# ExecuteScalar aneb jak mit v params celou query a dat ji jenom parametry

**ExecuteScalar neco vraci. Pokud ji dam gen. argument<int> pak vraci int:**

**Volani, nejprve query, pak jeji parametry:**

        public int ResetAndFillMessagesForExport(string query, DateTime from, DateTime to, int foreignLanguageAdsMotiveId, int productPlacement)

        {

            using (MediaDataExportDataModel model = CreateDbContext())

            {

                model.Command.CommandTimeout = 300;

                string command = string.Format(query, from.ToString("yyyMMdd"), to.ToString("yyyMMdd"), productPlacement, foreignLanguageAdsMotiveId);

                return model.SetCommand(command).ExecuteScalar<int>();

            }

**Query je natvrdo v params:**

SELECT \* FROM [MediaData3BGAuto].[dbo].[Params] WHERE Name ='NielsenExportQuery'

**Zneni: V {} jsou jednotlive parametry:**

truncate table Export.ACN\_MediaMessageForExport;

insert into Export.ACN\_MediaMessageForExport

select mm.Id

FROM Media.MediaMessage mm

JOIN Creative.Creative norm ON mm.NormCreativeId = norm.Id

join Media.MotiveVersion mv on norm.MotiveId = mv.MotiveId and mm.[AdvertisedFrom] >= mv.ActiveFrom and mm.[AdvertisedFrom] < mv.[ActiveTo]

JOIN Media.Motivlet mt ON mt.Id = mv.PrimaryMotivletId

JOIN Media.Category cat ON cat.Id = mt.CategoryId

join Media.[Owner] [o] on [o].[Id] = [mt].[OwnerId]

WHERE mm.AdvertisedFrom >= '{0}' --@from

AND mm.AdvertisedFrom < '{1}' --@to --prvni den daneho mesice

AND ISNULL(AdvertisementTypeId,255) != {2} --@productPlacement

AND norm.MotiveId != {3} --@foreignLanguageAdsMotiveId

AND (mm.MediaTypeId<>32

OR cat.Code = 'C.11.0.7.4')

OPTION (FORCE ORDER)

SELECT COUNT(MediaMessageId) FROM Export.ACN\_MediaMessageForExport

# Jak se pripojit na Stoupu pomoci AdoNet

(TeleridExportDemo.sln )

<Window

x:Class="TelerikExportDemo.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:TelerikExportDemo"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:telerik="http://schemas.telerik.com/2008/xaml/presentation"

Title="MainWindow"

Width="525"

Height="350"

mc:Ignorable="d">

<Grid>

<telerik:RadGridView

x:Name="RadGridView"

ItemsSource="{Binding Path=Items}"

SelectionMode="Extended"

SelectionUnit="FullRow"

GroupRenderMode="Flat"/>

</Grid>

</Window>

using System.Collections.ObjectModel;

using System.Configuration;

using System.Data.SqlClient;

using System.Windows;

namespace TelerikExportDemo

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

using (var conn = new SqlConnection())

{

var connSettings = ConfigurationManager.ConnectionStrings["Stoupa"];

if (connSettings?.ConnectionString != null)

{

conn.ConnectionString = ConfigurationManager.ConnectionStrings["Stoupa"].ConnectionString;

}

conn.Open();

var command = new SqlCommand(@"select top(3000)

ver.MotiveId motiveId,

ver.Id versionId,

ver.ActiveFrom ActiveFrom,

ver.ActiveTo ActiveTo,

let.Id letId,

own.Name Owner,

com.Name CompanyBrand,

prob.Name ProductBrand,

prod.Name ProductDetail,

cat.Name Category,

rol.Name Role,

mar.Name Market,

gen.Name Gender,

pla.Name Platform

from media.Motivlet let

join media.MotiveVersionToMotivlet mtm on let.Id = mtm.MotivletId

join media.MotiveVersion ver on mtm.MotiveVersionId = ver.Id

join media.Owner own on let.OwnerId = own.Id

join media.CompanyBrand com on let.CompanyBrandId = com.Id

join media.ProductBrand prob on let.ProductBrandId = prob.Id

join media.ProductDetail prod on let.ProductDetailId = prod.Id

join media.Category cat on let.CategoryId = cat.Id

join media.Telco tel on let.TelcoId = tel.Id

join media.Market mar on let.MarketId = mar.Id

join media.Gender gen on let.GenderId = gen.Id

join media.Platform pla on let.PlatformId = pla.Id

join media.Role rol on let.RoleId = rol.Id") {Connection = conn};

using (var reader = command.ExecuteReader())

{

while (reader.Read())

{

var motivlet = new Motivlet((int) reader[0], (int) reader[1], (int) reader[2], (short?) reader[3]);

Items.Add(motivlet);

}

}

}

RadGridView.DataContext = this;

RadGridView.ItemsSource = Items;

}

private ObservableCollection<Motivlet> Items { get; } = new ObservableCollection<Motivlet>();

private class Motivlet

{

public Motivlet(int ownerId, int productBrandId, int companyBrandId, short? categoryId)

{

OwnerId = ownerId;

ProductBrandId = productBrandId;

CompanyBrandId = companyBrandId;

CategoryId = categoryId;

}

public int MotiveId { get; set; }

public int OwnerId { get; }

public int? ProductBrandId { get; set; }

public int? CompanyBrandId { get; set; }

public short? CategoryId { get; set; }

public byte? TelcoId { get; set; }

public byte? MarketId { get; set; }

public byte? GenderId { get; set; }

public byte? PlatformId { get; set; }

public byte? RoleId { get; set; }

public short? ProductDetailId { get; set; }

}

}

}