

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)

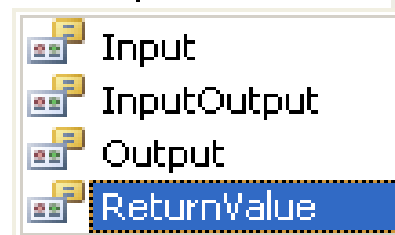
ADO.net et les procédures stockées

- 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`

```
= ParameterDirection.|
```



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 Ajouter_Click(object sender, EventArgs e)
{
    OracleCommand ObjAjout = new OracleCommand("GestionEmployes", conn);
    ObjAjout.CommandText = "GestionEmployes.INSERTION";
    ObjAjout.CommandType = CommandType.StoredProcedure;

    ObjAjout.Parameters.Add(new OracleParameter("num", OracleDbType.Int32, 4)).Direction = ParameterDirection.Input;
    ObjAjout.Parameters["num"].Value = TextNum.Text;

    ObjAjout.Parameters.Add(new OracleParameter("nom", OracleDbType.Varchar2, 30)).Direction = ParameterDirection.Input;
    ObjAjout.Parameters["nom"].Value = TextNom.Text;

    ObjAjout.Parameters.Add(new OracleParameter("prenom", OracleDbType.Varchar2, 30)).Direction = ParameterDirection.Input;
    ObjAjout.Parameters["prenom"].Value = TextPrn.Text;

    ObjAjout.Parameters.Add(new OracleParameter("salaire", OracleDbType.Int32, 8)).Direction = ParameterDirection.Input;
    ObjAjout.Parameters["salaire"].Value = TextSal.Text;

    ObjAjout.ExecuteNonQuery();
}
```

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

```
private void Ajouter_Click(object sender, EventArgs e)
{
    OracleCommand ObjAjout = new OracleCommand("GestionEmployes", conn);
    ObjAjout.CommandText = "GestionEmployes.INSERTION";
    ObjAjout.CommandType = CommandType.StoredProcedure;
```

Pour le paramètre nom on peut avoir les deux façons suivantes:

```
ObjAjout.Parameters.Add(new OracleParameter("nom", OracleDbType.Varchar2,
    30)).Direction = ParameterDirection.Input;
    ObjAjout.Parameters["nom"].Value = TextNom.Text;
```

Ou encore

```
OracleParameter pnom = new OracleParameter(":nom", OracleDbType.Varchar2,
    30);
    pnom.Value = TextNom.Text;
    ObjAjout.Parameters.Add(pnom).Direction = ParameterDirection.Input;
ObjAjout.ExecuteNonQuery()
}
```

ADO.net et les procédures stockées(paramètres en OUT)

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

```
    OracleCommand ObjetCmd = new OracleCommand("GestionEmploye", conn);
    ObjetCmd.CommandText = "GestionEmployes.chercher";
    ObjetCmd.CommandType = CommandType.StoredProcedure;
```

```
//paramètre de la recherche est en IN . On fait une recherche selon le nom.
```

```
    ObjetCmd.Parameters.Add(new OracleParameter("NOMR", OracleDbType.Varchar2,30)).Direction = ParameterDirection.Input;
    ObjetCmd.Parameters["NOMR"].Value = TextNom.Text;
```

Le résultat de la recherche est OUT. C'est un paramètre de type REF CURSOR.

```
    ObjetCmd.Parameters.Add(new OracleParameter("RESULTAT", OracleDbType.RefCursor)).Direction = ParameterDirection.Output;
    OracleDataAdapter oradap = new OracleDataAdapter(ObjetCmd);
```

```
    oradap.Fill(monDantaSet, "ListeEmp");
```

```
Lister(); // le DataBindings.
```

```
}
```

ADO.net et les procédures stockées(paramètres en OUT)

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

```
    OracleCommand ObjetCmd = new OracleCommand("GestionEmploye", conn);
    ObjetCmd.CommandText = "GestionEmployes.chercher";
    ObjetCmd.CommandType = CommandType.StoredProcedure;
```

Le résultat de la recherche est OUT. C'est un paramètre de type REF CURSOR.

Pour le paramètre en OUT on peut avoir les deux façons suivantes:

```
ObjetCmd.Parameters.Add(new OracleParameter(" REQUETE",
    OracleDbType.RefCursor)).Direction = ParameterDirection.Output;
    OracleDataAdapter oradap = new OracleDataAdapter(ObjetCmd);
```

Ou les trois lignes suivantes

```
OracleParameter orapam = new OracleParameter("REQUETE", OracleDbType.RefCursor);
orapam.Direction = ParameterDirection.Output;
ObjetCmd.Parameters.Add(orapam) ;
```

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;

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

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

}
```

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();

}
```

ADO.net et les procédures stockées

- Exercice 1:

Écrire le code qui correspond au bouton supprimer un employé. (le paramètre de la suppression est le numéro employé, il est en IN)

- Exercice 2:

Écrire le code correspondant à la mise à jour (modification). On modifie le salaire d'un employé. La modification se fait selon le numéro d'employé

Lab4: écrire le code C# qui fait appel aux procédures et fonctions que vous avez réalisés au dernier lab.