

# SÅDAN VIRKER C.R.U.D.

EN DEMO FRA WI

AF SØREN MAGNUSSON

# Indhold

## Introduktion 3

Hvorfor? 3

Hvad skal man kunne? 3

C.R.U.D. 3

Wireframe 3

Live demo 3

Sourcekode 3

## Overblik 4

Navigation 4

Opret hestesko 4

Ret - på hver hestesko 4

Slet - på hver hestesko 4

Formpanelet 5

## Markup 6

hestesko.aspx 6

## Create 8

hestesko.aspx.cs CodeBehind 8

## Update 1 10

hestesko.aspx.cs CodeBehind 10

## Update 2 12

hestesko.aspx.cs CodeBehind 12

## Delete 14

hestesko.aspx.cs CodeBehind 14

# Introduktion

CRUD er det man laver mest når man laver backend til websider. Man laver det altid, og mange gange i alle projekter. Men hvad er det så?

Dynamiske websider bruger ofte relationsdatabaser til at gemme de oplysninger der skal præsenteres for brugerne, eller som kommer fra brugerne. Disse data skal indtastes af nogen. Vi, webudviklere, gider jo ikke at sidde og skrive alle de herting ind for vores kunder. Det må de selv gøre. Jeg skal videre til et andet projekt!

## Hvorfor?

For at gøre det muligt for vores kunder, selv at vedligeholde oplysningerne på sitet, må vi lave nogen websider, som gør at de kan redigere indholdet af databasen.

Hver gang vi har en tabel (næsten) skal vi have en CRUD til at redigere den. Den simple CRUD gør kun det: redigerer en post ad gangen i en tabel. Mere avancerede CRUDs skal også forholde sig til om tabellen er afhængig af andre tabeller. Men her kigger vi på den simple først.

## Hvad skal man kunne?

Man skal kunne oprette en ny hestesko. Når man har oprettet en hestesko, skal man også kunne se den på siden. En hestesko i vores system, skal kunne redigeres. Endelig skal man også have mulighed for at slette en hestesko igen.

## C.R.U.D.

Det svarer til at vi i databasen udfører kommandoerne INSERT, SELECT, UPDATE og DELETE. Men det kan vores kunde og brugere netop IKKE (eller hvis det var mig selv, så gider jeg måske ikke).

Vi har altså brug for sider hvor man kan:

- indtaste en ny hestesko,
- se alle eksisterende hestesko,
- klikke på en hestesko for at redigere og derefter skrive om i det der allerede er om den hestesko og
- klikke på en hestesko for slette.

Man kan også lave det hele på en side, som jeg har valg i denne demo.

## Wireframe

Skematisk kan det se sådan her ud;  
[illustration fra omnigrapher]

## Live demo

De eksempler du ser i denne tekst kan du afprøve live på <http://asp-net-snippets.azurewebsites.net/CRUD/>.

## Sourcekode

Hvis du er interesseret i kildekoden, og det er man jo altid, så kan du finde den på GitHub: <https://github.com/Rts-wix/ASP-NET-Snippets/tree/master/CRUD>.  
Opsætning af databasen...

oprette = Create,  
vise eller indlæse = Read,  
redigere = Update og  
slet = Delete.

# Overblik

## Navigation

I venstre side sidder et panel som primært bruges til navigation.

Her har jeg placeret en repeater der udskriver alle poster i hestesko-tabellen. Der er ingen begrænsning, så alle poster vises. Dette er den første implementering af read. Om lidt skal vi se en til.

På hver linie, for hver post, hver hestesko, har jeg placeret to knapper; "Ret" og "Slet". Øverst i panelet, over repeateren har jeg lavet en knap til at "Opret(te) en ny hestesko".

Det er kun "Slet"-knappen der gør noget direkte. Og der kommer endda en javascript-confirm-box først. De andre knapper navigerer til side hvor man kan oprette og rette hestesko.

## Opret hestesko

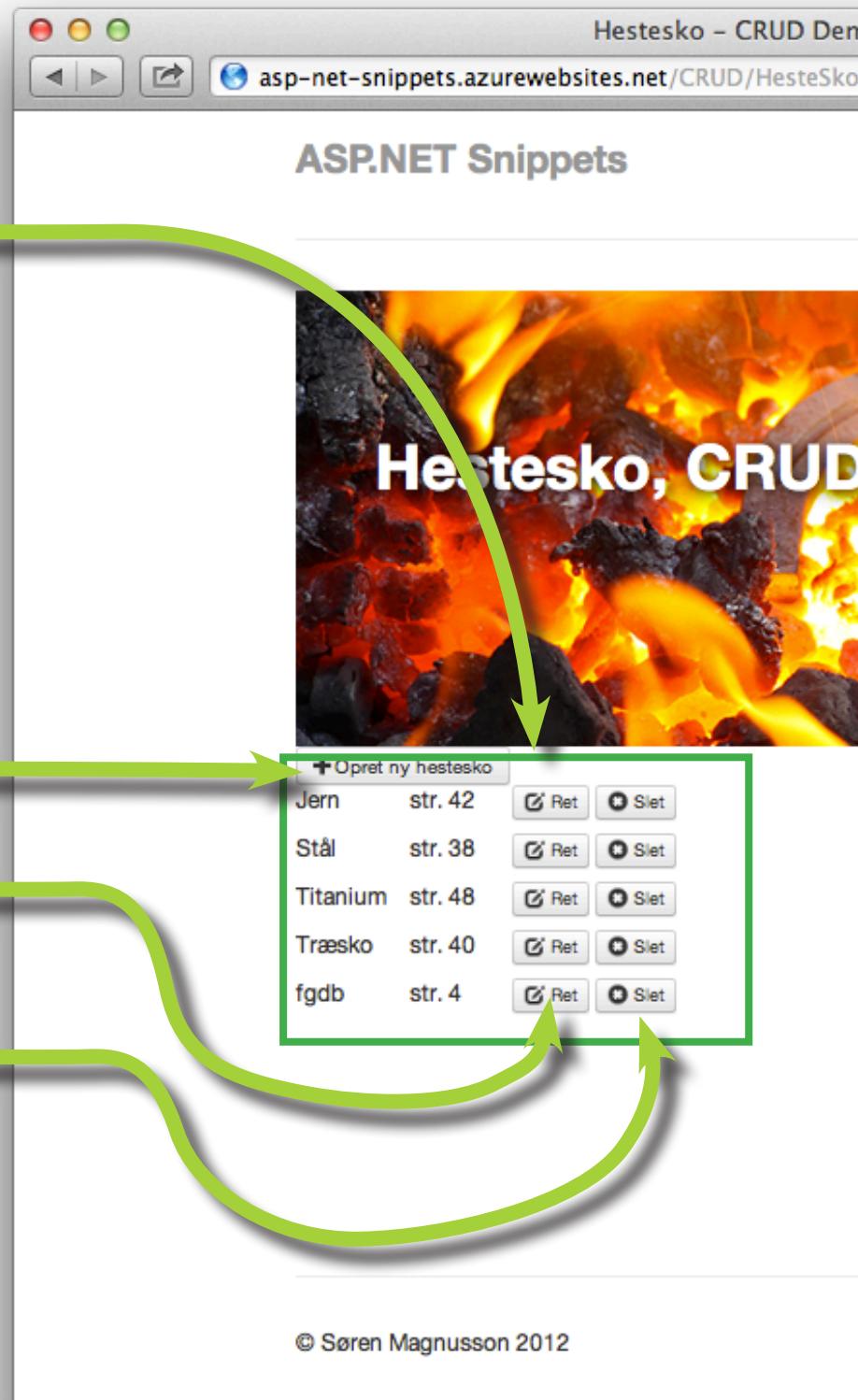
Åbner panelet til højre, med en tom formular og knappen "Opret" synlig.

## Ret - på hver hestesko

Åbner også panelet til højre, men indlæser den hestesko Ret-knappen sidder på, i formularen. I denne udgave vises knappen Gem.

## Slet - på hver hestesko

Sletter hesteskoen direkte i databasen (dog først efter at brugeren har accepteret i en confirm-box).



The screenshot shows a web browser window with the title "no - Asp.Net Snippets" and the URL ".aspx". The browser interface includes a back/forward button, a search bar labeled "Læser", and a download button. Below the title, there are two blue buttons: "Hjem" and "Kontakt". The main content area features a large image of a fire with glowing embers and a metal ring. Overlaid on the image is the word "demo". Below the image is a form with a green border. The form fields are:

- navn:
- størrelse:
- Beskrivelse:
- Gem

Green arrows point from the text "Formpanelet" to the green border of the form and to the "Gem" button.

### Formpanelet

I højre side er der et panel med en form i (med tre inputfelter og to knapper).

Her kan man udfylde de oplysninger som en hesteskø består af. I hvert fald som den ser ud på dette site, og i sitets database.

Panelset er default skjult (visibility = false), men når brugeren trykker på enten "Opret ny hesteskø" eller "Ret" gøres panelet synligt (visibility = true).

Under inputfelterne er der to knapper. De er også skjult, pr. default, men aktiveres en ad gangen når det enten er opret- eller gem-knappen der skal vises. Ud over at teksten er forskellig, har de to knapper også forskellige eventhandlere på onClick. Derfor udfører de også forskellige ting på databasen.  
Her viser jeg formen efter at jeg har trykket på "Ret" ud for nummer to hesteskø: "Stål".

# Markup

## hestesko.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile "~/sample.master" AutoEventWireup="true"
CodeFile="HesteSko.aspx.cs" Inherits="CRUD_Hestesko" %>

<asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolderContent" Runat="Server">
<form id="form1" runat="server">
<div id="top">
<h1>Hestesko, CRUD demo</h1>
</div>

<div class="row-fluid">
<div class="leftbox span5">
<asp:LinkButton ID="ButtonNy" runat="server" class="btn btn-small"
onclick="ButtonNy_Click" ><i class="icon-plus"></i>Opret ny hestesko</asp:LinkButton><br />

<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%$ ConnectionStrings:ConnectionString %>"
ProviderName="<%$ ConnectionStrings:ConnectionString.ProviderName %>"
SelectCommand="SELECT * FROM [ASPSNIPPETS_CRUD_Hestesko]">
</asp:SqlDataSource>

<asp:Repeater ID="Repeater1" runat="server" DataSourceID="SqlDataSource1"
onItemCommand="Repeater1_ItemCommand">
<ItemTemplate>
<div class="controls controls-row">
<div class="span3"><%# Eval("navn") %></div>
<div class="span3">str. <%# Eval("størrelse") %></div>
<asp:LinkButton ID="ButtonRet" runat="server" class="btn btn-mini"
CommandName="Ret" CommandArgument='<%# Eval("id") %>'>
<i class="icon-edit"></i> Ret</asp:LinkButton>
<asp:LinkButton ID="ButtonSlet" runat="server" class="btn btn-mini"
OnClientClick='<%# Eval("navn"), "return confirm(\\"Er du sikker på at du vil slette {0}\\");\\" %>'>
CommandName="Slet" CommandArgument='<%# Eval("id") %>'>
<i class="icon-remove-sign"></i> Slet</asp:LinkButton>
</div>
</ItemTemplate>
</asp:Repeater>
</div>
```

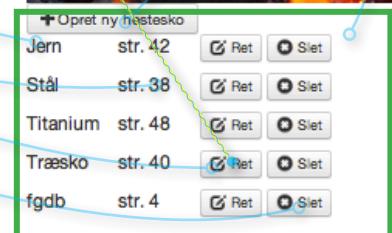
For at oprette en ny hestesko, skal  
brugeren først trykke på "opret ny  
hestesko".  
Så bliver formen synlig.

Et almindligt SqlDataSource og  
en repeater genererer en liste af de  
hestesko der er i databasen.

Repeateren har en eventhandler "onItem-  
Command". Den udføres når brugeren klicker  
på noget inden i repeateren.  
CommandName og CommandArgument på  
knapperne styrer hvad der skal ske og hvil-  
ken hestesko der skal pilles ved.

På slet-knappen er der en OnClient-  
Click eventhandler. Den udfører en  
JavaScript kommando i browseren, iden  
og kun hvis scriptet ikke  
returnerer false.

Her bruger vi den til at vise en con-  
firm-box, så man kan nå at fortryde.  
Bemærk at Eval kan også tage to para-  
metre, navnet på feltet, som man plejer,  
og en tekst hvor værdien skal sættes ind i.



Her på siden har jeg fjernet valideringen af indtastningerne i TextBox'ene fordi der mangler plads og fordi det ikke er det der er i fokus.

```
<div class="rightbox span7">
<asp:Panel ID="PanelForm" runat="server" Visible="False">
<div class="form-horizontal">
<div class="control-group">
<label class="control-label" for="TextBoxNavn">
    navn:
</label>
<div class="controls">
    <asp:TextBox ID="TextBoxNavn" runat="server"></asp:TextBox>
</div>
</div>
<div class="control-group">
<label class="control-label" for="TextBoxStørrelse">
    størrelse:
</label>
<div class="controls">
    <asp:TextBox ID="TextBoxStørrelse" runat="server"></asp:TextBox>
</div>
</div>
<div class="control-group">
<label class="control-label" for="TextBoxBeskrivelse">
    Beskrivelse:
</label>
<div class="controls">
    <asp:TextBox ID="TextBoxBeskrivelse" runat="server" Height="75px" Width="300px"></asp:TextBox>
</div>
</div>
<div class="control-group">
<div class="controls">
    <asp:Button ID="ButtonOpret" runat="server" Text="Opret" Visible="False"
        OnClick="ButtonOpret_Click" />
    <asp:Button ID="ButtonGem" runat="server" Text="Gem" Visible="False" OnClick="ButtonGem_Click" />
</div>
</div>
</div>
</asp:Panel>
</div>
</div>
</form>
</asp:Content>
```

Alle de finurlige class navne, rightbox, span7, form-horizontal, control-group, controls, btn, btn-mini, icon-edit m.fl. stammer fra Twitters bootstrap.

Formen kan skjules eller vises ved at sætte PanelForm.Visible til false eller true.

Tre TextBox' e til at skrive alt om hesteskoen i; TextBoxNavn, TextBoxStørrelse og TextBoxBeskrivelse. Fordi de er dynamiske asp controls kan vi også sætte deres værdi fra programmet, når men skal redigere en hestesco.

Jeg bruger den samme form både til at oprette og til at redigere hesteskø. Men jeg vil gerne have to forskellige knapper til Opret og Gem. Derfor har jeg bare lavet to knapper, som begge er skjult (Visibility = false) som default. Ligesom hele formen er. Når der er brug for det, kan jeg senere vise og skjule dem, som det passer mig.

navn: Stål

størrelse: 38

Beskrivelse:

# Create

## hestesko.aspx.cs CodeBehind

```
protected void ButtonNy_Click(object sender, EventArgs e)
{
    // sørge for at textboxe er tomme
    TextBoxNavn.Text = "";
    TextBoxStørrelse.Text = "";
    TextBoxBeskrivelse.Text = "";

    // vis panel og knap, skjul den anden knap
    PanelForm.Visible = true;
    ButtonOpret.Visible = true;
    ButtonGem.Visible = false;

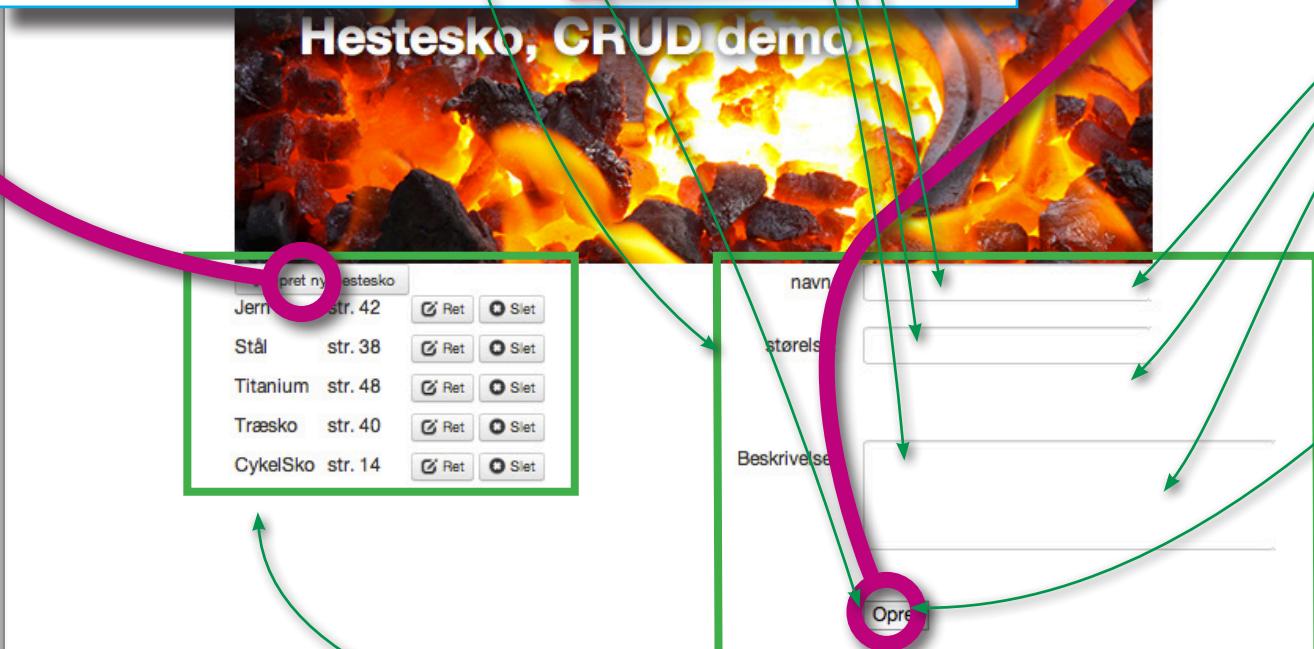
    // hvis brugeren trykker <Enter>
    // skal det have virkning som knappen
    form1.DefaultButton = "ButtonOpret";
}
```

Inden vi kan oprette hestesko skal formen vises.

Når vi skal lave en ny hestesko, skal TextBoxene være tomme.

PanelForm og ButtonOpret skal være synlige.

Når formen er synlig vil jeg gerne have at knappen i formen er default for når brugeren taster <Enter>. Her er det ButtonOpret.



© Søren Magnusson 2012

**ASPSNIPPETS\_CRUD\_Hestesko**

Column Name	Data Type	Allow Nulls
id	int	<input type="checkbox"/>
navn	varchar(50)	<input checked="" type="checkbox"/>
størrelse	int	<input checked="" type="checkbox"/>
beskrivelse	text	<input checked="" type="checkbox"/>

```
// C i CRUD
protected void ButtonOpret_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection();
    conn.ConnectionString =
        ConfigurationManager.ConnectionStrings["ConnectionString"].ToString();
    SqlCommand cmd = new SqlCommand();
    cmd.Connection = conn;

    cmd.CommandText =
        "INSERT INTO ASPSNIPPETS_CRUD_Hestesko "
        +"(navn, størelse, beskrivelse) VALUES (@navn, @størelse, @beskrivelse)";

    cmd.Parameters.Add("@navn", SqlDbType.VarChar).Value = TextBoxNavn.Text;
    cmd.Parameters.Add("@størelse", SqlDbType.VarChar).Value = TextBoxStørelse.Text;
    cmd.Parameters.Add("@beskrivelse", SqlDbType.VarChar).Value = TextBoxBeskrivelse.Text;

    conn.Open();
    cmd.ExecuteNonQuery();
    conn.Close();

    Repeater1.DataBind();
}

PanelForm.Visible = false;
ButtonOpret.Visible = false;
ButtonGem.Visible = false;

form1.DefaultButton = "ButtonNy";
}

```

Opret SqlConnection, SqlCommand osv

Selve SQL kommandoen INSERT. Læg mærke til @'erne i VALUES parentesen. Her erklærer vi at værdierne skal ind som parametre.

Parameter objekterne får værdi fra TextBoxene.

Åben forbindelsen, udfør SQL kommandoen og luk. Jeg bruger ExecuteNonQuery() fordi vi ikke skal have nogen data tilbage fra database. Kun indsætte.

Efter jeg har indsæt i databasen, vil jeg gerne have at man kan se det i listen til venstre, derfor beder jeg Repeater1 om at indlæse data igen.

Skjul formen og begge knapper.

Nu da formen er skjult, skal ButtonNy være default igen.

# Update 1

hestesko.aspx.cs CodeBehind

The screenshot shows a web application titled "Hestesko - CRUD Demo - Asp.Net Snippets". The main content area features a large image of a fire with the text "Hestesko, CRUD demo" overlaid. Below the image is a table listing five items:

	navn	størrelse	
1	Jern	str. 42	<input type="button" value="Ret"/> <input type="button" value="Slet"/>
2	Stål	str. 38	<input checked="" type="button" value="Ret"/> <input type="button" value="Slet"/>
3	Titanium	str. 48	<input type="button" value="Ret"/> <input type="button" value="Slet"/>
4	Træsko	str. 40	<input type="button" value="Ret"/> <input type="button" value="Slet"/>
5	fgdb	str. 4	<input type="button" value="Ret"/> <input type="button" value="Slet"/>

A green box highlights the row for "Stål". A pink circle with a black outline highlights the "Ret" button in this row. A green box surrounds the "Edit" form on the right, which contains fields for "navn" (navn: Stål), "størrelse" (størrelse: 38), and "Beskrivelse" (Beskrivelse: empty text area). A green arrow points from the "Ret" button to the "navn" field. Another green arrow points from the "Ret" button to the "størrelse" field. A third green arrow points from the "Ret" button to the "Beskrivelse" text area. A green arrow points from the "Beskrivelse" text area to the "Gem" button.

© Søren Magnusson 2012

ASPSNIPPETS_CRUD_Hestesko			
Column Name	Data Type	Allow Nulls	
id	int	<input type="checkbox"/>	
navn	varchar(50)	<input checked="" type="checkbox"/>	
størrelse	int	<input checked="" type="checkbox"/>	
beskrivelse	text	<input checked="" type="checkbox"/>	

Redigering af en hestesko er delt i to trin.

1. Hent alle oplysninger om hesteskoen og indlæs dem i TextBoxene på siden, så brugeren kan rykke rundt i teksten og tilføje, slette og adskifte det de synes.

2. Når brugeren er tilfreds, trykker hun/han på "Gem". Så skal alle data skrives ind i databasen med en UPDATE SQL-kommando.

Knapperne inden i repeateren kan ikke have deres egne eventhåndlere. Derfor håndteres alle klik via OnItemCommand-eventet. Det går til Repeater1\_ItemCommand() metoden. På knapperne har vi sat værdier for CommandName og CommandArgument. CommandName indeholder hvad der skal gøres. CommandArgument indeholder hvilken post der skal gøres noget ved. Her checker vi Repeater-CommandEventArgs'et i variablen e, for CommandName: e.CommandName.

```
protected void Repeater1_ItemCommand(object source, RepeaterCommandEventArgs e)
{
    // R i CRUD
    if (e.CommandName == "Ret")
    {
        SqlConnection conn = new SqlConnection();
        conn.ConnectionString =
            ConfigurationManager.ConnectionStrings["ConnectionString"].ToString();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conn;
        cmd.CommandText = "SELECT * FROM ASPSNIPPETS_CRUD_Hestesko WHERE id = @id";
        cmd.Parameters.Add("@id", SqlDbType.Int).Value = e.CommandArgument.ToString();
        conn.Open();
        SqlDataReader reader = cmd.ExecuteReader();
        if (reader.Read())
        {
            TextBoxNavn.Text = reader["navn"].ToString();
            TextBoxStørelse.Text = reader["størrelse"].ToString();
            TextBoxBeskrivelse.Text = reader["beskrivelse"].ToString();
        }
        conn.Close();
        ViewState["HesteskoId"] = e.CommandArgument.ToString();
        PanelForm.Visible = true;
        ButtonOpret.Visible = false;
        ButtonGem.Visible = true;
        form1.DefaultButton = "ButtonGem";
    }
}
```

Her er e.CommandName lig med "Ret" fordi brugeren trykkede på en af ret knapperne.

Opret SqlConnection, SqlCommand osv

Id'et til hesteskoen kommer ind igennem e.CommandArgument.

Jeg udfører SQL kommandoen, og får resultatet tilbage i en SqlDataReader.

Hvis der er nogen svar fra databasen, så er værdierne klar i reader'en, som også fungerer som collection-liste. Så kan værdierne sættes ind i TextBox'ene.

Nu skal ButtonGem være synlig og default.

Jeg gemmer lige hesteskokens id i ViewState, så kan jeg bruge den i næste trin, hvor jeg opdaterer de nye værdier i hesteskoen. ViewState virker lidt ligesom Session, men hvor Session er beregnet til at virke over hele sitet, på tværs af alle sider, så virker ViewState på en enkelt side, men imellem flere postback's

Metoden Repeater1\_ItemCommand fortsættes på side 14.

# Update 2

hestesko.aspx.cs CodeBehind

The screenshot shows a web application titled "Hestesko - CRUD Demo - Asp.Net Snippets". The main content area features a large image of a fire with the text "Hestesko, CRUD dæmo" overlaid. Below the image is a table listing five items:

	Upret ny hestesko		
Jern	str. 42	<input type="checkbox"/> Ret	<input checked="" type="radio"/> Slet
Stål	str. 38	<input checked="" type="checkbox"/> Ret	<input type="radio"/> Slet
Titanium	str. 48	<input checked="" type="checkbox"/> Ret	<input type="radio"/> Slet
Træsko	str. 40	<input checked="" type="checkbox"/> Ret	<input type="radio"/> Slet
fgdb	str. 4	<input checked="" type="checkbox"/> Ret	<input type="radio"/> Slet

To the right, a detailed view for "Stål" is shown in a modal dialog. The dialog contains fields for "navn" (Stål), "størrelse" (38), and "Beskrivelse" (empty). A green box highlights the "navn" field, and a green arrow points from it to the "navn" field in the table above. A pink circle highlights the "Gem" button at the bottom of the dialog.

© Søren Magnusson 2012

ASPSNIPPETS_CRUD_Hestesko			
Column Name	Data Type	Allow Nulls	
<input checked="" type="checkbox"/> id	int	<input type="checkbox"/>	
navn	varchar(50)	<input checked="" type="checkbox"/>	
størrelse	int	<input checked="" type="checkbox"/>	
beskrivelse	text	<input checked="" type="checkbox"/>	<input type="checkbox"/>

```
// U i CRUD .
protected void ButtonGem_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection(); Opret SqlConnection, SqlCommand osv
    conn.ConnectionString =
        ConfigurationManager.ConnectionStrings["ConnectionString"].ToString();
    SqlCommand cmd = new SqlCommand();
    cmd.Connection = conn;

    cmd.CommandText = "UPDATE ASPSNIPPETS_CRUD_Hestesko"
        + " SET navn = @navn, størrelse = @størrelse, beskrivelse = @beskrivelse"
        + " WHERE id = @id";
    Id'et har vi jo gemt i ViewState, se side 11.

    cmd.Parameters.Add("@id", SqlDbType.Int).Value = ViewState["HesteskoId"].ToString();

    cmd.Parameters.Add("@navn", SqlDbType.VarChar).Value = TextBoxNavn.Text;
    cmd.Parameters.Add("@størrelse", SqlDbType.VarChar).Value = TextBoxStørrelse.Text;
    cmd.Parameters.Add("@beskrivelse", SqlDbType.VarChar).Value = TextBoxBeskrivelse.Text;

    conn.Open();
    cmd.ExecuteNonQuery(); Åben, udfør, luk.
    conn.Close();

    Repeater1.DataBind(); Sørg for at Repeater1 henter de sidste nye data, så ændringerne vises i listen.

    PanelForm.Visible = false; Skjul PanelForm mv.
    ButtonOpret.Visible = false;
    ButtonGem.Visible = false;
    form1.DefaultButton = "ButtonNy";
}

} Når nu formen er skjult, skal knappen ButtonNy = "Opret ny hestesko" være default ved tryk på enter.
```

# Delete

hestesko.aspx.cs CodeBehind

The screenshot shows a web application titled "Hestesko - CRUD Demo - Asp.Net Snippets". The main page displays a banner with the text "Hestesko, CRUD demo" over a background of burning coals. Below the banner is a table listing five entries:

	Udskriv	Retur	Slet
Jern	str. 42	<input type="checkbox"/>	<input checked="" type="checkbox"/> Slet
Stål	str. 38	<input type="checkbox"/>	<input checked="" type="checkbox"/> Slet
Titanium	str. 48	<input type="checkbox"/>	<input checked="" type="checkbox"/> Slet
Træsko	str. 40	<input type="checkbox"/>	<input checked="" type="checkbox"/> Slet
fgdb	str. 4	<input type="checkbox"/>	<input checked="" type="checkbox"/> Slet

A green box highlights the "Slet" (Delete) button for the "Stål" entry. A pink arrow points from this button to a larger modal dialog on the right. The dialog contains fields for "navn" (name) set to "Stål", "størrelse" (size) set to "38", and a "Beskrivelse" (description) text area. A "Gem" (Save) button is at the bottom. A green arrow points from the "Gem" button back to the "Slet" button in the list.

© Søren Magnusson 2012

ASPSNIPPETS_CRUD_Hestesko			
Column Name	Data Type	Allow Nulls	
<input checked="" type="checkbox"/> id	int	<input type="checkbox"/>	
navn	varchar(50)	<input checked="" type="checkbox"/>	
størrelse	int	<input checked="" type="checkbox"/>	
beskrivelse	text	<input checked="" type="checkbox"/>	

Her fortsættes metoden Repeater1\_ItemCommand fra side 11.

```
// D i CRUD
else if (e.CommandName == "Slet")
{
    SqlConnection conn = new SqlConnection();
    conn.ConnectionString =
        ConfigurationManager.ConnectionStrings["ConnectionString"].ToString();
    SqlCommand cmd = new SqlCommand();
    cmd.Connection = conn;
    cmd.CommandText = "DELETE FROM ASPSNIPPETS_CRUD_Hestesko WHERE id = @id";
    cmd.Parameters.AddWithValue("@id", SqlDbType.Int).Value = e.CommandArgument.ToString();
    conn.Open();
    cmd.ExecuteNonQuery();
    conn.Close();
}

Repeater1.DataBind();
```

Her er e.CommandName lig med "Ret" fordi  
brugeren trykkede på en af ret knapperne.

Opret SqlConnection, SqlCommand osv

SQL Kommandoen DELETE sletter ekstirende poster.  
Husk "WHERE id = ..." ellers slettes ALLE!

Id'et til hesteskoen kommer ind igennem e.CommandArgument.

Åben, udfør, luk.

Sørg for at Repeater1 henter de sidste nye data, så ændringerne vises i listen.

Skjul PanelForm mv.

Når nu formen er skjult, skal knappen ButtonNy = "Opret ny hestesko"  
være default ved tryk på enter.

*Her er uddrag fra Collins English Dictionary, om hvad crud betyder...*

## **crud [krʌd] Slang**

(n)

1. a sticky substance, esp when dirty and encrusted
2. (Engineering / General Engineering) an undesirable residue from a process, esp one inside a nuclear reactor
3. something or someone that is worthless, disgusting, or contemptible
4. (sometimes preceded by the) a disease; rot

### **interj**

an expression of disgust, disappointment, etc.  
[earlier form of curd]

*Collins English Dictionary – Complete and Unabridged*  
© HarperCollins Publishers 1991, 1994, 1998, 2000, 2003