

Procédure pas à pas : Manipulation de données (C#)

<https://docs.microsoft.com/fr-fr/dotnet/framework/data/adonet/sql/linq/walkthrough-manipulating-data-csharp>

Cette procédure pas à pas fournit un scénario LINQ to SQL complet essentiel pour l'ajout, la modification et la suppression de données dans une base de données. Vous utiliserez une copie de l'exemple de base de données Northwind pour ajouter un client, modifier le nom d'un client et supprimer une commande.

Notes

Il est possible que pour certains des éléments de l'interface utilisateur de Visual Studio, votre ordinateur affiche des noms ou des emplacements différents de ceux indiqués dans les instructions suivantes. L'édition de Visual Studio dont vous disposez et les paramètres que vous utilisez déterminent ces éléments. Pour plus d'informations, consultez [Personnalisation de l'IDE](#).

Cette procédure pas à pas a été écrite à l'aide des paramètres de développement Visual C#.

Prérequis

Elle requiert les éléments suivants :

- Les fichiers sont stockés dans un dossier dédié, c:\linqtest6. Vous devez créer ce dossier avant de commencer la procédure pas à pas.
- Exemple de base de données Northwind.

Si cette base de données n'est pas disponible sur votre ordinateur de développement, vous pouvez la télécharger à partir du site de téléchargement Microsoft. Pour obtenir des instructions, consultez [téléchargement d'exemples de bases de données](#). Après avoir téléchargé la base de données, copiez le fichier northwnd.mdf dans le dossier c:\linqtest6.

- Fichier de code C# généré à partir de la base de données Northwind.

Vous pouvez générer ce fichier à l'aide de l'Concepteur Objet Relationnel ou de l'outil SQLMetal. Cette procédure pas à pas a été écrite à l'aide de l'outil SQLMetal, avec la ligne de commande suivante :

```
sqlmetal /code:"c:\linqtest6\northwind.cs" /language:csharp  
"C:\linqtest6\northwnd.mdf" /pluralize
```

Pour plus d'informations, consultez [SqlMetal.exe \(outil de génération de code\)](#).

Présentation

Cette procédure pas à pas se compose de six tâches principales :

- Création de LINQ to SQL la solution dans Visual Studio.
- Ajout du fichier de code de base de données au projet.
- Création d'un objet représentant un client.
- Modification du nom de contact d'un client.
- Suppression d'une commande.
- Soumission de ces modifications à la base de données Northwind.

Création d'une solution LINQ to SQL

Dans cette première tâche, vous allez créer une solution Visual Studio qui contient les références nécessaires pour générer et exécuter LINQ to SQL un projet.

Pour créer une solution LINQ to SQL

1. Dans le menu **fichier** de Visual Studio, pointez sur **nouveau**, puis cliquez sur **projet**.
2. Dans le volet **types de projets** de la boîte de dialogue **nouveau projet**, cliquez sur **visuel C#**.
3. Dans le volet **Modèles**, cliquez sur **Application console**.
4. Dans la zone **nom**, tapez **LinqDataManipulationApp**.
5. Dans la zone **emplacement**, vérifiez l'emplacement où vous souhaitez stocker vos fichiers projet.
6. Cliquez sur **OK**.

Ajout de références et de directives LINQ

Cette procédure pas à pas utilise des assemblies qui ne sont pas nécessairement installés par défaut dans votre projet. Si System.Data.Linq n'est pas répertorié comme une référence dans votre projet, ajoutez-le comme indiqué dans les étapes suivantes :

Pour ajouter System.Data.Linq

1. Dans **Explorateur de solutions**, cliquez avec le bouton droit sur **références**, puis cliquez sur **Ajouter une référence**.
2. Dans la boîte de dialogue **Ajouter une référence**, cliquez sur **.net**, sur l'assembly System. Data. Linq, puis sur **OK**.

L'assembly est ajouté au projet.

3. Ajoutez les directives suivantes au haut de Program.cs :

```
C#Copier
using System.Data.Linq;
using System.Data.Linq.Mapping;
```

Ajout du fichier de code Northwind au projet

Ces étapes supposent que vous avez utilisé l'outil SQLMetal pour générer un fichier de code à partir de l'exemple de base de données Northwind. Pour plus d'informations, consultez la section Composants requis au début de cette procédure pas à pas.

Pour ajouter le fichier de code Northwind au projet

1. Dans le menu **projet** , cliquez sur **Ajouter un élément existant**.
2. Dans la boîte de dialogue **Ajouter un élément existant** , accédez à `c:\lingtest6\northwind.cs`, puis cliquez sur **Ajouter**.

Le fichier northwind.cs est ajouté au projet.

Paramétrage de la connexion de base de données

Commencez par tester votre connexion à la base de données. Notez en particulier que le nom de la base de données (Northwnd) ne comporte pas de caractère i. Si vous générez des erreurs au cours des étapes suivantes, examinez le fichier northwind.cs pour déterminer comment la classe partielle Northwind est orthographiée.

Pour paramétrer et tester la connexion de base de données

1. Tapez ou collez le code suivant dans la méthode `Main` de la classe `Program` :

```
C#Copier
// Use the following connection string.
Northwnd db = new Northwnd(@"c:\lingtest6\northwnd.mdf");

// Keep the console window open after activity stops.
Console.ReadLine();
```

2. Appuyez sur F5 pour tester l'application à ce stade.

Une fenêtre de **console** s'ouvre.

Vous pouvez fermer l'application en appuyant sur entrée dans la fenêtre de **console** , ou en cliquant sur **arrêter le débogage** dans le menu **Déboguer** de Visual Studio.

Création d'une entité

La création d'une entité est une opération simple. Vous pouvez créer des objets (Customer, par exemple) à l'aide du mot clé `new`.

Dans cette section et les suivantes, vous apporterez des modifications uniquement au cache local. Aucune modification n'est envoyée à la base de données tant que vous n'avez pas appelé [SubmitChanges](#) vers la fin de cette procédure pas à pas.

Pour ajouter un nouvel objet d'entité Customer

1. Créez un Customer en ajoutant le code suivant avant `Console.ReadLine()`; dans la méthode `Main` :

C#Copier

```
// Create the new Customer object.
Customer newCust = new Customer();
newCust.CompanyName = "AdventureWorks Cafe";
newCust.CustomerID = "ADVCA";

// Add the customer to the Customers table.
db.Customers.InsertOnSubmit(newCust);

Console.WriteLine("\nCustomers matching CA before insert");

foreach (var c in db.Customers.Where(cust => cust.CustomerID.Contains("CA")))
{
    Console.WriteLine("{0}, {1}, {2}",
        c.CustomerID, c.CompanyName, c.Orders.Count);
}
```

2. Appuyez sur F5 pour déboguer la solution.
3. Appuyez sur entrée dans la fenêtre de **console** pour arrêter le débogage et poursuivez la procédure pas à pas.

Mise à jour d'une entité

Au cours des étapes suivantes, vous allez récupérer un objet Customer et modifier l'une de ses propriétés.

Pour modifier le nom d'un client

- Ajoutez le code suivant au-dessus de `Console.ReadLine()` :

C#Copier

```
// Query for specific customer.
// First() returns one object rather than a collection.
var existingCust =
    (from c in db.Customers
     where c.CustomerID == "ALFKI"
```

```
select c)
.First();

// Change the contact name of the customer.
existingCust.ContactName = "New Contact";
```

Suppression d'une entité

Vous pouvez supprimer la première commande à l'aide du même objet Customer.

Le code suivant montre comment couper les relations entre les lignes et supprimer une ligne de la base de données. Ajoutez le code suivant avant `Console.ReadLine` pour voir comment les objets peuvent être supprimés :

Pour supprimer une ligne

- Ajoutez le code suivant juste au-dessus de `Console.ReadLine();` :

```
C#Copier
// Access the first element in the Orders collection.
Order ord0 = existingCust.Orders[0];

// Access the first element in the OrderDetails collection.
OrderDetail detail0 = ord0.OrderDetails[0];

// Display the order to be deleted.
Console.WriteLine
    ("The Order Detail to be deleted is: OrderID = {0}, ProductID = {1}",
    detail0.OrderID, detail0.ProductID);

// Mark the Order Detail row for deletion from the database.
db.OrderDetails.DeleteOnSubmit(detail0);
```

Soumission des modifications à la base de données

La dernière étape requise pour la création, la mise à jour et la suppression d'objets consiste à soumettre les modifications à la base de données. Sans cette étape, vos modifications restent locales et n'apparaissent pas dans les résultats de requêtes.

Pour soumettre les modifications à la base de données

1. Insérez le code suivant juste au-dessus de `Console.ReadLine` :

```
C#Copier
db.SubmitChanges();
```

2. Insérez le code suivant (après `SubmitChanges`) pour afficher les effets avant/après de la soumission des modifications :

```
C#Copier
```

```
Console.WriteLine("\nCustomers matching CA after update");
foreach (var c in db.Customers.Where(cust =>
    cust.CustomerID.Contains("CA")))
{
    Console.WriteLine("{0}, {1}, {2}",
        c.CustomerID, c.CompanyName, c.Orders.Count);
}
```

3. Appuyez sur F5 pour déboguer la solution.
4. Appuyez sur entrée dans la fenêtre de **console** pour fermer l'application.

Notes

Une fois que vous avez soumis les modifications (ajouté le nouveau client), vous ne pouvez plus exécuter cette solution telle quelle. Pour l'exécuter à nouveau, modifiez le nom du client et l'ID client à ajouter.