ADO.NET: Le mode connecté

L'objet Connection

L'objet *Connection* sert à connecter physiquement un utilisateur à une source de données

Les classes d'accès en mode connecté à une base de données SQL Server sont situées dans l'espace de noms System.Data.SQLClient

SqlConnection maConnexion = new SqlConnection (maChaineDeConnexion); maConnexion.Open(); // Pour ouvrir la connexion

,,,

maConnexion.Close(); // Pour fermer la connexion; libère les ressources

IMPORTANT DE FERMER LA CONNEXION

La chaîne de connexion (1)

Authentification Windows:

```
maChaineDeConnexion = "Data Source=424-
SQL2017,5433;
```

Initial Catalog=NomDeLaBD; Integrated Security=True";

Data Source: 424-SQL2017,5433 si vous êtes au collège.

Initial Catalog: nom de la base de données

Integrated Security: vaut *true* si c'est la sécurité intégrée (c'est l'authentification *Windows* qui est utilisée).

La chaîne de connexion (2)

• Authentification *SQL Server*:

```
maChaineDeConnexion = "Data Source = 424-SQL2017,5433;
Initial Catalog = nomDeLaBD; User ID = nomUtilisateur;
Password = motDePasseUtilisateur ";
au 444 et 424 (Réseau rouge :Informatique)
```

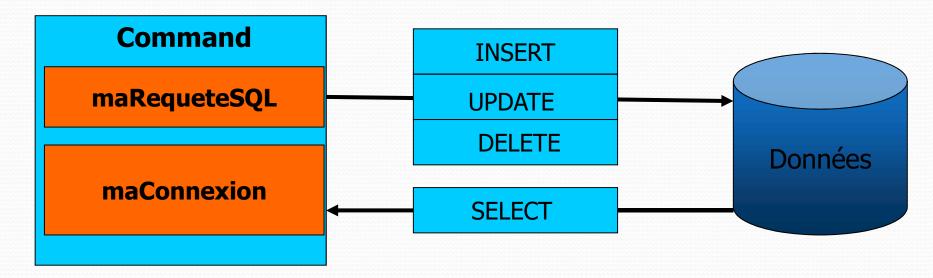
Data Source = tcp:424sql.cgodin.qc.ca,5433 À distance ou à partir du reseau bleu (interne)

L'objet Command

En mode connecté, l'objet *Command* sert à envoyer une commande (habituellement une requête SQL) à une source de données.

```
SqlConnection maConnexion = new SqlConnection (maChaineDeConnexion);
maConnexion.Open(); // Pour ouvrir la connexion
String maRequeteSQL = "maRequête"
SqlCommand maCommande= new SqlCommand(maRequeteSQL, maConnexion);
```

Pour exécuter une procédure stockée, il faut préciser le type de la commande en utilisant la propriété *CommandType* de l'objet *Command*.



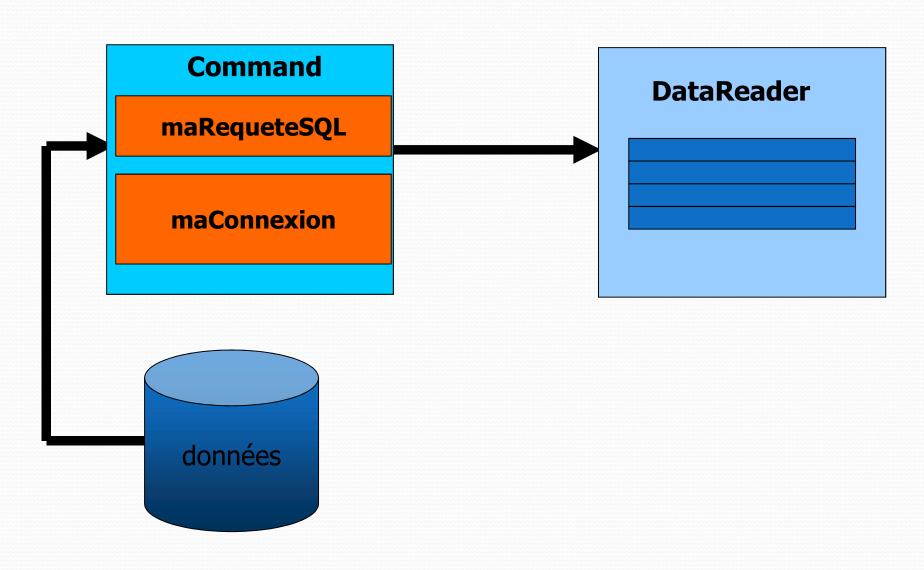
Méthodes de l'objet Command

- ExecuteNonQuery: Exécute une commande qui ne retourne pas de résultats en tant que tel (comme les requêtes *INSERT*, *UPDATE* ou *DELETE*). Cette méthode retourne le nombre d'enregistrements qui ont été affectés par la commande.
- ExecuteScalar: Exécute une commande qui retourne un seul résultat (comme la requête SELECT COUNT).
- ExecuteReader: Exécute une commande qui retourne plusieurs lignes de résultats (comme la requête SELECT habituelle). Chaque ligne est lue dans un objet de type DataReader qui permet de parcourir l'ensemble des lignes retournées par la commande.

L'objet DataReader

- Utilisé quand la commande retourne plusieurs lignes de résultats.
- Créé par l'exécution de la méthode ExecuteReader sur un objet Command.
- Permet un accès en lecture avant seulement
- Au point de départ, le curseur de l'objet DataReader est positionné avant la 1ère ligne de résultats. Pour que le curseur se positionne sur la ligne suivante, on utilise la méthode Read(). Cette méthode retourne true s'il y a une autre ligne à lire ou false dans le cas contraire.
- On peut lire chaque résultat situé sur une ligne à l'aide de la variable[indiceColonne ou nomColonne].
- A la fin, l'objet **DataReader** doit être fermé afin de libérer les ressources.

L'objet DataReader



• <u>Démonstration</u>