformation, "Required Field"
    Me.txtFromDateRptObsPerMonth.SetFocus
    Exit Sub
End If

If IsNull(Me.txtToDateRptObsPerMonth.Value) = False Then
    toDate = FormatDateTime(Me.txtToDateRptObsPerMonth.Value, vbShortDate)
    'date range must be within one year because the field observation goals for each job title
    'is for a one year time frame only.
    isValidDateRange = functions.IsDateDifferenceLessThanEqualToOneYear(fromDate, toDate)
    If isValidDateRange = False Then
        MsgBox "Please pick a 12 month date range or a calendar year.", vbInformation, "Required report criteria"
        Me.txtFromDateRptObsPerMonth.SetFocus
        Exit Sub
    End If
Else
    MsgBox "You must pick a date range. Please pick a date to.", vbInformation, "Required Field"
    Me.txtToDateRptObsPerMonth.SetFocus
    Exit Sub
End If

group = Me.cboGroupRptObsPerMonth.Value

Me.txtHiddenDateRangeFieldObsMonth.Value = Month(fromDate) & "/" & Year(fromDate) & " - " &
Month(toDate) & "/" & Year(toDate)

fromMonthNum = Month(fromDate)
fromYearNum = Year(fromDate)
toMonthNum = Month(toDate)
toYearNum = Year(toDate)
'start building the where conditions:
If whereQry = "" Then
    whereQry = "(obsMonthNumber >= " & fromMonthNum & " and obsYearNumber >= " & fromYearNum &
)" & _
    " and (obsMonthNumber <= " & toMonthNum & " and obsYearNumber <= " & toYearNum & ")"
Else
    whereQry = whereQry & " and (obsMonthNumber >= " & fromMonthNum & " and obsYearNumber >= " &
fromYearNum & ")" & _
    " and (obsMonthNumber <= " & toMonthNum & " and obsYearNumber <= " & toYearNum & ")"
End If

If whereQry = "" Then
    Select Case group
        Case "DC&M"
            DoCmd.OpenReport "RptObsPerMonthDcm", acViewPreview

```

```
Case "SC&M"  
    DoCmd.OpenReport "RptObsPerMonthScm", acViewPreview  
Case "Transmission"  
    DoCmd.OpenReport "RptObsPerMonthTrans", acViewPreview  
Case "Grid Operations"  
    DoCmd.OpenReport "RptObsPerMonthGridOps", acViewPreview  
Exit Sub  
End Select  
Else  
  
Select Case group  
Case "DC&M"  
    DoCmd.OpenReport "RptObsPerMonthDcm", acViewPreview, , whereQry  
Case "SC&M"  
    DoCmd.OpenReport "RptObsPerMonthScm", acViewPreview, , whereQry  
Case "Transmission"  
    DoCmd.OpenReport "RptObsPerMonthTrans", acViewPreview, , whereQry  
Case "Grid Operations"  
    DoCmd.OpenReport "RptObsPerMonthGridOps", acViewPreview, , whereQry  
  
Exit Sub  
End Select  
End If  
End Sub
```

Field Observation System DEVELOPMENT - [Field Observation Employee Management Form]

Tahoma 8 B I U

File Edit View Insert Format Records Tools Window Help

**SOUTHERN CALIFORNIA EDISON**  
An EDISON INTERNATIONAL Company

**Employee Management** [Main Menu](#)

Search Employees First Name [ ] Last Name [ ] Active [ ]

Reset Search

Double click on a record below to view details in the Update Existing Employee Form:

ID	First Name	Last Name	Active	Group	AOR Location	AOR Type	Division	Region	District_	Job Title_
15	Wendy	Johnson	N	GridOps	Whittier	Grid Ops TM	North West Division			GS
13	Kenneth	Elway	Y	DC&M			South East Division	Desert	Foothill	DM
7	SCMMan	M3	Y	SC&M	Management	CFF				M3
6	TransmissionMan	GM	Y	Transmission	Metro West					GM
5	Gridman	GMCM	Y	GridOps	Southern	Grid Ops Operator	South East Division			GMCM
4	Director	Person	Y	GridOps	Northern	Grid Ops Operator	North West Division			GMCM
3	Jane Doe	Doe	Y	DC&M			North West Division	North Coast	Ventura	SGS
2	John	Doe	Y	DC&M			North West Division	North Coast	Thousand Oaks	SGS
1	Brandon	Drake	Y	DC&M			North West Division	North Coast	Thousand Oaks	DM

**Save New Employee Form**

First Name\*\* [ ]

Last Name\*\* [ ]

Is Active\*\* [ Y ]

Group [ ]

Division [ ]

Region [ ]

District [ ]

Job Title\*\* [ ]

Save New Employee

Clear

**Update Existing Employee Form**

First Name\*\* [ ]

Last Name\*\* [ ]

Is Active\*\* [ ]

Group [ ]

Division [ ]

Region [ ]

District [ ]

Job Title\*\* [ ]

Update Existing Employee

Clear

Form View

Figure 4: Full screen shot of the field observation employee management form.

VBA and SQL Code below is used to save a new employee record to the database. So when the user clicks on the button named "Save New Employee" in Figure 4 above the code below will be executed:

```
Private Sub cmdInsertNewEmployeeRecord_Click()
    Me.InsertNewEmployeeRecord
    Me.lstEmployees.Requery
End Sub
```

```
Public Sub InsertNewEmployeeRecord()
On Error GoTo errHand
```

```
'req fields are first name, last name, is active, and job title id
Dim firstName As String, lastName As String, isActive As String, jobTitleID As Long
Dim rs As New ADODB.Recordset, sql As String, res As Integer
```

```
If IsNull(Me.txtFirstNameInsert.Value) = True Then
    MsgBox "Please enter an employee first name.", vbInformation, "Required Field"
    Me.txtFirstNameInsert.SetFocus
```

```
Exit Sub
End If
```

```
If IsNull(Me.txtLastNameInsert.Value) = True Then
    MsgBox "Please enter an employee last name.", vbInformation, "Required Field"
    Me.txtLastNameInsert.SetFocus
    Exit Sub
End If
```

```
If IsNull(Me.cboIsActiveInsert.Value) = True Then
    MsgBox "Please select the active status of the employee.", vbInformation, "Required Field"
    Me.cboIsActiveInsert.SetFocus
    Exit Sub
End If
```

```
If IsNull(Me.cboJobTitleIDInsert.Value) = True Then
    MsgBox "Please select the job title of the employee.", vbInformation, "Required Field"
    Me.cboJobTitleIDInsert.SetFocus
    Exit Sub
End If
```

```
firstName = Me.txtFirstNameInsert.Value
lastName = Me.txtLastNameInsert.Value
isActive = Me.cboIsActiveInsert.Value
jobTitleID = Me.cboJobTitleIDInsert.Value
```

```
Dim dupEmpMsg As String
```

```
dupEmpMsg = functions.IsDuplicateEmployeeRecord(firstName, lastName)
If dupEmpMsg = "" Then
    res = MsgBox("Are you sure you want to SAVE this employee record?", vbYesNoCancel, "Save employee record confirmation")
    If res = vbNo Or res = vbCancel Then
        MsgBox "New employee record will NOT be saved to the database.", vbOKOnly, "New employee record NOT saved"
        'Me.ClearInsertForm
        Exit Sub
    End If
Else
    res = MsgBox(dupeEmpMsg & vbCrLf & vbCrLf & "Do you still want to SAVE this employee record?", vbYesNoCancel, "Save employee record confirmation")
    If res = vbNo Or res = vbCancel Then
        MsgBox "New employee record will NOT be saved to the database.", vbOKOnly, "New employee record NOT saved"
        'Me.ClearInsertForm
        Exit Sub
    End If
End If
```

```
sql = "Select * from Employee"
```

```
rs.Open sql, CurrentProject.Connection, adOpenDynamic, adLockOptimistic
rs.AddNew
    rs("firstName").Value = firstName
    rs("lastName").Value = lastName
    rs("isActive").Value = isActive
    rs("rptJobTitleID").Value = jobTitleID
rs.Update
rs.Close
Set rs = Nothing
```

```
Me.lstEmployees.Requery
Me.ClearInsertForm
'Me.cboSearch_Click
'Me.lstEmployees.Requery
MsgBox "New employee record has been SAVED successfully", vbInformation
Me.lstEmployees.Requery
Exit Sub
```

```
'commented 6/25/2009 Me.lstEmployees.Requery
```

```
errHand:
```

```
    MsgBox "Error Number: " & Err.Number & vbCrLf & "Error Description: " & Err.Description
```

```
    Exit Sub
```

```
End Sub
```

## Field Observation Performance Goals

<i>Group</i>	<i>AOR Location</i>	<i>AOR Type</i>	<i>Division</i>	<i>Region</i>	<i>District</i>	<i>Job Title</i>	<i>Target Observations</i>	<i>Observations Completed</i>	<i>Percent Completed</i>
DC&M	N/A	N/A	North West Division	North Coast	Ventura	SGS	12	1	8.3%
GridOps	Northern	Grid Ops Operator	North West Division	N/A	N/A	GMCM	12	2	16.7%
GridOps	Southern	Grid Ops Operator	South East Division	N/A	N/A	GMCM	12	1	8.3%

Figure 5: Full screen shot of the field observation performance goals report.