

D'autres SGBD ?

```
Imports System.Data.SqlClient
Imports System.Data.Odbc
Imports System.Data.OleDb
Imports System.Data.OracleClient ' A obtenir gratuitement chez MicroSoft

Dim NomDBComplet As String ' Chemin d'accès et nom de fichier DB
Dim NomDuServeur As String = "... ' ".\SQLEXPRESS" par exemple
```

La fameuse chaîne de connexion ... Elle est un des ingrédients permettant de réaliser une application SGBD polyvalente : Access, SQL Serveur, MySql, Oracle, ...

Les autres ingrédients sont les classes interfaces `IDb ... : IDbConnection, IDbCommand, ...`

Toutefois, la programmation à l'aide de ces interfaces peut présenter quelques particularités qui rendent la mise au point plus difficile.

Il est souvent plus aisé de recompiler l'application après avoir adapté la chaîne de connexion et avoir réalisé un Recherche/Remplace, dans toute la solution, des noms des types utilisés (cf. ci-après).

```
' DB pilotée par MySql, ou un n'importe quel système pour lequel on peut régler un DSN (y compris
Access, SQL Serveur). L'Imports utilisé est System.Data.Odbc:
ChaîneConnexion = NomDuServeur ' Le nom du serveur est la chaîne DSN, exemple : "DSN=TestAccess;"
```

```
' DB pilotée par MicroSoft Jet (moteur Access). L'Imports utilisé est System.Data.OleDb :
ChaîneConnexion = "Provider=Microsoft.JET.OLEDB.4.0; Data Source=" & NomDBComplet & ";"
```

```
' DB pilotée par Oracle. L'Imports utilisé est System.Data.OracleClient :
' ChaîneConnexion = "...
```

```
' DB pilotée par SQL Serveur. L'Imports utilisé est System.Data.SqlClient :
ChaîneConnexion = "Data Source=" & NomDuServeur & "; Initial Catalog=" & NomDBComplet &
";Integrated Security=True;"
```

```
' DB SQLExpress en fichier attaché. L'Imports utilisé est System.Data.SqlClient :
ChaîneConnexion = "Data Source=" & NomDuServeur & "; AttachDbFilename=" & NomDBComplet &
";Integrated Security=True"
```

Il ne faut trop s'inquiéter de l'élaboration de la chaîne de connexion : il suffit de la lire dans les propriétés de la connexion.

Avant de réaliser un Recherche/Remplace, dans toute la solution, des noms des types utilisés, il faut s'assurer de ne pas avoir commis l'erreur de nommer ces procédures ...SQL... (juste pour que les noms des variables et des procédures restent identiques d'une version à l'autre de l'application ... mais là, ok ..., je pinaille ... :D)):

```
NomDuServeurSQL
OuvreSQLConnexion( . . . )
CreeSQLCommande( . . . )
```

Il fallait les nommer :

```
NomDuServeur
OuvreConnexion( . . . )
CreeCommande( . . . )
```

Dans l'application Stock, seuls les types suivants sont utilisés :

```
Dim MaConnexion As SqlConnection
Dim MaCommande As New SqlCommand
```

Mais beaucoup d'autres types peuvent être utiles dans d'autres applications :

```
Dim MaTransaction As SqlTransaction
Dim MonAdapteur As SqlDataAdapter
Dim MesParametres As SqlParameter
Dim MesDonnees As SqlDataReader
...
```

Tous ces noms de types sont identiques à l'exception des quelques premières lettres qui rappellent leurs origines, ou plus exactement qui désignent « l'import » d'où ils proviennent.

Le Recherche/Remplace à réaliser consiste à remplacer ces premières lettres. Par exemple, pour une DB Access, il faut faire le Recherche/Remplace de `Sql` par `OLEDB`. Tous les types « DB » du programme sont ensuite réglé pour une DB Access.

```
Dim MaConnexion As OleDbConnection
Dim MaCommande As New OleDbCommand
...
Dim MaTransaction As OleDbTransaction
Dim MonAdapteur As OleDbDataAdapter
Dim MesParametres As OleDbDataParameter
Dim MesDonnees As OleDbDataReader
```

Il est donc possible de migrer l'application d'un SGBD à un autre en très peu d'opérations :

- Placer l'Imports requis
- Régler la chaîne de connexion à partir de celle obtenue dans les propriétés de la connexion (par le Gestionnaire de serveurs)
- Affecter les variables `NomServeur` et `NomDBComple` (s'il n'est pas acquis en cours d'exécution)
- Faire le Recherche/Remplace approprié

Attention, il faut aussi que les codes SQL contenus dans les codes VB soient écrits en SQL « standard ». Ces codes ne doivent pas intégrer des ordres SQL qui ne sont pas gérés par tous les SGBD.

Il existe 4 lieux d'utilisation du langage SQL dans l'application (il n'y en existe que rarement plus) :

1. `ChargeDonnees()`
2. `SauveModif()`
3. `SauveAjout()`
4. `SauveSupprime()`

Il faut utiliser SQL « standard » :

Certains SGBD supportent les batteries de requêtes (tel SQL Serveur), d'autres pas (tel Access). La batterie de requêtes est l'écriture de plusieurs requêtes séparées par un point-virgule. La demande d'exécution porte sur toute la batterie en une seule opération.

Par exemple, la batterie suivante est permise sous SQL Serveur mais elle ne l'est pas sous Access. Traitée par un `DataReader`, elle permettrait d'alimenter une table des Articles au stock insuffisant et une table des Articles au stock excédentaire.

```
"SELECT * FROM TProduit WHERE Stock < StockMin; SELECT * FROM TProduit WHERE Stock > StockMax;"
```

Lorsqu'on veut pouvoir migrer d'un SGBD à un autre, il faut programmer la constitution de ces tables en plusieurs requêtes distinctes et exécutées distinctement.