ADO.net et les procédures stockées

- Rappels:
  - OracleCommand: Quelques propriétés.

CommandText	**Obtient ou définit l'instruction SQL ou la procédure stockée à exécuter sur la base de données**
CommandType	**Obtient ou définit une valeur indiquant la manière dont la propriété CommandText doit être interprétée (instruction SQL ou PROCEDURE)**
Connection	**Obtient ou définit l'objet OracleConnection utilisé par cette instance de OracleCommand.**
Parameters	**Spécifie les paramètres de la requête SQL ou de la procédure stockée**

ADO.net et les procédures stockées

- Rappels:
  - OracleParameter: Quelques constructeurs

OracleParameter (string, OracleDbType)	String désigne le nom du paramètre, le OracleDbtype désigne le type de données du Pramètre (un OracleType.) : public OracleParameter(string parameterName, OracleDbType oraType)
OracleParameter(string, OracleDbType, int)	Même que le précédent, sauf qu'on indique la taille du paramètre
OracleParameter(string, OracleDbType, ParameterDirection)	Le ParameterDirection indique si le paramètre est en IN ou Out ou IN OUT .Utilisé lorsque nous avons une procédure stockée

ADO.net et les procédures stockées

- Rappels:
  - OracleDataAdapter: Quelques constructeurs

OracleDataAdapter()	**Instancie un objet OracleDataAdapter**
OracleDataAdapter(String, OracleConnection)	**Instancie un objet un objet OracleConnection avec la commande SQL SELECT et une connexion à la BD.(Ici, le SELECT est passé par le OracleDataAdapter)**
OracleDataAdapter(OracleCommand)	**Initialise une nouvelle instance de la classe OracleDataAdapter avec l'instruction SQL SELECT spécifiée.(Ici, le SELECT est contenu dans le OracleCommand)**

ADO.net et les procédures stockées

- On utilise les mêmes références, les mêmes espaces de noms, les mêmes objets (OracleCommand, OracleDataAdapter ..).
- Pour exécuter une procédure stockées, il faut:
  - Indiquer au programme qu'il s'agit d'une procédure stockée:

```
OracleCommand OBjCommand = new OracleCommand("GestionEmploye", conn);
OBjCommand.CommandText = "Afficher"; // Nom de la procedure
OBjCommand.CommandType = CommandType.StoredProcedure;
```

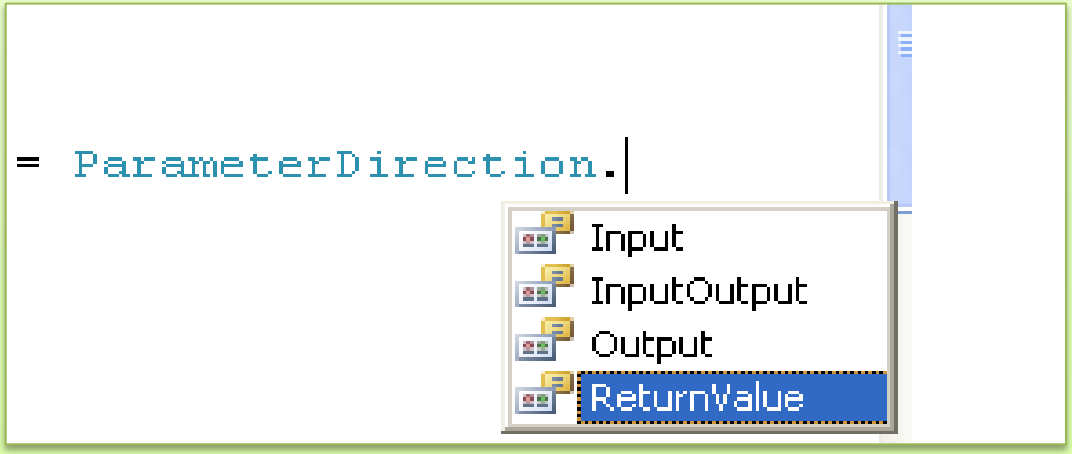
  - indiquer comment le programme va exécuter la procédure (en IN ou OUT ou autre)

ADO.net et les procédures stockées

- Les paramètres de la procédure sont les paramètres de l'objet OracleParameter
- La propriété Direction de OracleParameter indique si le paramètre est un paramètre d'entrée, de sortie ou de valeur de retour.

Exemple:

- ObjSupprimer.Parameters.Add(new OracleParameter("num", OracleDbType.Int32, 4)).Direction



ADO.net et les procédures stockées

Paramètres de la procédure	Paramètre de OracleParameter
IN	Input
Out	Output
Cas d'une fonction :La fonction retourne un type de variable.	ReturnValue
IN OUT	InputOutput

ADO.net et les procédures stockées

- Lorsque les paramètre de la procédure stockée sont en IN alors, la propriété de ParameterDirection est Input
- Lorsque les paramètre de la procédure stockée sont en OUT alors, la propriété de ParameterDirection est Output
- Lorsqu'il s'agit d'une fonction qui retourne une valeur alors, la propriété de ParameterDirection est ReturnValue

ADO.net et les procédures stockées: Cas d'une insertion (paramètres en IN)

```
private void Inserer_Click(object sender, EventArgs e)
    {
        OracleCommand oraAjout = new OracleCommand("GESTIONPRODUITS", conn);
        oraAjout.CommandText = "GESTIONPRODUITS.INSERTION";
        oraAjout.CommandType = CommandType.StoredProcedure;

        OracleParameter orapamNum = new OracleParameter("PNUM", OracleDbType.Int32);
        orapamNum.Direction = ParameterDirection.Input;
        orapamNum.Value = Numproduit.Text;
        oraAjout.Parameters.Add(orapamNum);

        OracleParameter orapamnumDesc = new OracleParameter("PDESCRIPTION", OracleDbType.Varchar2,50);
        orapamnumDesc.Direction = ParameterDirection.Input;
        orapamnumDesc.Value = Descriptionproduit.Text;
        oraAjout.Parameters.Add(orapamnumDesc);

// Le reste des paramètres
        oraAjout.ExecuteNonQuery();
    }
```

ADO.net et les procédures stockées: Cas d'une fonction qui retourne un REF CURSOR

```
OracleCommand oraliste = new OracleCommand("GESTIONPRODUITS", conn);
        oraliste.CommandText = "GESTIONPRODUITS.LISTER";
        oraliste.CommandType = CommandType.StoredProcedure;

       // pour une fonction, le paramètre de retour doit être déclaré en premier.
        OracleParameter OrapameResultat = new OracleParameter("RESULTAT", OracleDbType.RefCursor);
        OrapameResultat.Direction = ParameterDirection.ReturnValue;
        oraliste.Parameters.Add(OrapameResultat);

       // déclaration du paramètre en IN
        OracleParameter OrapamDesc = new OracleParameter("PDESCRIPTION", OracleDbType.Varchar2);
        OrapamDesc.Value = textDescription.Text;
        OrapamDesc.Direction = ParameterDirection.Input;
        oraliste.Parameters.Add(OrapamDesc);
```

pour remplir le DataSet, on declare un OracleDataAdapter pour lequel
on passe notre OracleCommand qui contient TOUS les paramètres.

```
OracleDataAdapter orAdater = new OracleDataAdapter(oraliste);
```

Suite du code (remplir le DataSet -→orAdater.Fill(monDataSet, "Produits");

Remarquez que: Le paramètre de retour est passé en premier.

Cas d'une procédure avec un paramètre de type REF CURSOR en OUT et un paramètre en IN

```
OracleCommand Oracmd = new OracleCommand("GESTIONPRODUITS", conn);
        Oracmd.CommandText = "GESTIONPRODUITS.CHERCHER";
        Oracmd.CommandType = CommandType.StoredProcedure;

   OracleParameter OraDesc = new OracleParameter("PDESCRIPTION", OracleDbType.Varchar2);
   OraDesc.Value = textDescription.Text;
   OraDesc.Direction = ParameterDirection.Input;
   Oracmd.Parameters.Add(OraDesc);

  OracleParameter orapamres = new OracleParameter("RES", OracleDbType.RefCursor);
   orapamres.Direction = ParameterDirection.Output;
    Oracmd.Parameters.Add(orapamres);

   OracleDataAdapter orAdater = new OracleDataAdapter(Oracmd);
```

Suite du code pour remplir le DataSe --→orAdater.Fill(monDataSet, "Produits");

Remarquez que les paramètres sont passés dans l'ordre de leur présence dans la procédure.

Utilisation d'une fonction qui retourne un REF CURSOR avec un DataReader

```
OracleCommand oraliste2 = new OracleCommand("GESTIONPRODUITS", conn);
        oraliste2.CommandText = "GESTIONPRODUITS.LISTER2";
        oraliste2.CommandType = CommandType.StoredProcedure;

          OracleParameter OrapameResultat = new OracleParameter("RESULTAT",
          OracleDbType.RefCursor);

  OrapameResultat.Direction = ParameterDirection.ReturnValue;
        oraliste2.Parameters.Add(OrapameResultat);

        OracleDataReader Oraread = oraliste2.ExecuteReader();
        while (Oraread.Read())
        {
           ListeDescription.Items.Add(Oraread.GetString(0));
           ListeDescription.SelectedIndex = 0;
        }
```

ADO.net et les procédures stockées(cas d'une fonction)

```
private void TotalEmp_Click(object sender, EventArgs e)
      {
  OracleCommand ObTotal = new OracleCommand("GestionEmployes", conn);
         ObTotal.CommandText = "GestionEmployes.TOTAL";
         ObTotal.CommandType = CommandType.StoredProcedure;
```

On peut également utiliser ces trois lignes

```
OracleParameter Oratotal = new OracleParameter ("TOTALEMP", OracleDbType.Int32,
    2);
    Oratotal.Direction = ParameterDirection.ReturnValue;
    ObTotal.Parameters.Add(Oratotal);

ObTotal.ExecuteScalar();
 textTotal.Text = ObTotal.Parameters["TOTALEMP"].Value.ToString();

      }
```