

TP 1

Objectifs :

- Utiliser des commandes en mode déconnecté.

Guide de travail :

- ✓ Rédaction d'un rapport.
- ✓ Technologie ADO.NET, Fournisseur d'accès OLEDB

PARTIE 1:

AFFICHAGE D'INFORMATIONS RELATIVES À DES LIVRES.

On veut consulter les données contenues dans une table **Livre** dans une base de données **Biblio** :

- Livre (CodeL, Titre, Auteur, NbExemplaires).

a. Créer la base de données **Biblio** et la table Livre sous Sql Server.

b. Pour accéder à cette base de données, en mode déconnecté, créer les objets suivants :

- une connexion con.

```
SqlConnection con = new SqlConnection();
```

- une commande CmdSelect.

```
SqlCommand CmdSelect = new SqlCommand();
```

- un dataset ds non typé.

```
DataSet ds = new DataSet();
```

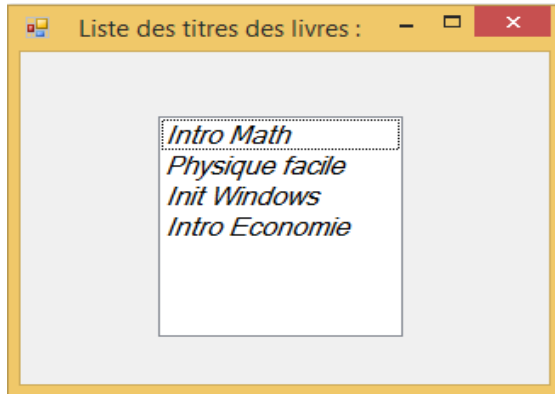
- un dataAdapter da.

```
SqlDataAdapter da = new SqlDataAdapter();
```

- Mappage (ou mapping) de table.

```
CmdSelect.CommandText = "Select * From Livre";  
da.SelectCommand = CmdSelect;  
da.Fill(ds, "Livre");
```

c. Créer un projet VS et le formulaire suivant permettant d'afficher les titres de tous les livres dans un ListBox :



d. Dans le même projet ajouter un deuxième formulaire permettant d'afficher dans un DataGridView les livres d'un auteur choisi dans un Combo :

Se servir de la méthode Select pour effectuer une recherche :

```
DataRow[] drTab = ds.Tables["Livre"].Select("Auteur='" + CboAuteur.Text + "'");
```

PARTIE 2:

AFFICHAGE D'INFORMATIONS RELATIVES A DES THEMES ET DES LIVRES.

On veut consulter les données contenues dans les deux tables **Thème** et **Livre** dans la base de données **Biblio** :

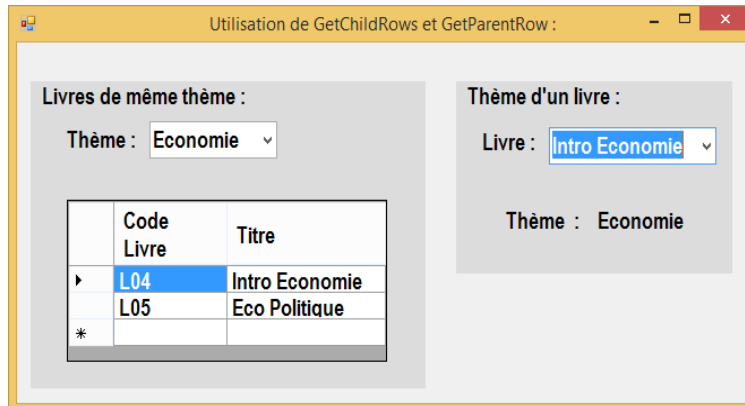
- Thème (CodeTh, IntituléTh).
- Livre (CodeL, Titre, Auteur, NbExemplaires, #CodeTh).

On veut afficher dans un même formulaire :

- La liste des livres d'un thème choisi dans un combo.
- Le thème d'un livre choisi dans un combo.

Travail à réaliser :

- a. Créer, dans la base de données **Biblio**, la table **Thème** et modifier la table **Livre** pour ajouter la clé étrangère #CodeTh.
- b. Créer un projet VS et le formulaire suivant :



. Créer les objets et variables nécessaires et le code réalisant le travail demandé.

- Les objets à créer :

```
SqlConnection con = new SqlConnection();
DataSet ds = new DataSet();
SqlDataAdapter daLivre = new SqlDataAdapter();
SqlDataAdapter daThème = new SqlDataAdapter();
```

- Mapping des tables Thème et Livre :

```
con.ConnectionString = strcon;

daLivre.SelectCommand = new SqlCommand();
daLivre.SelectCommand.Connection = con;
daLivre.SelectCommand.CommandText = "Select * From Livre";
daThème.SelectCommand = new SqlCommand();
daThème.SelectCommand.Connection = con;
daThème.SelectCommand.CommandText = "Select * From Thème";
```

```
daLivre.Fill(ds, "Livre");
daThème.Fill(ds, "Thème");
```

- Spécification des clés primaires :

```
DataColumn[] cols_L = {ds.Tables["Livre"].Columns["CodeL"]};
ds.Tables["Livre"].PrimaryKey = cols_L;
DataColumn[] cols_Th = {ds.Tables["Thème"].Columns["CodeTh"]};
ds.Tables["Thème"].PrimaryKey = cols_Th;
```

- Créations de la relation :

```
DataRelation drel = new DataRelation("Thème-Livre",
    ds.Tables["Thème"].Columns["CodeTh"],
    ds.Tables["Livre"].Columns["CodeL"]);

ds.Relations.Add(drel);
```

- Récupération des livres (fils) d'un thème :

```
foreach (DataRow drLivre in drThème.GetChildRows("Thème-Livre"))
{ . . . }
```

- Récupération du thème (Parent) d'un livre :

```
... = drLivre.GetParentRow("Thème-Livre")[1].ToString();
```

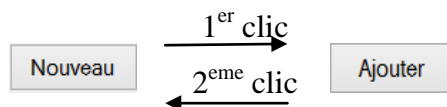
PARTIE 3:

Il s'agit de développer une application de gestion de bibliothèque en mode déconnecté. On va commencer par faire la gestion des Adhérents. On se servira de la table **Adhérent** dont la structure est la suivante :

- Adhérent (CodeA, NomA, Adresse, DateInscription).

Travail à réaliser :

a. Créer un projet VS nommé « **Gestion Bibliothèque** » et le formulaire suivant permettant de faire la mise à jour de la table Adhérent :



b. Créer un **DataSet typé dsBiblio**. Utiliser une instance de celui-ci :

```
dsBiblio dsB = new dsBiblio();
```

c. Créer un objet **BindingSource** qui sera lié à la table Adhérent, puis associer ses champs aux **TextBox(s)** respectifs à l'aide des **DataBindings** :

```
BindingSource AdhérentBS = new BindingSource();

AdhérentBS.DataSource = dsB.Adhérent;

TxtCodeA.DataBindings.Add("Text", AdhérentBS, "CodeA");

...

```

d. Écrire le code correspondant à chaque bouton réalisant le traitement attendu en utilisant les objets créés.

N.B. Le BindingSource pourra être utilisé aussi bien pour la navigation, la recherche et la mise à jour.

PARTIE 4:

Reprendre le travail fait dans l'activité 9, pour réaliser la mise à jour des deux tables Thème et Livres :

Rappel :

- Thème (CodeTh, IntituléTh).
- Livre (CodeL, Titre, Auteur, NbExemplaires, #CodeTh).

Travail à réaliser :

- Créer, dans la base de données **Biblio**, les deux tables Thème et Livre si vous ne l'avez pas encore fait.
- Modèles des formulaires MajThème et MajLivre:

MajThème_Déconnecté

Mise à jour Thème

Code Thème : Nouveau

Intitulé Thème : Modifier

Supprimer

<< < > >>

Recherche

Rechercher

Code Thème :

Quitter

MajLivre_Déconnecté

Mise à jour Livre

Code Livre : Nouveau

Titre : Modifier

Auteur : Supprimer

Nb Exemplaires :

Thème :

<< < > >>

Recherche

Rechercher

Code Adhérent :

Quitter

- Créer les objets nécessaires et le code réalisant les travaux attendus.

PARTIE 5:

Il s'agit maintenant de réaliser l'emprunt de livres par des adhérents.

Travail à réaliser :

a. Créer dans le projet VS « **Gestion Bibliothèque** » le formulaire suivant :

Saisie Emprunt

Code Adhérent : A02
Nom : Fikri
Thème : Economie
Livres : Intro Economie, Eco Politique
Date d'emprunt : 10/12/2014
Emprunts déjà existant de l'Adhérent :

	Titre	Date Emprunt	Date Retour
▶	L02	10/11/2014 00...	20/11/2014 00...
*	L05	10/12/2014 19...	

Enregistrer
Quitter

B. Créer les objets nécessaires et le code réalisant les travaux demandés :

Spécifications :

- Au choix d'un adhérent, son nom et ses emprunts passés sont affichées.
- Au choix d'un thème, la liste de tous les livres de ce thème est affichée.
- Après le choix d'un livre et d'une date d'emprunt, Le bouton « **Enregistrer** » permet d'enregistrer l'emprunt dans la table Emprunt et de l'ajouter dans le DataGridView.

PARTIE 6:

Il s'agit maintenant de réaliser le retour d'emprunts de livres par des adhérents.

Travail à réaliser :

a. Créer dans le projet VS « **Gestion Bibliothèque** » le formulaire suivant :

SaisieRetour_Déconnecté

Saisie Retour

Code Adhèrent : A03 Date de Retour : 02/12/2014

Nom : Naji

Emprunts de l'Adhèrent : Enregistrer

	Code Livre	Titre	Date Emprunt	Date Retour
	A03	L02	04/11/2014 00:...	20/11/2014 20.
▶	A03	L04	24/11/2014 00:...	
*				

Quitter

B. Créer les objets nécessaires et le code réalisant les travaux demandés :

Spécifications :

- Au choix d'un adhérent, son nom et ses emprunts passés sont affichées.
- Après sélection de l'emprunt à retourner et le choix d'une date de retour, Le bouton « **Enregistrer** » permet d'enregistrer le retour dans la table Emprunt et d'afficher la date de retour de l'emprunt sélectionné dans le DataGridView.