EF Core

ORM

- Object-relational mapping (ORM) is a way to align programming code with database structures. ORM uses metadata descriptors to create a layer between the programming language and a relational database.

- تسمح للمطورين بالعمل مع DB باستخدام dot.net object

WHAT ORM?

- ORM stands for Object to Relational Mapping
    1. Object: It is class's that we have in our programming language (c#, python etc.)
    2. Relational: this is Relational Database Manager System like MS-SQL, MySQL etc.
    3. Mapping: This is the part which bridges between objects and tables.
- ORM is a technique that lets you query and manipulate data from a database using an object-oriented paradigm.

WITHOUT ORM

```
book_list = new List();
sql = "SELECT book FROM library WHERE author = 'Bhrugen'";
data = query(sql); // This is just a dummy ...
while (row = data.next())
{
    book = new Book();
    book.setAuthor(row.get('author');
    book_list.add(book);
}
```

WITH ORM

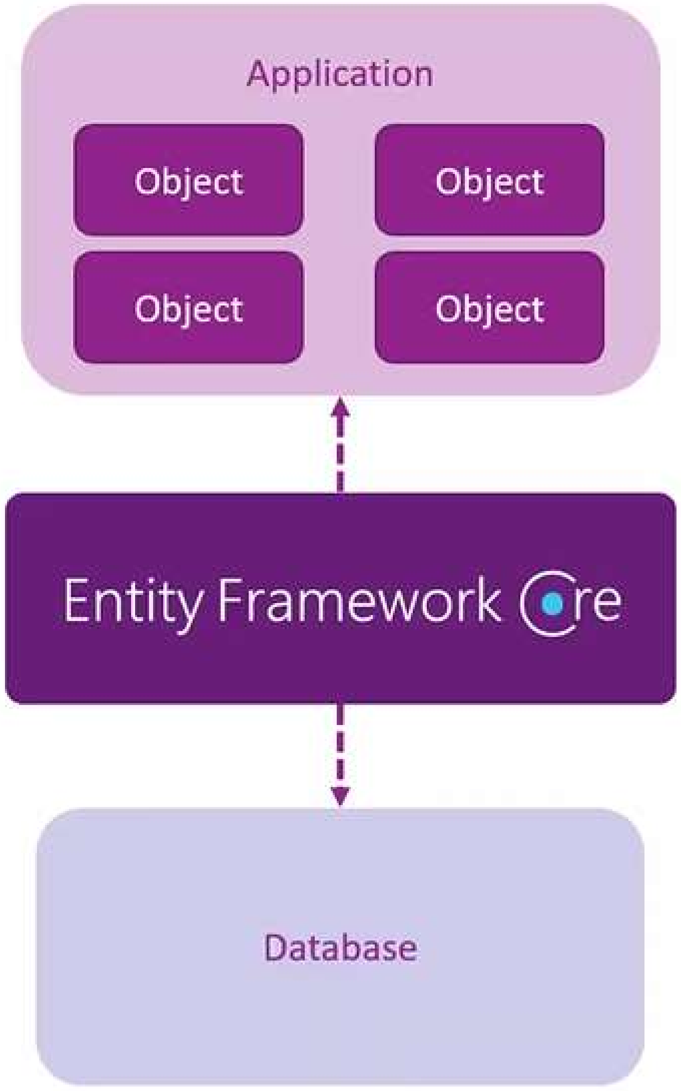
```
book_list = BookTable.query(author = "Bhrugen");
```

ORM

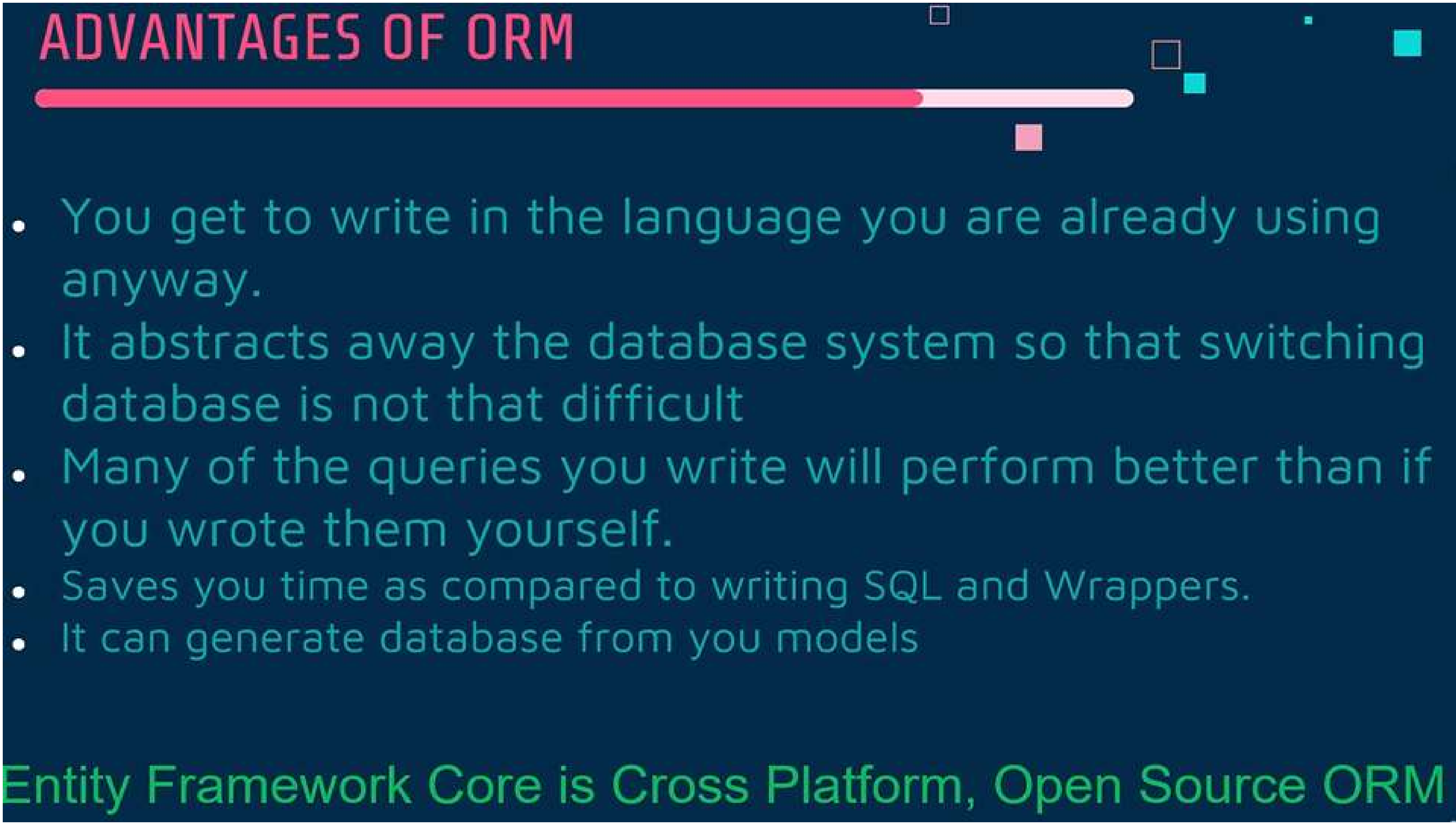
- يعني بدل من استخدام الاجراء المخزن او التعليمات نفسها، يتم التعامل مع قاعدة البيانات كأنها شروط ضمن الدوت نت

ORM

- سابقا تم استخدام ado dot net للتعامل مع قواعد البيانات
- سابقا تم تفضيل استخدام الاجراء المخزن ع DB لتجنب sql injection
- يتم التعامل مع DB بطريقة بسيطة بدون تعمق باستخدام EF
- Ado اسرع بالأداء راي معظم المطورين
- لكن framework التعامل معها أبسط واسهل وتفضل على Ado
- تتعامل مع معظم أنواع قواعد البيانات



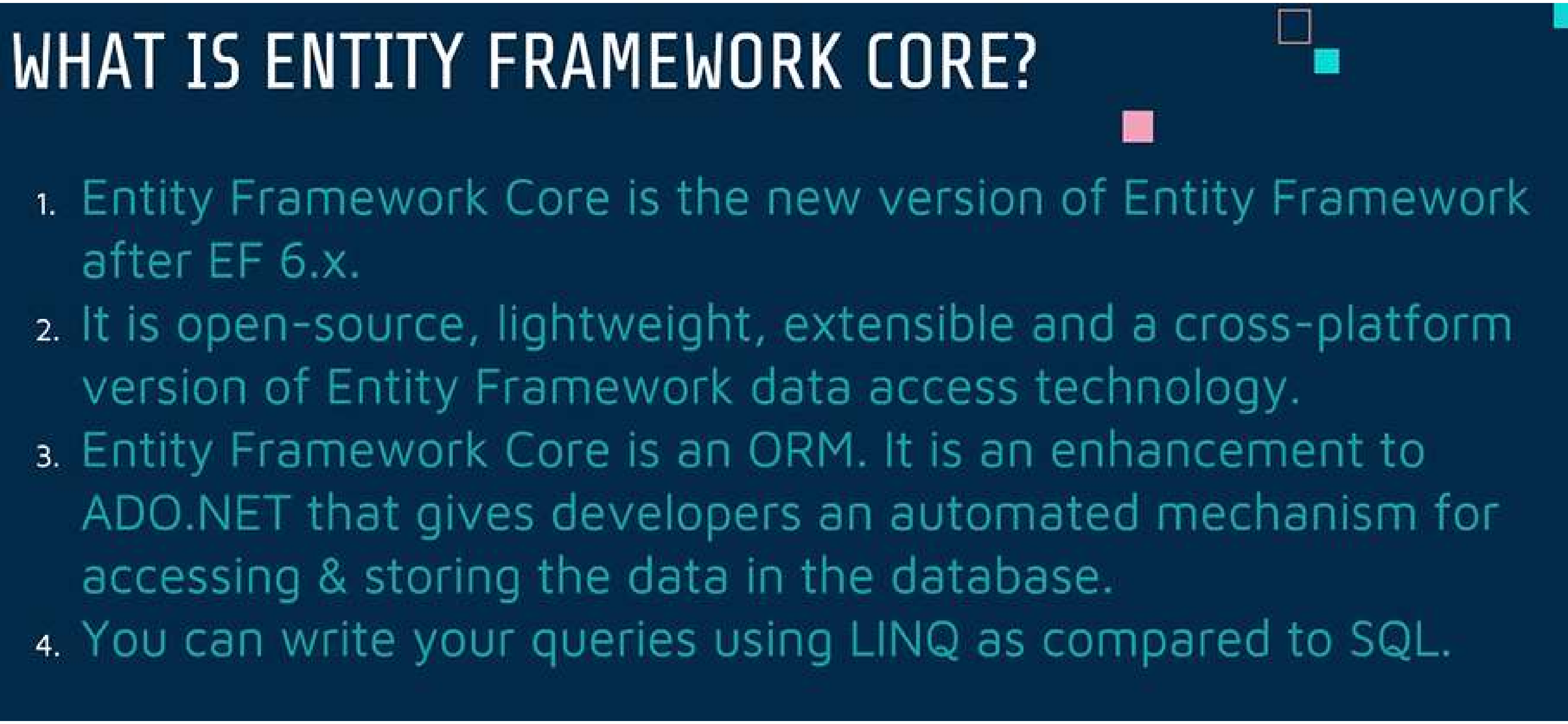
طبقة بين التطبيق وقاعدة البيانات.



ADVANTAGES OF ORM

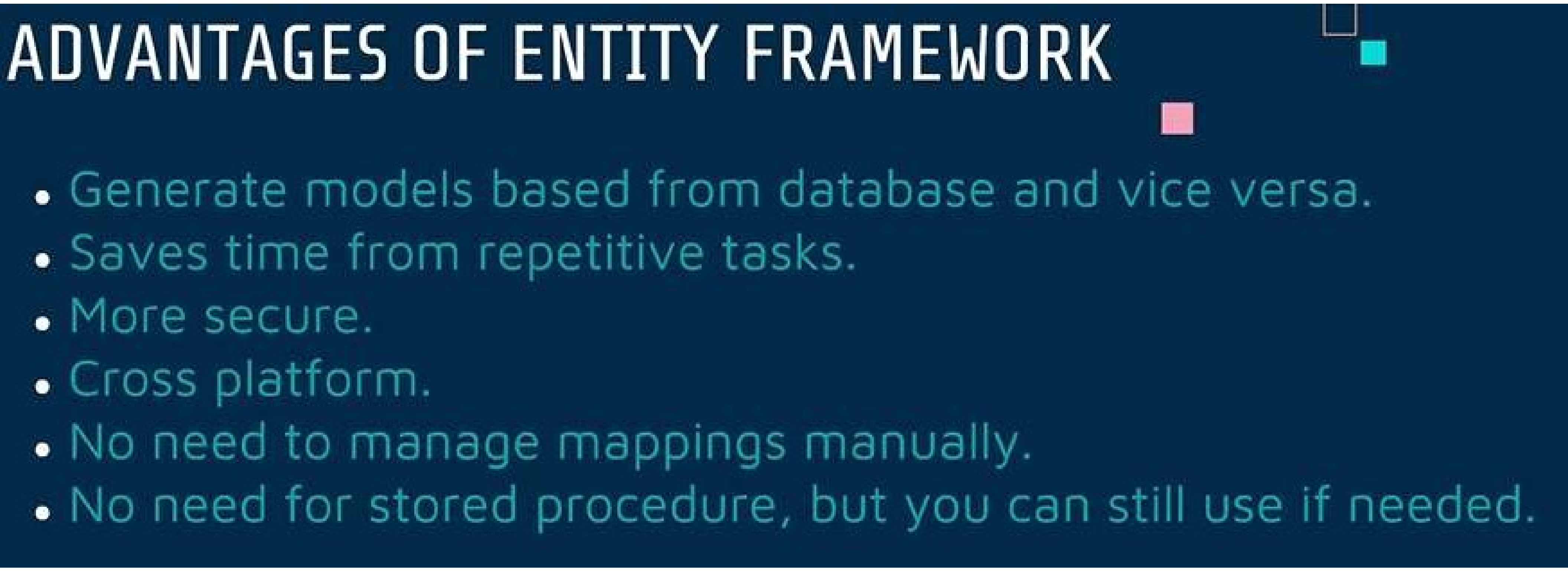
- You get to write in the language you are already using anyway.
- It abstracts away the database system so that switching database is not that difficult
- Many of the queries you write will perform better than if you wrote them yourself.
- Saves you time as compared to writing SQL and Wrappers.
- It can generate database from you models

Entity Framework Core is Cross Platform, Open Source ORM



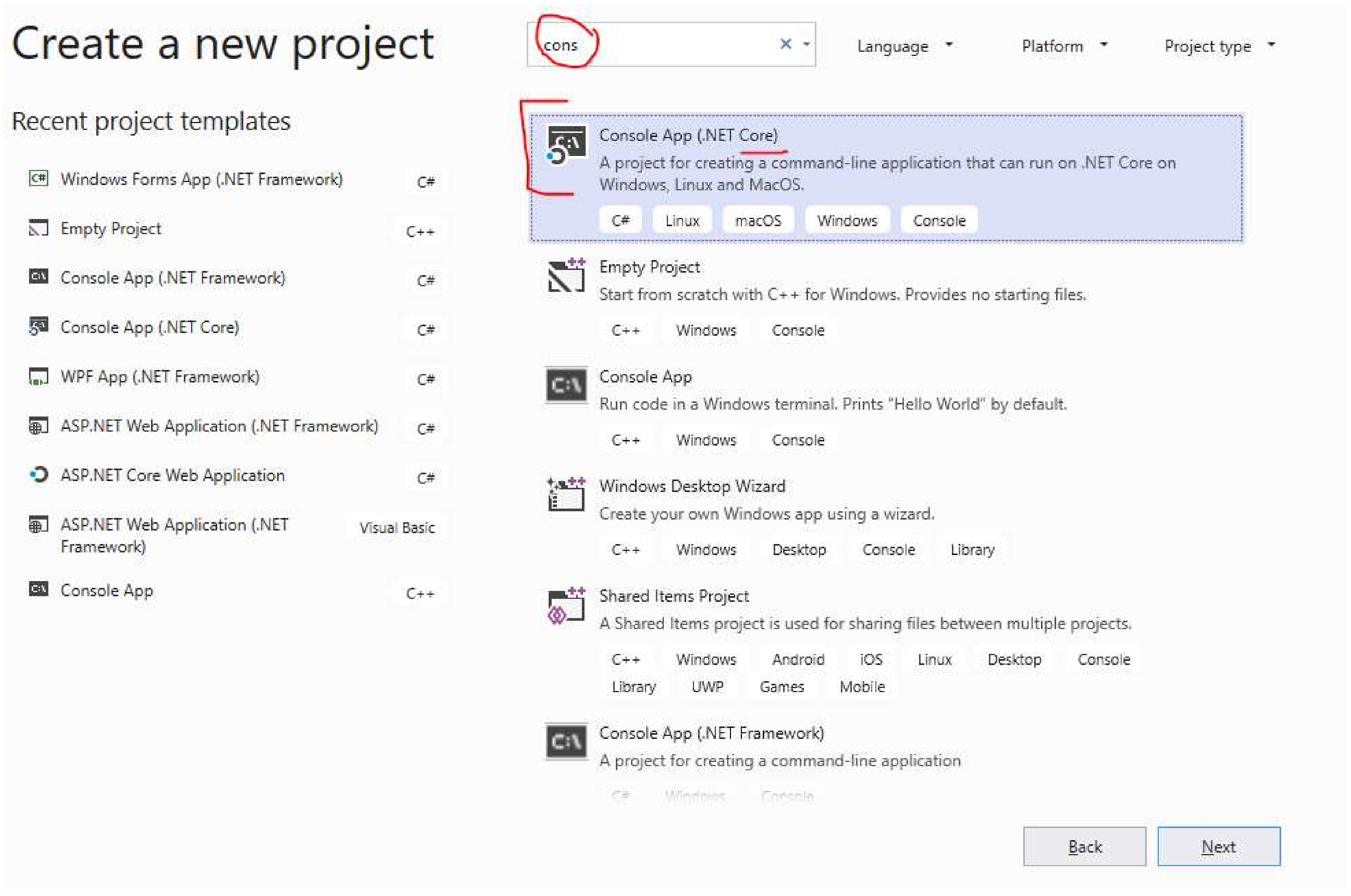
WHAT IS ENTITY FRAMEWORK CORE?

1. Entity Framework Core is the new version of Entity Framework after EF 6.x.
2. It is open-source, lightweight, extensible and a cross-platform version of Entity Framework data access technology.
3. Entity Framework Core is an ORM. It is an enhancement to ADO.NET that gives developers an automated mechanism for accessing & storing the data in the database.
4. You can write your queries using LINQ as compared to SQL.



ADVANTAGES OF ENTITY FRAMEWORK

- Generate models based from database and vice versa.
- Saves time from repetitive tasks.
- More secure.
- Cross platform.
- No need to manage mappings manually.
- No need for stored procedure, but you can still use if needed.


Create a new project

cons

Language Platform Project type

Recent project templates

Windows Forms App (.NET Framework)	C#
Empty Project	C++
Console App (.NET Framework)	C#
Console App (.NET Core)	C#
WPF App (.NET Framework)	C#
ASP.NET Web Application (.NET Framework)	C#
ASP.NET Core Web Application	C#
ASP.NET Web Application (.NET Framework)	Visual Basic
Console App	C++

Console App (.NET Core)
A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.
C# Linux macOS Windows Console

Empty Project
Start from scratch with C++ for Windows. Provides no starting files.
C++ Windows Console

Console App
Run code in a Windows terminal. Prints "Hello World" by default.
C++ Windows Console

Windows Desktop Wizard
Create your own Windows app using a wizard.
C++ Windows Desktop Console Library

Shared Items Project
A Shared Items project is used for sharing files between multiple projects.
C++ Windows Android iOS Linux Desktop Console Library UWP Games Mobile

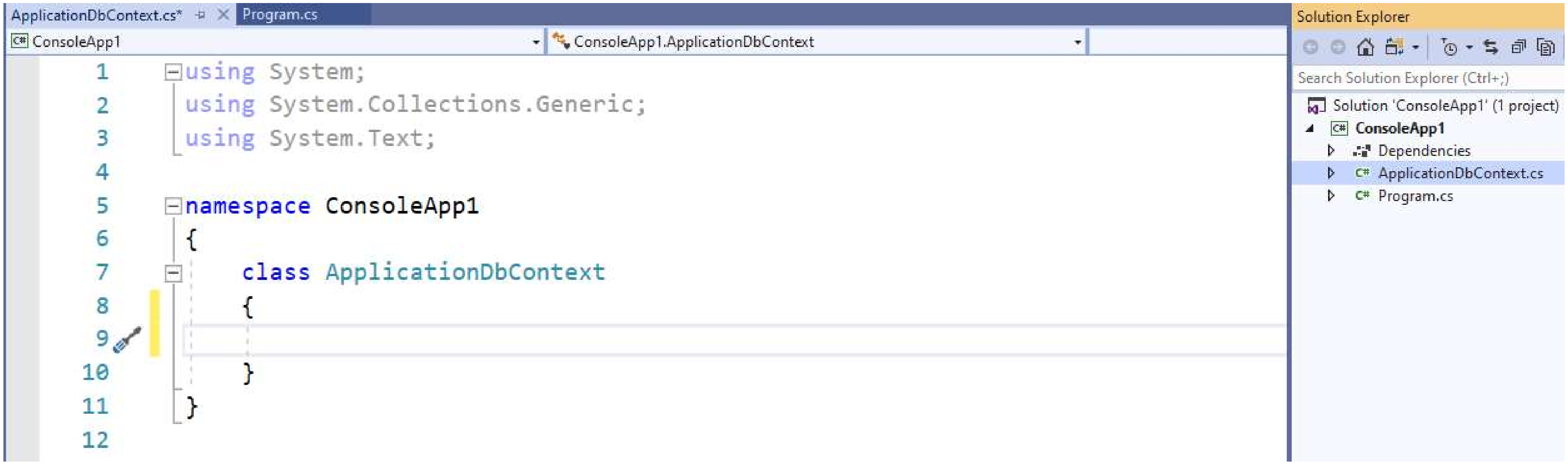
Console App (.NET Framework)
A project for creating a command-line application
C# Windows Console

Back Next

Microsoft.EntityFrameworkCore
Microsoft.EntityFrameworkCore.SqlServer
Microsoft.EntityFrameworkCore.Tools

نقوم بعمل صف يمثل حلقة الوصل بين Application and Database.

ApplicationDbContext



```
using System;
using System.Collections.Generic;
using System.Text;

namespace ConsoleApp1
{
    class ApplicationDbContext
    {

    }
}
```

ApplicationDbContext.cs*

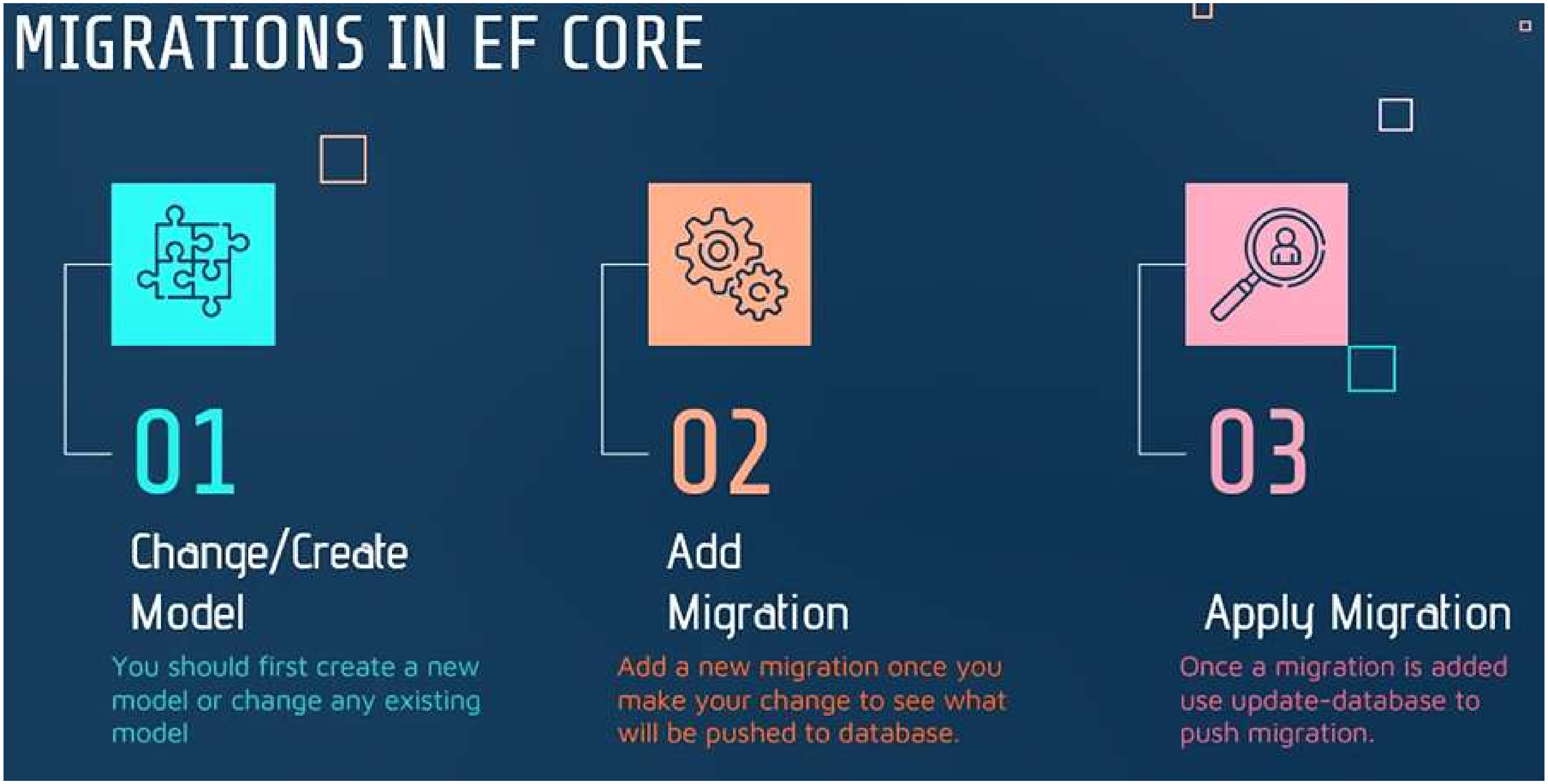
```
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Text;

namespace ConsoleApp1
{
    class ApplicationDbContext:DbContext
    {
        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            optionsBuilder.UseSqlServer("Data Source=.;Initial Catalog=SCSDBTest;Integrated 
Security=True");

        }
    }
}
```

Class Employee

```
Public class Employee
    {
        public int Id { get; set; }
        public string Name { get; set; }
    }
```



MIGRATIONS IN EF CORE

01

Change/Create Model

You should first create a new model or change any existing model

02

Add Migration

Add a new migration once you make your change to see what will be pushed to database.

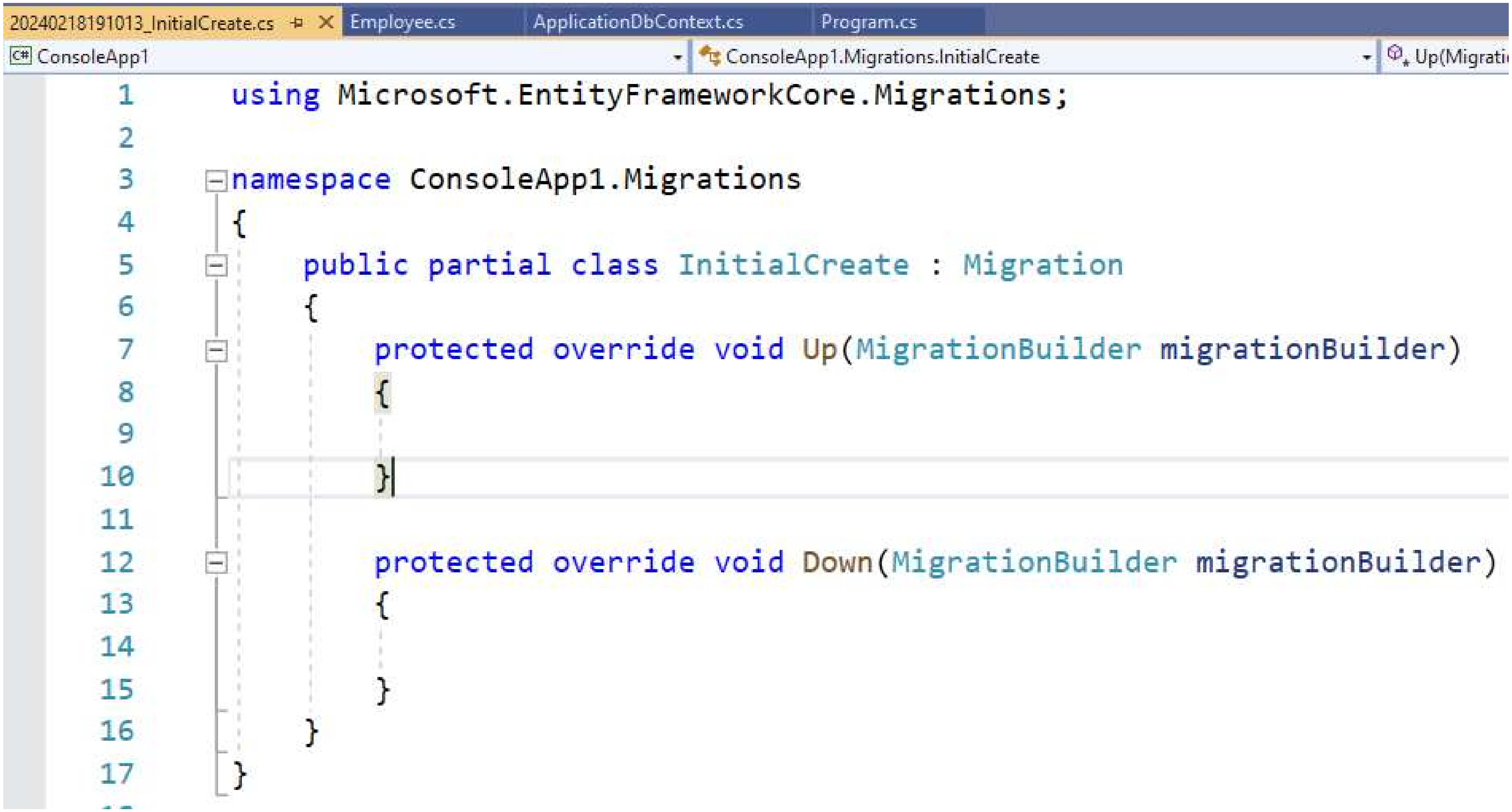
03

Apply Migration

Once a migration is added use update-database to push migration.

migration from console pakage عمل

```
PM> add-migration InitialCreate
```



20240218191013_InitialCreate.cs | Employee.cs | ApplicationDbContext.cs | Program.cs

ConsoleApp1 | ConsoleApp1.Migrations.InitialCreate | Up(Migrati

```
using Microsoft.EntityFrameworkCore.Migrations;

namespace ConsoleApp1.Migrations
{
    public partial class InitialCreate : Migration
    {
        protected override void Up(MigrationBuilder migrationBuilder)
        {

        }

        protected override void Down(MigrationBuilder migrationBuilder)
        {

        }
    }
}
```

Remove migration

لجعل الجدول يظهر مباشرة ضمن الاب نكتب

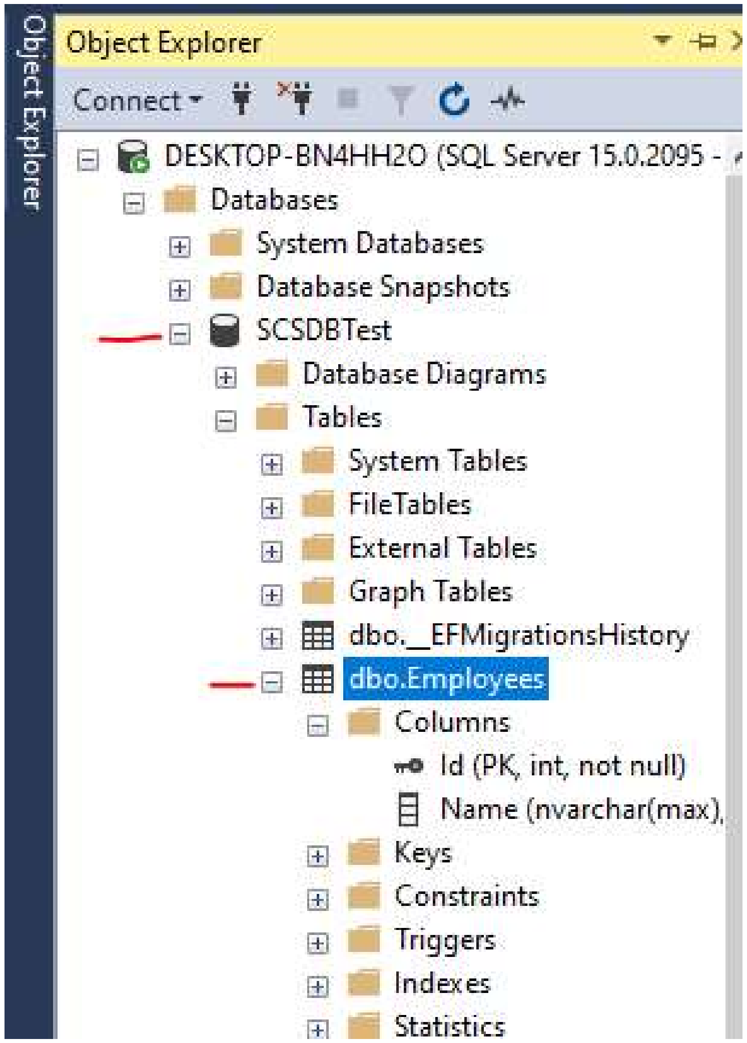
```
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Text;

namespace ConsoleApp2
{
    class ApplicationDbContext:DbContext
    {
        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            optionsBuilder.UseSqlServer("Data Source=.;Initial 
Catalog=SCSDBTest;Integrated Security=True");
        }
        public DbSet<Employee> Employees { get; set; }
    }
}
```

```
PM> add-migration InitialCreate
```

```
using Microsoft.EntityFrameworkCore.Metadata;
using Microsoft.EntityFrameworkCore.Migrations;
namespace ConsoleApp2.Migrations
{    public partial class InitialCreate : Migration
    {        protected override void Up(MigrationBuilder migrationBuilder)
        {            migrationBuilder.CreateTable(
                name: "Employees",
                columns: table => new
                {
Id = table.Column<int>(nullable: false)
.Annotation("SqlServer:ValueGenerationStrategy",SqlServerValueGenerationStrategy.IdentityColumn),
                    Name = table.Column<string>(nullable: true)
                },
                constraints: table =>
                {
                    table.PrimaryKey("PK_Employees", x => x.Id);
                });
        }
        protected override void Down(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.DropTable(  name: "Employees");
        }
    }
}
```

```
update-database
```



Object Explorer

Connect

DESKTOP-BN4HH2O (SQL Server 15.0.2095 -

Databases

System Databases

Database Snapshots

SCSDBTest

Database Diagrams

Tables

System Tables

FileTables

External Tables

Graph Tables

dbo.__EFMigrationsHistory

dbo.Employees

Columns

Id (PK, int, not null)

Name (nvarchar(max),

Keys

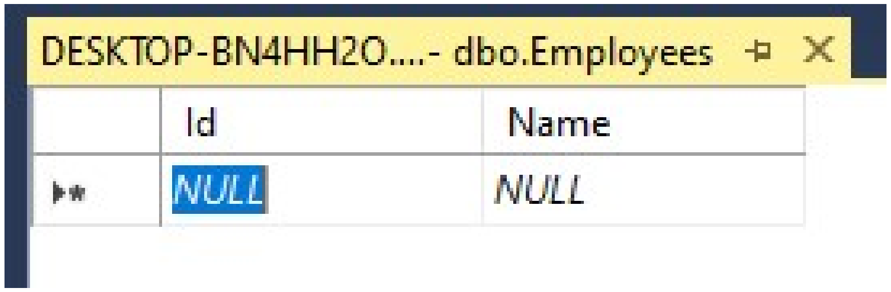
Constraints

Triggers

Indexes

Statistics

قاعدة البيانات والجدول فارغ



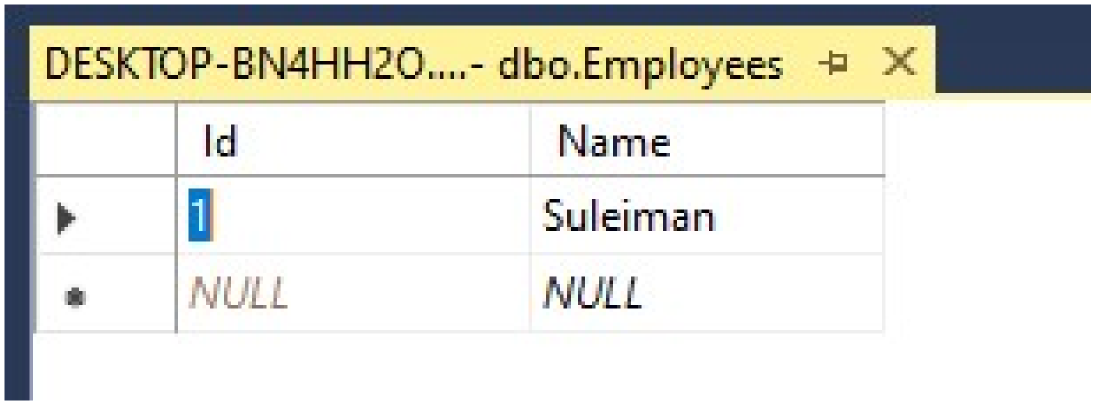
DESKTOP-BN4HH2O....- dbo.Employees

	Id	Name
▸*	NULL	NULL

تم انشاء قاعدة البيانات والجدول
نقوم بإضافة سجل ضمن الجدول

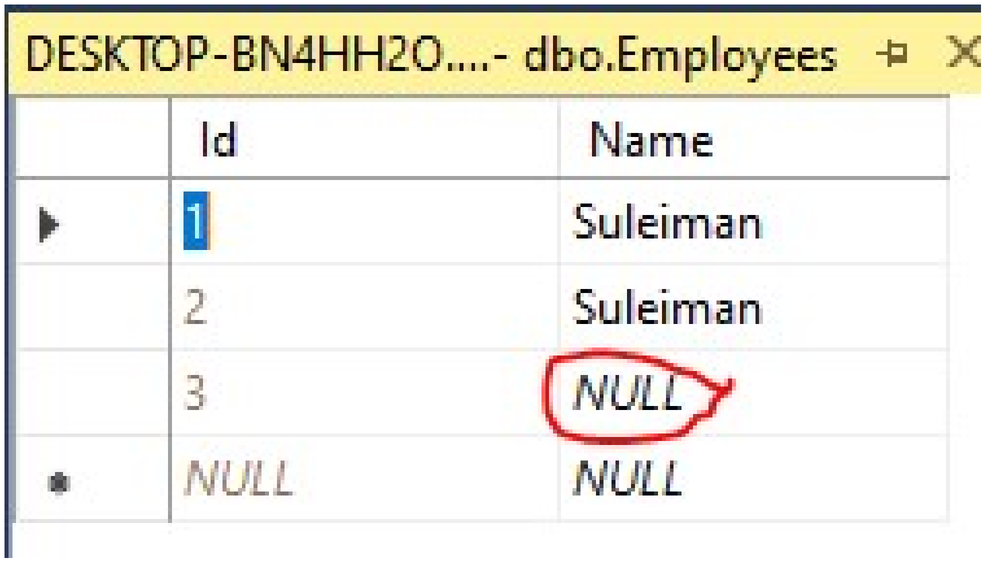
```
static void Main(string[] args)
       {
           var _context = new ApplicationDbContext();
           var employee = new Employee
           {
               Name="Suleiman"
           };
           _context.Employees.Add(employee); // add to the memory
           _context.SaveChanges(); // confirm to db
       }
```

Run F5


DESKTOP-BN4HH2O....- dbo.Employees

	Id	Name
▶	1	Suleiman
●	NULL	NULL

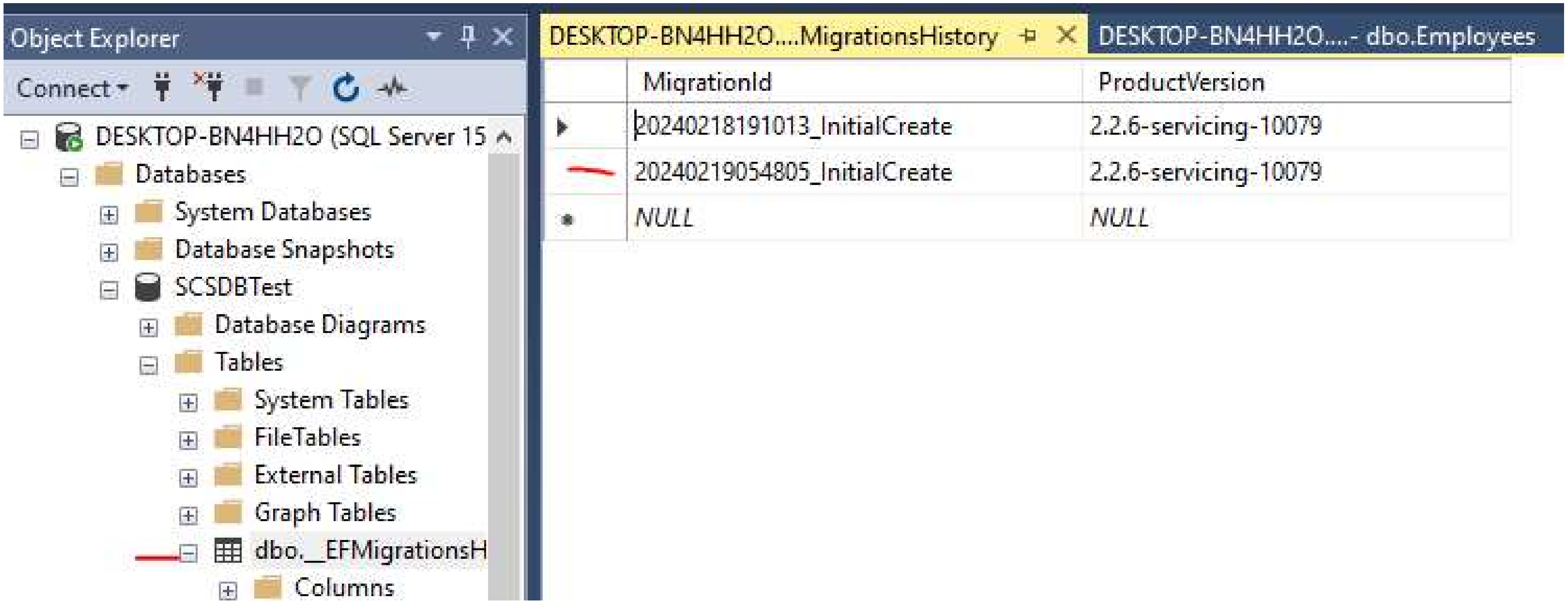
```
static void Main(string[] args)
        {
            var _context = new ApplicationDbContext();
            var employee = new Employee();
            //{
            //    Name="Suleiman"
            //};
            _context.Employees.Add(employee); // add to the memory
            _context.SaveChanges(); // confirm to db
        }
```



DESKTOP-BN4HH2O....- dbo.Employees

	Id	Name
▶	1	Suleiman
	2	Suleiman
	3	NULL
●	NULL	NULL

جدول أسماء ملفات المايغريشن التي قمنا بعملها في المشروع



MigrationId	ProductVersion
20240218191013_InitialCreate	2.2.6-servicing-10079
20240219054805_InitialCreate	2.2.6-servicing-10079
NULL	*NULL*