



***LINQ***

*LINQ To SQL*

# ***LINQ To SQL***

- Le but de *LINQ To SQL* est de transformer les enregistrements de la base de données en objets de manière à ce que ceux-ci soient utilisables par *LINQ to Objects*.

# ***LINQ To SQL***

- Les objets créés par *LINQ to SQL* sont des entités.
  - Les tables sont transformées en classes d'objets (ou entités)
  - Les champs sont transformés en propriétés (ou en attributs)
  - Les objets de différents types sont associés entre eux (il existe des propriétés de navigation qui permettent de naviguer entre les objets)

# ***DataContext***

- Le ***DataContext*** est le lien entre la vraie base de données ***SQL*** et ***LINQ To SQL***.
- Il s'apparente au ***TableAdapter*** (dans un ***DataSet***).

# ***DataContext***

- Le ***DataContext*** joue trois (3) rôles importants:
  - Récupère, dans la base de données, les enregistrements de la table
  - Transforme ces enregistrements en entités (en objets) utilisables par *LINQ to Objects*
  - Enregistre, dans la vraie base de données, les entités qui ont été modifiées dans le ***DataContext***

# Rôle 1 du *DataContext*

- Récupère , à partir de la base de données, les enregistrements de la table

Par exemple, la requête *LINQ* suivante:

```
from unClient in monDataContext.client  
orderby unClient.cliNo  
select unClient;
```

# Rôle 1 du *DataContext*

Chaque fois qu'on exécute une requête *LINQ*, il y a une interaction avec la vraie base de données.

Les données lues sont placées directement dans le *DataContext*.

Par conséquent, les données lues reflètent toujours ce qu'il y a dans la vraie base de données au moment de l'exécution de la requête *LINQ*.

## Rôle 2 du *DataContext*

- Transforme les enregistrements en entités utilisables par *LINQ To Objects*

Par exemple, à la suite de la requête *SQL* précédente, les enregistrements issus de la requête *SQL* vont être transformés en entités et vont être stockés dans le *DataContext*.



## Rôle 2 du *DataContext*

- Il est important de comprendre, qu'avec *LINQ To SQL*, il n'existe pas de structures intermédiaires.
- Par exemple, lorsqu'on travaille avec un *Dataset*, il existe des structures intermédiaire: les *DataTables*. Les *DataTables* contiennent les données.
- Dans le cas de *Linq to SQL*, c'est la requête *LINQ* qui récupère les données. Il n'est donc pas nécessaire de créer des *DataTables*.

## Rôle 3 du *DataContext*

- Enregistre, dans la vraie base de données, les entités qui ont été modifiées dans le *DataContext*.

Par exemple:

```
var leClient1000 = from unClient in  
monDataContext.client  
where unClient.cliNo == 1000  
select unClient;  
foreach (unClient in leClient1000)  
    unClient.cliVille = "Montréal";  
monDataContext.SubmitChanges();
```

Avant d'enregistrer la modification, une requête UPDATE va être créée.