

## L'approche de modélisation DataBaseFirst

### Les étapes de connexion dans une approche DataBaseFirst:

- 1- Créer une base de données nommée "ApplicationTP2DbFirst"
- 2- Créer les tables

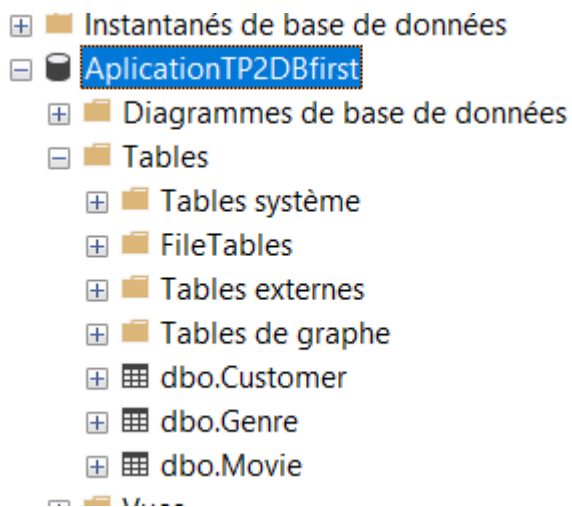


Figure 1. Résultat de la création de la base de données et ses tables

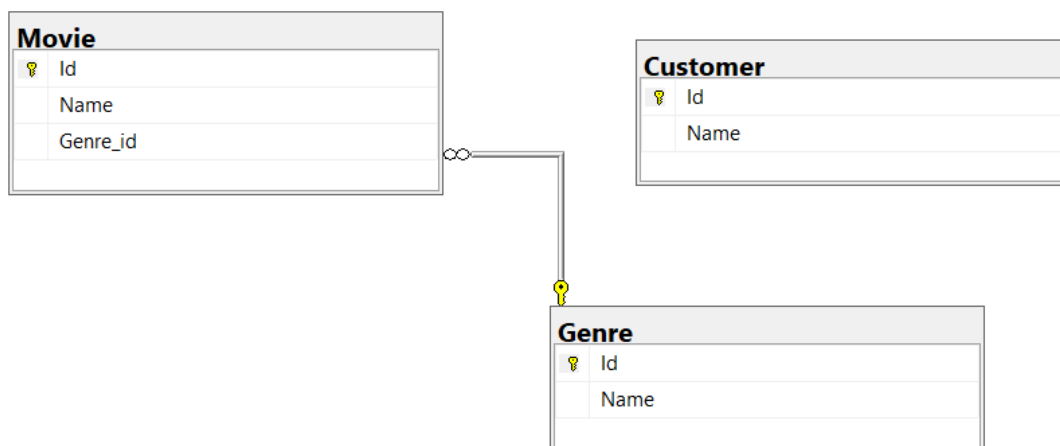


Figure 2. Diagramme de la base de données

## 2-Installer les packages

- \*Microsoft.EntityFrameworkCore.Tools
- \*Microsoft.EntityFrameworkCore.SqlServer

## 3-Aller maintenant dans le fichier Appsettings.json et configurer la chaîne de connexion à la base de données.

```
{  
  "AllowedHosts": "*",  
  "ConnectionStrings": {  
    "DefaultConnection": "Server=LAPTOP-HHN8V023\\SQLEXPRESS;Database=AplicationTP2DBfirst;Integrated Security=True;TrustServerCertificate=True"  
  }  
}
```

Figure 3:Chaîne de connexion

## 4-Passer dans le console de gestionnaire de package cette requête

```
Scaffold-DbContext Name="DefaultConnection"  
Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models
```

## 5- Comme résultat, on va arriver à créer les classes et les relations entre eux

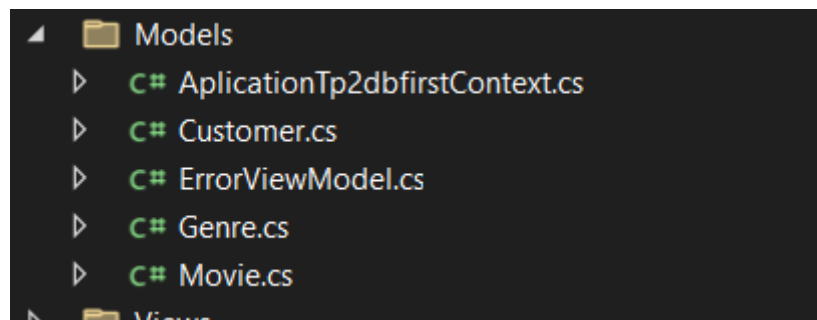


Figure 4:Création des classes

```

protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    modelBuilder.Entity<Customer>(entity =>
    {
        entity.ToTable("Customer");

        entity.Property(e => e.Id).ValueGeneratedNever();
    });

    modelBuilder.Entity<Genre>(entity =>
    {
        entity.ToTable("Genre");

        entity.Property(e => e.Id).ValueGeneratedNever();
    });

    modelBuilder.Entity<Movie>(entity =>
    {
        entity.ToTable("Movie");

        entity.Property(e => e.Id).ValueGeneratedNever();
        entity.Property(e => e.GenreId).HasColumnName("Genre_id");

        entity.HasOne(d => d.Genre).WithMany(p => p.Movies)
            .HasForeignKey(d => d.GenreId)
            .HasConstraintName("FK_Movie_Genre");
    });
}

```

Figure 5: Élaboration des relations entre les tables