Aplicação ASP.NET MVC 4 Usando Banco de Dados

Neste exemplo simples, vamos desenvolver uma aplicação **ASP.NET MVC** para acessar o banco de dados **Northwind**, que está armazenado no servidor SQL Server e, listar todas as categorias.

Para tal, temos que executar as seguintes tarefas:

1. Verificar se o banco de dados Northwind existe no servidor SQL Server.
2. Caso não exista, temos que cria-lo. Para isso, execute o script que está no meu site.
3. Verifique os campos da tabela Categorie.
4. Crie os Procedimentos no banco de dados Northwind

Stored procedures no Northwind

```
USE NORTHWIND
GO

CREATE PROCEDURE [dbo].[stpInserirCategory]
@CategoryName varchar(15),
@Description Text
AS
Insert Into Categories (CategoryName, Description)
     Values (@CategoryName, @Description)

CREATE PROCEDURE stpEditarCategory
@CategoryID int,
@CategoryName varchar(15),
@Description Text
AS
UPDATE Categories Set CategoryName = @CategoryName,
                Description  = @Description
            WHERE CategoryID = @CategoryID
GO


CREATE PROCEDURE stpDeletarCategory
@CategoryID int
AS
DELETE FROM Categories WHERE CategoryID = @CategoryID
GO
```

5. Crie uma aplicação ASP.NET MVC 4.
6. Crie o modelona pasta → Models, de acordo com o código a seguir:

Modelo em C#

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Web;
using System.Web.Configuration;

namespace MvcComDB.Models
{
```

```
public class CategoryRepository
{
    //Criar uma lista de categorias
    public List<Category> GetAllCategories()
    {
        ConnectionStringSettings getString = WebConfigurationManager.ConnectionStrings["nwind"] as
ConnectionStringSettings;
        if (getString != null)
        {
            string sSQL = "select CategoryID, CategoryName, Description from Categories";
            using (SqlConnection con = new SqlConnection(getString.ConnectionString))
            {
                List<Category> lst = new List<Category>();

                SqlDataReader r = null;
                SqlCommand cmd = new SqlCommand(sSQL, con);
                con.Open();

                r = cmd.ExecuteReader(CommandBehavior.CloseConnection);
                while (r.Read())
                {
                    Category category = new Category();
                    category.CategoryID = Convert.ToInt16(r["CategoryID"]);
                    category.CategoryName = r["CategoryName"].ToString();
                    category.Description = r["DEscription"].ToString();

                    lst.Add(category);
                }
                return lst;
            }
        }
        return null;
    }

    public Category GetCategory(int id)
    {
        ConnectionStringSettings getString = WebConfigurationManager.ConnectionStrings["nwind"] as
ConnectionStringSettings;
        if (getString != null)
        {
            String sSQL1 = "select CategoryID, CategoryName, Description from Categories ";
            String sSQL2 = "WHERE CategoryID = " + id.ToString();
            String sSQL = sSQL1 + sSQL2;
            using (SqlConnection con = new SqlConnection(getString.ConnectionString))
            {
                SqlDataReader r = null;
                SqlCommand cmd = new SqlCommand(sSQL, con);
                con.Open();

                Category category = new Category();

                r = cmd.ExecuteReader(CommandBehavior.CloseConnection);
                while (r.Read())
                {
                    category.CategoryID = Convert.ToInt16(r["CategoryID"]);
                    category.CategoryName = r["CategoryName"].ToString();
                    category.Description = r["Description"].ToString();
                }
                return category;
            }
```

```
        }
        return null;
    }

    public void InserirCategory(Category category)
    {
        ConnectionStringSettings getString = WebConfigurationManager.ConnectionStrings["nwind"] as
ConnectionStringSettings;
        if (getString != null)
        {
            using (SqlConnection con = new SqlConnection(getString.ConnectionString))
            {
                SqlCommand cmd = new SqlCommand("stpInserirCategory",con);
                cmd.CommandType = CommandType.StoredProcedure;
                cmd.Parameters.AddWithValue("@CategoryName", category.CategoryName);
                cmd.Parameters.AddWithValue("@Description", category.Description);

                try
                {
                    con.Open();
                    cmd.ExecuteNonQuery();
                }
                catch (SqlException ex)
                {
                    throw new Exception("Erro: " + ex.Message);
                }
                finally {
                    con.Close();
                }
            }
        }
    }
    public void EditCategory(Category category)
    {
        ConnectionStringSettings getString = WebConfigurationManager.ConnectionStrings["nwind"] as
ConnectionStringSettings;
        if (getString != null)
        {
            using (SqlConnection con = new SqlConnection(getString.ConnectionString))
            {
                SqlCommand cmd = new SqlCommand("stpEditarCategory", con);
                cmd.CommandType = CommandType.StoredProcedure;
                cmd.Parameters.AddWithValue("@CategoryID", category.CategoryID);
                cmd.Parameters.AddWithValue("@CategoryName", category.CategoryName);
                cmd.Parameters.AddWithValue("@Description", category.Description);

                try
                {
                    con.Open();
                    cmd.ExecuteNonQuery();
                }
                catch (SqlException ex)
                {
                    throw new Exception("Erro: " + ex.Message);
                }
                finally
                {
                    con.Close();
                }
            }
```

```
        }
    }
    public void DeleteCategory(int id)
    {
        ConnectionStringSettings getString = WebConfigurationManager.ConnectionStrings["nwind"] as
ConnectionStringSettings;
        if (getString != null)
        {
            using (SqlConnection con = new SqlConnection(getString.ConnectionString))
            {
                SqlCommand cmd = new SqlCommand("stpDeletarCategory", con);
                cmd.CommandType = CommandType.StoredProcedure;
                cmd.Parameters.AddWithValue("@CategoryID", id);

                try
                {
                    con.Open();
                    cmd.ExecuteNonQuery();
                }
                catch (SqlException ex)
                {
                    throw new Exception("Erro: " + ex.Message);
                }
                finally
                {
                    con.Close();
                }
            }
        }
    }
    public class Category
    {
        public int CategoryID { get; set; }
        public string CategoryName { get; set; }
        public string Description { get; set; }
    }
  }
}
```

7. No arquivo web.config, acrescente a linha de código a seguir:

```
<add name="nwind" connectionString="Data Source=(Local);Initial
                    Catalog=Northwind; Integrated Security=true;"/>
```

8. Crie o controle, na pasta Controllers, de acordo com o código a seguir:

Controle em C#

```
using MvcComDB.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace MvcComDB.Controllers
{
```

```
public class CategoryController : Controller
{
    //
    // GET: /Category/

    public ActionResult Index()
    {
        var model = _db.GetAllCategories();

        return View(model);

    }

    //
    // GET: /Category/Details/5

    CategoryRepository _db = new CategoryRepository();

    public ActionResult Details(int id)
    {
        var model = _db.GetCategory(id);

        return View(model);
    }

    //
    // GET: /Category/Create

    public ActionResult Create()
    {
        return View();
    }

    //
    // POST: /Category/Create

    [HttpPost]
    public ActionResult Create(MvcComDB.Models.CategoryRepository.Category category,
FormCollection collection)
    {
        try
        {
            if (!ModelState.IsValid) return View();

            _db.InserirCategory(category);

            return RedirectToAction("Index");
        }
        catch
        {
            return View();
        }
    }

    //
    // GET: /Category/Edit/5

    public ActionResult Edit(int id)
    {
        var model = _db.GetCategory(id);
```

```
            return View(model);
        }

        //
        // POST: /Category/Edit/5

        [HttpPost]
        public ActionResult Edit(CategoryRepository.Category category, FormCollection collection)
        {
            try
            {
                if (!ModelState.IsValid) return View();

                _db.EditCategory(category);

                return RedirectToAction("Index");
            }
            catch
            {
                return View();
            }
        }

        //
        // GET: /Category/Delete/5

        public ActionResult Delete(int id)
        {
            var model = _db.GetCategory(id);

            return View(model);
        }

        //
        // POST: /Category/Delete/5

        [HttpPost]
        public ActionResult Delete(int id, FormCollection collection)
        {
            try
            {
                // TODO: Add delete logic here
                if (!ModelState.IsValid) return View();

                _db.DeleteCategory(id);

                return RedirectToAction("Index");
            }
            catch
            {
                return View();
            }
        }
    }
}
```

9. Crie as Views

Views → Index

```
@model IEnumerable<MvcComDB.Models.CategoryRepository.Category>

@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>

<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table>
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.CategoryID)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.CategoryName)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Description)
        </th>
        <th></th>
    </tr>

@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.CategoryID)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.CategoryName)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Description)
        </td>
        <td>
            @Html.ActionLink("Edit", "Edit", new  { id=item.CategoryID }) |
            @Html.ActionLink("Details", "Details", new { id=item.CategoryID }) |
            @Html.ActionLink("Delete", "Delete", new { id=item.CategoryID })
        </td>
    </tr>
}

</table>
```

View → Create

```
@model MvcComDB.Models.CategoryRepository.Category

@{
    ViewBag.Title = "Create";
}

<h2>Create</h2>

@using (Html.BeginForm()) {
```

```
    @Html.ValidationSummary(true)

    <fieldset>
        <legend>Category</legend>
        <div class="editor-label">
            @Html.LabelFor(model => model.CategoryID)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.CategoryID)
            @Html.ValidationMessageFor(model => model.CategoryID)
        </div>
        <div class="editor-label">
            @Html.LabelFor(model => model.CategoryName)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.CategoryName)
            @Html.ValidationMessageFor(model => model.CategoryName)
        </div>

        <div class="editor-label">
            @Html.LabelFor(model => model.Description)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.Description)
            @Html.ValidationMessageFor(model => model.Description)
        </div>

        <p>
            <input type="submit" value="Create" />
        </p>
    </fieldset>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

View → Edit

```
@model MvcComDB.Models.CategoryRepository.Category

@{
    ViewBag.Title = "Edit";
}

<h2>Edit</h2>

@using (Html.BeginForm()) {
    @Html.ValidationSummary(true)

    <fieldset>
        <legend>Category</legend>

        @Html.HiddenFor(model => model.CategoryID)

        <div class="editor-label">
```

```
            @Html.LabelFor(model => model.CategoryName)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.CategoryName)
            @Html.ValidationMessageFor(model => model.CategoryName)
        </div>

        <div class="editor-label">
            @Html.LabelFor(model => model.Description)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.Description)
            @Html.ValidationMessageFor(model => model.Description)
        </div>

        <p>
            <input type="submit" value="Save" />
        </p>
    </fieldset>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

View → Detalhes

```
@model MvcComDB.Models.CategoryRepository.Category

@{
    ViewBag.Title = "Details";
}

<h2>Details</h2>

<fieldset>
    <legend>Category</legend>

    <div class="display-label">
        @Html.DisplayNameFor(model => model.CategoryName)
    </div>
    <div class="display-field">
        @Html.DisplayFor(model => model.CategoryName)
    </div>

    <div class="display-label">
        @Html.DisplayNameFor(model => model.Description)
    </div>
    <div class="display-field">
        @Html.DisplayFor(model => model.Description)
    </div>
</fieldset>
<p>
    @Html.ActionLink("Edit", "Edit", new { id=Model.CategoryID }) |
    @Html.ActionLink("Back to List", "Index")
```

```
</p>
```

View → Delete

```
@model MvcComDB.Models.CategoryRepository.Category

@{
    ViewBag.Title = "Delete";
}

<h2>Delete</h2>

<h3>Are you sure you want to delete this?</h3>
<fieldset>
    <legend>Category</legend>

    <div class="display-label">
        @Html.DisplayNameFor(model => model.CategoryName)
    </div>
    <div class="display-field">
        @Html.DisplayFor(model => model.CategoryName)
    </div>

    <div class="display-label">
        @Html.DisplayNameFor(model => model.Description)
    </div>
    <div class="display-field">
        @Html.DisplayFor(model => model.Description)
    </div>
</fieldset>
@using (Html.BeginForm()) {
    <p>
        <input type="submit" value="Delete" /> |
        @Html.ActionLink("Back to List", "Index")
    </p>
}
```

10. No arquivo web.config, acrescente a seguinte linha de código

Web.config

```
<connectionStrings>
    <add name="nwind" connectionString="Data Source=(Local);Initial Catalog=Northwind;Integrated
            Security=true;"/>
</connectionStrings>
```