



TreeView i Access II

Jeg vil i denne anden udgave af treeview føljetonen begynde på et mere brugbart eksempel med et TreeView. Vi laver en ny database med én tabel der indeholder træ strukturen.

Skrevet den **11. feb 2009** af **nih** I kategorien **Databaser / Access** | ★★★★★

Historie

24/8 2004 - Original

26/8 2004 - Eksempel lagt til download

http://home1.stofanet.dk/nih/Eksperten/TreeView_Eksperten.zip

Som i nok kan se af eksempel 1. er strukturen noget låst. Vi kan ikke give direktøren en telefon :), det er først nede i medarbejder tabellen vi kan referere til tabellen med telefon numre.

Derfor vil jeg bygge dette eksempel på een table hvor der kan tilknyttes personer til ALLE noder og der kan tilføjes et vilkårligt antal niveauer

Tabel struktur:

tNode: Nodeid, ParrentNodeld, NodeNavn

***Nodeid er autonummer og nøglefelt

***ParrentNodeld er langtheltal og standartværdi = null (ingenting)

***nodeNavn er alm. tekstfelt

tPerson: PersonId, nodeld, PersStilling, PersNavn

Fidusen er at alle poster i tNode UDEN en parrentId er rodNoder i vores træ. Herefter vil under noder være forbundet via parrentId.

Relationerne

I vinduet til relationer tilføjes alle tabeller dog skal nodetabelen tilføjes en ekstra gang - vi skal lave et selfJoin.

Den vigtige: tNode.nodeld til tNode_1.parentId og MEGET VIGTIG kaskade vis sletning af relaterede poster.

tPerson.nodeld til tNode.nodeld

Det hele med referentiel integritet - det er jo en database :);

Dette er ikke en artikel om normalisering, og det er heller ikke noget jeg er jysk mester i, så jeg vil undgå at kommentere ovenstående - men modtager gerne kommentarer til forbedringer.

Inden vi går videre er det en fordel at indtaste nogle test data i nodetabellen.

formularene

frmMain:

postKilde: tNode

TreeViewControl: navn = ctITree

3 knapper: navn = "btnNySubNode" , "btnNyRootnode" , "btnSletNode"

En tekstboks med kontrolelementkilde: nodeNavn

En subform med tabellen tPerson (fås nemmest ved at trække tabellen ind på formen)

Jeg kan anbefale at ændre treeView controlens udseende vha. dets egenskabsark (properties) som fåes fram ved at dobbeltklikke på controlen.

Så kommer koden:

Åben vba editoren fra frmMain ved trykke på knappen "programkode"

Vi starter med at definerer en global variabel objTree

Dernæst 2 procedure som bruges senere:

```
Option Compare Database
```

```
Option Explicit
```

```
Dim objTree As TreeView
```

```
Sub gaaTil(nodeld As Long)
```

```
    Dim rs As Object
```

```
    Set rs = Me.Recordset.Clone
```

```
    rs.FindFirst "[nodeld] = " & nodeld
```

```
    If Not rs.EOF Then Me.Bookmark = rs.Bookmark
```

```
End Sub
```

```
Sub loadTree(spld)
```

```
    On Error GoTo Ehandler
```

```
    Dim rs As New ADODB.Recordset
```

```
    Dim sql
```

```
    Dim n As Node
```

```
    If IsNull(spld) Then
```

```
        sql = "select * from tNode where isnull(parrentNodeld)"
```

```
    Else
```

```
        sql = "select * from tNode where parrentNodeld=" & spld
```

```
    End If
```

```
    rs.Open sql, CurrentProject.Connection
```

```
    Do While Not rs.EOF
```

```
        With objTree
```

```
            If IsNull(spld) Then
```

```
                Set n = .Nodes.Add(, "id" & rs("nodeld"), rs("nodeNavn"))
```

```

    Else
        Set n = .Nodes.Add("id" & spId, tvwChild, "id" & rs("nodeid"), rs("nodeNavn"))
    End If
End With

Call loadTree(rs("nodeid"))
rs.MoveNext
Loop

If Not rs Is Nothing Then
    rs.Close
End If
Set rs = Nothing
Exit Sub
Ehandler:
If Err.Number = 35602 Then
    'Vi har tegnet noden og springer den over
    Resume Next
Else
    MsgBox Err.Number & Err.Description
End If
End Sub

```

Sub gaaTil(nodeId As Long) vil jeg ikke kommentere - den går til en post på formen bestemt af nodeId.

Sub loadTree(spId) er en rekursiv procedure, dvs. den kalder sig selv - efter at have tegnet en node kalder den sig selv med den nye nodes nodeId og dermed den næste nodes parentId. Errorhandleren fanger fejl hvis noden findes i forvejen så løber den bare videre.

Vi fortsætter med hændelser på formen:

```

Private Sub Form_AfterInsert()
    objTree.Nodes.Clear
    loadTree Null
End Sub

Private Sub Form_Current()
    Dim n As Node
    If Not Me.NewRecord Then
        For Each n In objTree.Nodes
            If CLng(Mid(n.Key, 3)) = Me!nodeId Then
                n.Selected = True
                Exit For
            End If
        Next n
    End If
End Sub

Private Sub Form_Load()

    Set objTree = ctlTree.Object
    Call loadTree(Null)

```

End Sub

Form_AfterInsert() tegner træet efter at der er indsat en post
Form_Current() følger med på træet hvis brugeren navigere rundt i posterne uden at bruge træet
Form_Load() sætter objTree så vi kan arbejde med træet via den og loader træet vha. vores procedure loadTree

Læg mærke til vores kald af den recursive procedure argumentet er 'null', hvilket betyder at den begynder med en rodNode.

Hændelsesprocedure til knapperne:

```
Private Sub btnNyRootnode_Click()  
    Dim nyNode As Node  
    Dim nyNodeid As Long  
    Dim Navn As String  
    'prompt bruger for navn  
    Navn = InputBox("Indtast navn")  
    'indset i tabel  
    DoCmd.SetWarnings False  
    DoCmd.RunSQL "insert into tNode (nodeNavn) " & _  
        "values (" & Navn & ")"  
    DoCmd.SetWarnings True  
    'opdater form  
    Me.Requery  
    'tegn træ  
    nyNodeid = DMax("nodeid", "tNode")  
    Set nyNode = objTree.Nodes.Add(, , "id" & nyNodeid, Navn)  
  
    'Vælg den nye node og find den på formularen  
    nyNode.Selected = True  
    Call ctlTree_NodeClick(nyNode)
```

End Sub

```
Private Sub btnNySubNode_Click()  
    Dim NyParrentid As Long, nyNodeid As Long  
    Dim Navn As String  
    Dim nyNode As Node  
  
    Navn = InputBox("Indtast navn")  
  
    NyParrentid = Mid(objTree.SelectedItem.Key, 3)  
    DoCmd.SetWarnings False  
    DoCmd.RunSQL "insert into tNode (parentNodeid, NodeNavn) " & _  
        "values (" & NyParrentid & ", " & Navn & ")"  
    DoCmd.SetWarnings True  
    objTree.SelectedItem.Expanded = True  
    Me.Requery  
    nyNodeid = DMax("nodeid", "tNode")  
    Set nyNode = objTree.Nodes.Add("id" & NyParrentid, tvwChild, "id" & nyNodeid, Navn)  
  
    'Vælg den nye node og find den på formularen
```

```
nyNode.Selected = True
Call ctlTree_NodeClick(nyNode)
End Sub

Private Sub btnSlet_Click()

DoCmd.RunSQL "delete * from tNode where nodeld = " & _
Mid(objTree.Nodes(objTree.SelectedItem.Index).Key, 3)

objTree.Nodes.Remove ctlTree.SelectedItem.Index

End Sub
```

Og endelig nodeKlik

```
Private Sub ctlTree_NodeClick(ByVal Node As Object)
Dim id As Long
id = Mid(Node.Key, 3)
gaaTil id
End Sub
```

Det var et eksempel på brugen af treeViewet.

Jeg håber det har åbnet for flere muligheder i jeres egne databaser.
Nodetabelen kan jo sammenkædes med flere tabeller så mulighederne er mange.

Eksemplet kan downloades her:

http://home1.stofanet.dk/nih/Eksperten/TreeView_Eksperten.zip

Niels

Kommentar af -anders- d. 26. aug 2004 | 1

Mægtig god artikel, gælder mig til at afprøve det, et eks. til download, ville dog være en rar ting for os "dødlige"

Kommentar af thomasjepsen d. 29. aug 2004 | 2

Vældig godt eksempel, selvom det måske ikke helt kan klassificeres som 'artikel'! Men for min skyld kan dette forum sagtens benyttes til tips og eksempler også.

Dejligt at man kan downloade et færdigt eks. Jeg hader manuelt arbejde :o)

Kommentar af charlotterj d. 22. nov 2004 | 3

super med download af eksemplet