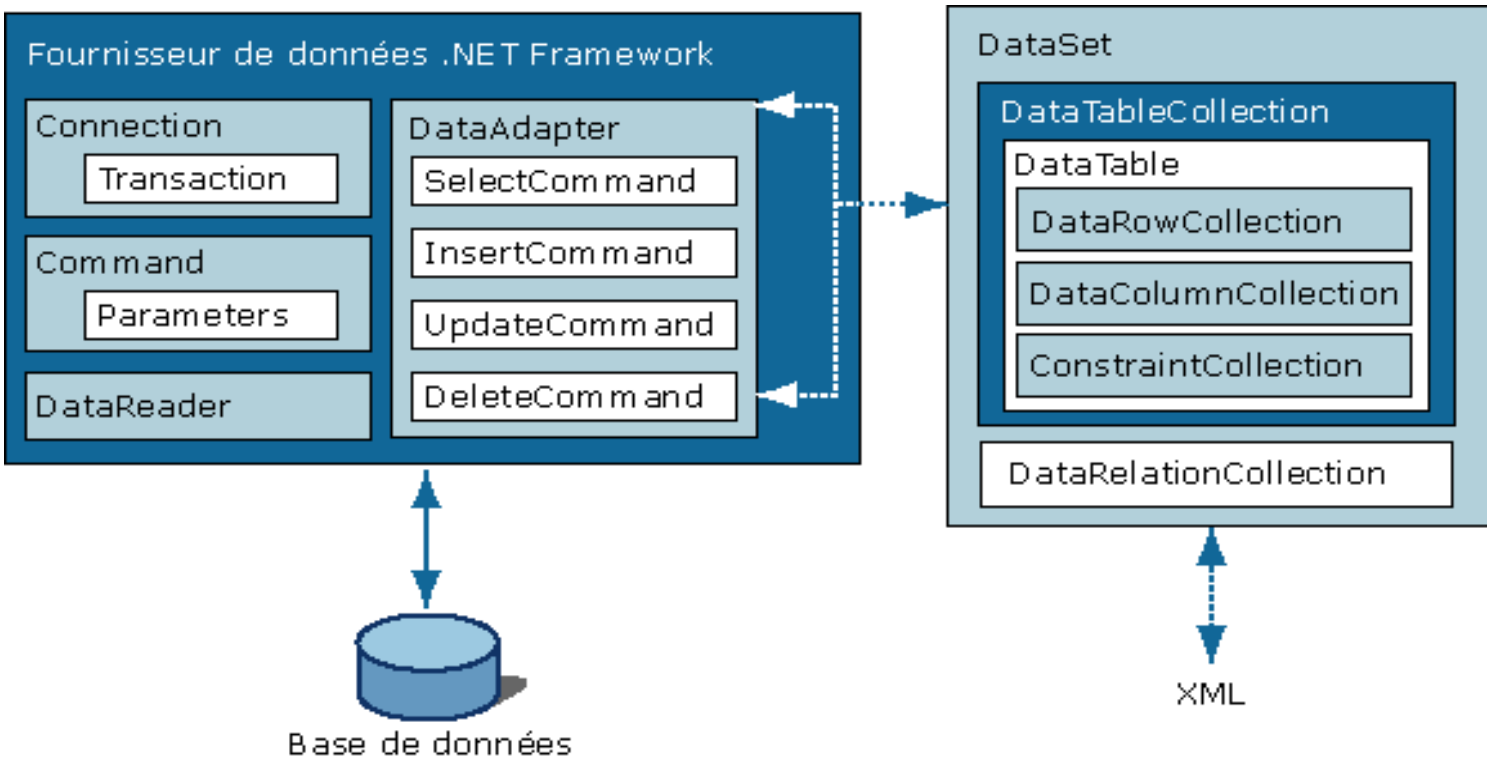
ADO.net Architecture



Rappels

- L’objet OracleConnection;
- L’objet OracleCommand ;
- L’objet OracleDataReader;

Le DataSet

- Définition :

Un DataSet est vue comme un cache de données avec une structure similaire à une base de données relationnelle. (table, colonnes, contraintes..)

Même si un DataSet est similaire à une base de données, cependant il n'interagit pas directement avec celle-ci.

Pour interagir avec une base de donnée, un DataSet a besoin d'un DataAdapter.

Le DataSet

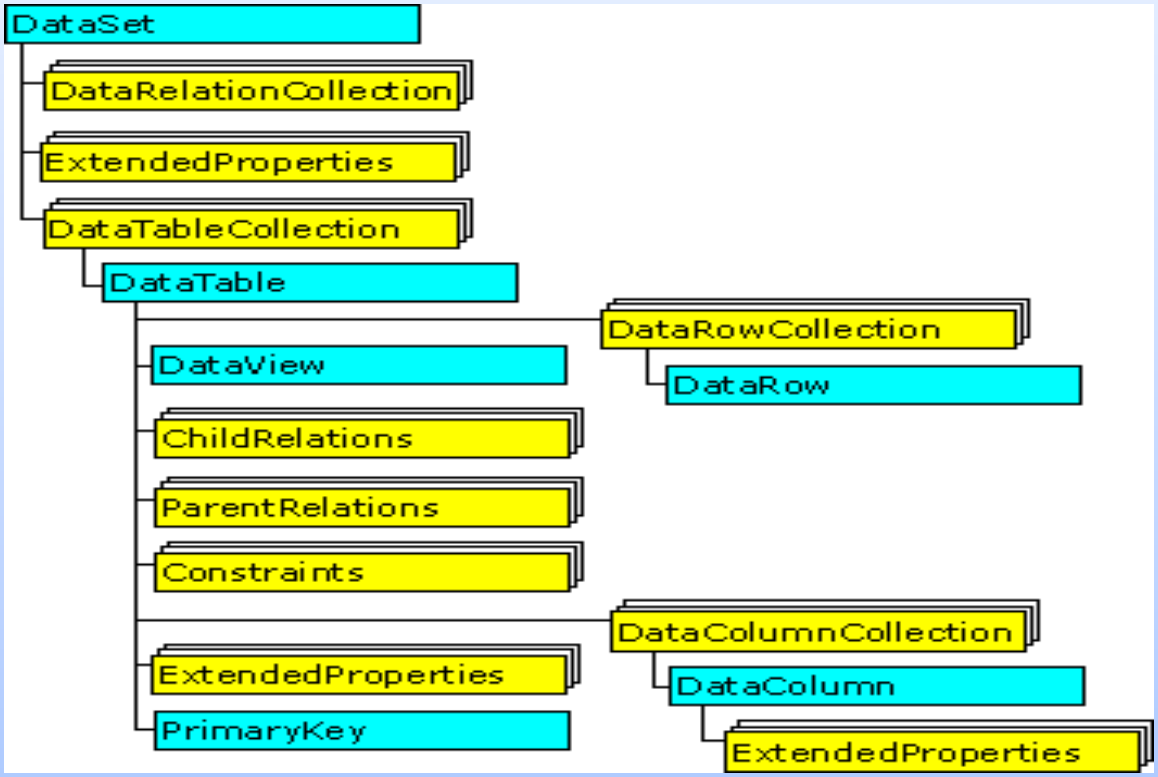
- Les données en provenance d'une base de données, d'un fichier XML ou d'une autre entrée de l'utilisateur peuvent être stockées dans un DataSet. Lorsque des modifications sont apportées au DataSet, celles-ci peuvent être suivies et vérifiées avant de mettre à jour les données source
- Creation d'un DataSet:

```
DataSet monDataSet = new DataSet();
```

Le DataSet

- Propriétés importantes:
  - DataSetName : définit ou obtient le nom du DataSet en cours.
  - Tables: permet d'obtenir les tables contenues dans le DataSet
- Méthodes importantes de l'objet DataSet
  - AcceptChanges (): Valide toutes les modifications apportées à ce **DataSet** depuis son chargement ou depuis le dernier appel à **AcceptChanges**.
  - Clear(): Efface toutes les données de **DataSet** en supprimant toutes les lignes de l'ensemble des tables.
  - GetChanges() : Obtient une copie du DataSet contenant l'ensemble des modifications qui lui ont été apportées depuis son dernier chargement ou depuis l'appel à AcceptChanges.
  - Les autres voir http://msdn.microsoft.com/fr-fr/library/system.data.dataset_methods(VS.80).aspx

Le DataSet: structure



L’objet OracleDataAdapter

Définition:

un objet OracleDataAdapter est un objet ODP qui permet de remplir un DataSet et permet d’effectuer les mises à jour de la base de données par le DataSet

Propriétés importantes du DataAdapter:

DeleteCommand:Obtient ou définit une instruction SQL ou une procédure stockée utilisée pour supprimer de nouvelles dans la base de données.

InsertCommand:Obtient ou définit une instruction SQL ou une procédure stockée utilisée pour insérer de nouvelles lignes dans la base de données

.

SelectCommand:Obtient ou définit une instruction SQL ou une procédure stockée utilisée pour sélectionner des lignes enregistrements dans la base de données.

UpdateCommand: Obtient ou définit une instruction SQL ou une procédure stockée utilisée pour mettre à jour des enregistrements dans la base de données.

Pour plus de détails voir: http://download.oracle.com/docs/html/B28089_01/OracleDataAdapterClass.htm#i1001593

L’objet OracleDataAdapter

- Méthodes importantes:
  - Fill ():Ajoute ou actualise des lignes de DataSet pour qu'elles correspondent à celles de la source de données.
  - Update () : Appelle les instructions INSERT, UPDATE ou DELETE respectives pour chaque ligne insérée, mise à jour ou supprimée dans DataSet.
  - Dispose ():
- Creation d’un OracleDataAdapter:

```
string sql = "select *  from Livres";
OracleDataAdapter monAdapter = new OracleDataAdapter(sql, conn);
```

- Remplisage du DataSet:

```
monAdapter.Fill(monDataSet, "Resultatlivre");
```

L’objet OracleDataAdapter

- Exemple :

```
public Form1()
      {
         InitializeComponent();
      }
       private OracleConnection conn = new
    OracleConnection();
     DataSet monDataSet =new DataSet();


----    connection
```

L’objet OracleDataAdapter

```
private void Afficher_Click(object sender, EventArgs e)
      {



         string sql = "select *  from Livres";

         OracleDataAdapter monAdapter = new OracleDataAdapter(sql, conn);
         //remplisage du DataSet
         monAdapter.Fill(monDataSet, "Resultatlivre");
         maSource = new BindingSource(monDataSet, "Resultatlivre");
         dataGridView1.DataSource = maSource;
}
```

Vider le DataSet avant de le remplir

```
if (monDataSet.Tables.Count > 0)
        {
       monDataSet.Tables["Resultatlivre"].Clear();
//Ou
  //monDataSet.Clear();
      }
```

Le DataBinding

Le but du data binding c'est de permettre de lier un contrôle (UI) avec une ou plusieurs sources de données.

Bindingsource: peut s'appliquer sur un dataTable

Exemple:

maSource = new BindingSource(monDataSet, "Resultatlivre");

Resultatlivre représente une table dans le DataSet.

Databinding: exemple

textBox1.DataBindings.Add("Text", dataSet1, "Customers.FirstName");

une façon de lier textBox1 à la colonne Customers.FirstName du DataSet1

Le DataBinding

S'il existe une deuxième textBox lié à une colonne de la même table du même DataSet alors le **BindingContext** est informé qu'il s'agit de la même liaison.

```
textBox2.DataBindings.Add("Text", dataSet1,
    "Customers.LastName");
```

Pour parcourir les contenus des textBox à l'aide des boutons suivant ou précédent vous pouvez utiliser la propriété Position du **BindingContext** .

```
this.BindingContext[dataSet1,"Customers"].Position+=1;
```

ou

```
textBox1.BindingContext[dataSet1,"Customers"].Position+=1;
```

Exemple

```
private void afficher_Click(object sender, EventArgs e)
    {
       string sql ="select * from etudiants";
       OracleDataAdapter monAdapter = new OracleDataAdapter(sql, conn);

       monAdapter.Fill(monDataSet,"ListeEtudiants");
       lister();
    }
    private void lister()
    {
       Textnumad.DataBindings.Add("Text", monDataSet, "ListeEtudiants.numad");
       TextNom.DataBindings.Add("Text", monDataSet, "ListeEtudiants.Nom");
       TextPrenom.DataBindings.Add("Text", monDataSet,"ListeEtudiants.PRENEOM");
       }
```

Exemple: Suite

```
private void suivant_Click(object sender, EventArgs e)
  {
  //this.BindingContext[monDataSet,"ListeEtudiants"].Position+=1;
  Textnumad.BindingContext[monDataSet, "ListeEtudiants"].Position += 1;
  }


  private void precedent_Click(object sender, EventArgs e)
  {
    // les deux lignes suivantes font la même chose
     //this.BindingContext[monDataSet, "ListeEtudiants"].Position -= 1;
     Textnumad.BindingContext[monDataSet, "ListeEtudiants"].Position -= 1;
```

Dissocier les contrôles

- On peut dissocier les contrôles d'une source de données en utilisant la méthode Clear() associée au DataBinding
- Exemple:

```
private void vider()
     {
//dissocier les contrôles – le faire pour tous les text box

     textNumad.DataBindings.Clear();
      textNom.DataBindings.Clear();

//effacer le contenu ---- le faire pour tous les text box
    textNumad.Clear();
    textNom.Clear();
}
```

L’objet OracleParameter

- Représente un paramètre pour l’objet OracleCommand
- Déclaration: OracleParameter NomPrametre = new OracleParameter();

Propriétés importantes:

- parameterName: nom du paramètre associé à un nom de colonne
- OracleDbType : type de données oracle
- Size: longueur du paramètre.

Exemple:

```
OracleParameter paramNumad = new
    OracleParameter(":numad", OracleDbType.NVarchar2, 3);
```

Exemple

```
private void Ajouter_Click(object sender, EventArgs e)
    {
        OracleCommand oranIns = new OracleCommand("insert into
    ETUDIANTS(numad, nom, PRENEOM, ville) values
    (:numad,:nom,:PRENEOM,:ville)", conn); // paramètre de la requête

        //Déclaration des paramètres pour OracleParameter
        OracleParameter paramNumad = new OracleParameter(":numad",
    OracleDbType.NVarchar2, 3);
        OracleParameter paramNom = new OracleParameter(":nom",
    OracleDbType.NVarchar2, 30);
        OracleParameter paramPrn = new OracleParameter(":PRENEOM",
    OracleDbType.NVarchar2, 30);
        OracleParameter paramVille = new OracleParameter (":ville",
    OracleDbType.NVarchar2,30);
```

Exemple suite

```
//affectation des valeurs à chacun des
    paramètres
        paramNumad.Value = textNumad.Text;
        paramNom.Value = textNom.Text;
        paramPrn.Value = textPrn.Text;
        paramVille.Value = textville.Text;
```

Exemple suite

attribution des paramètres (avec leurs valeurs) aux paramètres de l'OracleCommand **OraIns**

l'ordre dans lequel cette affectation est faîte doit être la même que celle que vous avez dans votre requête SQL INSERT

```
oranIns.Parameters.Add(paramNumad);
oranIns.Parameters.Add(paramNom);
oranIns.Parameters.Add(paramPrn);
oranIns.Parameters.Add(paramVille);
```

Exécution de la requête Insertion:

oranIns.ExecuteNonQuery();

Exercice

- Donner le code qui permet de faire une modification de la ville (un paramètre) pour un étudiant donné. (numad est un paramètre)

Solution

```
private void Modifier_Click(object sender, EventArgs e)
      {

          OracleCommand oraUpdate = new OracleCommand("update etudiants set ville
      =:ville where numad =:Numad " ,conn);
                    OracleParameter paramville = new OracleParameter(":ville",
      OracleDbType.NVarchar2, 30);
          paramville.Value = textville.Text;
          oraUpdate.Parameters.Add(paramville);


          OracleParameter paramnumad = new OracleParameter(":numad",
      OracleDbType.NVarchar2, 3);
          paramnumad.Value = textNumad.Text;
          oraUpdate.Parameters.Add(paramnumad);
//execution de la requête
          oraUpdate.ExecuteNonQuery();
```