

Working with Real Data Using Entity Framework Core



Gill Cleeren

CTO Xebia Microsoft Services Belgium

@gillcleeren

Overview



Introducing Entity Framework Core

Adding EF Core to the application

Using migrations

Adding seed data



Introducing Entity Framework Core





**Nearly all web applications
you build will need data from a database.**





While we can use low-level ADO.NET combined with SQL statements, we will use Entity Framework Core.



Introducing Entity Framework Core

ORM

LINQ

Lightweight & cross-platform

Open-source

SQL Server & other relational and non-relational DB support

Code-first



EF Core



What EF Core Does for You

Class

```
public class Pie
{
    public int PieId { get; set; }
    public string? Name { get; set; }
    public string? Description { get; set; }
}
```

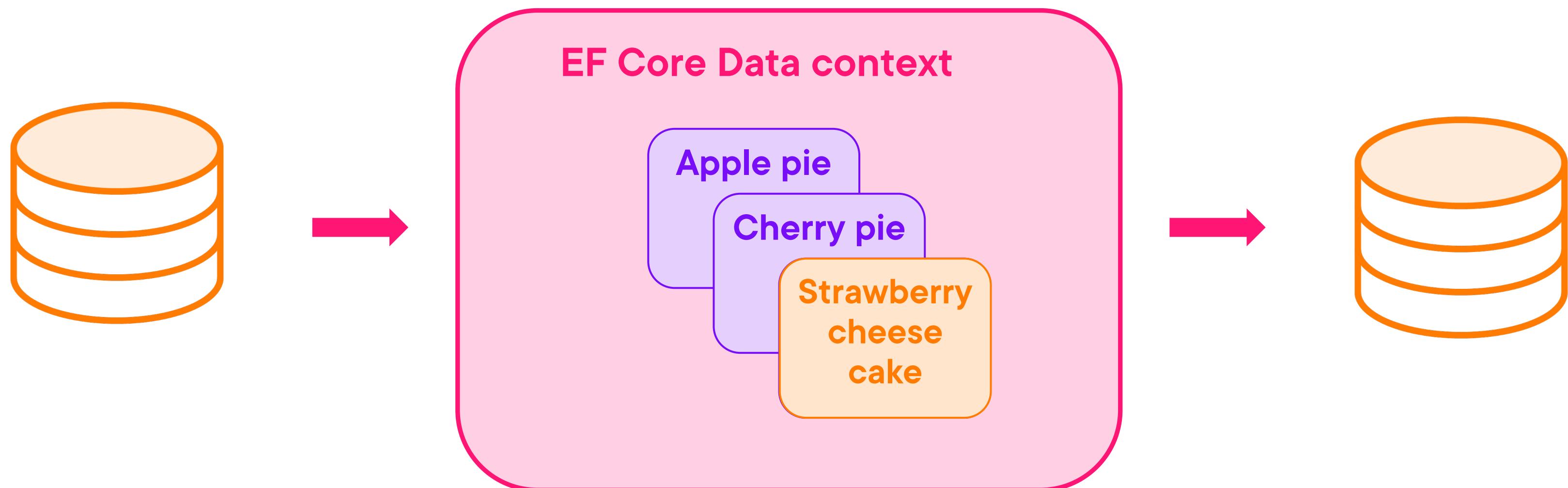


Table

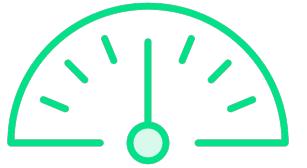
Pield	Int (PK)
Name	String
Description	string



The EF Core Change Tracker



Using EF Core



Speed of development



Can work with SQL statements



But... can sometimes be less performant than raw SQL



Adding EF Core to the Application



Adding EF Core to the Application

Packages

Domain classes

Database context

Application configuration



`Microsoft.EntityFrameworkCore.SqlServer`

◀ **SQL Server package**

`Microsoft.EntityFrameworkCore.Tools`

◀ **Helper package for Package Manager Console**



Adding EF Core to the Application

Packages

Domain classes

Database context

Application configuration



Domain Classes

```
public class Pie
{
    public int PieId { get; set; }
    public string Name { get; set; }
    public string? ShortDescription { get; set; }
    public decimal Price { get; set; }
    public int CategoryId { get; set; }
    public Category Category { get; set; }
}
```



Creating the Mapping

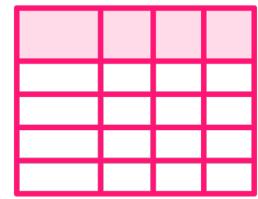


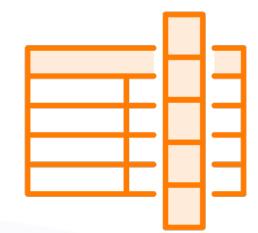
Table name and column name



Field will become primary key



CategoryId will become foreign key



Column types used in database



Adding EF Core to the Application

Packages

Domain classes

Database context

Application configuration



The Database Context

```
public class BethanysPieShopDbContext : DbContext
{
    public BethanysPieShopDbContext
        (DbContextOptions<BethanysPieShopDbContext> options)
        : base(options)
    {
    }

    public DbSet<Pie> Pies { get; set; }
}
```



Adding EF Core to the Application

Packages

Domain classes

Database context

Application configuration



```
{  
  "ConnectionStrings": {  
    "BethanysPieShopDbContextConnection":  
      "Server=(localdb)\\mssqllocaldb;  
      Database=BethanysPieShop;  
      Trusted_Connection=True;  
      MultipleActiveResultSets=true"  
  }  
}
```

Adding the Connection String

appSettings.json

Read automatically by default



```
builder.Services.AddDbContext<BethanysPieShopDbContext>(  
    options => {  
        options.UseSqlServer(  
            builder.Configuration["ConnectionStrings:BethanysPieShopDbContextConnection"]);  
    }  
);
```

Registering the Database Context

AddDbContext is an extension method



Demo



Adding the required packages

Creating the DbContext

Changing the application configuration



```
_bethanysPieShopDbContext.Pies.Include(c => c.Category).Where(p => p.IsPieOfTheWeek);
```

Querying for Data Using LINQ



Adding New Items

```
foreach (ShoppingCartItem? shoppingCartItem in shoppingCartItems)
{
    var orderDetail = new OrderDetail
    {
        Amount = shoppingCartItem.Amount,
        PieId = shoppingCartItem.Pie?.PieId,
        Price = shoppingCartItem.Pie?.Price
    };

    order.OrderDetails.Add(orderDetail);
}

_bethanysPieShopDbContext.Orders.Add(order);

_bethanysPieShopDbContext.SaveChanges();
```



Demo



Creating the repository



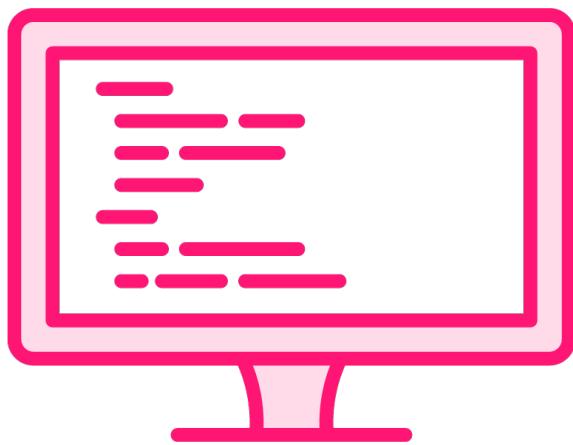
Using Migrations



**Using EF Core Migrations,
code can be generated to
bring the database in sync
with code model.**



Creating an Initial Migration



Database migration

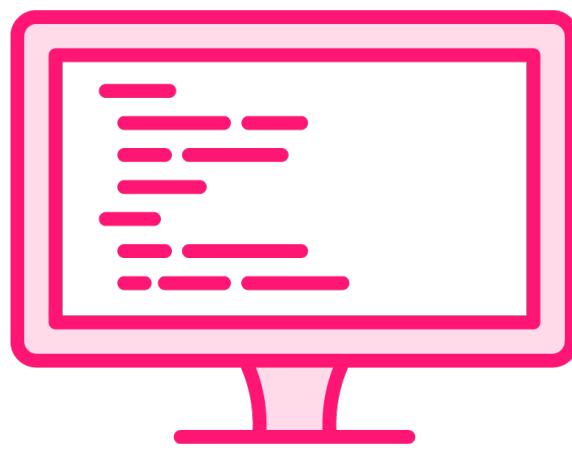
Package Manager Console

Commands

```
>add-migration <MigrationName>
```



Creating the Database



Database migration
Package Manager Console



Create database



Database

Commands
>update-database



Demo



Creating the initial migration
Creating the database



Adding Seed Data



Demo



Adding seed data



Summary



EF Core is a lightweight ORM

Use LINQ to interact with the database

Migrations are used to bring model and database in sync



Up Next:

Navigating through the site

