# LINQ & Entity Framework – Exercices

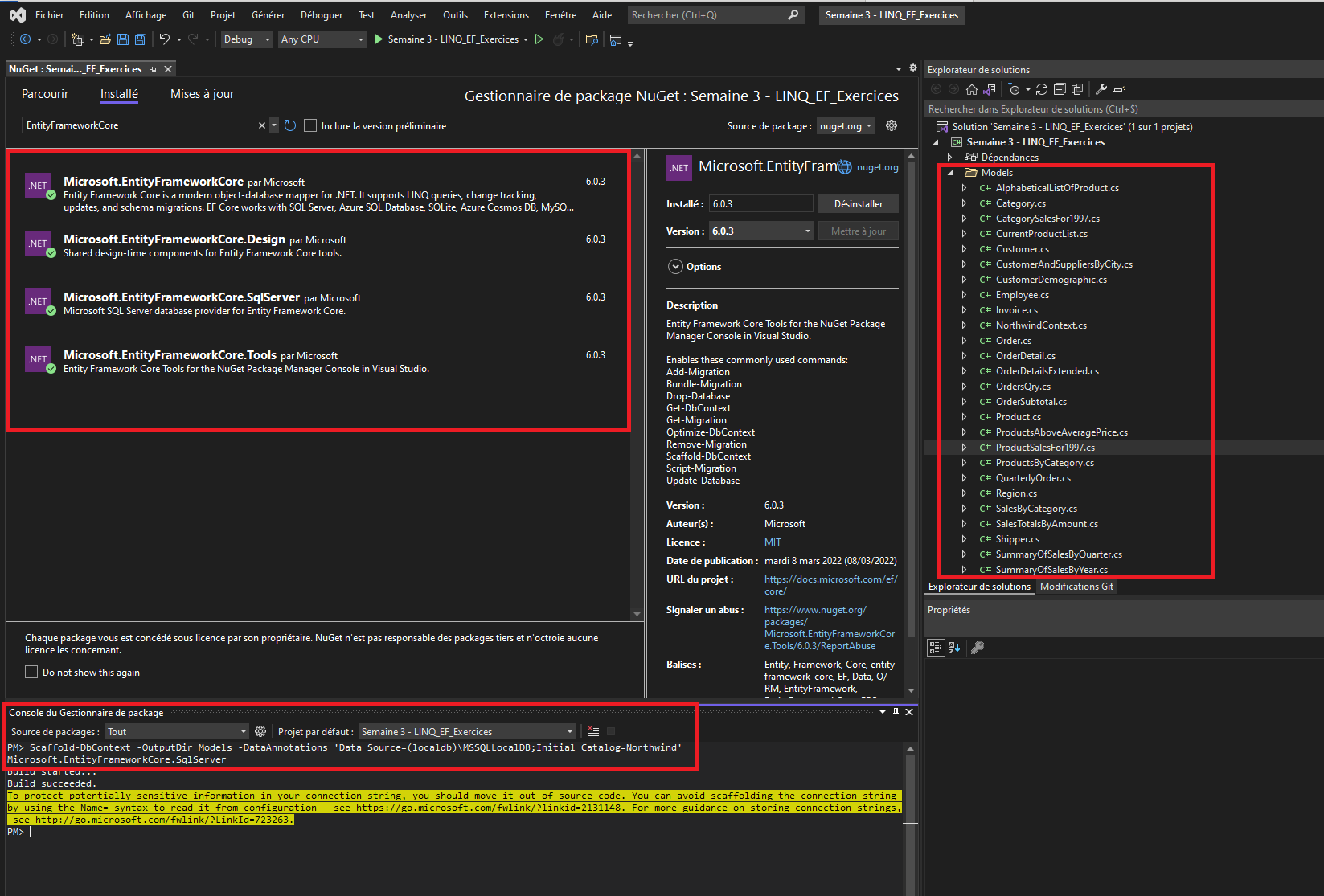
## Objectifs

* Créer un modèle à partir d’une base de données existante
* Manipuler la base de données au travers de LINQ (ajout, suppression, recherche)
* Utiliser les propriétés de navigation

## A – Création des entités (DAO) à partir de la base de données

1. Créer un projet Application Console
2. Créer la base de données Northwind à partir du fichier « Northwind4SqlServer.sql»
   1. Récupérer sur MooVin le fichier « Northwind4SQLServer.sql» dans les ressources
   2. Exécuter un query avec ce fichier SQL dans Visual Studio
      1. Outils –> SQL SERVER -> Nouvelle requête/query
      2. Copier-coller le code du fichier «Northwind4SQLServer.sql»
      3. Exécuter le code du query (bouton play vert en haut à gauche de la fenêtre query)
3. Voir la base de données Northwind
   1. Dans Visual Studio, vous pouvez manipuler les bases de données SQL SERVER via l’explorateur d’objets SQL SERVER
   2. Affichage -> Explorateur d’objets SQL SERVER
   3. Vous devez vous connecter à une instance SQL Server. Nous utiliserons toujours l’instance : **(localdb)\MSSQLLocaldb**
   4. Vérifier que vous avez bien une base de données Northwind et afficher les données d’une table
4. Ajouter les packages NuGet nécessaires pour EntityFramework
   1. Clic droit sur le projet -> Gérer les packages NuGet
   2. Ajouter les packages
      1. Microsoft.EntityFrameworkCore
      2. Microsoft.EntityFrameworkCore.Design
      3. Microsoft.EntityFrameworkCore.Tools
      4. Microsoft.EntityFrameworkCore.SqlServer
5. Créer les classes-entités depuis la DB
   1. **Dans une console du Package Manager (PM)**
      1. Outils – Gestionnaire de Package NuGet -> Console
   2. Lancez la commande (sur une seule ligne) :
6. **Scaffold-DbContext -OutputDir Models 'Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=Northwind' Microsoft.EntityFrameworkCore.SqlServer**

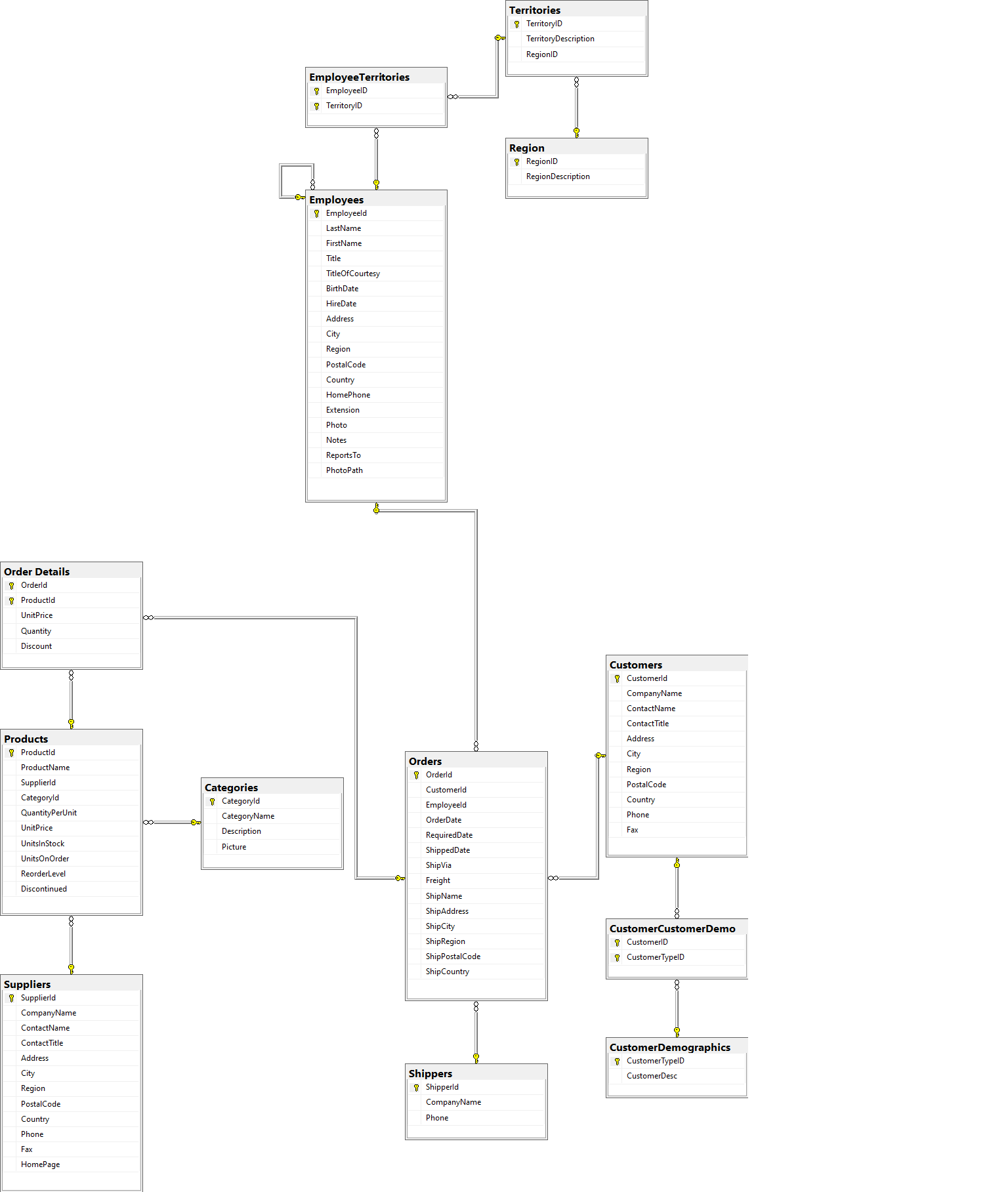
En image :



1. Regardez ce qui a été généré par la commande dans le dossier Models
   1. Regardez également les fichier C# générés
      1. Le fichier étendant DbContext (point d’entrée pour manipuler la DB)
      2. Les POCOS créés -> les entités correspondant aux tables
      3. Pq les classes sont- elles partials ?
      4. Pq les propriétés de navigation sont-elles virtual ?

## B – Queries

Un petit schéma de la DB :



Afin de pouvoir manipuler la base de données via l’Entity Framework, il vous faudra dans votre M*ain()* créez un objet du type NorthwindContext (la classe générée étendant DbContext). Le nom de cette classe peut être différent si vous avez changé le nom par défaut dans l’assistant. Essayez maintenant de réaliser les requêtes LINQ suivantes :

1. Activer le lazy loading avec utilisation d’un proxy
   1. Installer le package EntityFrameworkCore.Proxies
   2. Aller dans votre fichier DbContext Nortwind généré lors de l’étape A.
   3. Modifier la méthode OnConfiguring comme ceci :

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

if (!optionsBuilder.IsConfigured)

{

optionsBuilder.UseSqlServer("Data Source=(localdb)\\MSSQLLocalDB;Initial Catalog=Northwind;**MultipleActiveResultSets=True**")

**.UseLazyLoadingProxies()**

**.LogTo(Console.WriteLine, LogLevel.Information)**

**.EnableSensitiveDataLogging();**

}

}

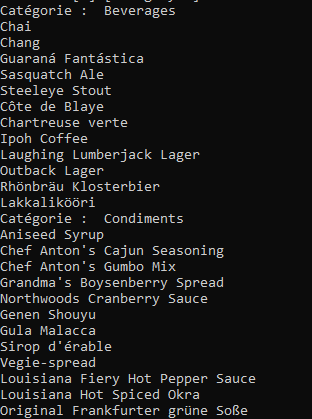
1. Lister tous les *Customers* habitants dans une ville saisie au clavier.

Une image contenant texte

Description générée automatiquement

1. Afficher les produits de la catégorie Beverages et Condiments. Utilisez **le lazy loading**  (pas d’include !)

Résultat attendu :



1. Afficher les produits de la catégorie Beverages et Condiments. Utilisez **le eager loading** ! (avec include). Le résultat est identique à la requête précédente.
   1. Désactiver le lazy loading -> retirer .UseLazyLoadingProxies()
   2. Remarquez que vous avez maintenant un seul select !
2. Donnez pour un client donné saisi au clavier (LILAS par ex) la liste de ces commandes (de la plus récente à la plus ancienne) et qui ont été livrées ( il y a une date de livraison). Les champs renvoyés par ce query sont le ID du client «CustomerID », la date de la commande « OrderDate » et la date de livraison « ShippedDate ».

Une image contenant texte

Description générée automatiquement

1. Afficher le total des ventes par produit (ID produit -> Total) trié par ordre de numéro produit.

Une image contenant texte

Description générée automatiquement

…..

1. Afficher tous les employés (leur nom) qui ont sous leur responsabilité la région « Western »

Une image contenant texte, périphérique, sombre, ensemble

Description générée automatiquement

1. Quels sont les territoires gérés par le supérieur de « Suyama Michael »

Une image contenant texte

Description générée automatiquement

## C – Updates

1. Mettez en majuscule le nom de tous les clients
2. Vérifiez que l’update a bien été réalisé en faisant un query qui affiche les noms des clients

## D – Insert

1. Ajoutez une catégorie à partir d’un nom saisi au clavier
2. Vérifiez que l’ajout a bien été effectué en DB

## E – Deletes

1. Supprimez la catégorie ajoutée à l’exercice précédent
2. Vérifiez que la suppression a bien été faite en DB
3. Supprimez un employé et réassignez toutes ses commandes (« orders ») à un autre employé. Vous demanderez les «id » des employés au clavier. Essayez de supprimer DAVOLIO (employeeId ==1) et de réassigner ses commandes à FULLER (employeeId ==2)
4. Vérifiez