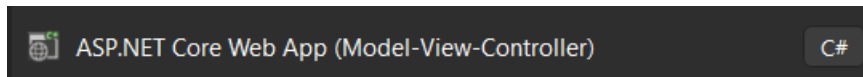


# How to Migrate/Create a Database using Microsoft Entity Framework

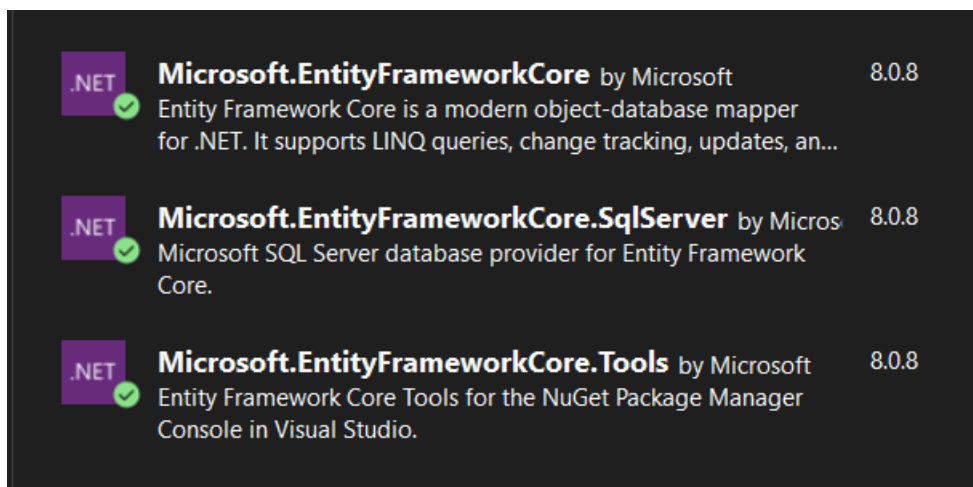
USING CODE TO CREATE THE DATABASE AND TABLES

REMINDER: This database is local and you need to add instructions on how to migrate the database in the readme if using the local SQL server.

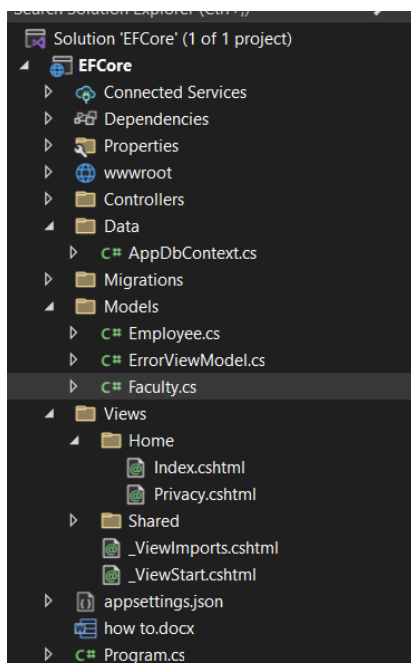
Create a Asp.net Core Web App project.



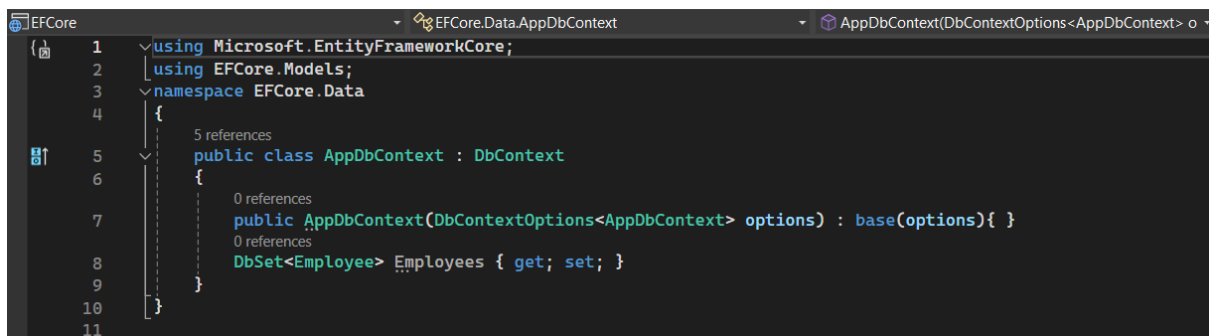
Install these packages using NuGet package manager.



Create a 'Data' folder.

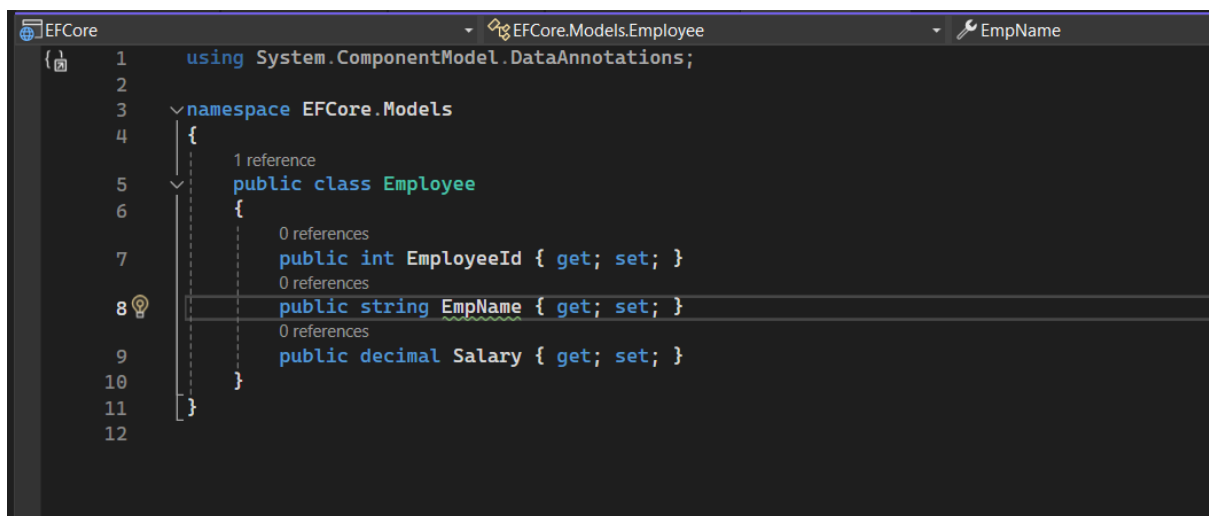


Add a .cs file called AppDbContext (make the DbSet variables public)



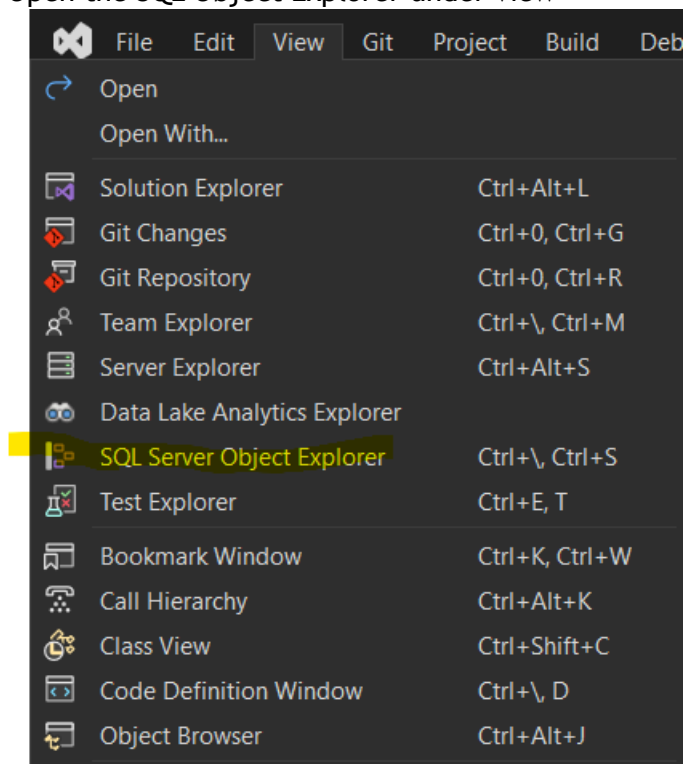
```
1 using Microsoft.EntityFrameworkCore;
2 using EFCore.Models;
3 namespace EFCore.Data
4 {
5     public class AppDbContext : DbContext
6     {
7         public AppDbContext(DbContextOptions<AppDbContext> options) : base(options) { }
8         DbSet<Employee> Employees { get; set; }
9     }
10 }
11
```

Under 'Models' create a file called Employee

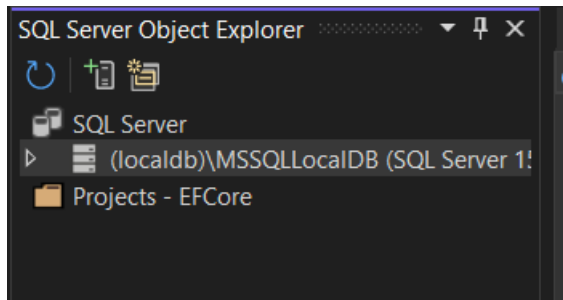


```
1 using System.ComponentModel.DataAnnotations;
2
3 namespace EFCore.Models
4 {
5     public class Employee
6     {
7         public int EmployeeId { get; set; }
8         public string EmpName { get; set; }
9         public decimal Salary { get; set; }
10     }
11 }
12
```

Open the SQL Object Explorer under view

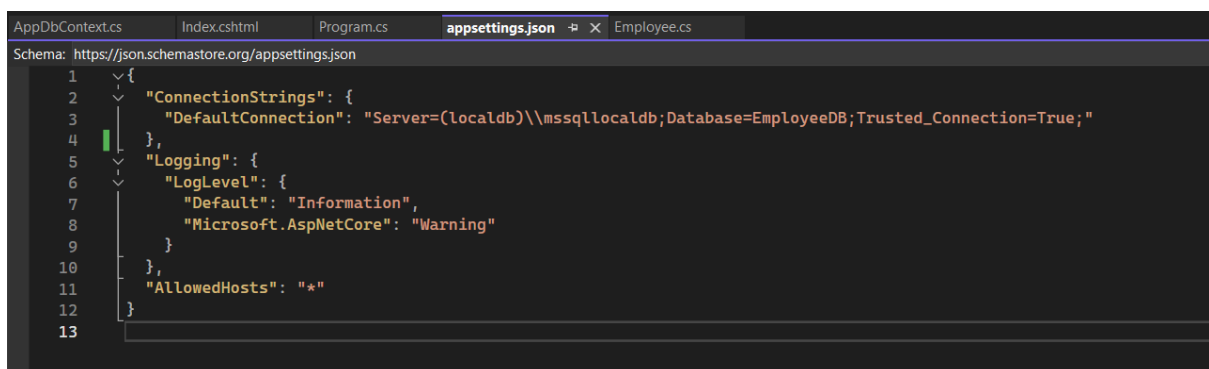


Open it and keep it open.



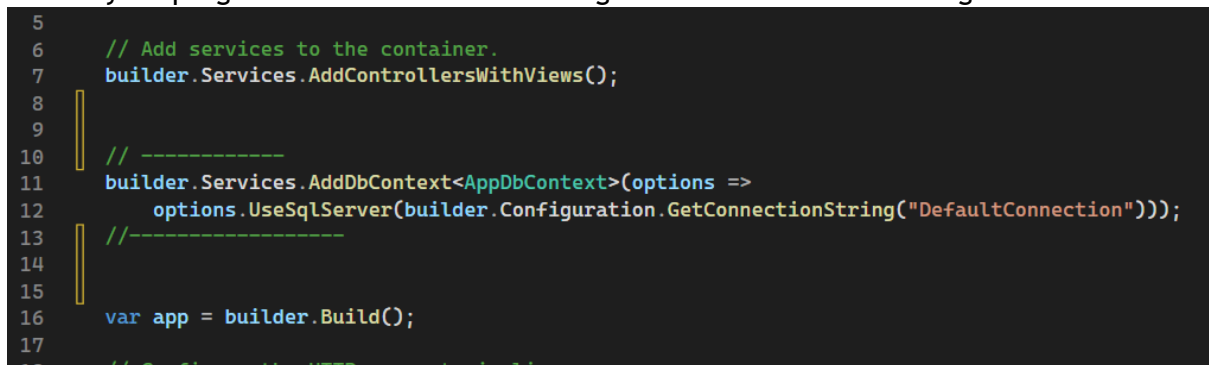
Now open appsettings and it should something like this

EmployeeDB will be the name of the database created.

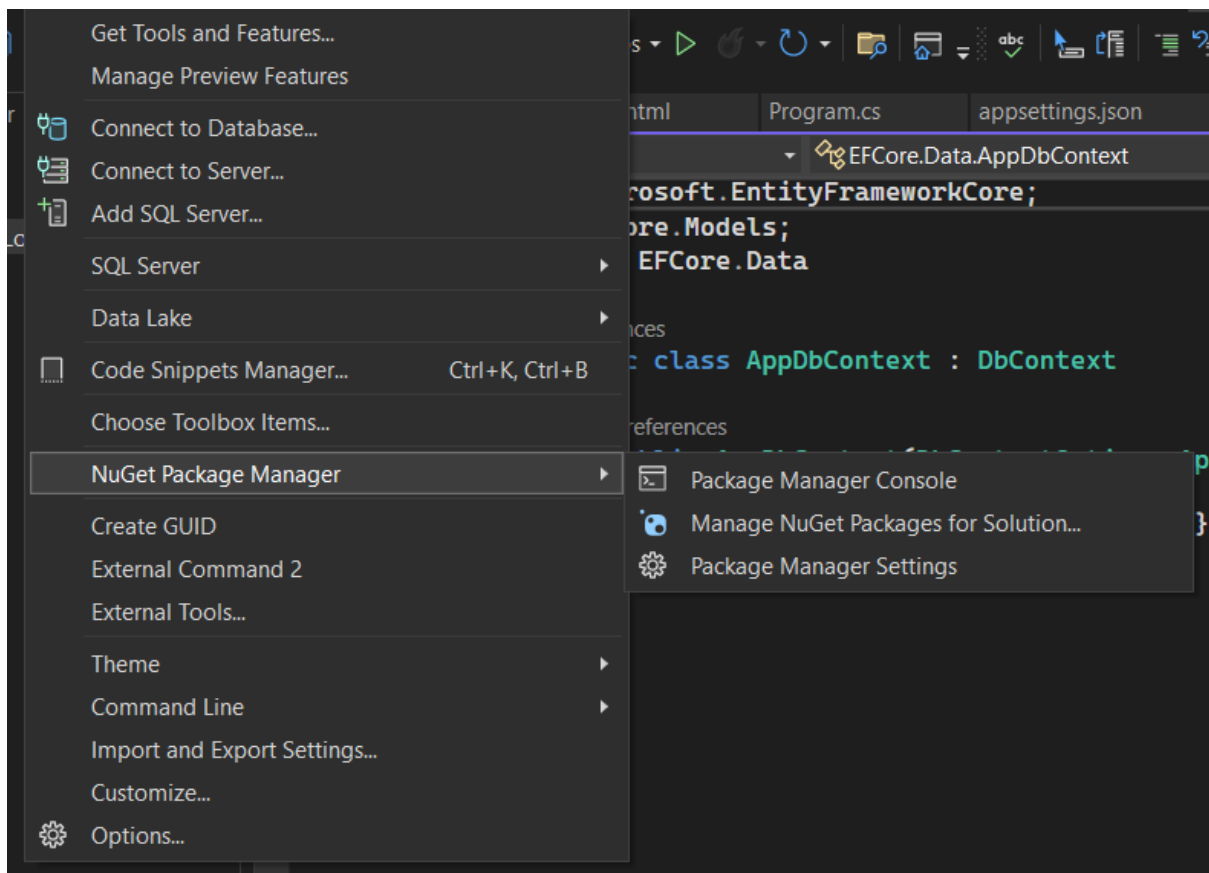


\\mssqllocaldb should match what is shown in the server object explorer.

Now in your program.cs file add the following code between two existing lines of code



Now open the package manager Console. (Under Tools/Nuget Package Manager)



Open the package manager console and type.

```
PM> Add-migration IntialCreate
```

Either it will produce an error succeed

```
An error occurred while accessing the Microsoft.Extensions.Hosting services. Continuing without the application service provider. Error: Failed to load configuration from file 'C:\Users\...
Unable to create a 'DbContext' of type ''. The exception 'Unable to resolve service for type 'Microsoft.EntityFrameworkCore.DbContextOptions'1[EFCore.Data.AppDbContext]' while attempting to activate 'EFCore.Data.AppDbContext'.' was thrown while attempting to create an instance. For the different patterns supported at design time, see https://go.microsoft.com/fwlink/?linkid=851728
```

Figure 1 Error (error in your appsettings file or your employee key was not typed correctly. (it wants classnameld))

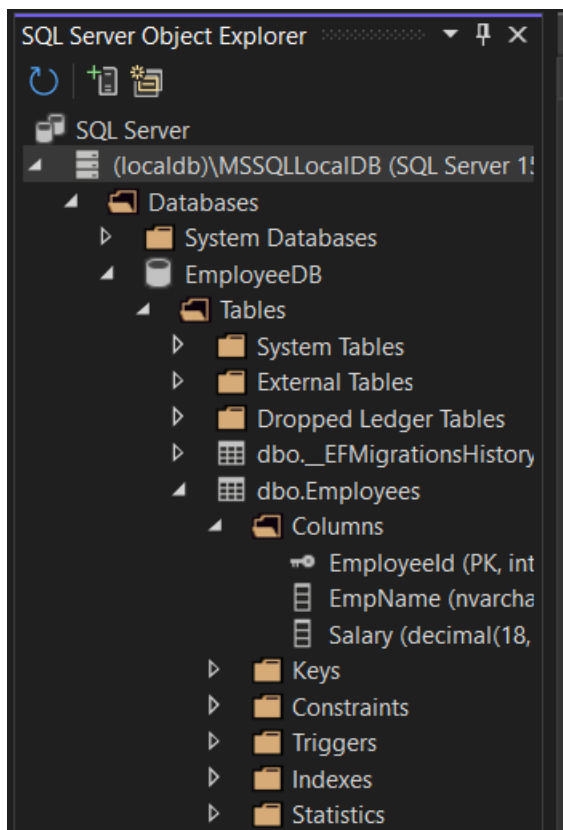
```
PM> Add-migration IntialCreate
Build started...
Build succeeded.
Microsoft.EntityFrameworkCore.Model.Validation[30000]
No store type was specified for the decimal property 'Salary' on entity type 'Employee'. This will cause values to be silently truncated if they do not fit in the default precision and scale. Explicitly specify the SQL server column type that can accommodate all the values in 'OnModelCreating' using 'HasColumnType', specify precision and scale using 'HasPrecision', or configure a value converter using 'HasConversion'.
Microsoft.EntityFrameworkCore.Model.Validation[30000]
No store type was specified for the decimal property 'Salary' on entity type 'Employee'. This will cause values to be silently truncated if they do not fit in the default precision and scale. Explicitly specify the SQL server column type that can accommodate all the values in 'OnModelCreating' using 'HasColumnType', specify precision and scale using 'HasPrecision', or configure a value converter using 'HasConversion'.
No store type was specified for the decimal property 'Salary' on entity type 'Employee'. This will cause values to be silently truncated if they do not fit in the default precision and scale. Explicitly specify the SQL server column type that can accommodate all the values in 'OnModelCreating' using 'HasColumnType', specify precision and scale using 'HasPrecision', or configure a value converter using 'HasConversion'.
To undo this action, use Remove-Migration.
```

Figure 2 Succeed

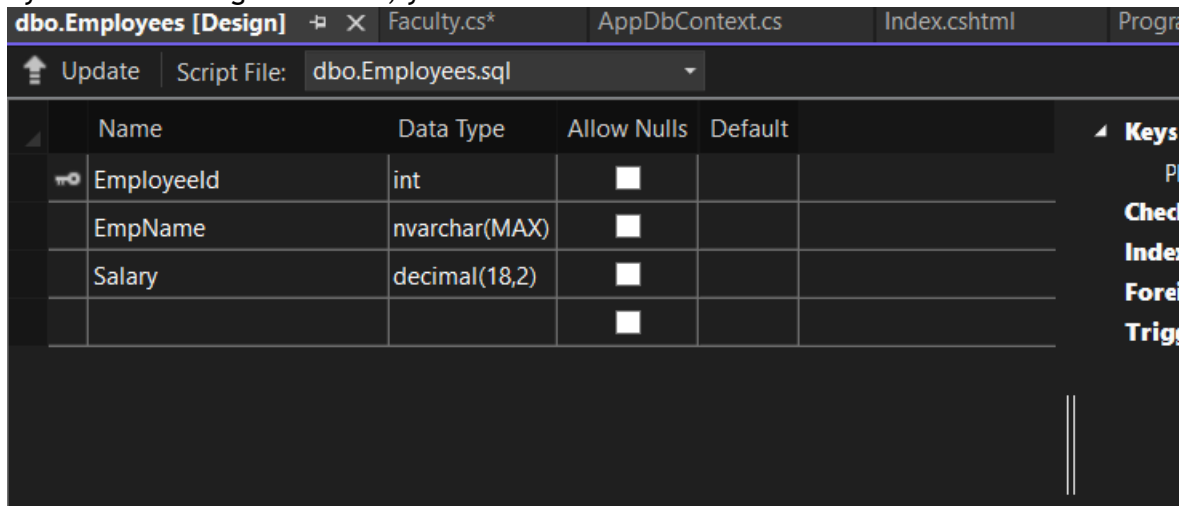
After Succeeding you want to type

```
PM> update-database
```

On success it should be reflected in your SQL Server Object Explorer (refresh it)



By double clicking on a table, you can see the table view



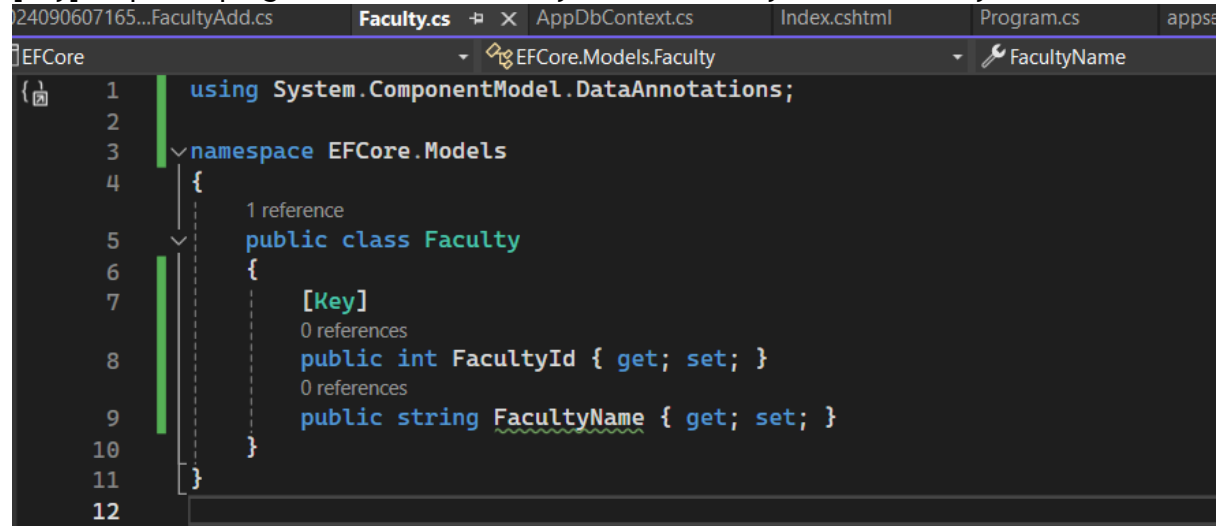
Well Done! You have successfully migrated (created) a database using the entity framework.

## Add more migrations.

When you need new tables for your database

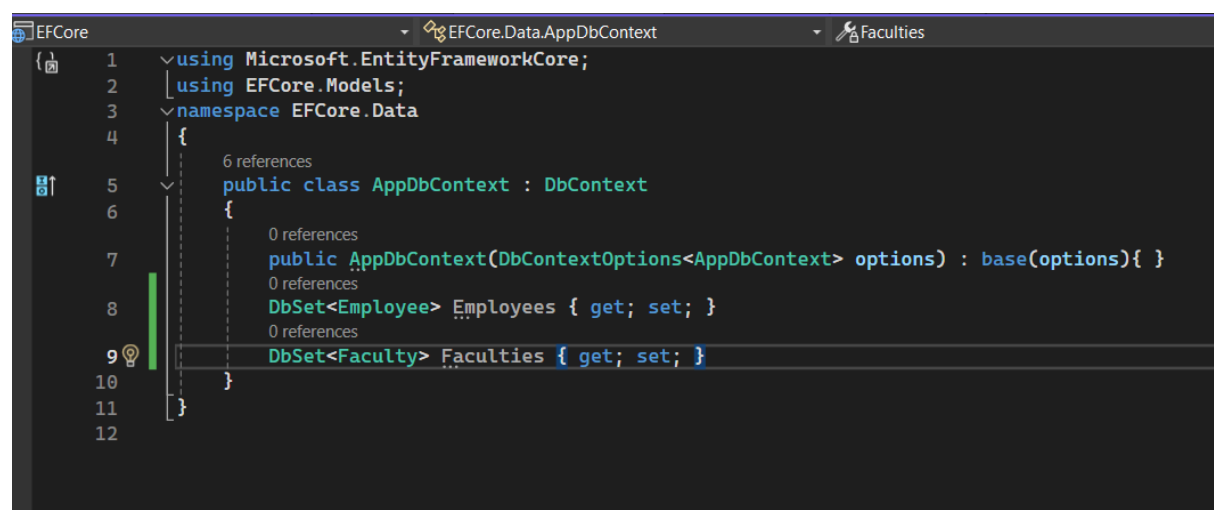
Create a new model class and follow what you did earlier for the Employee class.

[Key] helps the program understand that you want FacultyId to be the key.



```
1 using System.ComponentModel.DataAnnotations;
2
3 namespace EFCore.Models
4 {
5     1 reference
6     public class Faculty
7     {
8         [Key]
9         0 references
10        public int FacultyId { get; set; }
11        0 references
12        public string FacultyName { get; set; }
13    }
14 }
```

You then add it to the AppDbContext



```
1 using Microsoft.EntityFrameworkCore;
2 using EFCore.Models;
3 namespace EFCore.Data
4 {
5     6 references
6     public class AppDbContext : DbContext
7     {
8         0 references
9         public AppDbContext(DbContextOptions<AppDbContext> options) : base(options){ }
10        0 references
11        DbSet<Employee> Employees { get; set; }
12        0 references
13        DbSet<Faculty> Faculties { get; set; }
14    }
15 }
```

Then run 'add-migration AddFaculty' then 'update-database'.

This should reflect in your database.

## Revert Migration

If you want to go back before you added a table, you use  
`update-database [nameofmigration]`