



# ADO.NET

*ActiveX Data Objects*

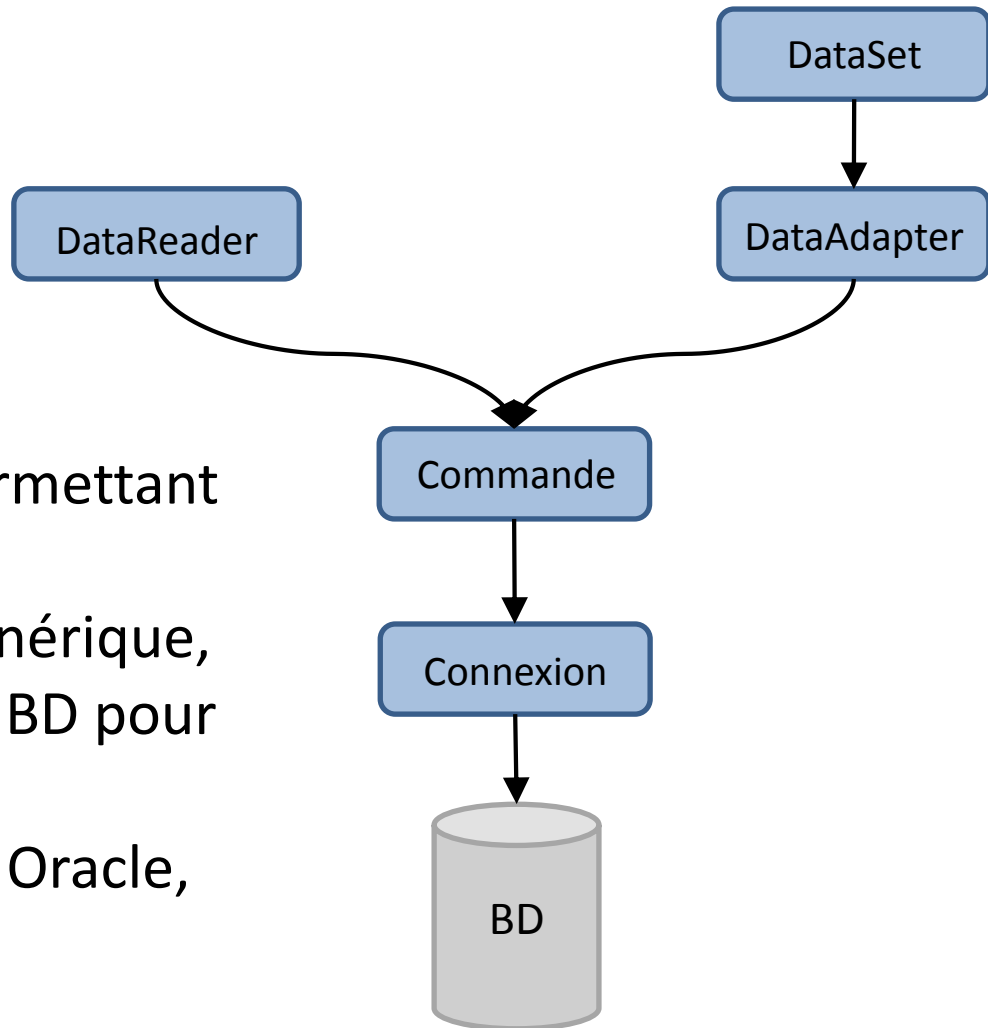


# Sources de données

- Bases de données
  - Sql Server
  - Oracle
  - ODBC
  - ...
- Excel
- Fichiers XML
- Fichiers texte...

# ADO.NET Data Providers

- Ensemble de classes permettant d'agir avec une BD
- **Aucun** Data Provider générique, spécifiques aux types de BD pour des optimisations
- Exemples: SQL, OLE DB, Oracle, ODBC





# Classes connectées (connection-based)

- Récupération/mise à jour des données directement dans la BD
  - Implémentées par les fournisseurs de données
  - Exemples:

Classe	Version Sql Server
Connection	SqlConnection
Command	SqlCommand
DataReader	SqlDataReader
DataAdapter	SqlDataAdapter

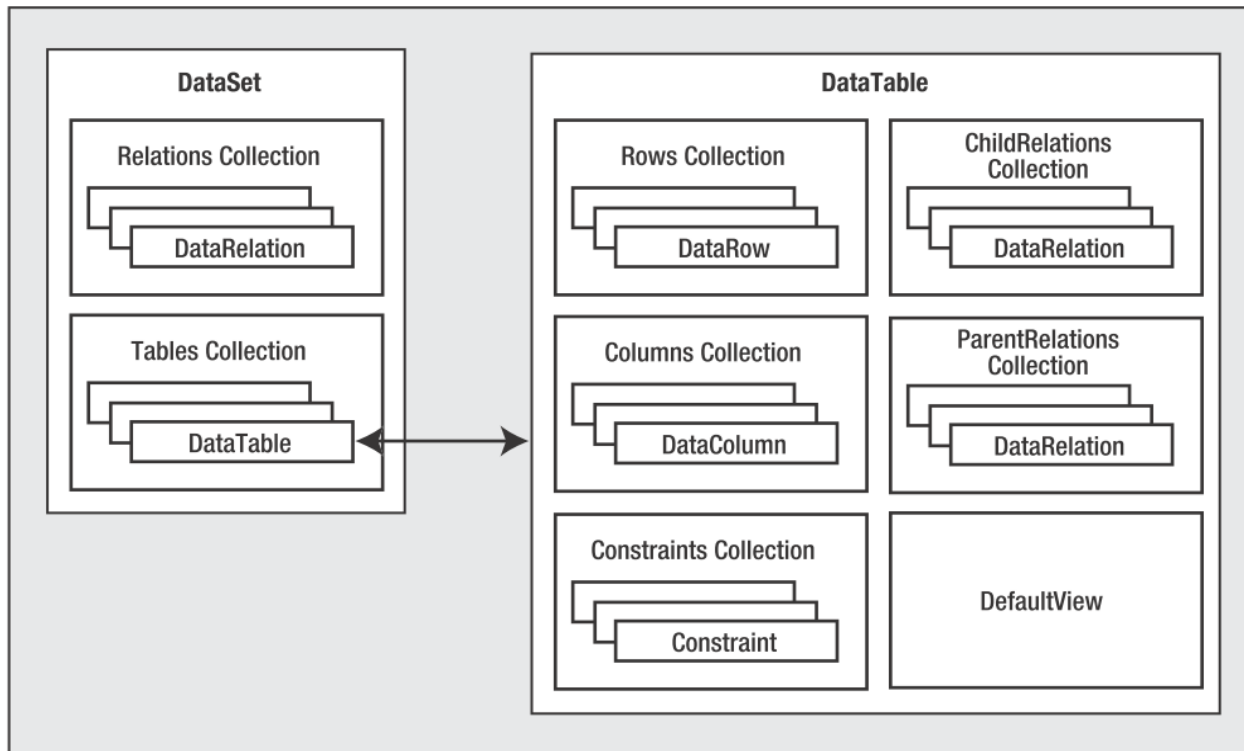


# Classes déconnectées (content-based)

- Récupération/mise à jour des données « offline »
- Synchronisation avec la BD en se servant des classes connectées
- Classes ADO.NET (implémentées par Microsoft)
- Indépendantes de la base de données
- Exemples:
  - DataSet (représentation dans la mémoire d'une BD)
  - DataTable
  - DataRow
  - DataColumn
  - ...

# DataSet

- Représentation en mémoire d'une base de données
- Rempli à l'aide d'un DataAdapter





# DataSet fortement typé

```
// DataSet dsFaible est faiblement typé  
int identifiant = dsFaible.Tables["Produits"].Rows[0]["idProduit"].ToInt32();  
  
// DataSet dsFort est fortement typé  
int identifiant = dsFort.Produits[0].idProduit;
```

## Avantages:

- Simplicité du code
- Utilisation de Intellisense



# LINQ





# Qu'est-ce que LINQ?

- Un modèle de programmation qui permet un accès unifié à n'importe quelle source de données.
- Implémentations disponibles:
  - LINQ to Objects
  - LINQ to ADO.NET
    - LINQ to Entities
    - LINQ to SQL
    - LINQ to DataSet
  - LINQ to XML



# Un exemple

```
var req =  
    from c in Clients  
    where c.Pays == "Italie"  
    select c.NomCompagnie;
```

La variable *req* contient le résultat de la requête, on peut parcourir les valeurs renvoyées:

```
foreach ( string nom in req ) {  
    Console.WriteLine( nom );  
}
```



## Un exemple (suite)

Où étaient stockées les données?

- `Client[] Clients;`
- `DataSet ds = recupDS();`  
`DataTable Clients = ds.Tables["Clients"];`
- `DataContext db = new DataContext(ConnectionString);`  
`Table<Client> Clients = db.GetTable<Client>();`
- ...



# DÉMO