Code Academy

Lecturer

Rokas Slaboševičius

Entity Framework

Data

Today you will learn

01 Lazy Loading vs Eager Loading

02 Micro-orm vs ORM

03 Transactions

04 Stored procedures

Lazy Loading vs Eager Loading

Lazy loading:

When parental objects are retrieved from the database, the children are not retrieved together and are retrieved from the DB when requested.

Lazy Loading vs Eager Loading

Lazy loading configuration:

Download the Microsoft.EntityFrameworkCore.Proxies package.

```
protected override void OnConfiguring(DbContextOptionsBuilder options)
    => options
    .UseLazyLoadingProxies()
    .UseSqlServer($"Server=localhost;Database=EntityFrameworkCore;Trusted_Connection=True;");
```

Also, the classes you want to lazy load need to be set to virtual so that EF can override and create its own instance during lazy loading.

```
public class Book
{
    public Guid Id { get; set; }
    public string Name { get; set; }
    public virtual List<Page> Pages { get; set; }
    public virtual List<Category> Categories { get; set; }
```

Lazy Loading vs Eager Loading

Lazy loading:

On Select Book objects from the database

```
static void Main(string[] args)
{
    using var dbContext = new BookContext();
    var books = dbContext.Books.Select(x => x);


    foreach (var book in books)
    {
        Console.WriteLine(book.Name);
        foreach(var page in book.Pages)
        {
            Console.WriteLine(page.Content);
        }
    }
}
```

the following query is generated:

"SELECT [b].[Id], [b].[Name] FROM [Books] AS [b]"

Lazy Loading vs Eager Loading

Eager loading:

When parent objects are extracted from the database, the children are extracted together.

Lazy Loading vs Eager Loading

Eager loading:

On Select Book objects from the database

```
static void Main(string[] args)
{
    using var dbContext = new BookContext();
    var books = dbContext.Books.Include("Pages").Select(x => x);


    foreach (var book in books)
    {
        Console.WriteLine(book.Name);
        foreach(var page in book.Pages)
        {
            Console.WriteLine(page.Content);
        }
    }
}
```

the following query is generated:

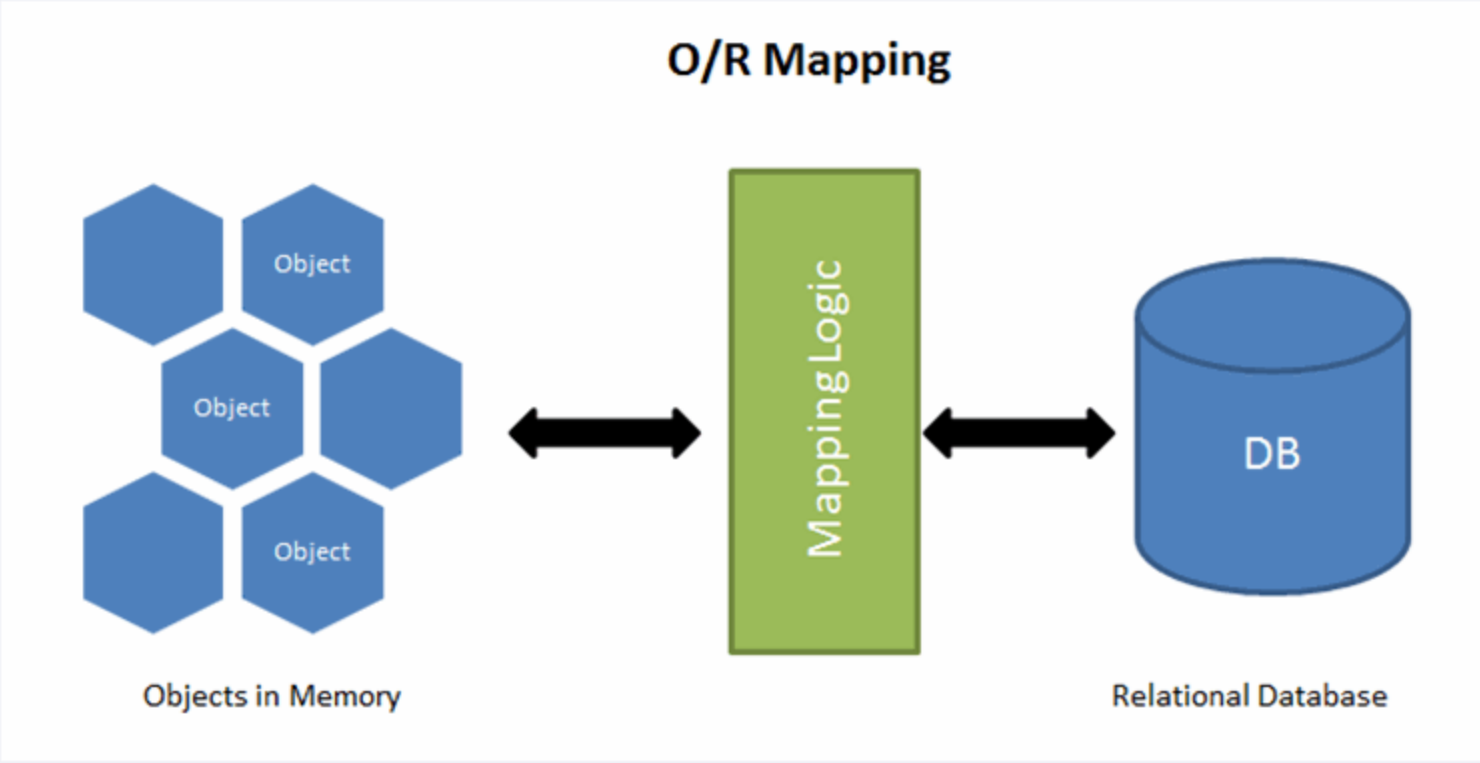
"SELECT [b].[Id], [b].[Name], [p].[Id], [p].[BookId], [p].[Content], [p].[Number]

FROM [Books] AS [b]

LEFT JOIN [Pages] AS [p] ON [b].[Id] = [p].[BookId]

ORDER BY [b].[Id], [p].[Id]"

Micro-orm vs ORM



Micro-orm vs ORM

	Micro ORM	ORM
Map queries to objects	✔	✔
Caching results	✘	✔
Change tracking	✘[1]	✔
SQL generation	✘[2]	✔
Identity management	✘	✔
Association management	✘	✔
Lazy loading	✘	✔
Unit of work support	✘	✔
Database migrations	✘	✔

Micro-orm vs ORM

Micro-orm limits:

Caching: Level 2 caching* is not supported. If you want level 2 caching, you will have to implement that functionality yourself.

Relationships: Micro-orm does not support one-to-one/many-to-many relationships between tables, so storing one object will not save its children in the same way as pulling parent objects from DB, the children will not be pulled together. It is possible to get around this limitation by merging the tables yourself, but this will require several queries to do the same thing.

Migrations: migrations are also not supported in the micro-orm functionality, whereas almost all ORM tools support migrations. Migration functionality can be achieved with additional tools, but this again requires additional third-party applications.

*Second level caching: https://docs.microsoft.com/en-us/archive/msdn-magazine/2011/september/data-points-second-level-caching-in-the-entity-framework-and-appfabric

Micro-orm vs ORM

Advantages of Micro-orm:

1. Simplicity
2. Speed

Micro-orm vs ORM

Most ORM frameworks are very large and complex, with hundreds of functionalities, and programmers often use only a fraction of them. The size and complexity of the framework comes with its own downsides - slower speed, higher required configuration.

With micro-orms, the configuration is very minimal and we create only the functionality we need.

Micro-orm vs ORM

Speed comparisons:

Method	Duration
Hand-coded (using a `SqlDataReader`)	47ms
Dapper `ExecuteMapperQuery`	49ms
PetaPoco	52ms
NHibernate SQL	104ms
Entity framework	631ms

Micro-orm vs ORM

When to choose a micro-orm?

1. When you want to achieve high speed communication with the DB.
2. When developing a "temporary" app - a common scenario in startup development - the first version of the app is written in haste, without regard to quality, in the hope of getting funding and rewriting the app neatly.
3. Legacy code with SqlDataReader(handwritten queries) - changing to micro-orm will make the code neater, easier to maintain and change.

When not to choose a micro-orm?

1. A large application is planned for the long term
2. You want to use links

Sql Stored procedures

Transaction:

A transaction is a logical unit of work that contains one or more SQL statements. A transaction is an atomic unit. The effects of all the SQL statements in a transaction can be either all committed (applied to the database) or all rolled back (undone from the database)

Stored Procedure:

SQL Server stored procedures are used to group one or more Transact-SQL statements into logical units.

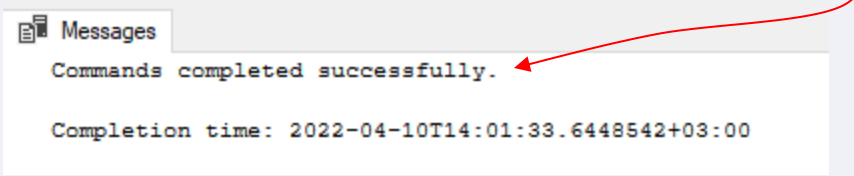
Sql Stored procedures

A Stored procedure can be thought of as a method in C#. It is a set of commands and it is invoked using a name and passing parameters (if required)

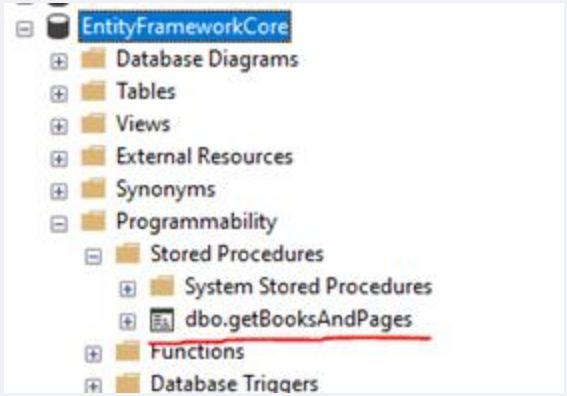
Sql Stored procedures

With the CREATE PROCEDURE function we describe what SP we want to create, by running this query we should get the following

```
CREATE PROCEDURE getBooksAndPages
AS
 BEGIN
    SELECT [b].[Id], [b].[Name], [p].[Id], [p].[BookId], [p].[Content], [p].[Number]
    FROM [Books] AS [b]
    LEFT JOIN [Pages] AS [p] ON [b].[Id] = [p].[BookId]
    ORDER BY [b].[Id], [p].[Id];
END;
```



Let's refresh and see if SP has appeared



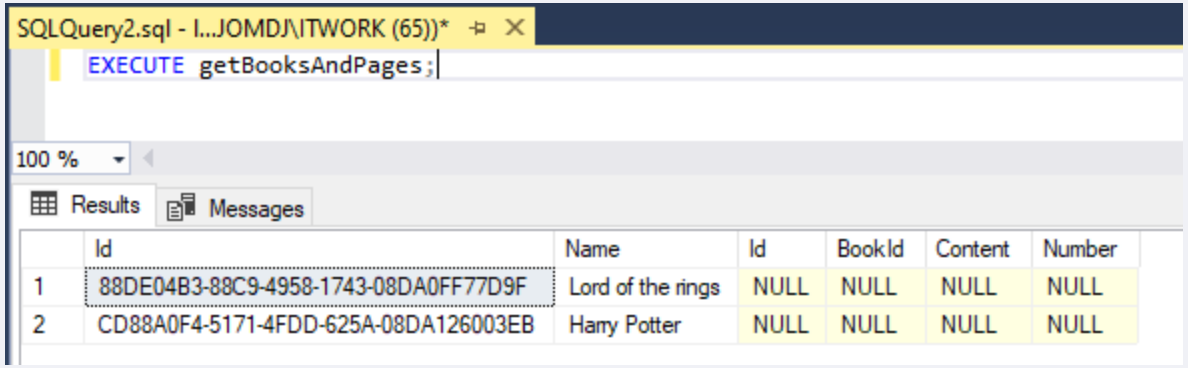
Sql Stored procedures

To run the SP you created, just write:

"EXECUTE <spName>"

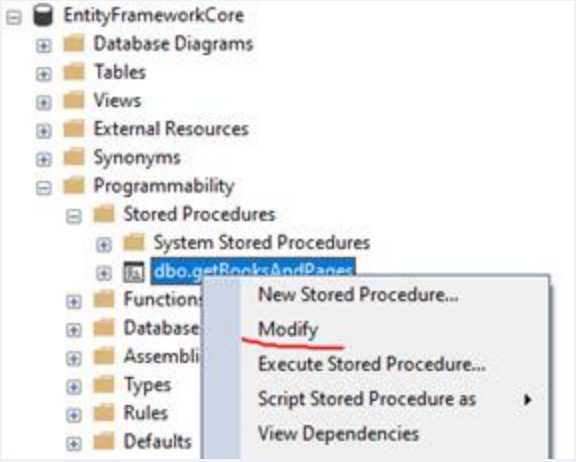
or

"EXEC <spName>"



	Id	Name	Id	BookId	Content	Number
1	88DE04B3-88C9-4958-1743-08DA0FF77D9F	Lord of the rings	NULL	NULL	NULL	NULL
2	CD88A0F4-5171-4FDD-625A-08DA126003EB	Harry Potter	NULL	NULL	NULL	NULL

Sql Stored procedures



To modify, select Modify

After receiving this window and modifying the SP as desired, we run the function:

SQLQuery3.sql - I...JOMDJ\ITWORK (73)) | SQLQuery2.sql - I...JOMDJ\ITWORK (65))*

```
USE [EntityFrameworkCore]
GO
/****** Object:  StoredProcedure [dbo].[getBooksAndPages]    Script Date: 2022-04-10 14:05:45 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[getBooksAndPages]
AS
 BEGIN
    SELECT [b].[Id], [b].[Name], [p].[Id], [p].[BookId], [p].[Content], [p].[Number]
    FROM [Books] AS [b]
    LEFT JOIN [Pages] AS [p] ON [b].[Id] = [p].[BookId]
    ORDER BY [b].[Id], [p].[Id];
END;
```

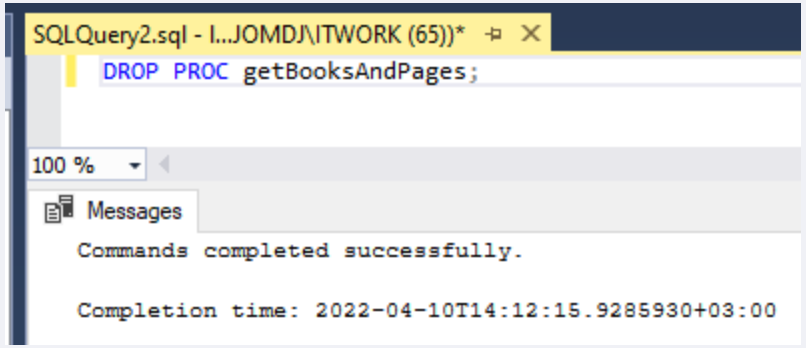
Sql Stored procedures

To delete an SP, write

"DROP PROCEDURE <spName>"

or

"DROP PROC <spName>"



Sql Stored procedures

We create an SP with parameters

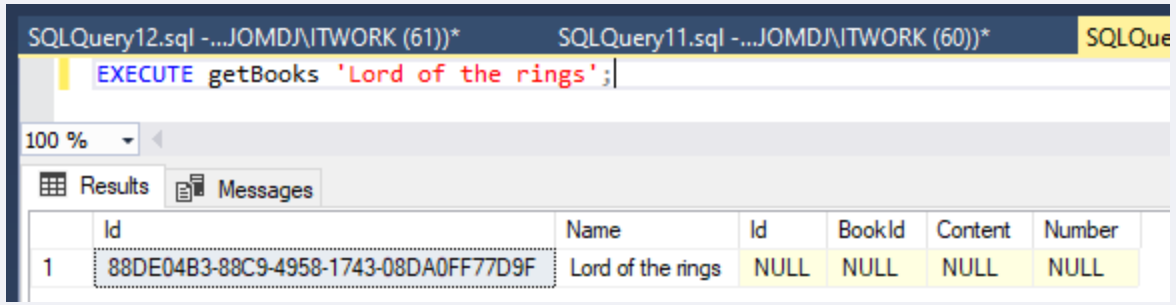
Add the @book_name parameter and use it in the WHERE clause. You can also create more parameters via comma.

```
USE [EntityFrameworkCore]
GO
/****** Object:  StoredProcedure [dbo].[getBooks]    Script Date: 2022-04-10 15:28:26 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[getBooks](@book_name AS NVARCHAR(100))
AS
BEGIN
    SELECT [b].[Id], [b].[Name], [p].[Id], [p].[BookId], [p].[Content], [p].[Number]
    FROM [Books] AS [b]
    LEFT JOIN [Pages] AS [p] ON [b].[Id] = [p].[BookId]
    WHERE [b].[Name] = @book_name
    ORDER BY [b].[Id], [p].[Id]
END;
```

Sql Stored procedures

Execute'iname SP with parameter:

It would also be possible to pass more parameters via comma if required



Sql Stored procedures

We can also create optional parameters by assigning them an initial value

```
USE [EntityFrameworkCore]
GO
/****** Object:  StoredProcedure [dbo].[getBooks]    Script Date: 2022-04-10 15:28:26 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[getBooks](@book_name AS NVARCHAR(100), @number AS DECIMAL = 10)
AS
BEGIN
    SELECT [b].[Id], [b].[Name], [p].[Id], [p].[BookId], [p].[Content], [p].[Number]
    FROM [Books] AS [b]
    LEFT JOIN [Pages] AS [p] ON [b].[Id] = [p].[BookId]
    WHERE [b].[Name] = @book_name
    ORDER BY [b].[Id], [p].[Id]
END;
```

Sql Stored procedures

Such an SP can be execute'int with or without the second parameter

```
EXECUTE getBooks 'Lord of the rings';
```

100 %

Results | Messages

	Id	Name	Id	BookId	Content	Number
1	88DE04B3-88C9-4958-1743-08DA0FF77D9F	Lord of the rings	NULL	NULL	NULL	NULL

SQLQuery12.sql -...JOMDJ\ITWORK (61))* SQLQuery11.sql -...JOMDJ\ITWORK (60))* SQLQuery

```
EXECUTE getBooks 'Lord of the rings', 50;
```

100 %

Results | Messages

	Id	Name	Id	BookId	Content	Number
1	88DE04B3-88C9-4958-1743-08DA0FF77D9F	Lord of the rings	NULL	NULL	NULL	NULL

Task 1

- Rework the tasks in the first DB lecture to use **stored procedures**
- Queries that used some kind of conditional sentences have to get the values via SP parameters.

Headline

www.youtube.com

Useful information