How to connect a database and add update delete record

Technology

How to Connect to a Database and Add/Update/Delete Record In this tutorial I will explain to you on how to connect to an Access database and allow you to Add/Update/Delete a record. To fully understand these tutorials please download the source code How to Add/Update/Delete Record using MS Access Database. This source code is part of the Hotel Reservation System that I am currently working. At the end of this tutorial you will learn the basic of database programming. I would like, however, to emphasize especially for beginners that one way to learn programming is to know how to debug a program and devote some of your time to reading.

Don't be frightened on how short or long an article should be. The important is at the end of the tutorial you will learn something NEW! If you already know the topic, then don't bother to study this again. Table of Contents 1. Introduction 2. Let's get started 3. Database Connection 4. Add and Update a Record 5. Delete a Record 6. Final Thoughts Introduction Before I started learning VB. NET one of the topic that I search for in the internet is on how to connect to the database and make some changes to the table. Although there's a lot of results, but I cannot find one that suit to my needs.

Most of the tutorial is using drag and drop features of vb. net editor. Well, this is okay in most cases but what if you'd like to manipulate the data by code? So, I created this tutorial so that beginner programmer will learn from this. Let's get started It is very important that you use your common sense to understand the logic of database programming. There's a lot of features built-in to Visual Basic Editor that most programmer especially beginner who overlook it. One of the favorite tools I usually used is the DEBUGGER. If you

only knew how important a debugger is, then you do not even need to study this tutorial.

Why? Because you can jump right away to the source code and start firing the F8 command from your keyboard and analyze every line as you step through the code. Anyway beginner is a beginner. You need to start from scratch. If you have already downloaded the source code, then open it in the visual basic . net editor by double clicking the " HowtoAddUpdateDeleteRecord. sln". If you want to know what is the object that runs the first time you start the program (by pressing F5) then double click the " My Project" at the Solution Explorer. Look at the Startup Form.

You will see that the value is " frmCustomersList". Now, click this object in the Solution Explorer and click the View Code at the toolbar. Look for the Load event similar below:
Private Sub frmCustomersList_Load(ByVal sender As System.
Object, ByVal e As System. EventArgs)Handles MyBase. Load   sSql = " SELECT CustomerID, CompanyName, ContactName, ContactTitle, Address FROM Customers ORDER BY CustomerID ASC"   Call FillList() FillListView(lvList, GetData(sSql)) End Sub frmCustomersList_Load is the second procedure that runs when you hit the F5 Key from your keyboard.

If you'd like to know how this code is executed then press F8. Believe it or not F8 is the answer to all your programming question. And I really mean it. When I started programming all I do is to search for free source code and start using the debugging tool. That's why Visual Basic is being named as Rapid Application Development or RAD. If you follow the debugger the first line it executes is

the Private Sub frmCustomersList_Resize(ByVal senderAs Object, ByVal e As System. EventArgs) then followed by frmCustomersList_Load which is actually the important procedure to note here.

Another important debugging tool is " Toggle Breakpoint". You will be prompted to your code if one of the line is marked by toggle break point. This can be done by pressing the F9 key or clicking the Debug menu then Toggle Breakpoint. This tool is important if the form is already loaded and you want to tract the execution of a code say within a command button. For example. Open the form frmCustomersList and double click the add button and move the up arrow key once and press F9. You willl have a picture as shown below: [inline: Toggle Breakpoint. jpg]

Now, when you run the program and click the Add button you will be directed to the code editor window. This case you will see what is happening when you are executing the program. Isn't it nice? Database Connection In order to connect to the database you need a connection string like this: Public Const cnString As String = " Provider= Microsoft. Jet. OLEDB. 4. 0; Persist Security Info= False; Data Source=.. /data/sample. mdb" Then open it by using this command: Dim cnHotel As OleDbConnection cnHotel = New OleDbConnection With cnHotel If . State = ConnectionState.

Open Then . Close() . ConnectionString = cnString . Open() End With You need this whether you use OleDbDataReader, ExecuteNonQuery or OleDbCommandBuilder to read or write into the database table. To know more about this class just click this command and press F1 key to open the help files. Be sure you installed the MSDN. Since you have already open the connection to your database this is now the time to fill the ListView with

data. This can be done by calling a function like: FillListView(lvList, GetData(sSql)) The line of code will then execute a function: Fill ListView control   with   data

Public Sub FillListView(ByRef lvList As ListView, ByRef myData As OleDbData Reader)   Dim itmListItem As ListViewItem   Dim strValue As String

Do While myData.   Read   itmListItem   = New ListViewItem()   strValue   = IIf(myData. IsDBNull(0), "", myData. GetValue(0))   itmListItem. Text = strValue For shtCntr = 1 To myData. FieldCount() - 1   If myData. IsDBNull(shtCntr) Then   itmListItem.   SubItems.   Add("") Else itmListItem. SubItems. Add(myData. GetString(shtCntr))   End If

Next shtCntr lvList. Items. Add(itmListItem) Loop End Sub Again in order to see how this code is being executed just run the program using the debugging tool (either F8 or F9). The rest of the procedure is executed only when they are called. For example, the code below is executed only when you click the Add button. Private Sub btnAdd_Click(ByVal sender As System. Object, ByVal e As System.   EventArgs) HandlesbtnAdd.   Click Dim CustomerID As String frmCustomers. State = gModule. FormState. adStateAddMode   For Each sItem As ListViewItem In lvList.

SelectedItems   CustomerID = sItem. Text Next frmCustomers. CustomerID = CustomerID frmCustomers. ShowDialog() Call FillList() End Sub This code will open the form frmCustomers in add mode and will execute also its own Load Event. If you want to open the form frmCustomers in edit mode, then just double click the item in a ListView. The code being executed are:

Private Sub lvList_DoubleClick(ByVal sender As Object, ByVal e As System.

EventArgs) HandleslvList.   DoubleClick   Dim CustomerID As String
For Each sItem As ListViewItem In lvList.

SelectedItems   CustomerID = sItem. Text Next With frmCustomers .
State = gModule. FormState. adStateEditMode   . CustomerID =
CustomerID . ShowDialog() Call FillList() End With frmCustomers = Nothing
End Sub The two procedure seems carry the same concept, by opening a
form, except they vary on the button invoke for execution. The line
frmCustomers. State = gModule. FormState. adStateAddMode will tell the
target form to open the connection to the database in add mode and
frmCustomers. State = gModule. FormState. adStateEditMode ill open the
database in edit mode. Add and Update a Record Now, how to save the data
in textboxes within the form? This can be done by calling a procedure
calledbtnSave_Click. This procedure is fired when the Save button is clicked.
Private Sub btnSave_Click(ByVal sender As System.

Object, ByVal e As System.   EventArgs) HandlesbtnSave.   Click
Dim dt As DataTable   =   dsCustomers.   Tables("   Customers")
If txtCustomerID. Text = "" Or txtCompanyName. Text = "" Then
MsgBox(" Please fill up Customer ID or Company Name information. ",
MsgBoxStyle.

Critical)   Exit Sub End If Try If State = gModule. FormState.
adStateAddMode Then   ' add a row Dim newRow As DataRow
newRow = dt. NewRow() newRow(" CustomerID") = txtCustomerID. Text
dt. Rows. Add(newRow) End If With dt . Rows(0)(" CustomerID") =
txtCustomerID.   Text   .   Rows(0)("   CompanyName")   =
txtCompanyName.   Text   .   Rows(0)("   ContactName")   =

IIf(txtContactName. Text = "", System. DBNull. Value, txtContactName. Text)

. Rows(0)(" ContactTitle") = IIf(txtContactTitle. Text = "", System.

DBNull. Value, txtContactTitle. Text)   . Rows(0)(" Address") =
IIf(txtAddress. Text = "", System. DBNull. Value, txtAddress. Text)   .
Rows(0)(" City") = IIf(txtCity. Text = "", System. DBNull. Value, txtCity. Text)
. Rows(0)(" Region") = IIf(txtRegion. Text = "", System. DBNull. Value,
txtRegion. Text)   . Rows(0)(" PostalCode") = IIf(txtPostalCode. Text
= "", System. DBNull. Value, txtPostalCode. Text)   . Rows(0)("
Country") = IIf(txtCountry. Text = "", System. DBNull. Value, txtCountry.

Text)   . Rows(0)(" Phone") = IIf(txtPhone. Text = "", System.
DBNull. Value, txtPhone. Text)   . Rows(0)(" Fax") = IIf(txtFax. Text
= "", System. DBNull. Value, txtFax. Text)   daCustomers.
Update(dsCustomers, " Customers")   MsgBox(" Record successfully
saved.   ",   MsgBoxStyle.   Information)   End With
Catch ex As OleDbException MsgBox(ex. ToString) End Try End Sub The code
for adding and updating a table is the same except that if you are in add
mode you just simply add this command: If State = gModule.

FormState. adStateAddMode Then ' add a row Dim newRow As DataRow
newRow = dt. NewRow() newRow(" CustomerID") = txtCustomerID. Text dt.
Rows. Add(newRow) End If This way you do not need to create a separate
command to insert and update a table. Delete a Record Let us go back
to frmCustomersList form and delete a record. The procedure before will be
fired   after   clicking   a   Delete   button:
Private Sub btnDelete_Click(ByVal sender As System.

Object, ByVal e As System.   EventArgs) HandlesbtnDelete.   Click

Dim CustomerID As String For Each sItem As ListViewItem In lvList.

SelectedItems   CustomerID   =   sItem.   Text   Next

If CustomerID   ;; "" Then   'Delete   the   selected   record

Dim strDeleted As Boolean strDeleted = ExecNonQuery(" DELETE Customers.

CustomerID FROM Customers WHERE CustomerID= '" & CustomerID & "'")

If strDeleted = " True" Then MsgBox(" Record's deleted. ", MsgBoxStyle.

Information)   Call FillList() Else MsgBox(strDeleted) End If Else

MsgBox(" Please select record to delete. ", MsgBoxStyle. Critical)   End If

End Sub The important line here is the strDeleted = ExecNonQuery(" DELETE

Customers.

CustomerID FROM Customers WHERE CustomerID= '" & CustomerID & "'")

which call the function ExecNonQuery and deletes a record based on the SQL

Statement. Final Thoughts The above tutorial will simply teach you on how to

connect to a database and make some changes to the database table. It is

very important that you read first some tutorials about programming before

you dive into the source code if you're just starting out. If you really wanted

to learn faster, then I recommend a book which is my reference also with this

article. This book is called Beginning VB 2008 Databases: From Novice to

Professional (Beginning: