62T01 Grundlæggende Objektorienteret Programmering

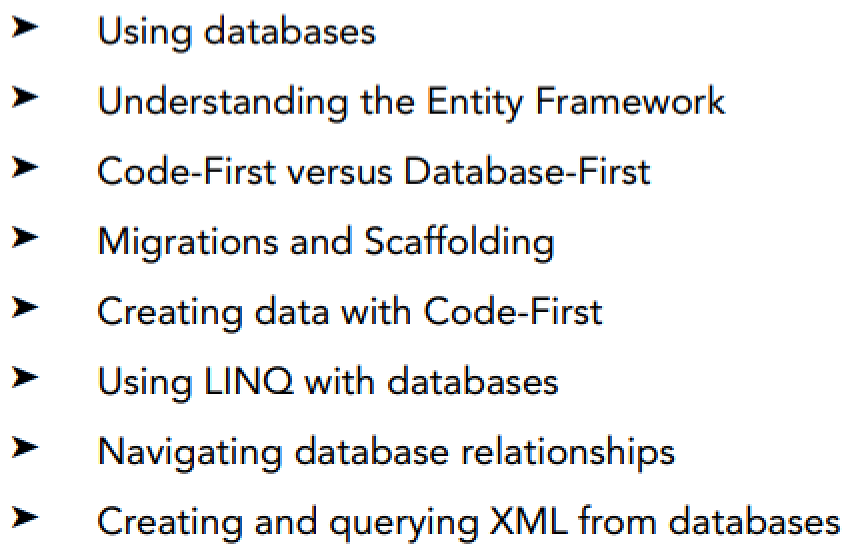
Kap. 17

Databaser

- Oversigt:
  - Connected layer
  - Disconnected layer
  - EF Core (Entity Framework)
    - Nu: Kun ”code-first”

Databaser

- Databaser



- ➤ Using databases
- ➤ Understanding the Entity Framework
- ➤ Code-First versus Database-First
- ➤ Migrations and Scaffolding
- ➤ Creating data with Code-First
- ➤ Using LINQ with databases
- ➤ Navigating database relationships
- ➤ Creating and querying XML from databases

Databaser

- Databaser
  - Er persistent data
    - (ligesom filer og serialisering af objekter)
  - ER PROFFESIONELT
  - Benytter SQL (Structured Query Language)
    - Benytter tabeller med rækker og kolonner
    - ER Model (Entity – Relationship model)
    - Entiteter er en abstraktion af REAL Data (kunde, ordrer, osv..)
      - Men hov! Kender vi det et andet sted fra?

Databaser

- Databaser
  - At arbejde med SQL kan kræve professionel indsigt
    - Ind fra højre kommer:
    - Entity Framework (EF):
    - Giver en C# OBJEKT abstraktion ovenpå databasen
      - (Det betyder dog IKKE, at der ikke er behov for databasespecialister, men at en del mindre komplicerede lagringsopgaver kan løses med EF)

Databaser

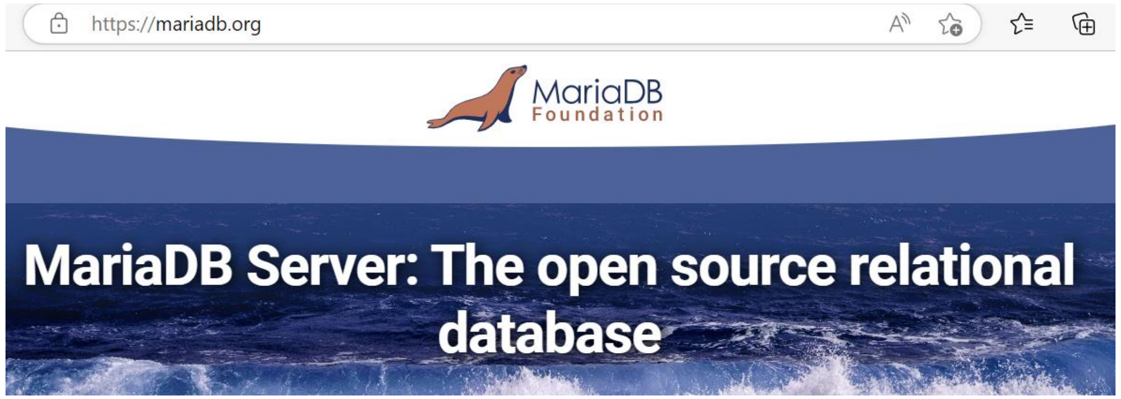
- Databaser
  - At arbejde med SQL kan kræve professionel indsigt
    - Ind fra højre kommer:
    - Entity Framework (EF):
    - Giver en C# OBJEKT abstraktion ovenpå databasen
      - (Det betyder dog IKKE, at der ikke er behov for databasespecialister, men at en del mindre komplicerede lagringsopgaver kan løses med EF)

Databaser

- Databaser
  - At arbejde med SQL kan kræve professionel indsigt
    - INFO: Normalt vil man normalisere tabellerne
    - Formål:
    - At få mindre datamængder og hurtigere søgninger
    - Undgå redundans og inkonsistens i data
    - Læs f.eks. MySQL normalisering (vidas.dk)

Databaser

- Databaser
  - Database providere
    - SQLServer (Microsoft)
    - Oracle
    - Sybase
    - MySql (Oracle)
    - MariaDB (Helt gratis open source)
    - Med flere..



MariaDB Server: The open source relational database

Databaser

- Databaser
  - Entity Framework
  - ORM (Object Relational Mapping)
    - Mapper Objekter til tabeller på databaser
    - Properties (C#) mapper til data i tabeller

Databaser

- Databaser
  - Entity Framework
    - Code first princippet
      - Skab en klasse med automatiske properties -> Tabel ved hjælp af EF
      - Håndtering af alle 4 (5) operationer:

      - INSERT INTO
      - UPDATE
      - DELETE
      - SELECT FROM

      - + CREATE TABLE

Databaser

- Databaser
  - Entity Framework
    - Code first princippet
      - Skab en klasse med automatiske properties -> Tabel ved hjælp af EF
      - EF klarer håndtering af alle 4 (5) operationer:

      - INSERT INTO
      - UPDATE
      - DELETE
      - SELECT FROM

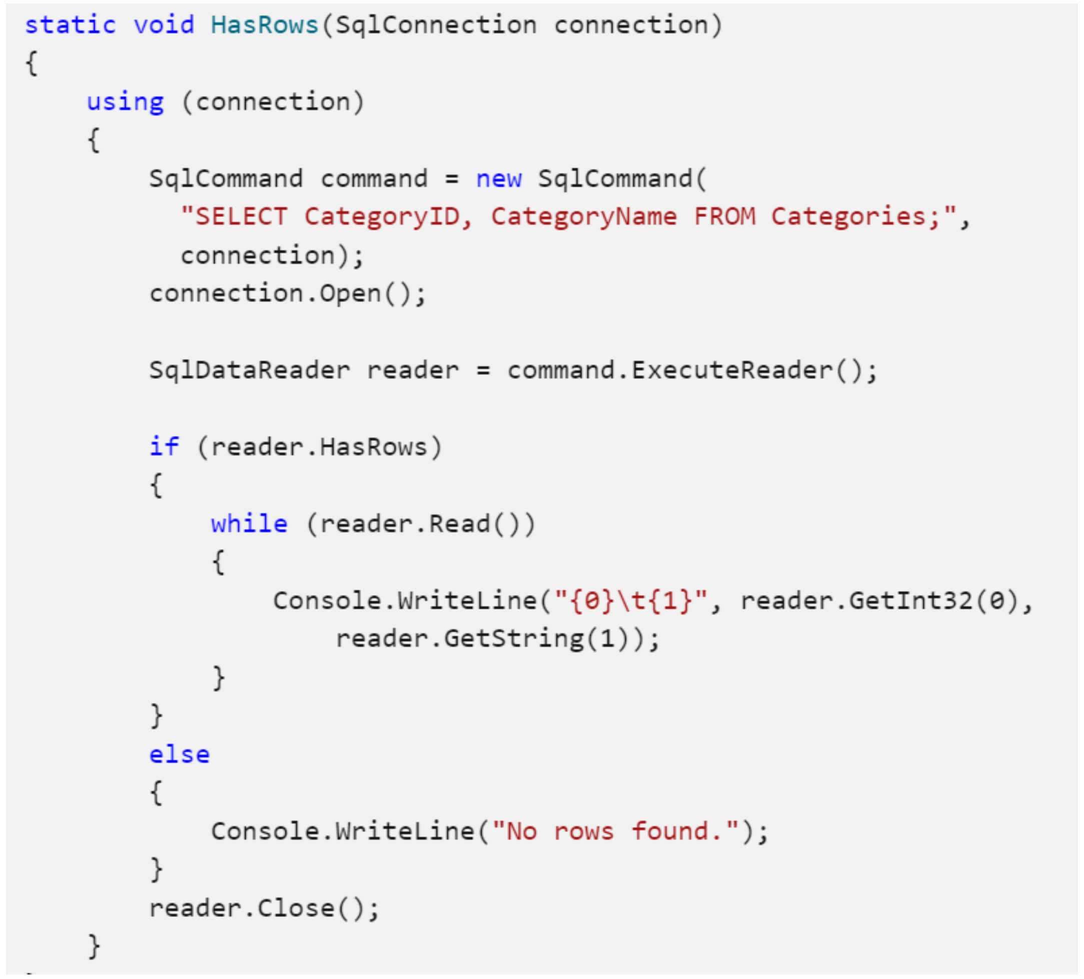
      - + CREATE TABLE

Databaser

- Databaser
  - Entity Framework
    - Database first princippet
      - Opret forbindelse til databasen (Connectionstring)
      - Udfør operationer
      - INSERT INTO, UPDATE, DELETE (ExecuteNonQuery)
      - SELECT FROM (ExecuteReader)

Databaser

- Databaser
  - Entity Framework
    - Database first princippet



```
static void HasRows(SqlConnection connection)
{
    using (connection)
    {
        SqlCommand command = new SqlCommand(
          "SELECT CategoryID, CategoryName FROM Categories;",
          connection);
        connection.Open();

        SqlDataReader reader = command.ExecuteReader();

        if (reader.HasRows)
        {
            while (reader.Read())
            {
                Console.WriteLine("{0}\t{1}", reader.GetInt32(0),
                    reader.GetString(1));
            }
        }
        else
        {
            Console.WriteLine("No rows found.");
        }
        reader.Close();
    }
```

Databaser

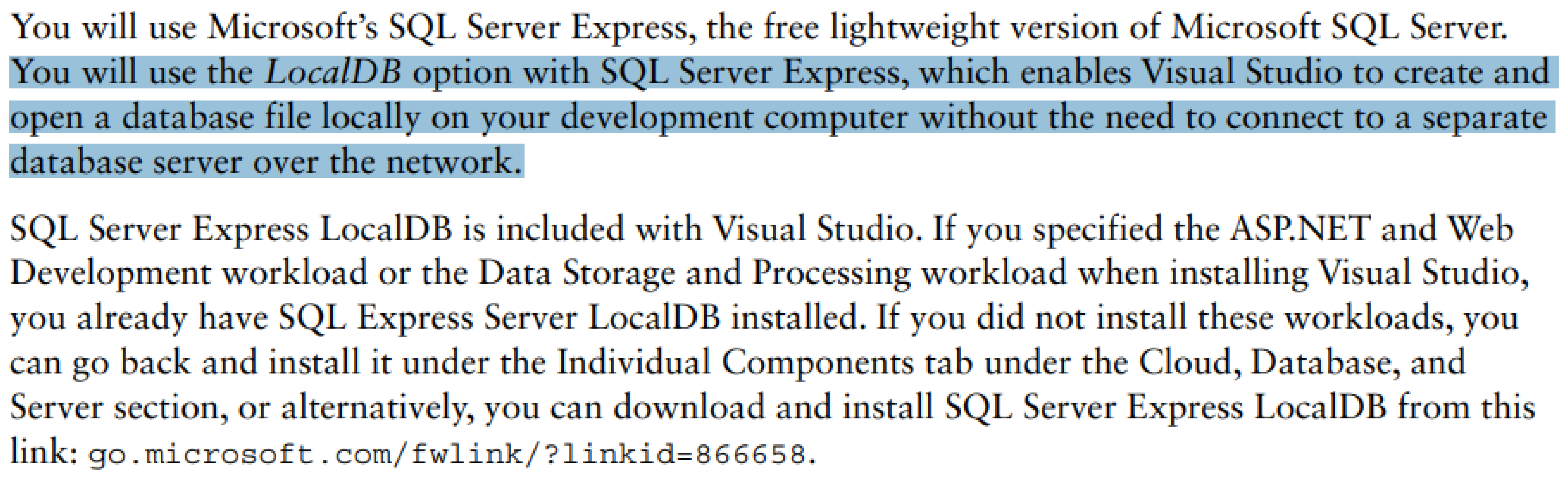
- Databaser
  - KONCEPT: Migrations (Code first)
    - At skabe klasser, der skaber database OBJEKTER udfra C# klasser
    - Efter en initial migrering, skal der, hver gang der skabes ændringer i database mapped klasser, udføres en NY MIGRATION for at opdatere databasen(ellers er tabelgrundlaget ikke overført fra C# klasserne)
    - EF (og Visual Studio) har et CLI (Command Line Interface) værktøj til dette

Databaser

- Databaser
  - KONCEPT: Scaffolding (Database first)
    - Det omvendte værktøj: Kigger ned i databasen og skaber de nødvendige C# klasser, der benyttes i EF

Databaser

- Databaser
  - Vi skal have en database – lokal DB
  - Er en del af Visual Studio

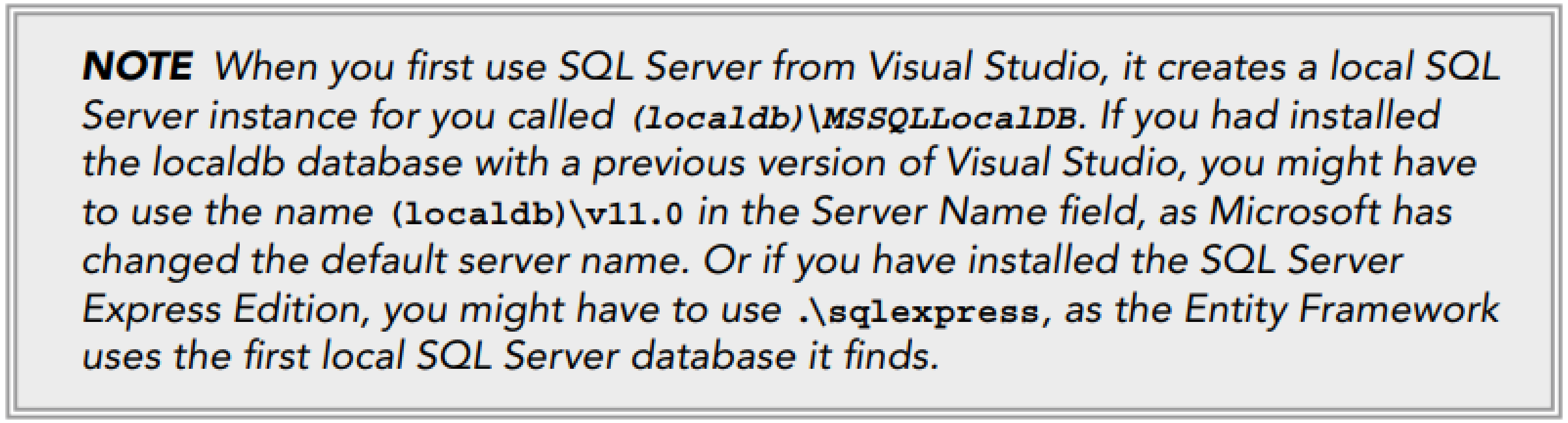


You will use Microsoft's SQL Server Express, the free lightweight version of Microsoft SQL Server. You will use the *LocalDB* option with SQL Server Express, which enables Visual Studio to create and open a database file locally on your development computer without the need to connect to a separate database server over the network.

SQL Server Express LocalDB is included with Visual Studio. If you specified the ASP.NET and Web Development workload or the Data Storage and Processing workload when installing Visual Studio, you already have SQL Express Server LocalDB installed. If you did not install these workloads, you can go back and install it under the Individual Components tab under the Cloud, Database, and Server section, or alternatively, you can download and install SQL Server Express LocalDB from this link: `go.microsoft.com/fwlink/?linkid=866658`.

Databaser

- Databaser
  - Vi skal have en database – lokal DB
  - Er en del af Visual Studio

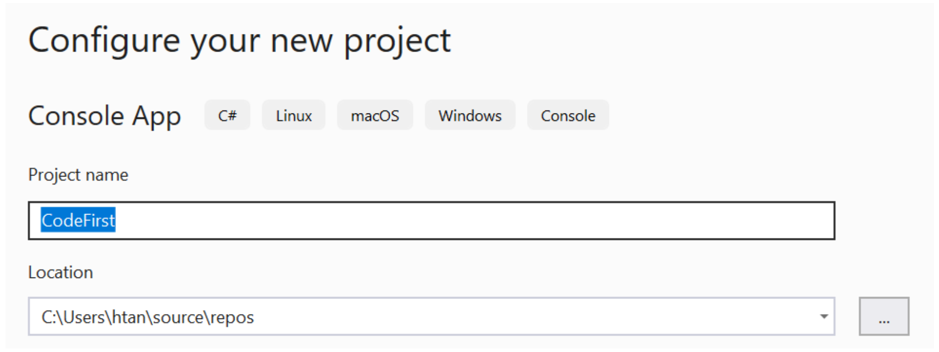


NOTE *When you first use SQL Server from Visual Studio, it creates a local SQL Server instance for you called* `(localdb)\MSSQLLocalDB`*. If you had installed the localdb database with a previous version of Visual Studio, you might have to use the name* `(localdb)\v11.0` *in the Server Name field, as Microsoft has changed the default server name. Or if you have installed the SQL Server Express Edition, you might have to use* `.\sqlexpress`*, as the Entity Framework uses the first local SQL Server database it finds.*

Databaser

- Databaser
  - Code first eksempel:

  - 1. Skab et almindeligt Console App projekt



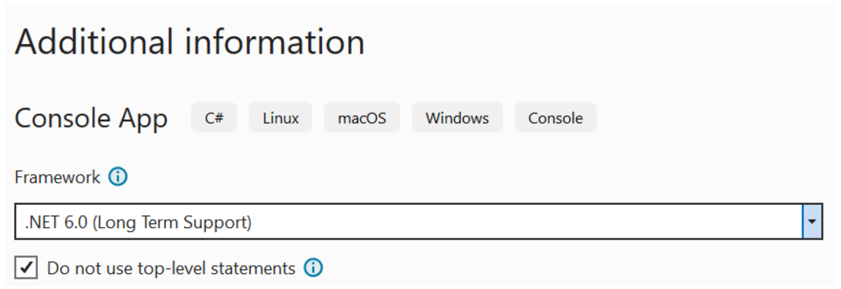
Configure your new project

Project name

CodeFirst

Location

C:\Users\htan\source\repos



Additional information

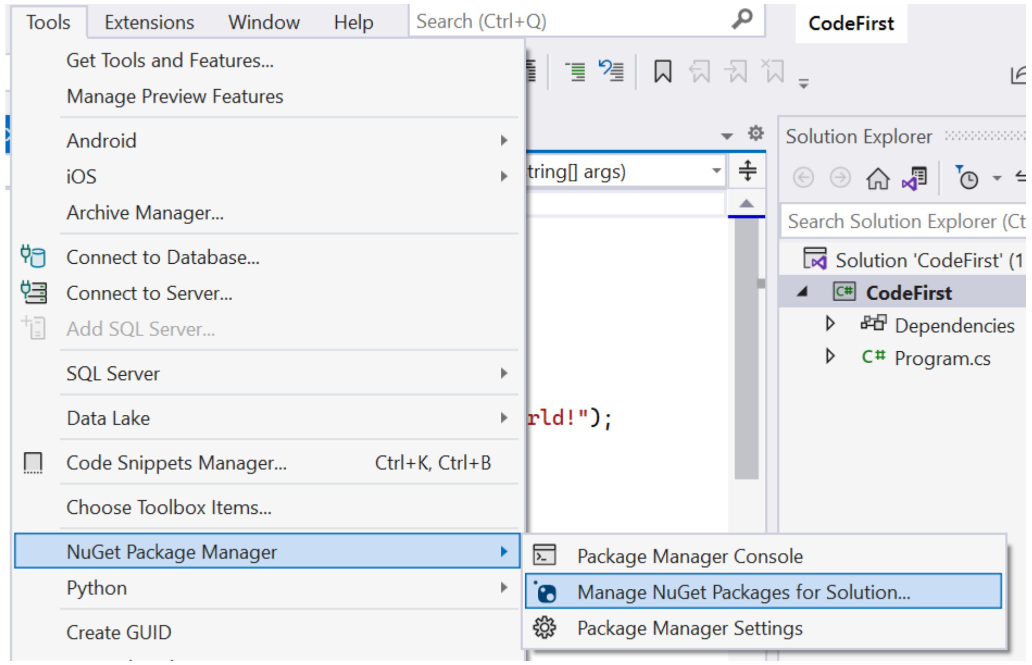
Framework

.NET 6.0 (Long Term Support)

☑ Do not use top-level statements

Databaser

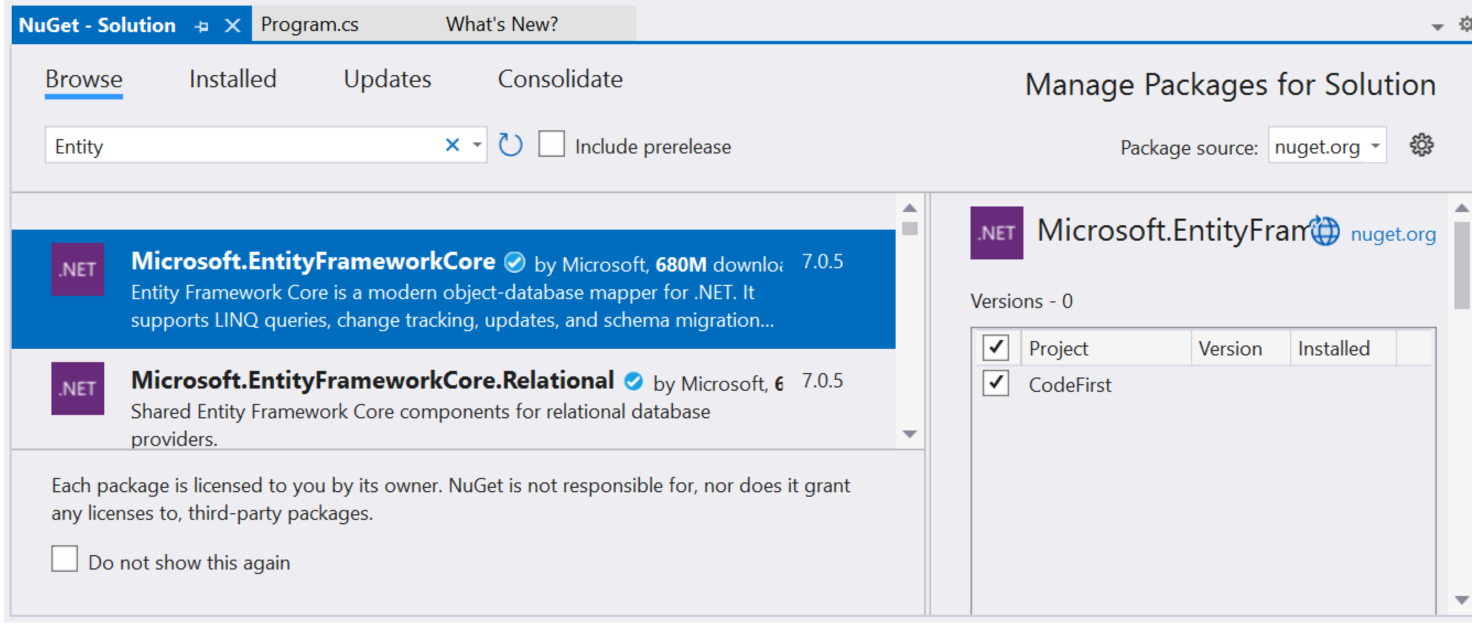
- Databaser
  - Code first eksempel:
  - 2. Adder nuGet pakken: Brug NuGet Package Manager (Tools)
    - Manage NuGet Packages for solution



```
tring[] args)
rld!");
```

Databaser

- Databaser
  - Code first eksempel:
  - 2. Adder nuGet pakken: Brug NuGet Package Manager (Tools)
    - Browse: ”Enity”:
    - Microsoft.FrameworkCore
    - Husk at addere projektet



Microsoft.EntityFrameworkCore by Microsoft, 680M downloa 7.0.5
Entity Framework Core is a modern object-database mapper for .NET. It supports LINQ queries, change tracking, updates, and schema migration...

Microsoft.EntityFrameworkCore.Relational by Microsoft, 6 7.0.5
Shared Entity Framework Core components for relational database providers.

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

☐ Do not show this again

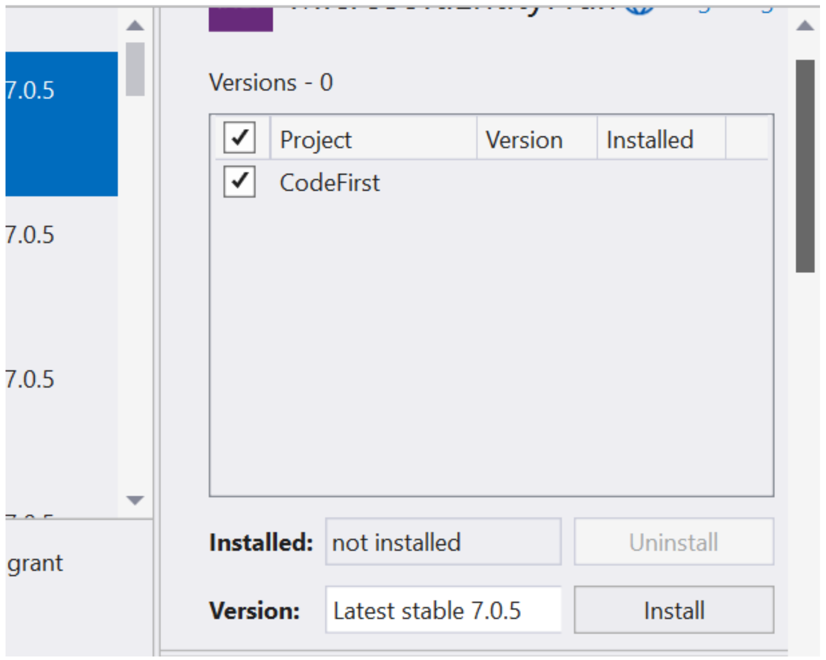
Microsoft.EntityFram nuget.org

Versions - 0

✓	Project	Version	Installed
✓	CodeFirst		

Databaser

- Databaser
  - Code first eksempel:
  - 2. Adder nuGet pakken: Brug NuGet Package Manager (Tools)
    - Browse: ”Enity”:
    - Microsoft.FrameworkCore
    - Husk at addere projektet
    - NU tryk Install:



7.0.5

7.0.5

7.0.5

grant

Versions - 0

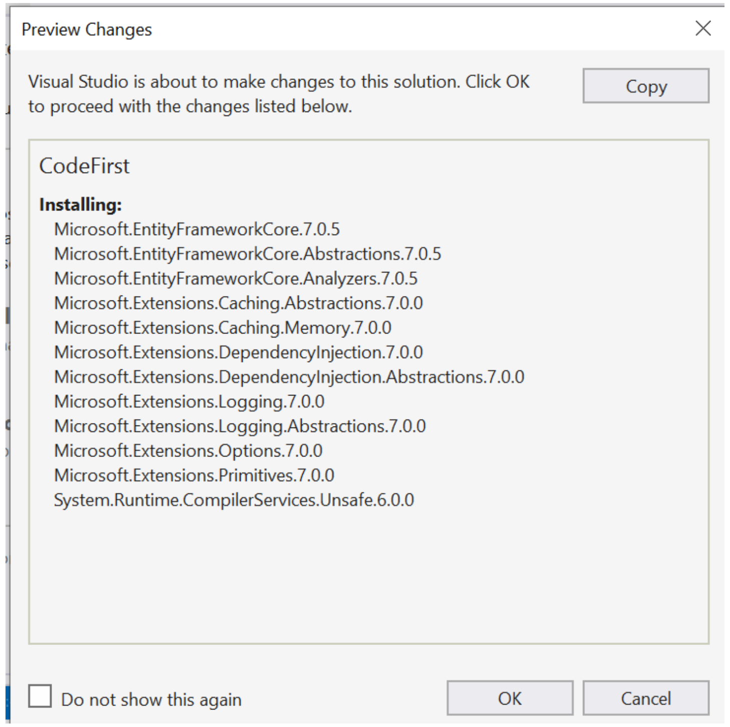
	Project	Version	Installed
✓	CodeFirst		

Installed: not installed

Version: Latest stable 7.0.5

Databaser

- Databaser
  - Code first eksempel:
  - 2. Adder nuGet pakken: Brug NuGet Package Manager (Tools)
    - OK



Visual Studio is about to make changes to this solution. Click OK to proceed with the changes listed below.

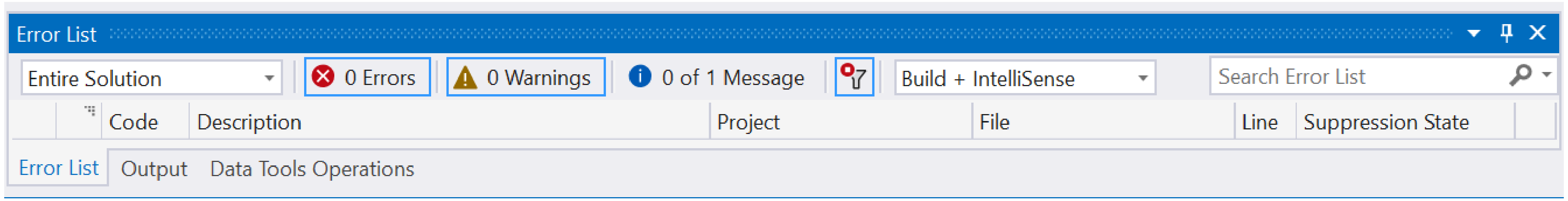
CodeFirst

Installing:

```
Microsoft.EntityFrameworkCore.7.0.5
Microsoft.EntityFrameworkCore.Abstractions.7.0.5
Microsoft.EntityFrameworkCore.Analyzers.7.0.5
Microsoft.Extensions.Caching.Abstractions.7.0.0
Microsoft.Extensions.Caching.Memory.7.0.0
Microsoft.Extensions.DependencyInjection.7.0.0
Microsoft.Extensions.DependencyInjection.Abstractions.7.0.0
Microsoft.Extensions.Logging.7.0.0
Microsoft.Extensions.Logging.Abstractions.7.0.0
Microsoft.Extensions.Options.7.0.0
Microsoft.Extensions.Primitives.7.0.0
System.Runtime.CompilerServices.Unsafe.6.0.0
```

Databaser

- Databaser
  - Code first eksempel:
  - 2. Adder nuGet pakken: Brug NuGet Package Manager (Tools)
    - Hurra (rimeligt forventet)


Error List

Entire Solution | 0 Errors | 0 Warnings | 0 of 1 Message | Build + IntelliSense | Search Error List

	Code	Description	Project	File	Line	Suppression State

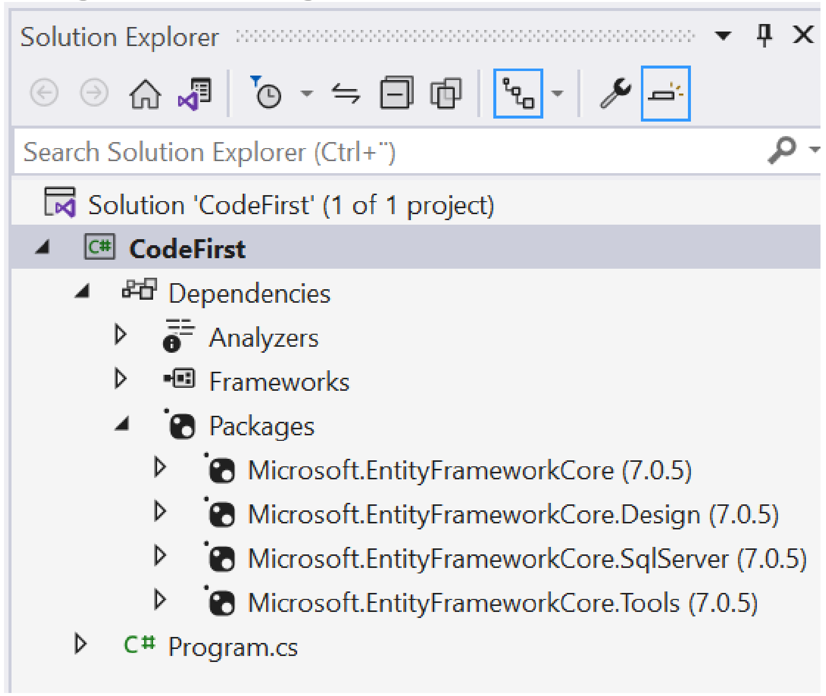
Error List | Output | Data Tools Operations

Databaser

- Databaser
  - Code first eksempel:
  - 2. Adder nuGet pakken: Brug NuGet Package Manager (Tools)
    - Nu gentages festen for:
      - Microsoft.EntityFrameworkCore.Design
      - Microsoft.EntityFrameworkCore.Tools
      - Microsoft.EntityFrameworkCore.SqlServer

Databaser

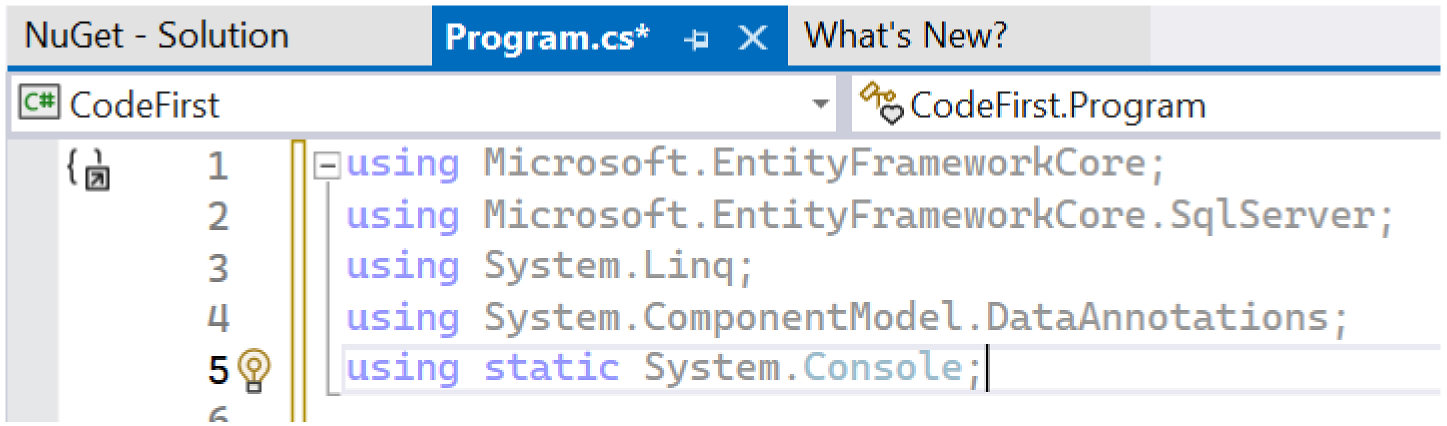
- Databaser
  - Code first eksempel:
  - 2. Adder nuGet pakken: Brug NuGet Package Manager (Tools)
    - Du kan checke installation af pakker
    - I Solution Explorer:



Solution 'CodeFirst' (1 of 1 project)
CodeFirst
Dependencies
Analyzers
Frameworks
Packages
Microsoft.EntityFrameworkCore (7.0.5)
Microsoft.EntityFrameworkCore.Design (7.0.5)
Microsoft.EntityFrameworkCore.SqlServer (7.0.5)
Microsoft.EntityFrameworkCore.Tools (7.0.5)
Program.cs

Databaser

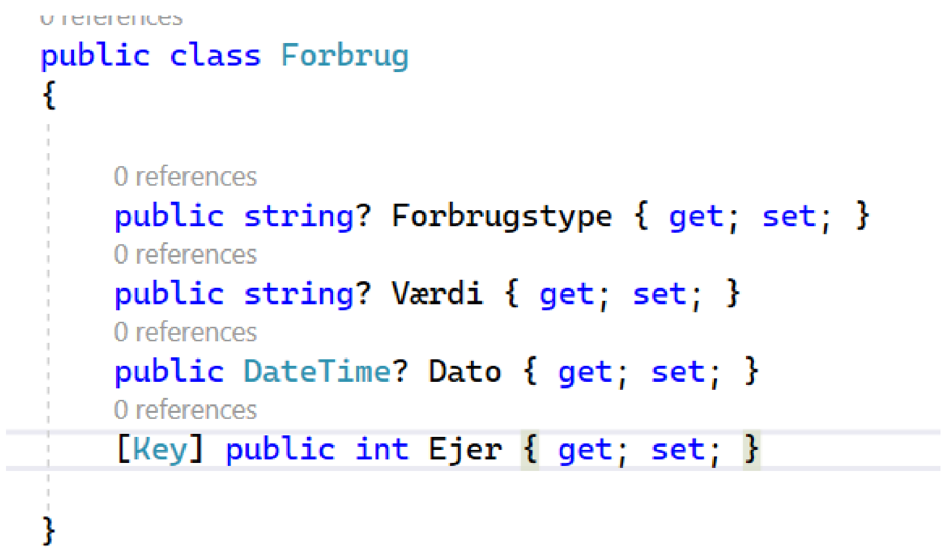
- Databaser
  - Code first eksempel:
  - 3. I Program.cs adder følgende namespaces



```
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.SqlServer;
using System.Linq;
using System.ComponentModel.DataAnnotations;
using static System.Console;
```

Databaser

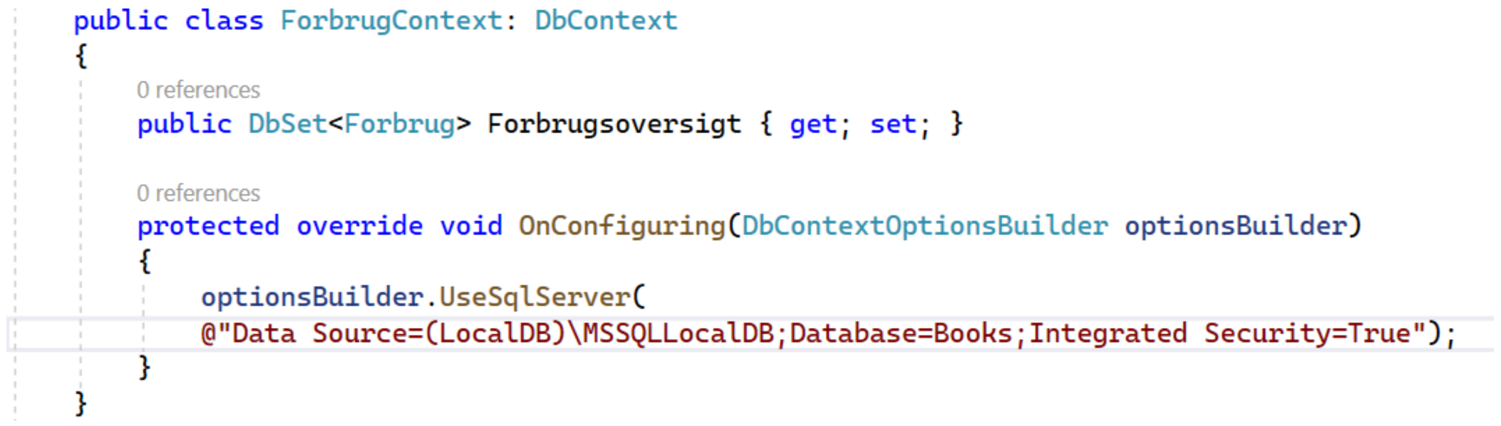
- Databaser
  - Code first eksempel:
  - 4. Skriv en klasse (som bliver til en tabel)



```
0 references
public class Forbrug
{

    0 references
    public string? Forbrugstype { get; set; }
    0 references
    public string? Værdi { get; set; }
    0 references
    public DateTime? Dato { get; set; }
    0 references
    [Key] public int Ejer { get; set; }

}
```

Databaser

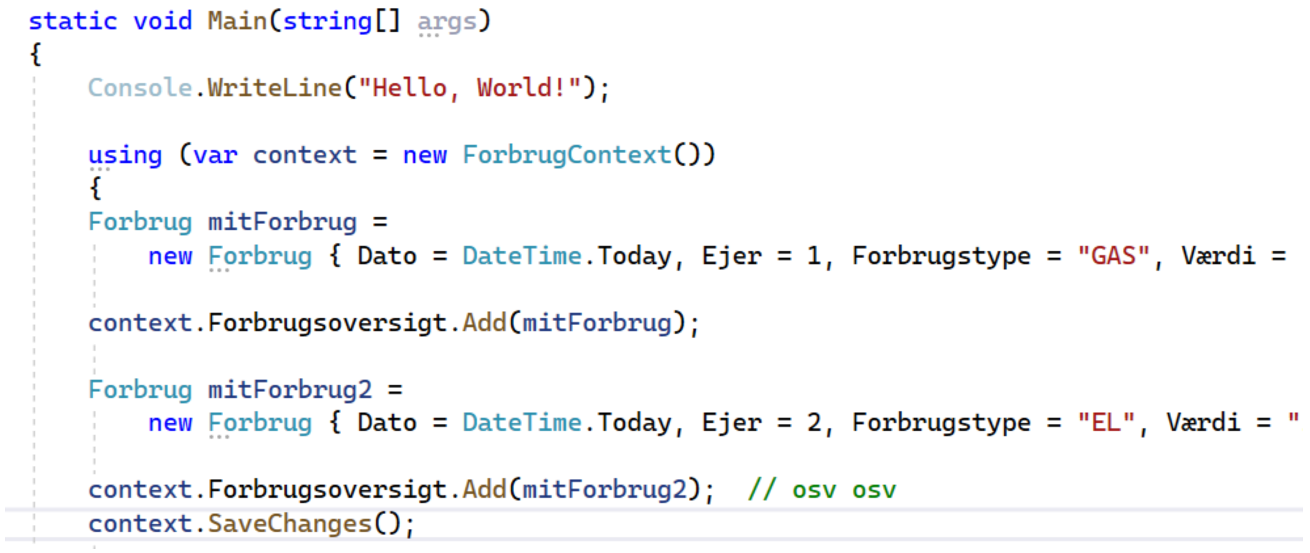
- Databaser
  - Code first eksempel:
  - 5. Skriv en klasse der håndterer transport af data til databasen
    - Klassen arver fra DbContext
    - Har et DbSet<>
    - Og override af OnConfiguring()



```
public class ForbrugContext: DbContext
{
    0 references
    public DbSet<Forbrug> Forbrugsoversigt { get; set; }

    0 references
    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
    {
        optionsBuilder.UseSqlServer(
        @"Data Source=(LocalDB)\MSSQLLocalDB;Database=Books;Integrated Security=True");
    }
}
```

Databaser

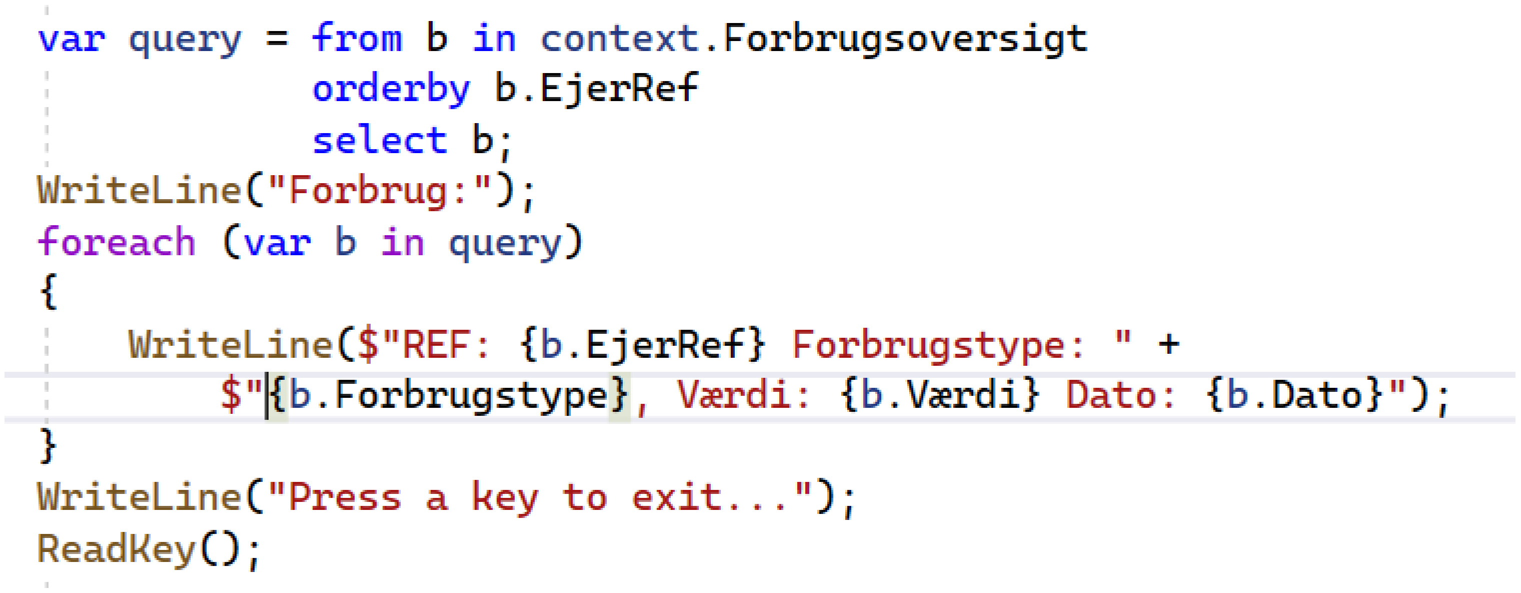
- Databaser
  - Code first eksempel:
  - 6. Skriv koden i Main, der skal oprette records:



```
static void Main(string[] args)
{
    Console.WriteLine("Hello, World!");

    using (var context = new ForbrugContext())
    {
    Forbrug mitForbrug =
        new Forbrug { Dato = DateTime.Today, Ejer = 1, Forbrugstype = "GAS", Værdi =

    context.Forbrugsoversigt.Add(mitForbrug);

    Forbrug mitForbrug2 =
        new Forbrug { Dato = DateTime.Today, Ejer = 2, Forbrugstype = "EL", Værdi = "

    context.Forbrugsoversigt.Add(mitForbrug2);  // osv osv
    context.SaveChanges();
```

Databaser

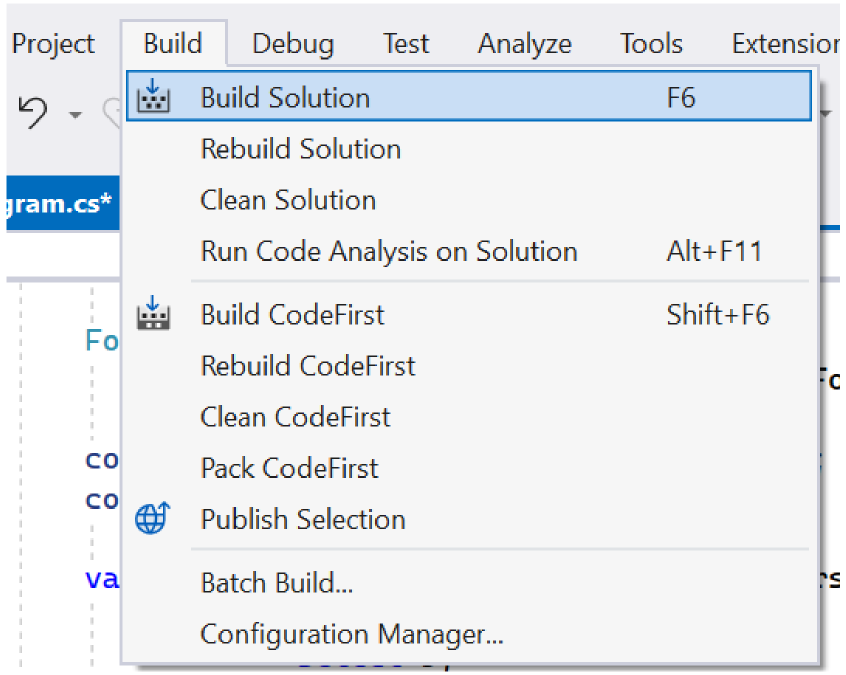
- Databaser
  - Code first eksempel:
  - 7. Skriv koden i Main, der skal hente records (LINQ benyttes – så nemt):



```
var query = from b in context.Forbrugsoversigt
            orderby b.EjerRef
            select b;
WriteLine("Forbrug:");
foreach (var b in query)
{
    WriteLine($"REF: {b.EjerRef} Forbrugstype: " +
        $"{b.Forbrugstype}, Værdi: {b.Værdi} Dato: {b.Dato}");
}
WriteLine("Press a key to exit...");
ReadKey();
```

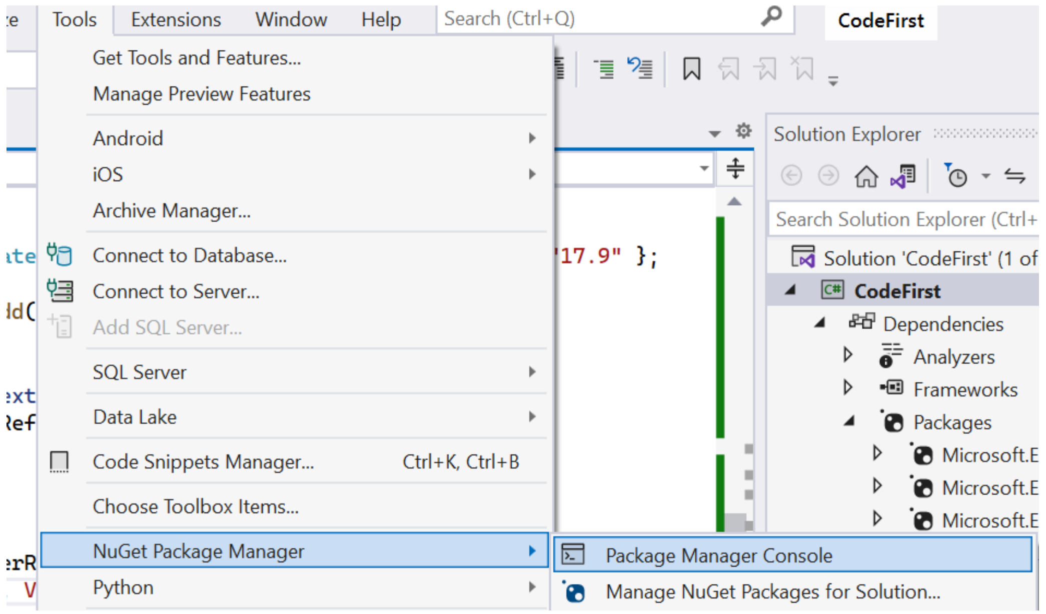
Databaser

- Databaser
  - Code first eksempel:
  - 8. Build koden (skal ikke afvikles – databasen skal skabes)



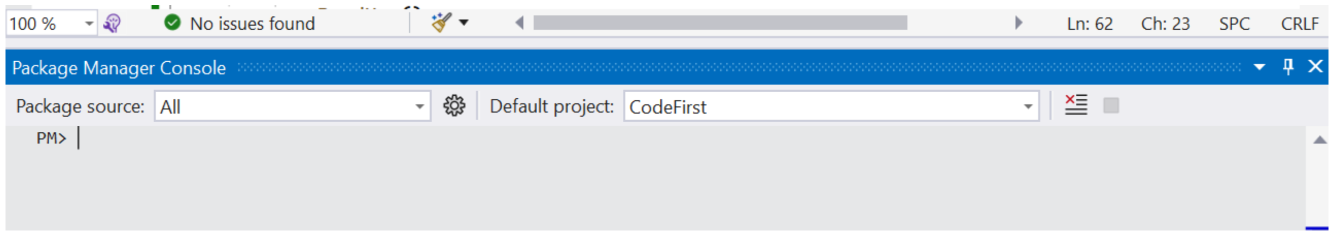
Databaser

- Databaser
  - Code first eksempel:
  - 9. Nu skal databasen skabes: Brug (Tools) Package Manager Console

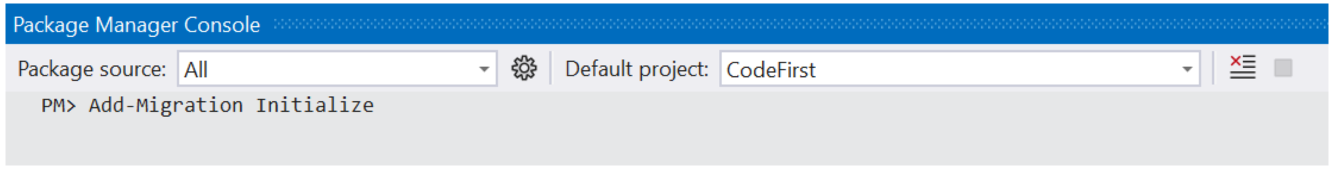


Databaser

- Databaser
  - Code first eksempel:
  - 10. I Package Manager Console: Skriv på PM>
    - ”Add-Migration Initialize”, tryk ”enter”



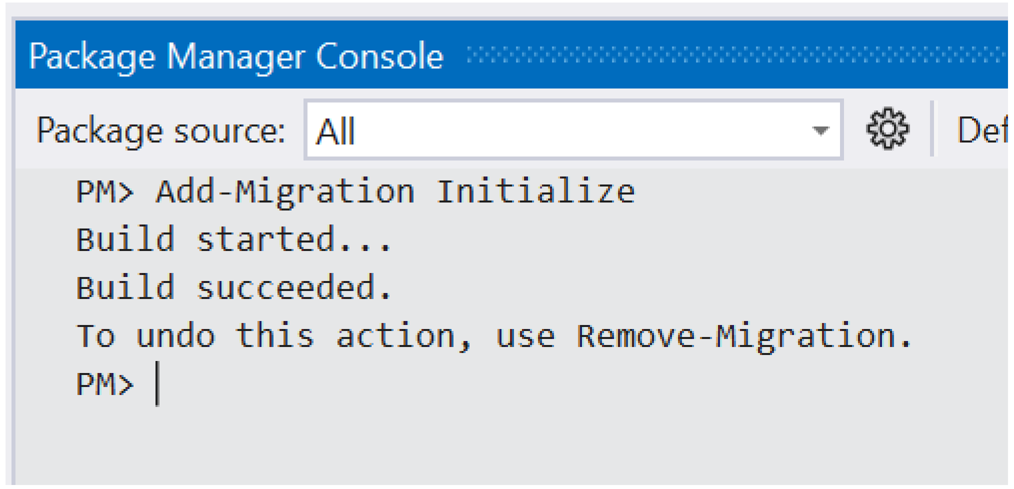
```
PM>
```



```
PM> Add-Migration Initialize
```

Databaser

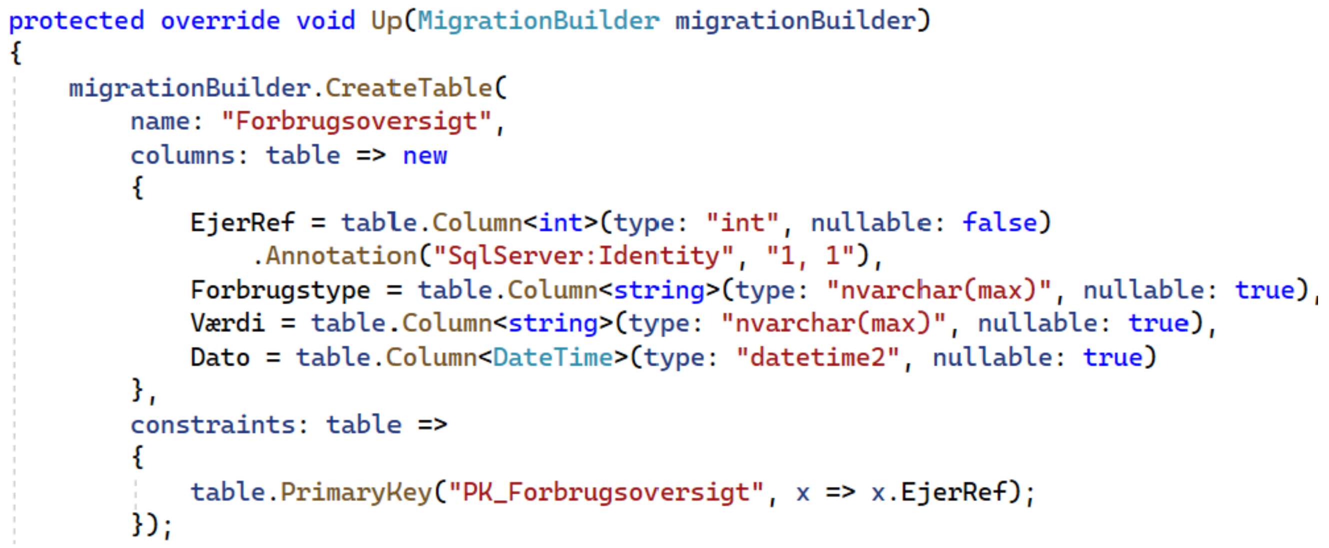
- Databaser
  - Code first eksempel:
  - 11. BINGO



```
PM> Add-Migration Initialize
Build started...
Build succeeded.
To undo this action, use Remove-Migration.
PM>
```

Databaser

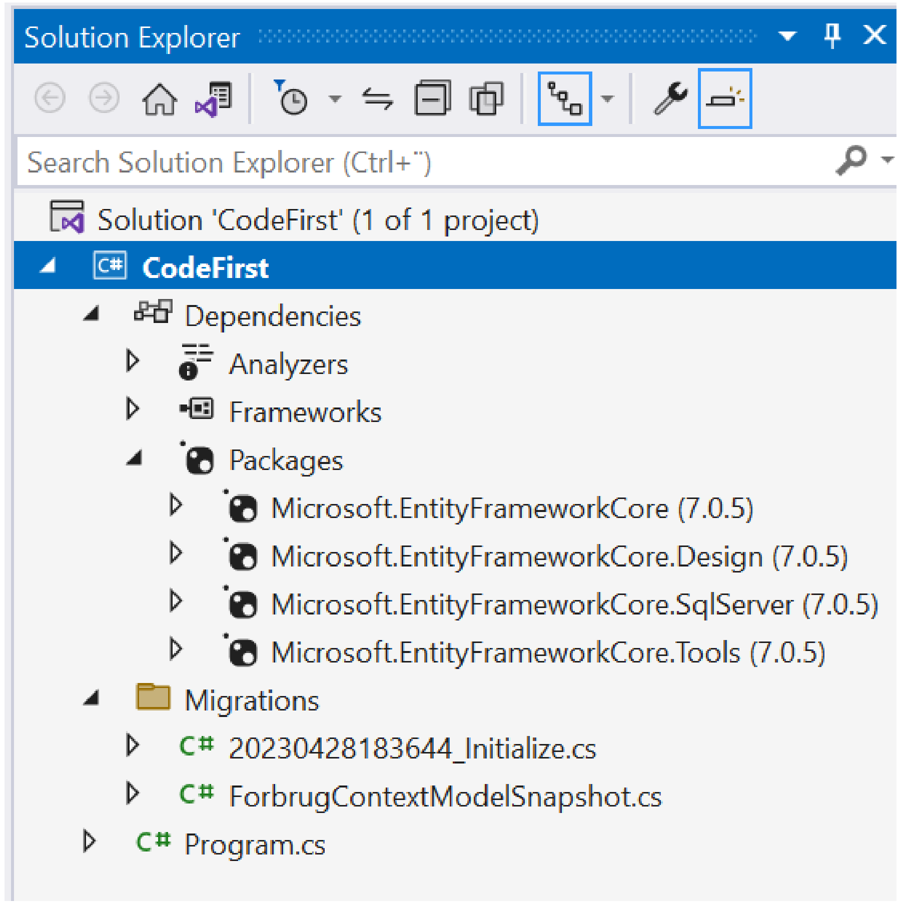
- Databaser
  - Code first eksempel:
  - 12. Der skabes en klasse bagved (uddrag)



```
protected override void Up(MigrationBuilder migrationBuilder)
{
    migrationBuilder.CreateTable(
        name: "Forbrugsoversigt",
        columns: table => new
        {
            EjerRef = table.Column<int>(type: "int", nullable: false)
                .Annotation("SqlServer:Identity", "1, 1"),
            Forbrugstype = table.Column<string>(type: "nvarchar(max)", nullable: true),
            Værdi = table.Column<string>(type: "nvarchar(max)", nullable: true),
            Dato = table.Column<DateTime>(type: "datetime2", nullable: true)
        },
        constraints: table =>
        {
            table.PrimaryKey("PK_Forbrugsoversigt", x => x.EjerRef);
        });
```

Databaser

- Databaser
  - Code first eksempel:
  - 13. Oversigt over migration:



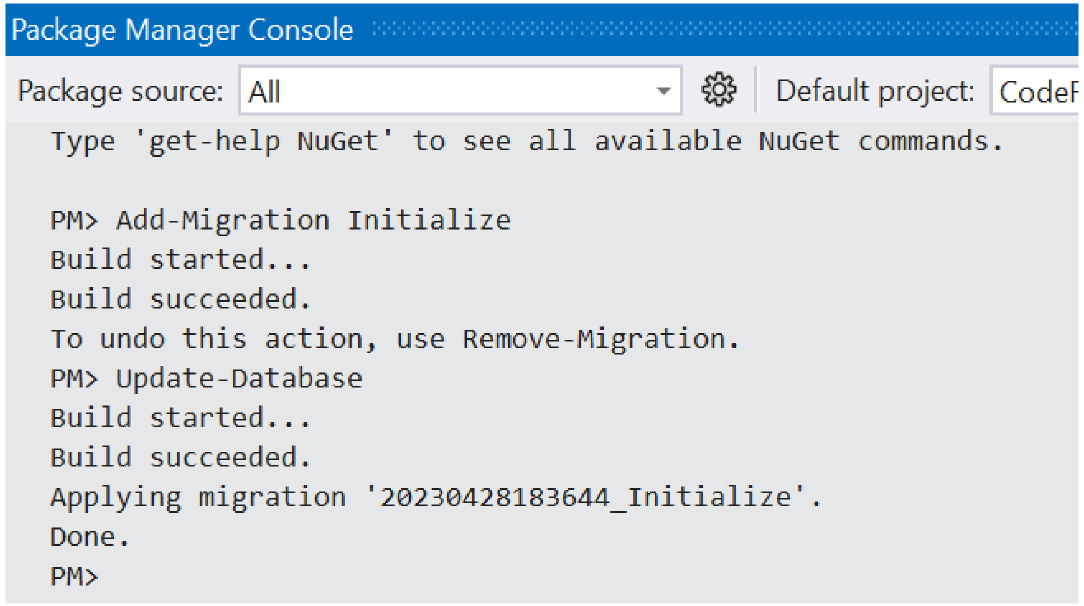
Solution Explorer

Search Solution Explorer (Ctrl+¨)

Solution 'CodeFirst' (1 of 1 project)
CodeFirst
Dependencies
Analyzers
Frameworks
Packages
Microsoft.EntityFrameworkCore (7.0.5)
Microsoft.EntityFrameworkCore.Design (7.0.5)
Microsoft.EntityFrameworkCore.SqlServer (7.0.5)
Microsoft.EntityFrameworkCore.Tools (7.0.5)
Migrations
20230428183644_Initialize.cs
ForbrugContextModelSnapshot.cs
Program.cs

Databaser

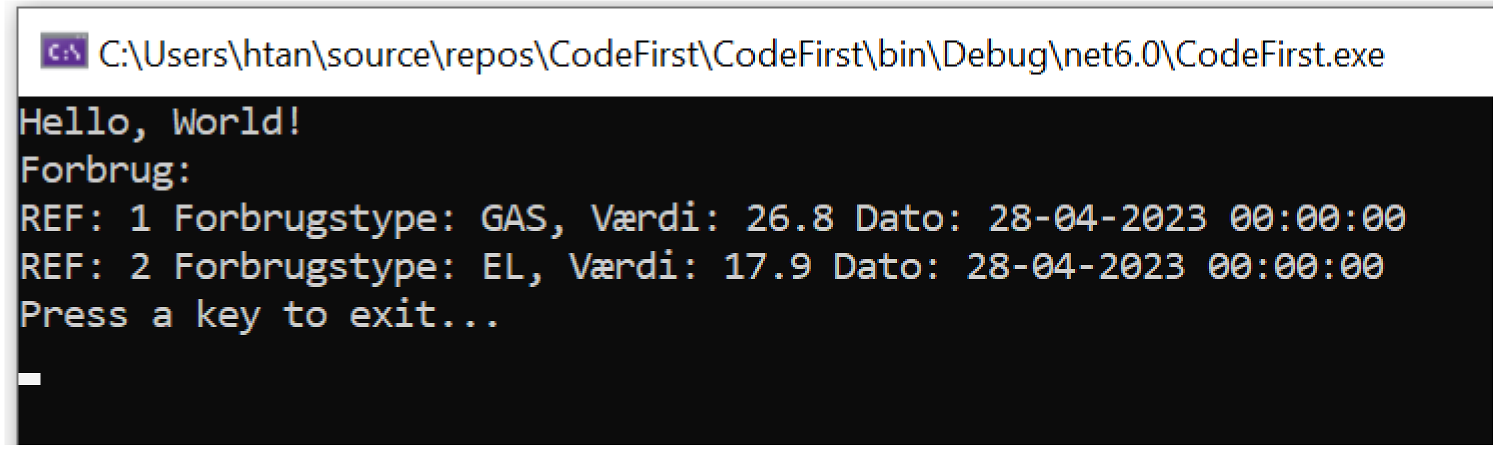
- Databaser
  - Code first eksempel:
  - 14. Update (skaber database objekter)
  - Ved PM> ” Update-Database”



```
Type 'get-help NuGet' to see all available NuGet commands.

PM> Add-Migration Initialize
Build started...
Build succeeded.
To undo this action, use Remove-Migration.
PM> Update-Database
Build started...
Build succeeded.
Applying migration '20230428183644_Initialize'.
Done.
PM>
```

Databaser

- Databaser
  - Code first eksempel:
  - 15. Kør programmet som normalt

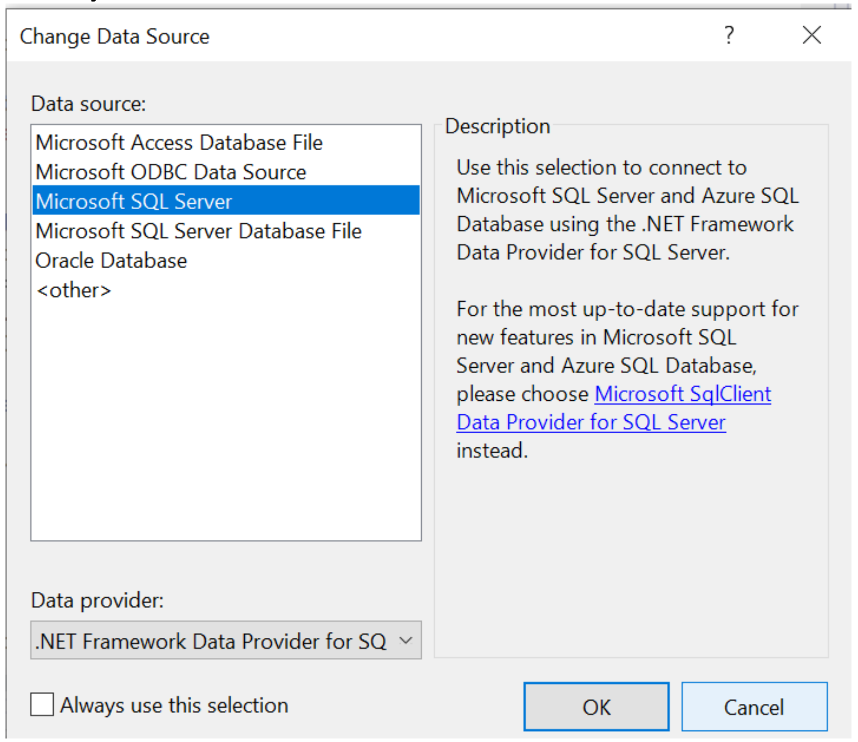


C:\Users\htan\source\repos\CodeFirst\CodeFirst\bin\Debug\net6.0\CodeFirst.exe

```
Hello, World!
Forbrug:
REF: 1 Forbrugstype: GAS, Værdi: 26.8 Dato: 28-04-2023 00:00:00
REF: 2 Forbrugstype: EL, Værdi: 17.9 Dato: 28-04-2023 00:00:00
Press a key to exit...
```

Databaser

- Databaser
  - Code first eksempel:
  - 16. Databasen kan findes via Server Explorer (normalt øverst til venstre i Visual Studio): Men inden da:
  - Tools | Connect To Database | Choose Data Source
  - Vælg Microsoft SQL Server

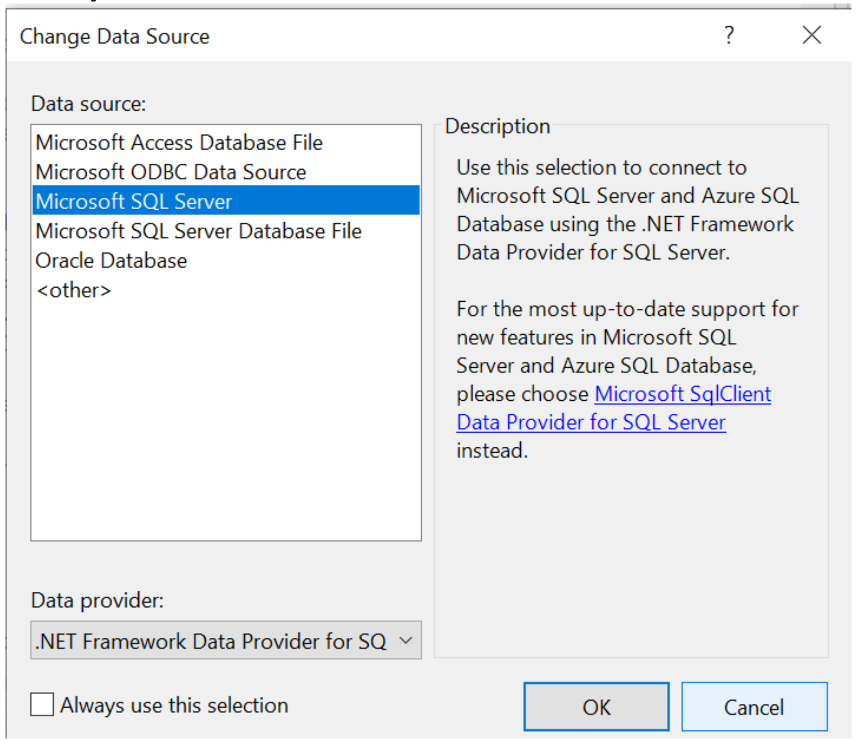

Change Data Source
?
Data source:
Microsoft Access Database File
Microsoft ODBC Data Source
Microsoft SQL Server
Microsoft SQL Server Database File
Oracle Database
<other>
Description
Data provider:
.NET Framework Data Provider for SQ
Always use this selection
OK
Cancel

Use this selection to connect to Microsoft SQL Server and Azure SQL Database using the .NET Framework Data Provider for SQL Server.

For the most up-to-date support for new features in Microsoft SQL Server and Azure SQL Database, please choose Microsoft SqlClient Data Provider for SQL Server instead.

Databaser

- Databaser
  - Code first eksempel:
  - 16. Databasen kan findes via Server Explorer (normalt øverst til venstre i Visual Studio): Men inden da:
  - Tools | Connect To Database | Choose Data Source
  - Vælg Microsoft SQL Server



Microsoft Access Database File
Microsoft ODBC Data Source
Microsoft SQL Server
Microsoft SQL Server Database File
Oracle Database
<other>

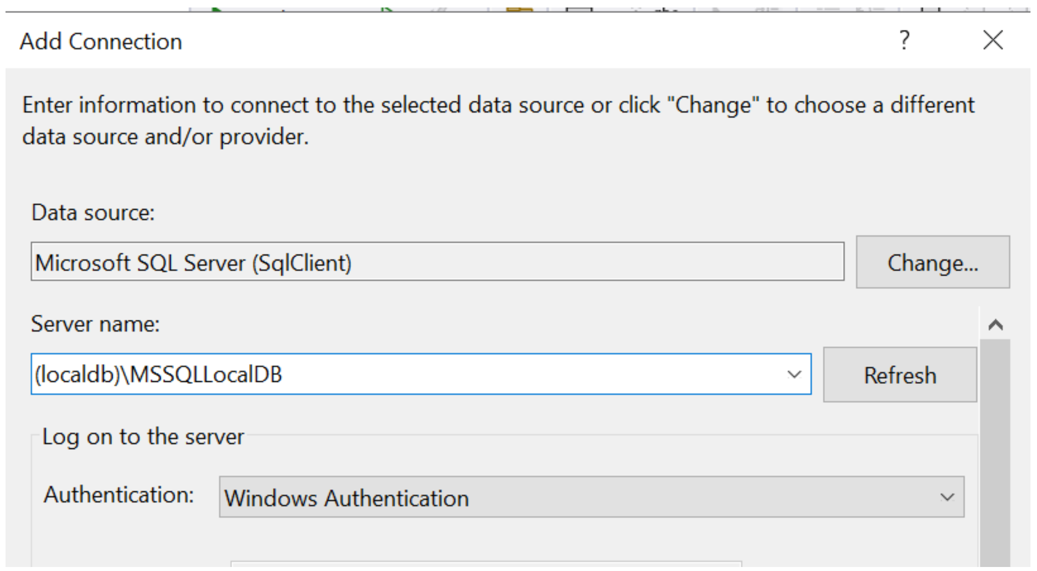
Use this selection to connect to Microsoft SQL Server and Azure SQL Database using the .NET Framework Data Provider for SQL Server.

For the most up-to-date support for new features in Microsoft SQL Server and Azure SQL Database, please choose Microsoft SqlClient Data Provider for SQL Server instead.

.NET Framework Data Provider for SQ

Databaser

- Databaser
  - Code first eksempel:
  - 16. Databasen kan findes via Server Explorer (normalt øverst til venstre i Visual Studio): Men inden da:
  - Skriv: (localdb)\MSSQLLocalDB


Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft SQL Server (SqlClient)

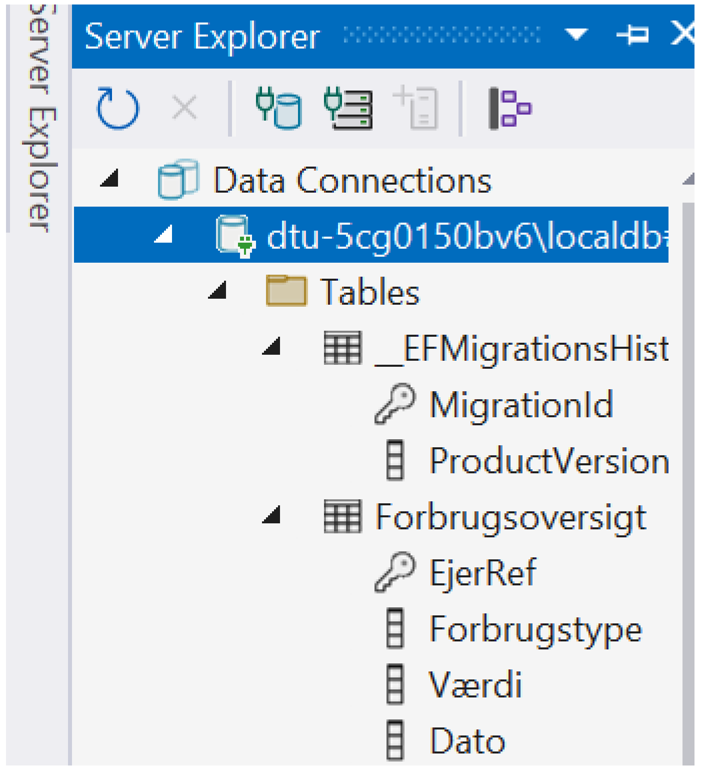
Server name: (localdb)\MSSQLLocalDB

Log on to the server

Authentication: Windows Authentication

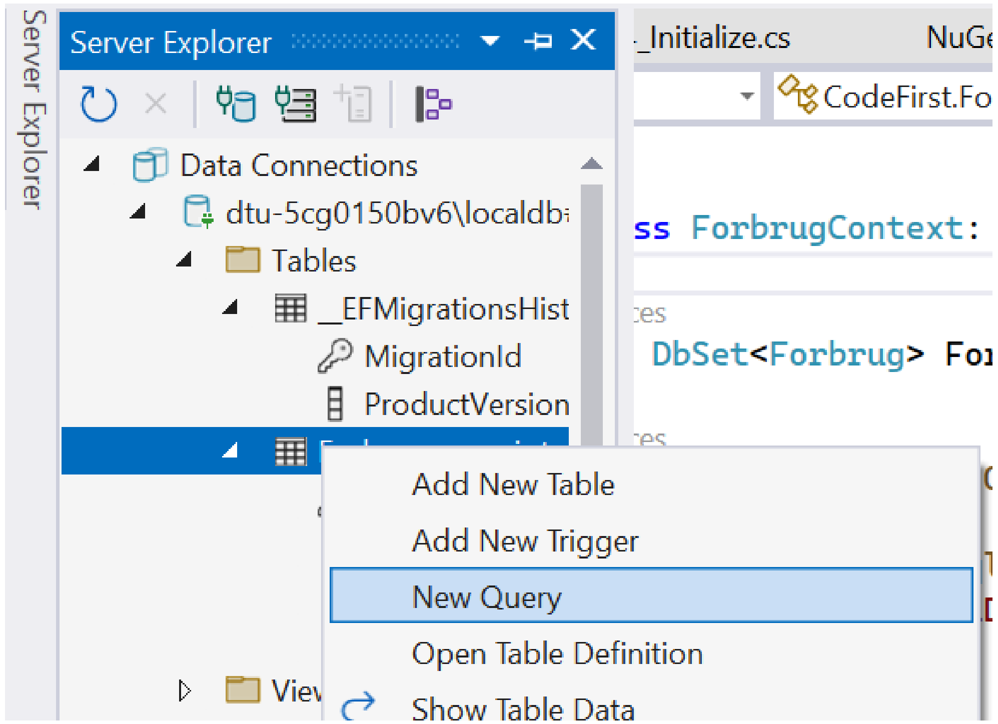
Databaser

- Databaser
  - Code first eksempel:
  - 17. Databasen (nu) kan findes via Server Explorer
  - (normalt øverst til venstre i Visual Studio):



Databaser

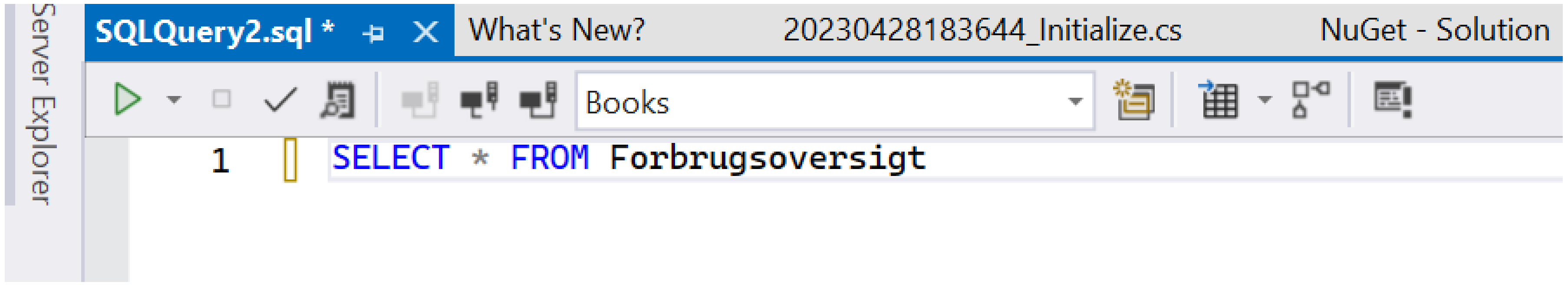
- Databaser
  - Code first eksempel:
  - 18. Læs records (data):
  - Højeklik på tabellen – Add Query:



```
ss ForbrugContext:
DbSet<Forbrug> Fo
```

Databaser

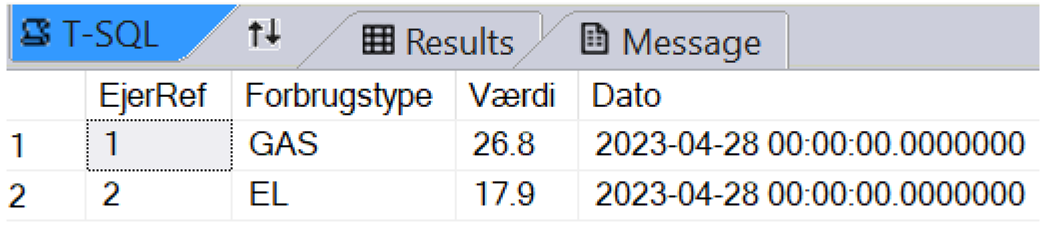
- Databaser
  - Code first eksempel:
  - 18. Læs records (data):
  - Højeklik på tabellen – Add Query:
  - Skriv SQL SELECT, Højreklik: Execute



```sql
SELECT * FROM Forbrugsoversigt
```

Databaser

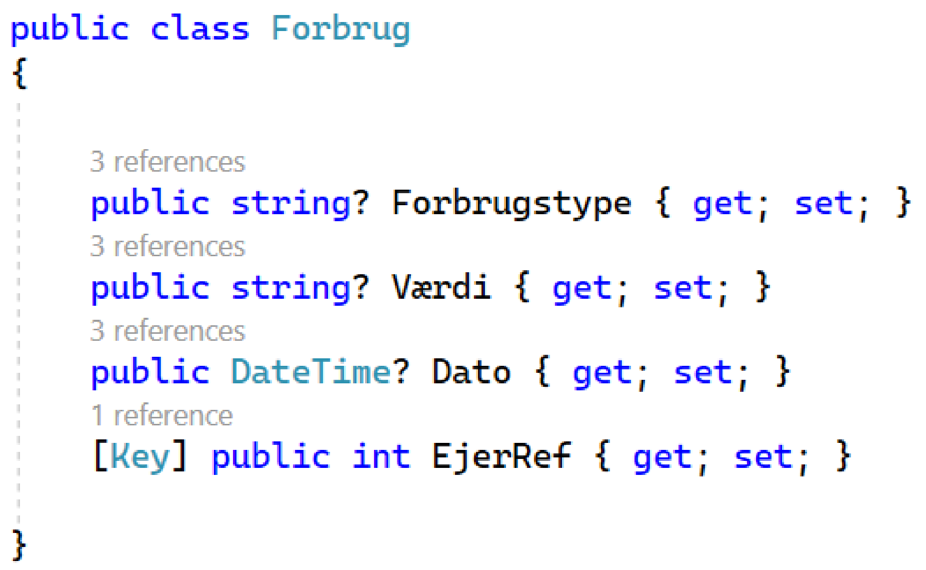
- Databaser
  - Code first eksempel:
  - 19. Læs records (data):



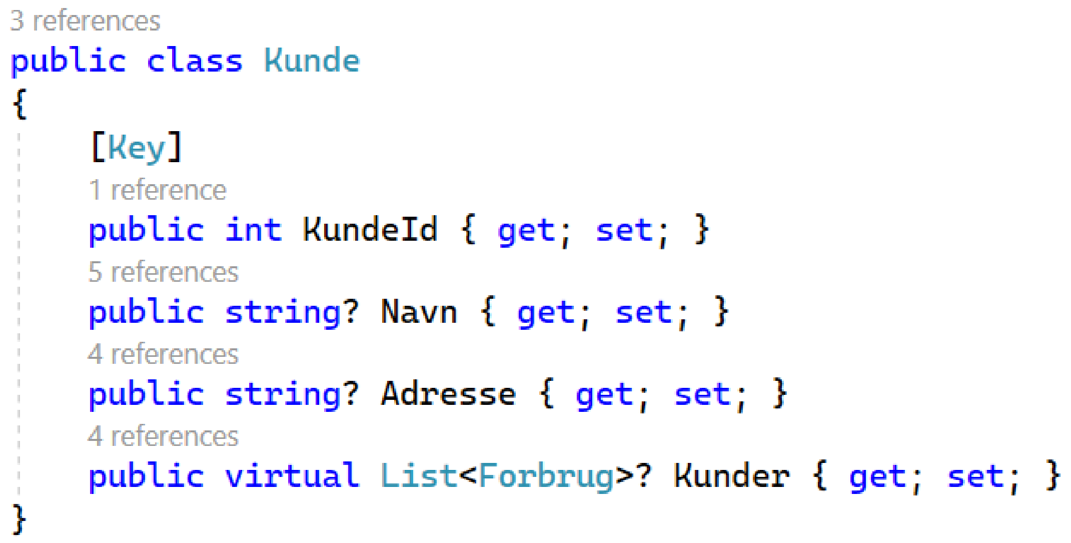
	EjerRef	Forbrugstype	Værdi	Dato
1	1	GAS	26.8	2023-04-28 00:00:00.0000000
2	2	EL	17.9	2023-04-28 00:00:00.0000000

Databaser

- Relationer mellem tabeller
  - En kunde har et tilknyttet forbrug:



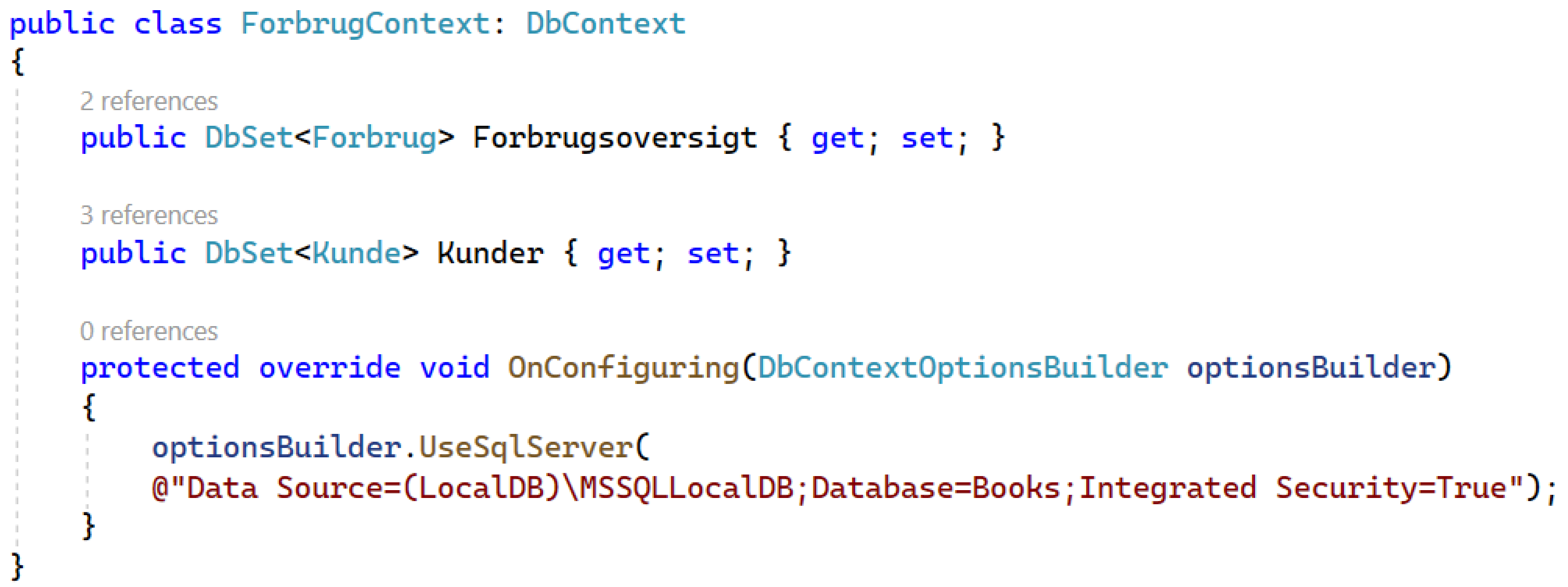
```
public class Forbrug
{

    3 references
    public string? Forbrugstype { get; set; }
    3 references
    public string? Værdi { get; set; }
    3 references
    public DateTime? Dato { get; set; }
    1 reference
    [Key] public int EjerRef { get; set; }

}
```



```
3 references
public class Kunde
{
    [Key]
    1 reference
    public int KundeId { get; set; }
    5 references
    public string? Navn { get; set; }
    4 references
    public string? Adresse { get; set; }
    4 references
    public virtual List<Forbrug>? Kunder { get; set; }
}
```

Databaser

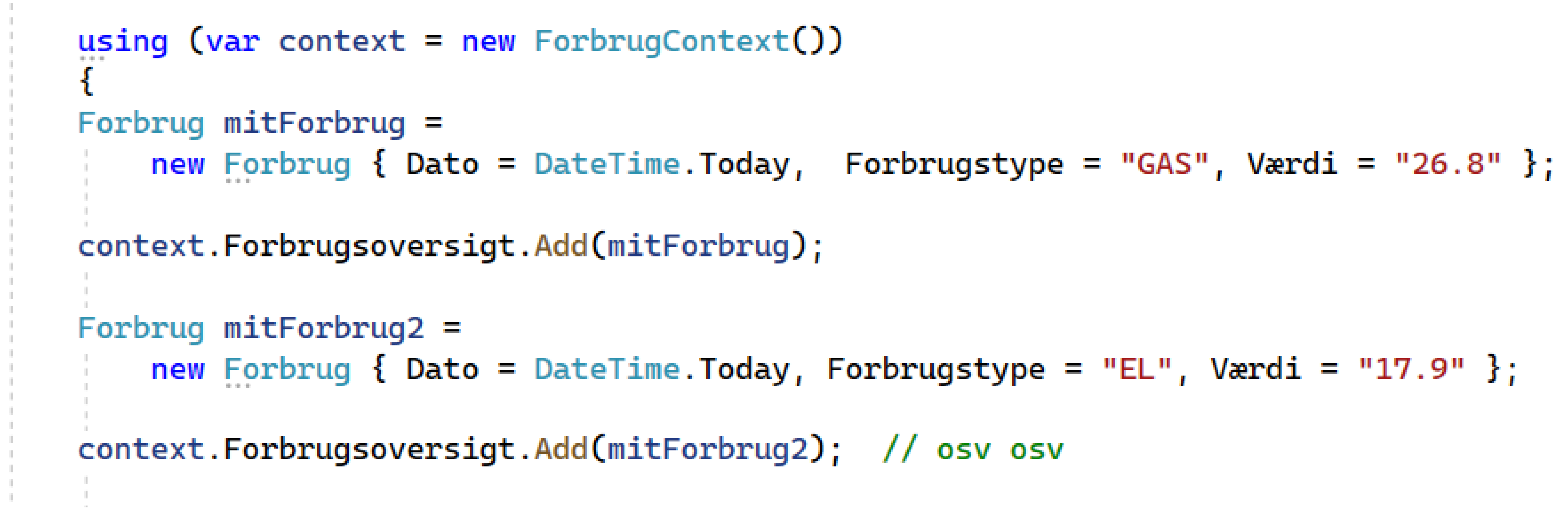
- Relationer mellem tabeller
  - Opdatering af ForbrugContext (DbContext)



```
public class ForbrugContext: DbContext
{
    2 references
    public DbSet<Forbrug> Forbrugsoversigt { get; set; }

    3 references
    public DbSet<Kunde> Kunder { get; set; }

    0 references
    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
    {
        optionsBuilder.UseSqlServer(
        @"Data Source=(LocalDB)\MSSQLLocalDB;Database=Books;Integrated Security=True");
    }
}
```

Databaser

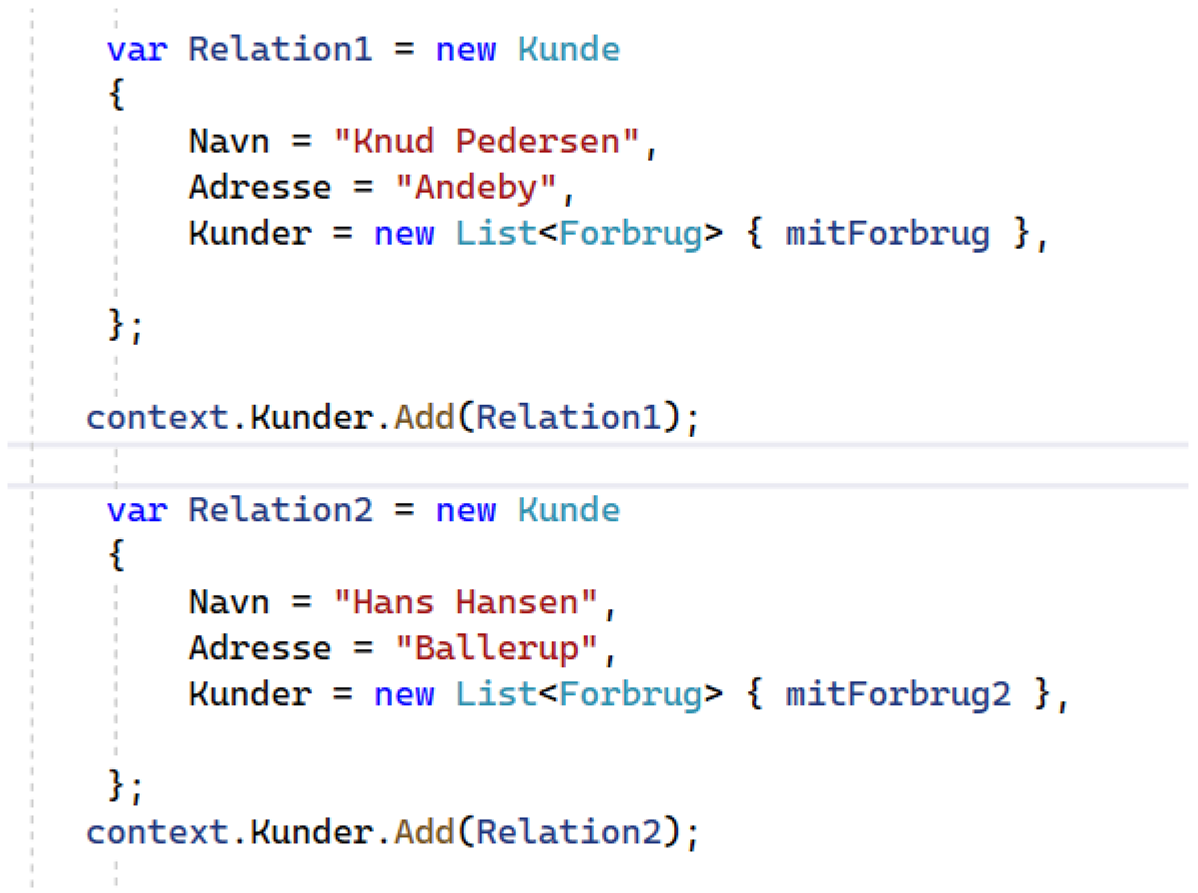
- Relationer mellem tabeller
  - 2 forbrugs records



```
using (var context = new ForbrugContext())
{
Forbrug mitForbrug =
    new Forbrug { Dato = DateTime.Today,  Forbrugstype = "GAS", Værdi = "26.8" };

context.Forbrugsoversigt.Add(mitForbrug);

Forbrug mitForbrug2 =
    new Forbrug { Dato = DateTime.Today, Forbrugstype = "EL", Værdi = "17.9" };

context.Forbrugsoversigt.Add(mitForbrug2);  // osv osv
```

Databaser

- Relationer mellem tabeller
  - 2 kunde records
  - med relation til forbrug



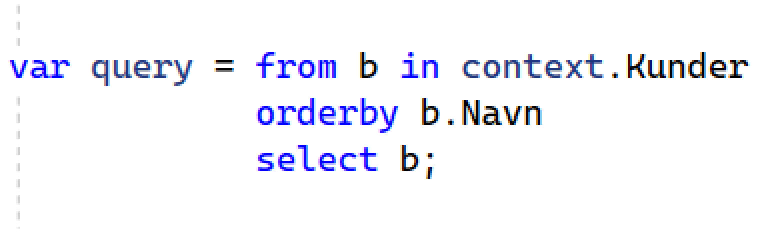
```
 var Relation1 = new Kunde
 {
     Navn = "Knud Pedersen",
     Adresse = "Andeby",
     Kunder = new List<Forbrug> { mitForbrug },

 };

context.Kunder.Add(Relation1);

 var Relation2 = new Kunde
 {
     Navn = "Hans Hansen",
     Adresse = "Ballerup",
     Kunder = new List<Forbrug> { mitForbrug2 },

 };
context.Kunder.Add(Relation2);
```

Databaser

- Relationer mellem tabeller
    - Husk:
    - `context.SaveChanges();`

Databaser

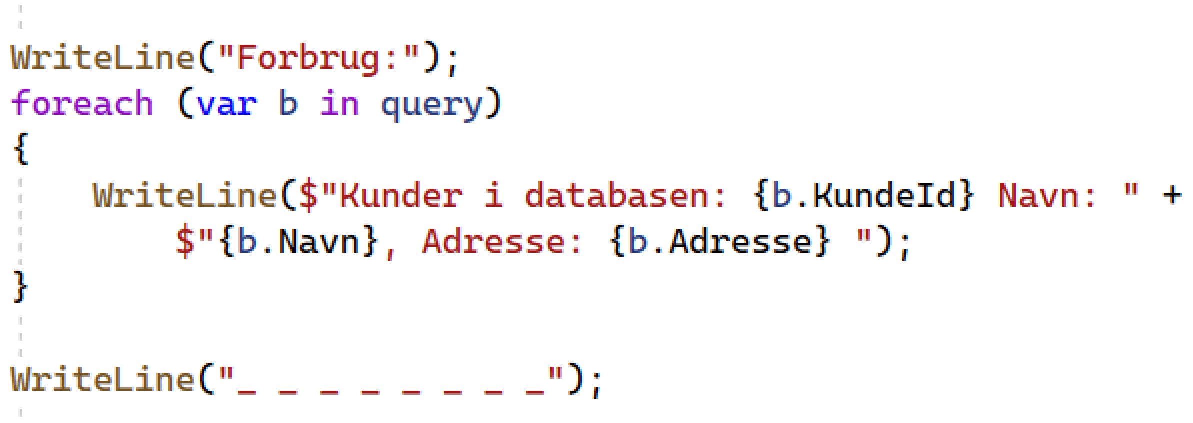
- Query :



```
var query = from b in context.Kunder
            orderby b.Navn
            select b;
```

Databaser

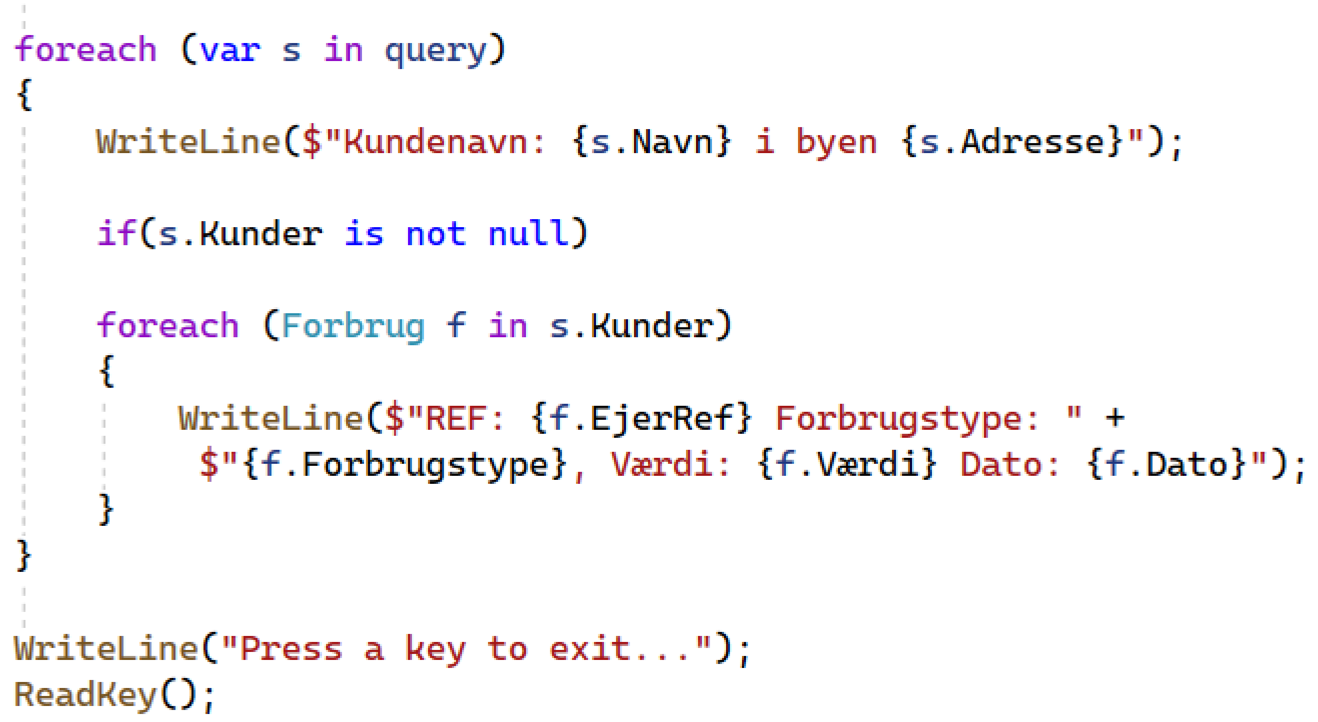
- Relationer mellem tabeller
  - query – kunder (kunder har et forbrug)



```
WriteLine("Forbrug:");
foreach (var b in query)
{
    WriteLine($"Kunder i databasen: {b.KundeId} Navn: " +
        $"{b.Navn}, Adresse: {b.Adresse} ");
}

WriteLine("_ _ _ _ _ _ _ _");
```

Databaser

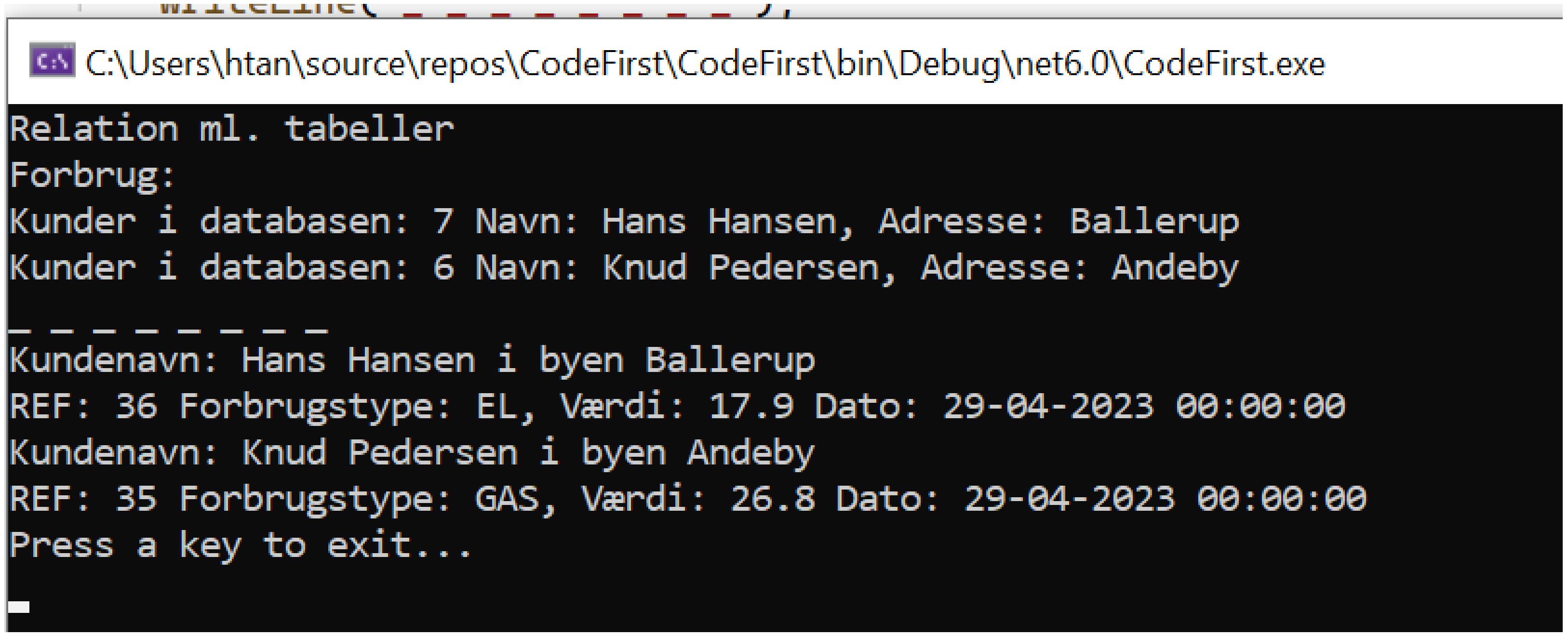
- Relationer mellem tabeller
    - query – udfra en kunde – find forbruget



```
foreach (var s in query)
{
    WriteLine($"Kundenavn: {s.Navn} i byen {s.Adresse}");

    if(s.Kunder is not null)

    foreach (Forbrug f in s.Kunder)
    {
        WriteLine($"REF: {f.EjerRef} Forbrugstype: " +
         $"{f.Forbrugstype}, Værdi: {f.Værdi} Dato: {f.Dato}");
    }
}

WriteLine("Press a key to exit...");
ReadKey();
```

Databaser

- Relationer mellem tabeller
  - output:

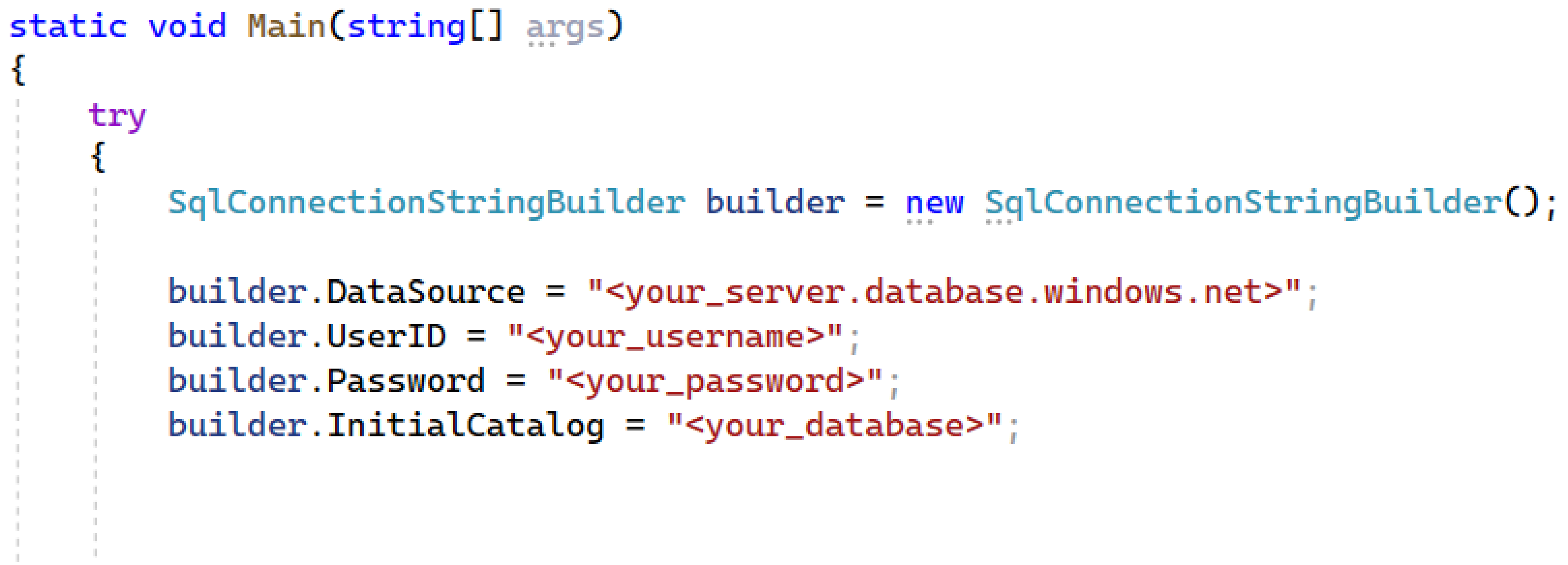


C:\Users\htan\source\repos\CodeFirst\CodeFirst\bin\Debug\net6.0\CodeFirst.exe

```
Relation ml. tabeller
Forbrug:
Kunder i databasen: 7 Navn: Hans Hansen, Adresse: Ballerup
Kunder i databasen: 6 Navn: Knud Pedersen, Adresse: Andeby

_ _ _ _ _ _ _ _
Kundenavn: Hans Hansen i byen Ballerup
REF: 36 Forbrugstype: EL, Værdi: 17.9 Dato: 29-04-2023 00:00:00
Kundenavn: Knud Pedersen i byen Andeby
REF: 35 Forbrugstype: GAS, Værdi: 26.8 Dato: 29-04-2023 00:00:00
Press a key to exit...
```

Databaser

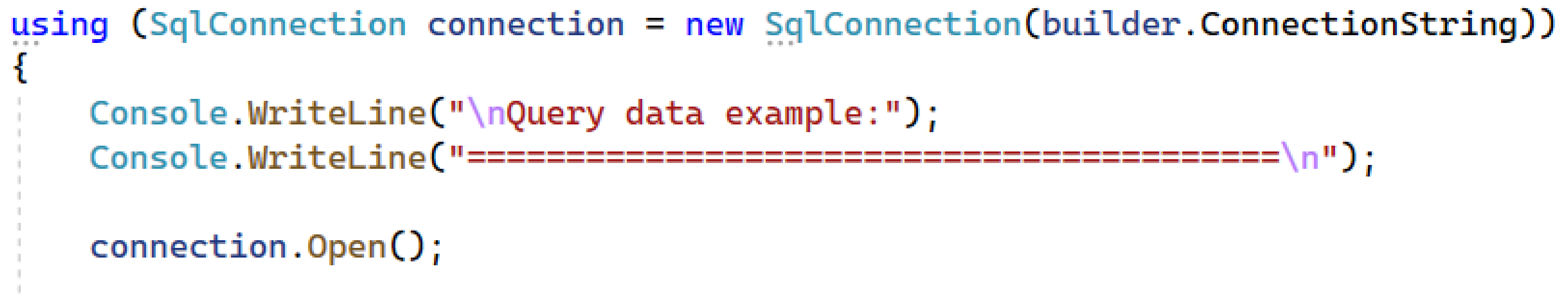
- Ordinær brug af databaser
- Bruges meget
- ConnectionString
- ??
- Kan diskuteres



```
static void Main(string[] args)
{
    try
    {
        SqlConnectionStringBuilder builder = new SqlConnectionStringBuilder();

        builder.DataSource = "<your_server.database.windows.net>";
        builder.UserID = "<your_username>";
        builder.Password = "<your_password>";
        builder.InitialCatalog = "<your_database>";
```

Databaser

- Ordinær brug af databaser
- Skabe connection



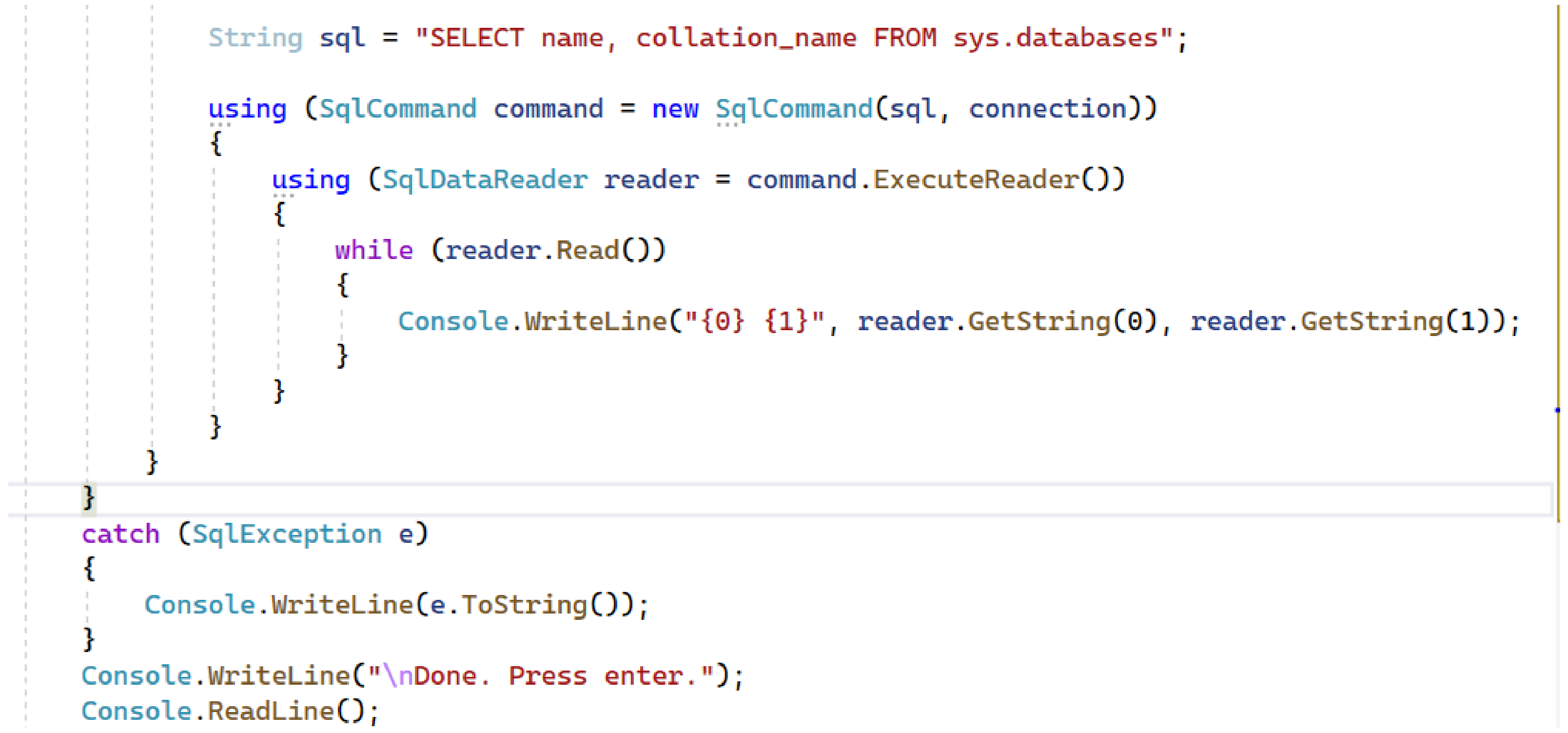
```
using (SqlConnection connection = new SqlConnection(builder.ConnectionString))
{
    Console.WriteLine("\nQuery data example:");
    Console.WriteLine("=========================================\n");

    connection.Open();
```

- Databasen skal altid åbnes

Databaser

- Ordinær brug af databaser
- Skabe SqlCommand
- Læse med en
- SqlDataReader

- Husk at benytte
- try-catch



```
                String sql = "SELECT name, collation_name FROM sys.databases";

                using (SqlCommand command = new SqlCommand(sql, connection))
                {
                    using (SqlDataReader reader = command.ExecuteReader())
                    {
                        while (reader.Read())
                        {
                            Console.WriteLine("{0} {1}", reader.GetString(0), reader.GetString(1));
                        }
                    }
                }
            }
        }
        catch (SqlException e)
        {
            Console.WriteLine(e.ToString());
        }
        Console.WriteLine("\nDone. Press enter.");
        Console.ReadLine();
```

Opgaver

- øvelse 17.1 – 17.4

- Vejledning