

Worksheet – Advanced LINQ

Lernziel:

Sie können komplexe Abfragen mit LINQ formulieren. Sie kennen grundlegenden Konzepte von .NET Lambdas und LINQ erklären und anwenden.

Aufgabe 1 – LINQ

Sie finden auf dem Netzwerkshare das Projekt LinqQuerySandbox. Implementieren Sie darin folgende LINQ-Queries:

- a) Überprüfen Sie mit Hilfe der LINQ-Methode Any, ob eine der Personen zufälligerweise Ihren Vornamen hat und schreiben Sie das Resultat auf die Konsole.
- b) Erstellen Sie mit Hilfe der LINQ-Methode SelectMany eine (flache) Liste aller Autos und zählen Sie die Anzahl Autos mit der LINQ-Methode Count.
- c) Überprüfen Sie, ob alle Personen mindestens ein Auto besitzen (All).
- d) Geben Sie alle von Managern gefahrene Automarken auf die Konsole aus (OfType, SelectMany).
- e) Geben Sie den ersten Buchstaben aller Vornamen aus, ohne einen Buchstaben doppelt aufzulisten (Select, Distinct).
- f) Geben Sie alle Personen aus, sortiert nach Vornamen, Nachnamen und Anzahl Autos.

Aufgabe 2 – More LINQ

- a) Gruppieren Sie die Personen mit Hilfe der LINQ-Methode `GroupBy` nach Vornamen und schreiben Sie auf die Konsole, wie oft welcher Vornamen vorkommt. Sortieren Sie diese Liste alphabetisch nach Vornamen (`OrderBy`).
- b) Bestimmen Sie die Nachnamen aller Personen, die mit Vornamen „Bill“ heissen (`Where`). Bestimmen Sie den alphabetisch ersten Nachnamen dieser Personen (`First` und `OrderBy`).
- c) Berechnen Sie, wie viele Autos eine Person durchschnittlich besitzt (`Average`, `Count`)

Schreiben Sie die häufigste Auto-Farbe auf die Konsole.

- d) Implementieren Sie eine eigene Kopie der LINQ-Methode `Select` als `AwesomeSelect` neu. Diese Extension-Method soll auf beliebige `IEnumerable<T>` angewendet werden können und eine neue Folge produzieren, die alle Elemente in transformierter Form enthält. Die Transformation soll als Lambda-Ausdruck übergeben werden können. Verwenden Sie in Ihrer Implementation `yield`.

Testen Sie Ihre Erweiterung, indem Sie einige Werte mit π multiplizieren:

```
int[] numbers = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
var result = numbers.AwesomeSelect(n => n *  $\pi$ ).ToArray();
```

Aufgabe 3 – Query Syntax

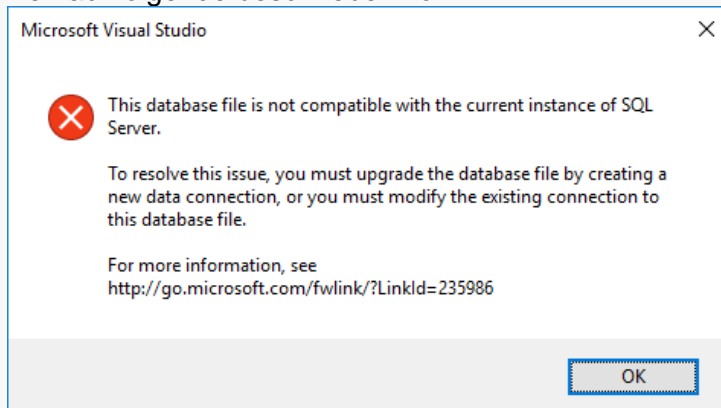
- a) Was ist der Unterschied zwischen der „*query syntax*“ und der „*method syntax*“? Was hat dies für Auswirkungen auf die Ausführung der Queries?
- b) Was versteht man unter den Konzepten der „*local queries*“ und der „*interpreted queries*“? Wann kommt welches Konzept typischerweise zum Einsatz?
- c) Implementieren Sie die Abfragen aus der ersten Aufgabe in Query Syntax.

Aufgabe 4 – LINQ to SQL

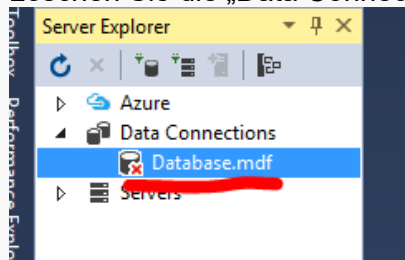
Sie finden auf dem Netzwerkshare das LINQtoSQLExample. Das Projekt enthält bereits die SQL-Datenbank Database.mdf.

Troubleshooting.

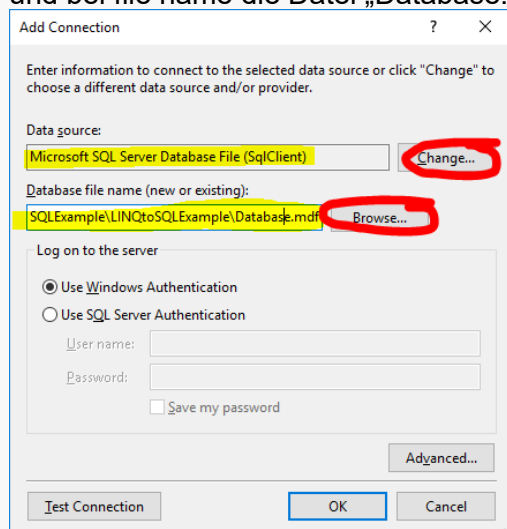
Sollten Sie die nachfolgend sichtbare Fehlermeldung angezeigt bekommen, gehen Sie bitte wie nachfolgende beschrieben vor:



1. Löschen Sie die „Data Connection“ zu der Datenbank aus dem Server Explorer



2. Fügen Sie eine neue Connection hinzu (Rechtsklick auf DataConnections, „Add Connection“).
3. Stellen Sie sicher dass bei Data source „Microsoft SQL Server Database File (SqlClient)“ und bei file name die Datei „Database.mdf“ ausgewählt ist.



Bestätigen Sie mit „OK“

Führen Sie nun die folgenden Schritte durch:

1. Stellen Sie sicher, dass der „Server Explorer“ angezeigt wird. (View -> Server Explorer)
2. Öffnen Sie die Datenbank mit einem Doppelklick auf Database.mdf.
3. Betrachten Sie den Inhalt der Tabelle Persons (Rechtsklick, „Show Table Data“)

Nun können Sie sich um den Datenbank-Zugriff mittels LINQ-to-SQL kümmern:

4. Führen Sie auf der Solution Restore NuGet Packages aus.
5. Weisen Sie der Variable `absolutePathToDb` Ihren Pfad zur Datei Database.mdf im ausführbaren Projektordner zu.
6. Starten Sie die Applikation und prüfen Sie ob der Datenbankzugriff läuft. Auf der Konsole müsste irgendwo folgender Block erscheinen:

```
Context 'Context' started tracking 'Persons' entity. Consider using 'DbContextOptions'
to see key values.
-----
Martin Kropp
Context 'Context' started tracking 'Persons' entity. Consider using 'DbContextOptions'
to see key values.
-----
Michael Schnyder
Context 'Context' started tracking 'Persons' entity. Consider using 'DbContextOptions'
to see key values.
-----
Simon Felix
Context 'Context' started tracking 'Persons' entity. Consider using 'DbContextOptions'
to see key values.
-----
Romeo Montague
Context 'Context' started tracking 'Persons' entity. Consider using 'DbContextOptions'
to see key values.
-----
Juliet Capulet
Context 'Context' started tracking 'Persons' entity. Consider using 'DbContextOptions'
to see key values.
-----
```

7. Erweitern Sie die Applikation so, dass der Name des Partners auf derselben Zeile ausgegeben wird.
8. Versuchen Sie herauszufinden, welche SQL-Abfragen LINQ-to-SQL hinter den Kulissen generiert.
9. Versuchen Sie zu jeder Person eine Liste der zugeordneten Autos auszugeben.
Hinweis: mit `Include(c => c.Cars)` können Sie die Cars einer Personenliste nachladen.
10. Versuchen Sie alle Personen zu finden, deren Vorname mit "Ma" beginnt. Verwenden Sie dazu die Methode `string.StartsWith()`. Welchen SQL-Code generiert LINQ-to-SQL dafür? Was passiert, wenn Sie als zweiten Parameter `StringComparison.InvariantCultureIgnoreCase` mitgeben? Weshalb?
11. Welche Werte gibt der folgende Code aus? Weshalb?

```
using (var dbx = new Context(_dbOptions.Options))
{
    var p = dbx.Persons.First();
    p.LastName = "Test";
    Console.WriteLine($"Inside same data context: {p.LastName}");
}
```

```
using (var dbx = new Context(_dbOptions.Options))
{
    var p = dbx.Persons.First();
    Console.WriteLine($"From another data context: {p.LastName}");
}
```

Versuchen Sie das Problem mit `dbx.SaveChanges();` zu beheben.

A14.

1. Quit devenv
2. Launch **Visual Studio Installer**
3. Modify VS2017

Visual Studio

Products

Installed



Visual Studio Enterprise 2017

Microsoft DevOps solution for productivity and coordination across teams of any size

[Release notes](#)

(15.0.26228.9)

Update

Launch



Repair

Uninstall

Modify

Available

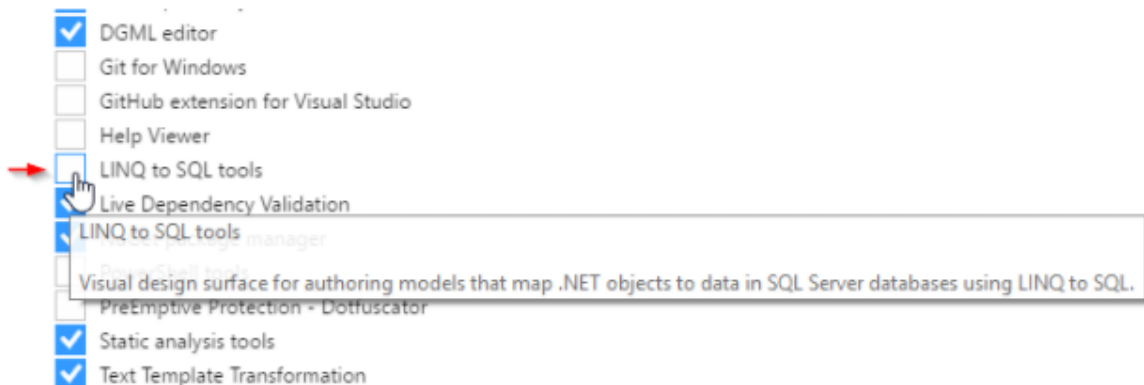
4. Click on the **Individual components** tab and scroll down to **LINQ to SQL tools** (within **Code tools** group.)

Modifying - Visual Studio Enterprise 2017 (15.0.26228.9)

Workloads

Individual components

Language packs



5. Click **Modify**.
6. Launch devenv again and you should be able to open **.dbml** in the designer we all love to hate.