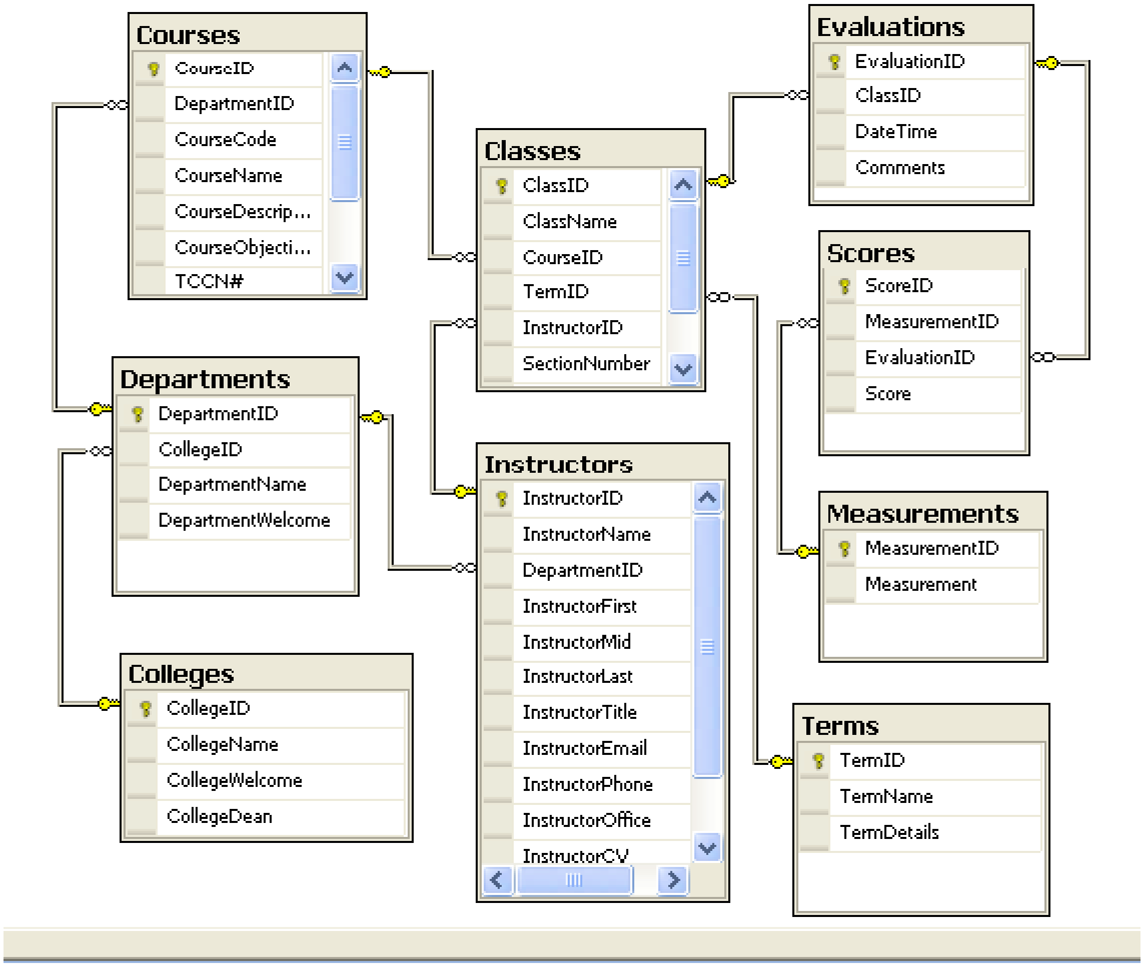
Handling Evaluations

Database design

Instead of hard coding Questions in the Evaluation table, the site uses linking tables.



For every class there are zero or more evaluations.
For every evaluation there are zero or more Score records, up to the number of Measurements available.
The score records are created dynamically based on the result of the evaluation.
The scores are individualized. If a value is not entered for a measurement, there is no record for it.
Needed for correct averages.

Creating the form

The page

```
    <asp:Panel ID="EvalPanel" runat="server">
    </asp:Panel>
    <br />
    <p>Evaluation Comments</p>
    <asp:TextBox ID="txtComments" runat="server" Width="100%" TextMode="MultiLine"
Height="100px"></asp:TextBox>
    <br />
    <br />
    <asp:Button ID="btnEvaluate" runat="server" Text="Submit" />
```

The panel (code behind)

```
    Private Sub ShowEvalForm() ' Shows the evaluation form
        ' Gets the number of measurements to list
        Dim measures As DataTable
        Dim MCount As Integer
        measures = DataAccess.GetMeasurements(MCount)

        ' Create arrays
        Dim myLabel(MCount) As Label ' Measurement
        Dim myRadio(MCount) As RadioButtonList ' Selection
        Dim myItem(8) As ListItem ' Selection items

        ' Create controls for Measurements and RadioButtonLists
        For n As Integer = 0 To MCount - 1
            myLabel(n) = New Label()
            myLabel(n).Text = measures.Rows(n)("Measurement").ToString()
            myRadio(n) = New RadioButtonList()
            myRadio(n).RepeatDirection = RepeatDirection.Horizontal

            ' Create and assign items to each radiobuttonlist
            For m As Integer = 1 To 7
                myItem(m) = New ListItem(m, m)
                myRadio(n).Items.Add(myItem(m))
            Next

            'Customize some buttons
            myItem(1).Text = "1-Low"
            myItem(4).Text = "4-Average"
            myItem(7).Text = "7-High"

            ' Add controls to EvalPanel
            EvalPanel.Controls.Add(myLabel(n))
            EvalPanel.Controls.Add(myRadio(n))
            EvalPanel.Controls.Add(New LiteralControl("<br />"))
        Next

    End Sub
```

Adding evaluations

Stored Procedure

```
CREATE PROCEDURE dbo.AddEvaluationScore(@Measurement INT, @Evaluation INT, @Score
INT)
AS
      INSERT INTO Scores (MeasurementID, EvaluationID, Score) VALUES (@Measurement,
@Evaluation, @Score)
```

Code behind for Click

```
    Protected Sub btnEvaluate_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnEvaluate.Click
        Dim EvaluationID As Integer

        Application.Lock()
        ' Add the Evaluation record and
        ' return EvaluationID in the output variable
        DataAccess.AddClassEvaluation(Request.QueryString("ClassID"), EvaluationID,
txtComments.Text)

        ' Loop through RadioButtonList controls in the panel
        ' Create score records for scores given
        Dim n As Integer = 1
        For Each Control In EvalPanel.Controls
            If
Control.GetType.ToString().Equals("System.Web.UI.WebControls.RadioButtonList") Then
                ' Only store scores for items answered, not as 0s
                If Control.SelectedIndex <> -1 Then
                    DataAccess.AddEvaluationScore(n, EvaluationID,
Control.SelectedIndex + 1)
                    n += 1
                End If
            End If
        Next
        Application.UnLock()
        statusLabel.Text = "Evaluation saved. "

        btnEvaluate.Enabled = False
        lnkViewEval.Text = "View Class Evaluation."
        lnkViewEval.NavigateUrl = Link.ToClass(Request.QueryString("CollegeID"),
Request.QueryString("DepartmentID"), Request.QueryString("CourseID"),
Request.QueryString("ClassID"))

    End Sub
```

Evaluation summary (for class)

Stored Procedure

```
CREATE PROCEDURE [dbo].[GetClassEvaluation](@Class INT, @EvalCount INT OUTPUT)

AS

SELECT Measurement, CAST(RawScore AS DECIMAL(2,1)) AS Average
FROM Measurements,
(SELECT Measurements.MeasurementID, AVG(CAST(Score AS DECIMAL(2,1))) as RawScore
            FROM Measurements, Scores, Evaluations
            WHERE (Measurements.MeasurementID = Scores.MeasurementID)
            AND Scores.EvaluationID = Evaluations.EvaluationID
            AND Evaluations.ClassID = @Class
            GROUP BY Measurements.MeasurementID) AS INSIDE
WHERE Measurements.MeasurementID = INSIDE.MeasurementID

      /* EvalCount represents the number of evaluations. Averages are based on the
number of scores for each Measurement. */
select @EvalCount = Count(EvaluationID) from Evaluations
WHERE ClassID = @Class
```

User Control

```
<asp:GridView ID="EvaluationGrid" runat="server" AutoGenerateColumns="False">
    <Columns>
        <asp:BoundField DataField="Measurement" HeaderText="Measurement">
        <HeaderStyle BackColor="#666666" ForeColor="#CCCCCC" />
        <ItemStyle BorderColor="Silver" CssClass="TableData" HorizontalAlign="Left"
            Width="400px" />
        </asp:BoundField>
        <asp:BoundField DataField="Average" HeaderText="Score">
        <HeaderStyle BackColor="#666666" ForeColor="#CCCCCC" />
        <ItemStyle BorderColor="Silver" CssClass="TableData"
HorizontalAlign="Right" />
        </asp:BoundField>
    </Columns>
</asp:GridView>

<asp:DataList ID="CommentList" runat="server">
    <ItemStyle Font-Italic="True" />
    <HeaderTemplate>
        <p>Evaluation Comments</p>
    </HeaderTemplate>
    <HeaderStyle Font-Bold="True" />
    <ItemTemplate>
        <p>
        <asp:Label ID="Label1" runat="server"
        Text='<%# Eval("Comments").ToString() %>'></asp:Label></p>
    </ItemTemplate>
</asp:DataList>
```

User control code behind

```
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
        Dim classID As String = Request.QueryString("ClassID")
        EvalCount = 0
        EvaluationGrid.DataSource = DataAccess.GetClassEvaluation(classID,
EvalCount)
        EvaluationGrid.DataBind()

        CommentList.DataSource = DataAccess.GetEvalComments(classID)
        CommentList.DataBind()
        If (CommentList.Items.Count) Then
            ' CommentList.HeaderTemplate = "Evaluation Comments"
        End If

    End Sub
```

Page

```
    <uc2:EvaluationSummary ID="EvaluationSummary1" runat="server" />
```

Code behind

```
        If EvaluationSummary1.EvalCount > 0 Then
            lblEvaluationHead.Text = "<br />Evaluation summary for " +
lblCourseCodeHead.Text + " (" + EvaluationSummary1.EvalCount.ToString() + "
evaluations)"
        Else
            lblEvaluationHead.Text = "No evaluations available"

        End If
```