

Série N 5

Parties abordées dans cette série :

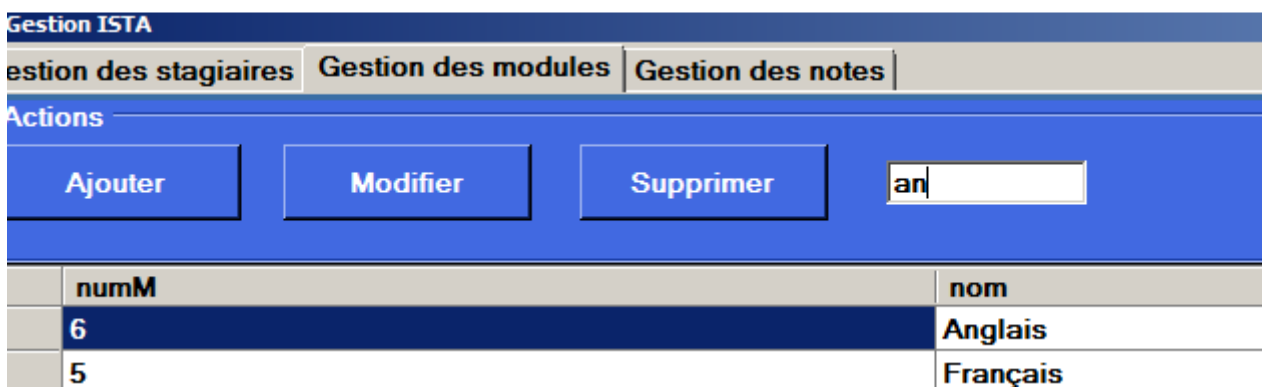
- **DataView**
- **DataSet et XML**
- **Gestion des erreurs**
- **Ecriture des erreurs en fichier de log**
- **Appel aux procédures stockées depuis l'application**
- **L'utilisation des transactions**

Nous allons appliquer les exercices ci-dessous sur l'application de gestion des notes liée à la base de données GestionNote qui contient les tables Etudiant (ou Stagiaire) / Matière et Evaluer

EXERCICE 01 : DataView

1. Créer deux objets de type DataView, un pour l'affichage des stagiaires et l'autre pour l'affichage des modules. Laisser le paramètre de filtrage vide et trier par le nom.
2. Affecter les DataView comme DataSource au DataGridView stagiaires et modules.
3. Ajouter un TextBox au-dessus des DataGridView stagiaire et module.
4. Au changement de texte dans le TextBox, mettre à jours le filtre du DataView pour filtrer avec le nom. (utiliser la propriété RowFilter du DataView)


Exemple :



numM	nom
6	Anglais
5	Français

EXERCICE 02 : DataSet et XML

1. Ajouter une interface nommé 'Configuration' à votre application.
2. Ajouter un GroupBox nommé 'Import/Export XML' dans cette interface.
3. Dans le GroupBox, ajouter deux boutons 'Import' et 'Export'
4. Au clic sur le bouton 'Export' sauvegarder votre DataSet dans un fichier XML
5. Au clic sur le bouton 'Import', charger le fichier XML dans votre DataSet

 OFPPT	DR : Oriental Etablissement : CFMNTIOE Formateur : Zakaria KADDARI	Module : ADO.NET << Travaux pratiques >>
---	---	---

- Assurez-vous que votre application fonctionne aussi avec le chargement de données depuis le fichier XML vers le DataSet.

EXERCICE 03 : Gestion des erreurs et écriture des erreurs en fichier de log.

Toute application doit gérer ses erreurs via les Exceptions avec affichage des erreurs dans des MessageBox pour les utilisateurs ainsi que dans des fichiers dite fichiers de log.

Ces fichiers de log sont un moyen de diagnostic très utile pour le développeur de l'application, le développeur peut consulter ces fichiers de log à tout moment, afin de déterminer la cause d'un crash, erreur ou problème.

- Mettre en place la gestion des erreurs via des Exception lors de la connection à l'image du code présenté dans le slide 40 dans le support de cours (attention : les numéros des erreurs dans ce slide ne sont pas tous correctes)
- Mettre en place l'enregistrement des erreurs dans un fichier de log nommé erreurs.log situé dans le même répertoire que votre application, et ceci sous la forme suivante :

Date et heure de l'erreur : NUM numéro de l'erreur – Message de l'erreur

EXERCICE 04 : Appel aux procédures stockées depuis l'application

- Créer une procédure stockée qui retourne le nombre de place disponibles dans chaque groupe pour une filière dont l'id est passé en paramètre (Le nombre max de stagiaires par groupe est 30)
- Faire appel à cette procédure depuis votre application, afficher le résultat dans un DataGridView dans le tableau de bord, le DataGridView doit avoir les colonnes suivantes : Libellé groupe, Nombre de place disponible
- Ajouter un ComboBox qui contient la liste des filières au-dessus du DataGridView, au choix d'une filière, mettre à jours le DataGridView avec les données relatives à la filière choisie.

EXERCICE 05 : L'utilisation des transactions

Le code d'utilisation des transactions présenté dans le slide 47 dans le support de cours est pour le cas du mode connecté, rien ne change dans le mode déconnecté sauf qu'il faut associer l'objet de transaction directement aux commandes InsertCommand, UpdateCommand et DeleteCommand de votre DataAdapter.

Si vous utiliser un CommandBuilder pour générer les trois commandes citées ci-dessus, ce qui est certainement le cas, il faut que :

- L'objet `CommandBuilder` soit accessible dans toute l'application, donc qu'il soit statique par exemple.
- Après l'instanciation de l'objet, il faut appeler les 3 commandes comme suite :

```
Votre_objet_command_builder.GetDeleteCommand();  
Votre_objet_command_builder.GetInsertCommand();  
Votre_objet_command_builder.GetUpdateCommand();
```

Ceci nous permet de générer le code des 3 commandes avant l'utilisation des transactions.

- Enfin, il faut associer votre objet transaction aux 3 commandes comme suite :

```
Votre_objet_command_builder.GetInsertCommand().Transaction = tran;  
Votre_objet_command_builder.GetUpdateCommand().Transaction = tran;  
Votre_objet_command_builder.GetDeleteCommand().Transaction = tran;
```

Travail à faire :

Intégrer les transactions dans votre application lors de toute opération de sauvegarde en base de données.