# Setting Up Code First Database

#DotNet6

## Install Packages

In The Package Manager Console:

`dotnet add package Microsoft.EntityFrameworkCore.Design --version 6.0.27`

`dotnet add package Microsoft.EntityFrameworkCore.SqlServer --version 6.0.27`

## Update .NET Tools- Usually Optional

In The Package Manager Console:

`dotnet tool uninstall --global dotnet-ef`

`dotnet tool install --global dotnet-ef --version 6.0.27`

## Define the Connecting String

replace [ServerName] and [DatabaseName] = [ProjectConnection] is any name you want to give the connection string

```
"ConnectionStrings": {
  "ProjectConnection": "Server=(localdb)\\[ServerName];Database=
[DatabaseName];MultipleActiveResultSets=true"
}
```

## Create a DBContext Class

[Product] is the Model to be used in the database

```
public class ProjectDBContext : DbContext
{
        public ProjectDBContext(DbContextOptions<ProjectDBContext> options) :
base(options) { }
        public DbSet<Product> Products => Set<Product>();
}
```

## Add DbContext to Program.cs

Make sure this snippet comes before the `var app = builder.Build();` line

```
builder.Services.AddDbContext<ProjectDBContext>(opts => {
    opts.UseSqlServer(
        builder.Configuration["ConnectionStrings:ProjectConnection"]);
});

builder.Services.AddScoped<ProjectRepoitory>();
```

## Create a Repository

```
public class ProjectRepoitory
{
    private ProjectDBContext context;
    public AssignmentRepoitory(ProjectDBContext ctx)
    {
            context = ctx;
    }
    public IQueryable<Product> Products => context.Products;
}
```

## Create the Migrations

In The Package Manager Console:
`dotnet ef migrations add Initial`

## Create Seed Data

`EnsurePopulated()` will be called on the app start to ensure we have at least some data in the database

```
public class SeedData
{
    public static void EnsurePopulated(IApplicationBuilder app)
    {
            //geting the context from app
        ProjectDBContext context = app.ApplicationServices
            .CreateScope().ServiceProvider.GetRequiredService<ProjectDBContext>
();
        //make sure we wait for migrations
        if (context.Database.GetPendingMigrations().Any())
```

```
        {
            context.Database.Migrate();
        }
        //check if Products Table is empty, then populate with seed data
        if (!context.Products.Any()) {
            context.Assignments.Add(new Assignment { });
        }
    }
}
```

## Add Seed Data to Program.cs

Make sure you call this before the `app.Run();` line

```
SeedData.EnsurePopulated(app);
```