

Exemple d'application LINQ to SQL utilisant des procédures stockées

Voici ce que nous voulons obtenir

The screenshot shows a web application interface with the following elements:

- Liste des catégories disponibles:** A dropdown menu.
- Liste des question dans la catégorie:** A large grey rectangular area, likely a placeholder for a list of questions.
- Form for adding a question:** A series of input fields on the left side:
 - Enoncé de la question:
 - Categorie:
 - Difficulté:
 - Reponse A:
 - Est bonne?:
 - Reponse B:
 - Est bonne?:
 - Reponse C:
 - Est bonne?:
 - Reponse D:
 - Est bonne?:
- Ajouter Question et les réponses:** A button at the bottom of the form.

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.Linq;

namespace KbkExempleLinq
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            const string chaine = "Data Source=M-INFO-SY\\SQLEXPRESS;Initial Catalog=BDjeur;User ID=patoche;Password=123456";
            private DataClasses1DataContext dcBd= new DataClasses1DataContext(chaine);

            private void Form1_Load(object sender, EventArgs e)
            {
                listerCategorie();
            }

            private void listerCategorie()
            {
                foreach (var listeCa in dcBd.listerCategorie())
                comboCategoTrivia.Items.Add(listeCa.nomCategorie);
                comboCategoTrivia.SelectedIndex = 0;
            }
        }
    }
}

```

```
private void comboCategoTrivia_SelectedIndexChanged(object sender, EventArgs e)
{
    string nomCat = comboCategoTrivia.SelectedItem.ToString();
    DGVques2.DataSource = dcBd.listerQuestion(nomCat);
}
```

```
private void ajouterQuestion()
{
    char ca = char.Parse(textCategorie.Text.ToString());
    char b1 = char.Parse(textEstbonne1.Text.ToString());
    char b2 = char.Parse(textEstbonne1.Text.ToString());
    char b3 = char.Parse(textEstbonne1.Text.ToString());
    char b4 = char.Parse(textEstbonne1.Text.ToString());
    char dif = char.Parse(textDifficulte.Text.ToString());
    dcBd.ajouterQuestionReponses(textEnonce.Text, ca, dif,
                                textRep1.Text, b1,
                                textRep2.Text, b2,
                                textRep3.Text, b3,
                                textRep4.Text, b4);
}
```

```
private void btnAjouterQuestion_Click(object sender, EventArgs e)
{
    ajouterQuestion();
}
}
```

Les procédures :

Ajouter questions et ses réponses :

```
CREATE procedure [dbo].[ajouterQuestionReponses] (@enonce varchar(100),
                                                @idcategorie char(1),@difficulte char(1),
                                                @repA varchar(100), @estbonneA char(1),
                                                @repB varchar(100),@estbonneB char(1),
                                                @repC varchar(100),@estbonneC char(1),
                                                @repD varchar(100),@estbonneD char(1)
                                                ) AS
```

```
Begin
declare @idQuetion int;
begin try
    begin transaction
    insert into QuestionTrivia(enonce,difficulte,idCategorie)
    values (@enonce,@difficulte,@idcategorie);

    select @idQuetion =@@IDENTITY;

    insert into ReponseTrivia(laReponse,estBonne,idQuestion)
    values(@repA, @estbonneA,@idQuetion);

    insert into ReponseTrivia(laReponse,estBonne,idQuestion)
    values(@repB, @estbonneB,@idQuetion);

    insert into ReponseTrivia(laReponse,estBonne,idQuestion)
    values(@repC, @estbonneC,@idQuetion);

    insert into ReponseTrivia(laReponse,estBonne,idQuestion)
    values(@repD, @estbonneD,@idQuetion);
    commit;

end try
begin catch
    rollback;
end catch
end;
```

Lister les catégories

```
CREATE procedure [dbo].[listeCategorie] as
BEGIN
select nomCategorie from categorieTrivia;
END;
```

Lister les questions selon le nom de la catégorie

```
ALTER procedure [dbo].[listerQuestion](@nomCategorie varchar(30)) as
begin
select enonce, difficulte, nomCategorie
from QuestionTrivia Q inner join CategorieTrivia C
on Q.idCategorie=c.idCategorie
where @nomCategorie =nomCategorie;

end;
```