

Exemple d'application

GESCONTACT

Gestion des Contacts

Structure de la base de données MySql

Table : groupe_contact

#	Nom	Type
1	id_gr_contact 	int(1)
2	libelle_gr_contact	varchar(30)

```
CREATE TABLE `groupe_contact` (  
  `id_gr_contact` int(1) NOT NULL,  
  `libelle_gr_contact` varchar(30) COLLATE utf8_unicode_ci NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;  
ALTER TABLE `groupe_contact`  
  ADD PRIMARY KEY (`id_gr_contact`);
```

Fichier XML : groupe_contact.xml

```
<?xml version="1.0" encoding="utf-8" ?>  
<structure>  
  <champs name="id_gr_contact" type="nombre" auto="oui" rang="0" tablerefname=""/>  
  <champs name="libelle_gr_contact" type="text" auto="non" rang="1" tablerefname=""/>  
</structure>
```

Classe : Groupe_contact.cs

```
using System;  
namespace GesContact  
{  
  public class Groupe_contact  
  {  
    public int Id_gr_contact{  
      set;get;  
    }  
    public string Libelle_gr_contact{  
      set;get;  
    }  
    public Groupe_contact()  
  }  
}
```

```

    {
    }
}
}

```

Table : contact

#	Nom	Type	Interclassement
1	id_contact 	int(2)	
2	nom_prenom	varchar(40)	utf8_unicode_ci
3	tel_contact	varchar(20)	utf8_unicode_ci
4	email_contact	varchar(30)	utf8_unicode_ci
5	id_gr_contact 	int(1)	

```

CREATE TABLE `contact` (
  `id_contact` int(2) NOT NULL,
  `nom_prenom` varchar(40) COLLATE utf8_unicode_ci NOT NULL,
  `tel_contact` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
  `email_contact` varchar(30) COLLATE utf8_unicode_ci NOT NULL,
  `id_gr_contact` int(1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

```

Clé primaire

```

ALTER TABLE `contact`
  ADD PRIMARY KEY (`id_contact`),
  ADD KEY `id_gr_contact` (`id_gr_contact`);
ALTER TABLE `contact`
  MODIFY `id_contact` int(2) NOT NULL AUTO_INCREMENT;

```

Clé étrangère

```

ALTER TABLE `contact`
  ADD CONSTRAINT `contact_ibfk_1` FOREIGN KEY (`id_gr_contact`) REFERENCES `groupe_contact`
  (`id_gr_contact`) ON UPDATE CASCADE;
COMMIT;

```

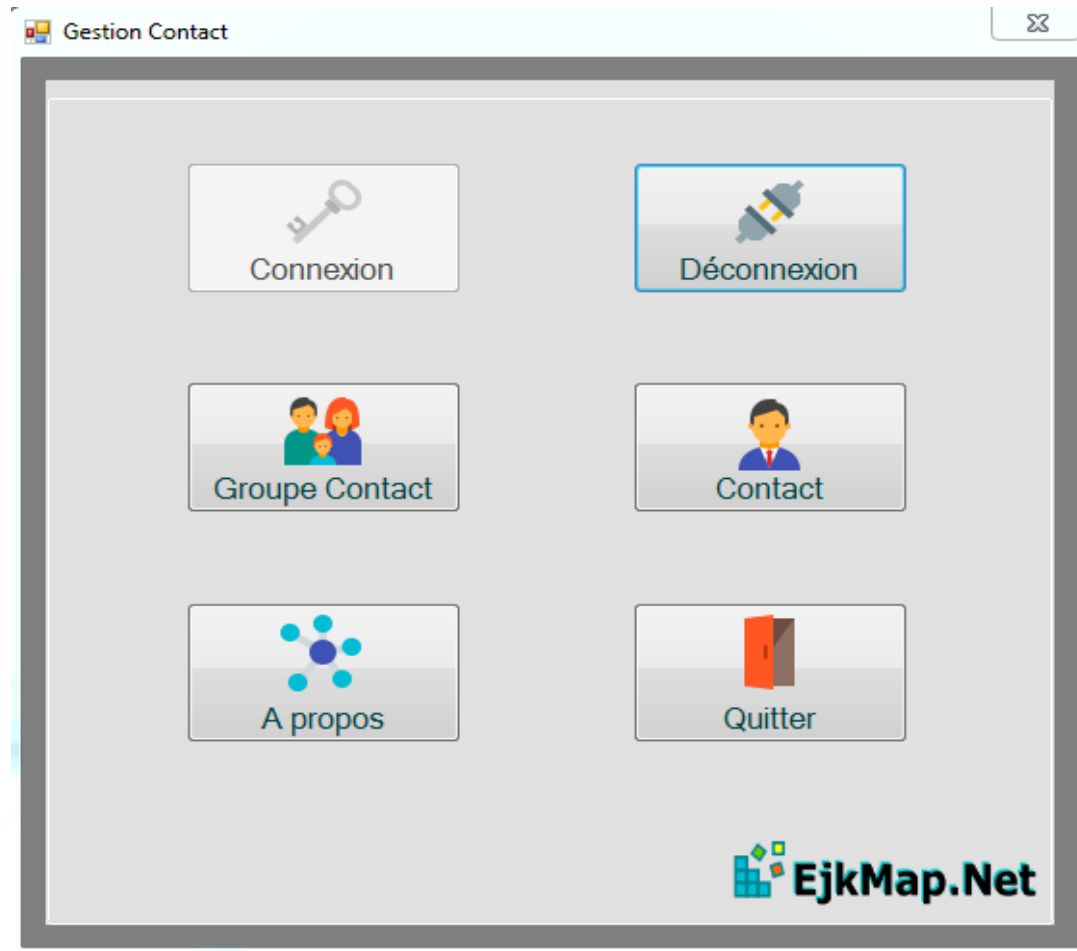
Fichier xml : contact.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<structure>
  <champs name="id_contact" type="nombre" auto="oui" rang="0" tablerefname=""/>
  <champs name="nom_prenom" type="text" auto="non" rang="1" tablerefname=""/>
  <champs name="tel_contact" type="text" auto="non" rang="2" tablerefname=""/>
  <champs name="email_contact" type="text" auto="non" rang="3" tablerefname=""/>
  <champs name="id_gr_contact" type="nombre" auto="non" rang="4" tablerefname="groupe_contact"/>
</structure>
```

Classe Contact.cs

```
using System;
namespace GesContact
{
    public class Contact
    {
        public int Id_contact{
            set;get;
        }
        public string Nom_prenom{
            set;get;
        }
        public string Tel_contact{
            set;get;
        }
        public string Email_contact{
            set;get;
        }
        public int Id_gr_contact{
            set;get;
        }
        public Contact() {}
    }
}
```

Formulaire principal



Codage :

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Windows.Forms;
using EjkMap;

namespace GesContact
{
    /// <summary>
    /// Description of MainForm.
    /// </summary>

    public partial class MainForm : Form
    {
        public MainForm()
        {
```

```

InitializeComponent();
}
void Btn_quitterClick(object sender, EventArgs e)
{
    this.Close();
}
void Btn_groupe_contactClick(object sender, EventArgs e)
{
    //Chargement du Formulaire Groupe Contact
    try{
        var myform=new FormGroupeContact();
        myform.ShowDialog();
    }
    catch(Exception ex){
        Data.AfficherMessage(ex.Message,"inf");
    }
}
void Btn_contactClick(object sender, EventArgs e)
{
    //Chargement du Formulaire Contact
    try{
        var myform=new FormContact();
        myform.ShowDialog();
    }
    catch(Exception ex){
        Data.AfficherMessage(ex.Message,"inf");
    }
}
void Btn_connexionClick(object sender, EventArgs e)
{
    //Connexion à la base de données
    try{
        Data.chaineconnexion="SERVER=localhost; DATABASE=gescontact; UID=gescontact;
PASSWORD='123456789'";
        Data.typedb="mysql";
        Data.Connexion();
        //Connexion OK
        Data.AfficherMessage("Connexion Ok","inf");
        //Activation des Bouttons
        btn_deconnexion.Enabled=true;
        btn_groupe_contact.Enabled=true;
        btn_contact.Enabled=true;
        //Déactivation du bouton connexion
        btn_connexion.Enabled=false;
    }
}

```

```

    }
    catch(Exception ex){
        Data.AfficherMessage(ex.Message,"inf");
    }
}
void Btn_deconnexionClick(object sender, EventArgs e)
{
    //Déconnexion
    try{
        Data.Deconnexion();
        //Déconnexion OK
        Data.AfficherMessage("Déconnexion Ok","inf");
        //Désactivation des Bouttons
        btn_deconnexion.Enabled=false;
        btn_groupe_contact.Enabled=false;
        btn_contact.Enabled=false;
        //Activation du bouton connexion
        btn_connexion.Enabled=true;
    }
    catch(Exception ex){
        Data.AfficherMessage(ex.Message,"inf");
    }
}
void Btn_aproposClick(object sender, EventArgs e)
{
    //Chargement du Formulaire Contact
    try{
        var myform=new Formapropos();
        myform.ShowDialog();
    }
    catch(Exception ex){
        Data.AfficherMessage(ex.Message,"inf");
    }
}
}
}
}

```

Formulaire Groupe Contact

Groupe Contact

Id Groupe

Libelle

id_gr_contact	libelle_gr_contact
1	Famille
2	Amis
3	Collègue de Travail
4	Professionnel

EjkMap.Net

Codage :

```
using System;
using System.Windows.Forms;
using EjkMap;

namespace GesContact
{
    /// <summary>
    /// Description of FormGroupeContact.
    /// </summary>
    public partial class FormGroupeContact : Form
    {
        //Paramètres de retour lors des appels externes
    }
}
```

```

public int Id_gr_contact_ret{
    set;get;
}
public string Lib_gr_contact_ret{
    set;get;
}
public FormGroupeContact()
{
    //
    // The InitializeComponent() call is required for Windows Forms designer support.
    //
    InitializeComponent();

    //
    // TODO: Add constructor code after the InitializeComponent() call.
    //
}
void FormGroupeContactLoad(object sender, EventArgs e)
{
    //Chargement de la liste des groupes de contact
    try{
        Data.Liste("Groupe_contact",lst_contact);
    }
    catch(Exception ex){
        Data.AfficherMessage(ex.Message,"err");
    }
}
void Btn_enregistrerClick(object sender, EventArgs e)
{
    //Enregistrement d'un nouveau groupe de contact
    try{
        var mygroupe=new Groupe_contact();
        //Hydratation
        mygroupe.Libelle_gr_contact=txt_lib_gr.Text.Trim();
        //Enregistrement
        if(Data.Enregistrer(mygroupe)){
            Data.AfficherMessage("Données ajoutées avec succès","inf");
            Data.Liste("Groupe_contact",lst_contact);
            //Formulaire vide
            btn_nouveau.PerformClick();
        }
        else {
            Data.AfficherMessage("Erreur détectée","inf");
        }
    }
}

```



```

        mygroupe=null;
    }
    catch(Exception ex){
        Data.AfficherMessage(ex.Message,"inf");
    }
}

void Btn_modifierClick(object sender, EventArgs e)
{
    //Modification d'un groupe de contact
    try{
        var mygroupe=new Groupe_contact();
        //Hydratation
        mygroupe.Id_gr_contact=int.Parse(txt_id_gr.Text); //Clé Primaire
        mygroupe.Libelle_gr_contact=txt_lib_gr.Text;
        //Enregistrement
        if(Data.Modifier(mygroupe)){
            Data.AfficherMessage("Données modifiées avec succès","inf");
            Data.Liste("Groupe_contact",lst_contact);
        }
        else {
            Data.AfficherMessage("Erreur détectée","inf");
        }
        mygroupe=null;
    }
    catch(Exception ex){
        Data.AfficherMessage(ex.Message,"inf");
    }
}

void Btn_supprimerClick(object sender, EventArgs e)
{
    //Suppression d'un groupe de contact
    try{
        var mygroupe=new Groupe_contact();
        mygroupe.Id_gr_contact=int.Parse(txt_id_gr.Text); //Clé Primaire
        DialogResult reponse;
        //Confirmation de la suppression
        reponse=MessageBox.Show("Voulez vous supprimer ce groupe de contacts ?", "Confirmation
Suppression", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
        if(reponse.ToString()=="Yes")
        {
            //Suppression
            if(Data.Supprimer(mygroupe)){
                Data.AfficherMessage("Données supprimées avec succès","inf");
                Data.Liste("Groupe_contact",lst_contact);
            }
        }
    }
    catch(Exception ex){
        Data.AfficherMessage(ex.Message,"inf");
    }
}

```

```

    }
    else {
        Data.AfficherMessage("Erreur détectée", "inf");
    }
}
mygroupe=null;
}
catch(Exception ex){
    Data.AfficherMessage(ex.Message, "inf");
}
}
void Lst_contactCellClick(object sender, DataGridViewCellEventArgs e)
{
    try{
        int ligne=lst_contact.CurrentRow.Index;
        txt_id_gr.Text=lst_contact.Rows[ligne].Cells[0].Value.ToString();
        txt_lib_gr.Text=lst_contact.Rows[ligne].Cells[1].Value.ToString();
        this.Id_gr_contact_ret=int.Parse(txt_id_gr.Text);
        this.Lib_gr_contact_ret=txt_lib_gr.Text;

    }
    catch(Exception ex){
        Data.AfficherMessage(ex.Message, "inf");
    }
}
void Btn_nouveauClick(object sender, EventArgs e)
{
    txt_id_gr.Text=txt_lib_gr.Text="";
}
void Btn_fermerClick(object sender, EventArgs e)
{
    this.Close();
}
}
}

```

Formulaire Contact

	id_contact	nom_prenom	tel_contact	email_contact	id_gr_contact	libelle_gr_contact
▶	1	ELJAHED Karim	1111111111	karim@mynodus.ma	1	Famille
	2	LAATAR Abdelmjid	2222222222	laatar@gmail.com	2	Amis
	3	IGHMOUR Hassna	3333333333	ighmour@menara.ma	3	Collègue de Travail
	4	HOUSSAILY Leila	4444444444	leila@gmail.com	3	Collègue de Travail
	5	HASSOUNI Mohammed	5555555555	hassouni@gmail.com	2	Amis
	6	Dr. BERRADA Maria	6666666666	maria@gmail.com	4	Professionnel
	7	Pharmacie OUALILI	7777777777	oualili@gmail.com	4	Professionnel
	8	Clinique la détente	8888888888	contact@detente.com	4	Professionnel
	9	Laboratoire la liberté	9999999999	contact@liberte.com	4	Professionnel
	10	AKANOUR Siham	1010101010	siham@akanour.com	1	Famille
	11	ELJAHED Salaheddine	1212121212	salah@akanour.com	1	Famille

Codage :

using System;

using System.Windows.Forms;

using EjkMap;

namespace GesContact

{

/// <summary>

/// Description of FormContact.

/// </summary>

public partial class FormContact : Form

{

public FormContact()

{

//

// The InitializeComponent() call is required for Windows Forms designer support.

//

InitializeComponent();

```

//
// TODO: Add constructor code after the InitializeComponent() call.
//
}
void FormContactLoad(object sender, EventArgs e)
{
    //Affichage de la liste des contacts
    try{
        Data.ListeAvecJointure("Contact",lst_contact);
    }
    catch(Exception ex){
        Data.AfficherMessage(ex.Message,"inf");
    }
}
void Btn_fermerClick(object sender, EventArgs e)
{
    this.Close();
}
void Btn_chercher_groupeClick(object sender, EventArgs e)
{
    //Chercher un groupe de contact
    try{
        var myform=new FormGroupeContact();
        myform.ShowDialog();
        //Utilisation des variables retournées par le formulaire groupe contact
        txt_id_gr.Text=myform.Id_gr_contact_ret.ToString();
        txt_lib_gr.Text=myform.Lib_gr_contact_ret;
    }
    catch(Exception ex){
        Data.AfficherMessage(ex.Message,"inf");
    }
}
void Btn_enregistrerClick(object sender, EventArgs e)
{
    try{
        //Déclaration d'une instance
        var mycontact=new Contact();
        //Hydratation
        mycontact.Nom_prenom=txt_nom.Text.Replace("\", "\\");
        mycontact.Tel_contact=txt_tel.Text;
        mycontact.Email_contact=txt_email.Text;
        mycontact.Id_gr_contact=int.Parse(txt_id_gr.Text);
        //Enregistrement
        if(Data.Enregistrer(mycontact)){

```

```

        Data.AfficherMessage("Données ajoutées avec succès","inf");
        //Rafraichissement de la liste
        Data.ListeAvecJointure("Contact",lst_contact);
        //Formulaire vide
        btn_nouveau.PerformClick();
    }
    else {
        Data.AfficherMessage("Erreur détectée","inf");
    }
    mycontact=null;
}
catch(Exception ex){
    Data.AfficherMessage(ex.Message,"inf");
}
}
void Btn_modifierClick(object sender, EventArgs e)
{
    try{
        //Déclaration
        var mycontact=new Contact();
        //Hydratation
        mycontact.Id_contact=int.Parse(txt_id_contact.Text);
        mycontact.Nom_prenom=txt_nom.Text.Replace("\", "\\");
        mycontact.Tel_contact=txt_tel.Text;
        mycontact.Email_contact=txt_email.Text;
        mycontact.Id_gr_contact=int.Parse(txt_id_gr.Text);
        //Enregistrement
        if(Data.Modifier(mycontact)){
            Data.AfficherMessage("Données modifiées avec succès","inf");
            //Rafraichissement de la liste
            Data.ListeAvecJointure("Contact",lst_contact);
        }
        else {
            Data.AfficherMessage("Erreur détectée","inf");
        }
        mycontact=null;
    }
    catch(Exception ex){
        Data.AfficherMessage(ex.Message,"inf");
    }
}
void Btn_supprimerClick(object sender, EventArgs e)
{
    try{

```

```

//Déclaration
var mycontact=new Contact();
//Hydratation de la clé primaire
mycontact.Id_contact=int.Parse(txt_id_contact.Text);
DialogResult reponse;
//Confirmation de la suppression
reponse=MessageBox.Show("Voulez vous supprimer ce Contact ?", "Confirmation
Suppression", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
if(reponse.ToString()=="Yes")
{
    if(Data.Supprimer(mycontact)){
        Data.AfficherMessage("Données supprimées avec succès", "inf");
        //Rafraichissement de la liste
        Data.ListeAvecJointure("Contact",lst_contact);
    }
    else {
        Data.AfficherMessage("Erreur détectée", "inf");
    }
}
mycontact=null;
}
catch(Exception ex){
    Data.AfficherMessage(ex.Message, "inf");
}
}
void Btn_nouveauClick(object sender, EventArgs e)
{
    //Formulaire vide
    txt_id_contact.Text=txt_nom.Text=txt_tel.Text=txt_email.Text="";
    txt_id_gr.Text=txt_lib_gr.Text="";
}
void Lst_contactCellClick(object sender, DataGridViewCellEventArgs e)
{
    try{
        //Récupération de la ligne sélectionné
        int ligne=lst_contact.CurrentRow.Index;
        txt_id_contact.Text=lst_contact.Rows[ligne].Cells[0].Value.ToString();
        //Rechercher le contact à partir de son ID
        var mycontact=new Contact();
        mycontact.Id_contact=int.Parse(txt_id_contact.Text); //Clé Primaire
        Data.Rechercheld(mycontact);
        if(mycontact.Nom_prenom==null)
        {
            Data.AfficherMessage("Contact introuvable, vérifier votre ID !", "exc");
        }
    }
}

```

```

    }
    else
    {
        txt_nom.Text=mycontact.Nom_prenom;
        txt_tel.Text=mycontact.Tel_contact;
        txt_email.Text=mycontact.Email_contact;
        txt_id_gr.Text=mycontact.Id_gr_contact.ToString();
        //Recherche du libellé du groupe de contact
        var mygroupe=new Groupe_contact();
        mygroupe.Id_gr_contact=mycontact.Id_gr_contact;
        Data.Rechercheld(mygroupe);
        txt_lib_gr.Text=mygroupe.Libelle_gr_contact;
        mygroupe=null;
    }
    mycontact=null;
}
catch(Exception ex){
    Data.AfficherMessage(ex.Message,"inf");
}
}
void Btn_filtre_adhClick(object sender, EventArgs e)
{
    try{
        //Création d'un Ordre SQL
        string reqsql="SELECT C.id_contact, C.nom_prenom, C.tel_contact, C.email_contact, C.id_gr_contact,
G.libelle_gr_contact ";
        reqsql+="FROM contact C, groupe_contact G WHERE C.id_gr_contact=G.id_gr_contact ";
        //Ajout du filtre
        reqsql+="AND C.id_gr_contact="+txt_id_gr.Text;
        Data.OrdreSql(reqsql,st_contact);
    }
    catch(Exception ex){
        Data.AfficherMessage(ex.Message,"inf");
    }
}
}

```

```
void Btn_listeClick(object sender, EventArgs e)
{
    try{
        //Affichage de tous les contacts
        Data.ListeAvecJointure("Contact",lst_contact);
    }
    catch(Exception ex){
        Data.AfficherMessage(ex.Message,"inf");
    }
}
}
```