420-KBA-LG, programmation de bases de données

Saliha Yacoub

ADO.NET

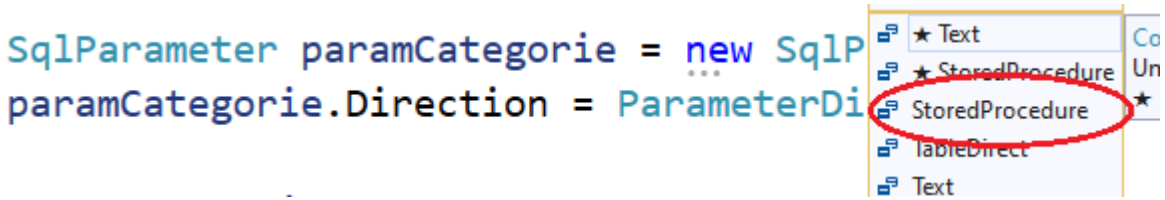
- Retour sur la dernière séance:
  - Point de vue des enseignants
  - Point de vue des étudiants.
- SqlParameter
  - Retour sur SqlCommand
  - Cas de paramètres en IN
  - Cas d’une fonctions stockée.

ADO.NET

La classe SqlCommand: Propriétés importantes:

Propriétés	Signification
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 procédure stockée).
Connection	Obtient ou définit l'objet SqlConnection utilisé par cette instance de SqlCommand.
Parameters	Spécifie les paramètres de la requête SQL ou de la procédure stockée

› Par défaut la propriété CommandType a la valeur Text (pour une requête SQL qui n'est pas une procédure



ADO.NET

La classe SqlParameter: Constructeurs:

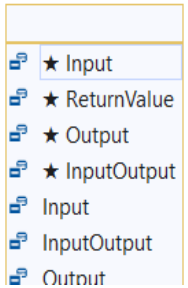
L'objet SqlParameter représente un paramètre pour l'SqlCommand ou une colonne du DataSet

SqlParameter()	Instancie un SqlParameter
SqlParameter (string, SqlDbType)	String désigne le nom du paramètre, le SqlDbtype désigne le type de données du Pramètre (un SqlType.) : public SqlParameter(string parameterName, SqlDbType)
SqlParameter(string, SqlDbType, int)	Même que le précédent, sauf qu'on indique la taille du paramètre
SqlParameter(string, SqlDbType, int, string)	Même que le précédent, sauf qu'on indique la taille du paramètre et le nom de la colonne source : à voir avec le DataSet
SqlParameter(string,SqlDbType,ParameterDirection)	Le ParameterDirection indique si le paramètre est en IN ou Out. Utilisé lorsque nous avons une procédure stockée

ADO.NET

Direction du paramètre:

```
SqlParameter categorie = new SqlParameter("@categorie", Sq
categorie.Direction = ParameterDirection.;
```



Paramètre de la procédure	Direction du SqlParameter
IN(par défaut)	Input (par défaut)
OUT	Output
Une fonction	ReturnValue

ADO.NET

La classe SqlParameter: Propriétés importantes

Direction	Obtient ou définit une valeur qui indique si le paramètre est un paramètre d'entrée uniquement, de sortie uniquement, bidirectionnel ou de valeur de retour d'une procédure stockée.
ParameterName	Obtient ou définit le nom de SqlParameter
SqlDbType	Spécifie le type de données Sql
Size	Obtient ou définit la taille maximale, en octets, des données figurant dans la colonne.
Value	Obtient ou définit la valeur du paramètre

ADO.NET

- Comment passer une procédure et ses paramètres ?
  - Étape 0: Le SqlCommad va permettre d'indiquer clairement qu'il s'agit d'une procédure par ses propriétés CommandText et CommandType:

```
SqlCommand cmdInsert = new SqlCommand("ajouterJoueur", sqlconn);
cmdInsert.CommandText = "ajouterJoueur";
cmdInsert.CommandType = CommandType.StoredProcedure;
```

ajouterJoueur est le nom de la procédure stockée

ADO.NET

- Comment passer une procédure et ses paramètres ?
  - Étape 1: définir un SqlParameter pour chaque paramètre de la procédure

```
SqlParameter parmNom = new SqlParameter("@nom", SqlDbType.VarChar, 30);
parmNom.Direction = ParameterDirection.Input;

SqlParameter parmPrenom = new SqlParameter("@prenom", SqlDbType.VarChar, 30);
parmPrenom.Direction = ParameterDirection.Input;

SqlParameter parmSalaire = new SqlParameter("@salaire", SqlDbType.Decimal);
parmSalaire.Direction = ParameterDirection.Input;

SqlParameter parmEquipe = new SqlParameter("@codeEquipe", SqlDbType.Char, 3);
parmEquipe.Direction = ParameterDirection.Input;
```

La procédure ajouterJoueur a 4 paramètres. Ils sont tous en IN

ADO.NET

- Comment passer une procédure et ses paramètres ?
  - Étape 2: donner des valeurs aux paramètres en utilisant le propriété Value de SqlParameter

```
parmNom.Value = 'Patoche';
parmPrenom.Value = 'Alain';
parmSalaire.Value = 100000;
parmEquipe.Value = 'MTL';
```

Les valeurs des paramètres peuvent être fournis par des zones de texte.

ADO.NET

- Comment passer une procédure et ses paramètres ?
  - Étape 3: Ajouter les paramètres avec leurs valeurs à SqlCommand avec la propriété Parameters de SqlCommand.

```
cmdInsert.Parameters.Add(parmNom);
cmdInsert.Parameters.Add(parmPrenom);
cmdInsert.Parameters.Add(parmSalaire);
cmdInsert.Parameters.Add(parmEquipe);
```

Les valeurs des paramètres peuvent être fournis par des zones de texte.

ADO.NET

- Comment passer une procédure et ses paramètres ?
  - Étape 4: Exécuter la procédure Stockée. Il faudra utiliser la méthode appropriée de SqlCommand.

```
cmdInsert.ExecuteNonQuery();
```

ajouterJoueur est le nom de la procédure stockée qui fait une insertion . Donc la méthode appropriée

ADO.NET

- Exemple de fonction sans paramètre qui retourne une valeur:

```
private void compter()
  {
        SqlCommand sqlcmdTotal  = new SqlCommand("totalJoueur", sqlconn);
       sqlcmdTotal l.CommandText = "totalJoueur";
       sqlcmdTotal.CommandType = CommandType.StoredProcedure;

      SqlParameter resultat = new SqlParameter("@totaljoueur", SqlDbType.Int);
      resultat.Direction = ParameterDirection.ReturnValue;

     sqlcmdTotal .Parameters.Add(resultat);
    sqlcmdTotal.ExecuteScalar();
    total.Text = resultat.Value.ToString();
 }
```

ADO.NET

› Exemple de fonction avec paramètre qui retourne une valeur:

```
private void nbJoueursEquipe() {
                SqlCommand cmdTotal = new SqlCommand("totalJoueurEquipe", sqlconn);
                cmdTotal.CommandText = "totalJoueurEquipe";
                cmdTotal.CommandType = CommandType.StoredProcedure;
//Paramètre en IN
            SqlParameter paramEquipe = new SqlParameter("@nomEquipe", SqlDbType.VarChar, 30);
            paramEquipe.Direction = ParameterDirection.Input;
            paramEquipe.Value = nomEquip; //'Canadiens'
//paramètre de retour
            SqlParameter resultat = new SqlParameter("@totajoueurs", SqlDbType.Int);
            resultat.Direction = ParameterDirection.ReturnValue;
//ajouter les paramètres au SqlCommand
            cmdTotal.Parameters.Add(resultat);
            cmdTotal.Parameters.Add(paramEquipe);
            cmdTotal.ExecuteScalar();
            totalParEquipe.Text = resultat.Value.ToString();
    }
```

ADO.NET

- Cas d’une fonction table:

Le cas d’une fonction table est traité comme une requête SQL SELECT quelconque.

Pour exécuter une fonction table dans Managment Studio on fait : select * from nomfonction().

CONCLUSION

QUESTIONS ??