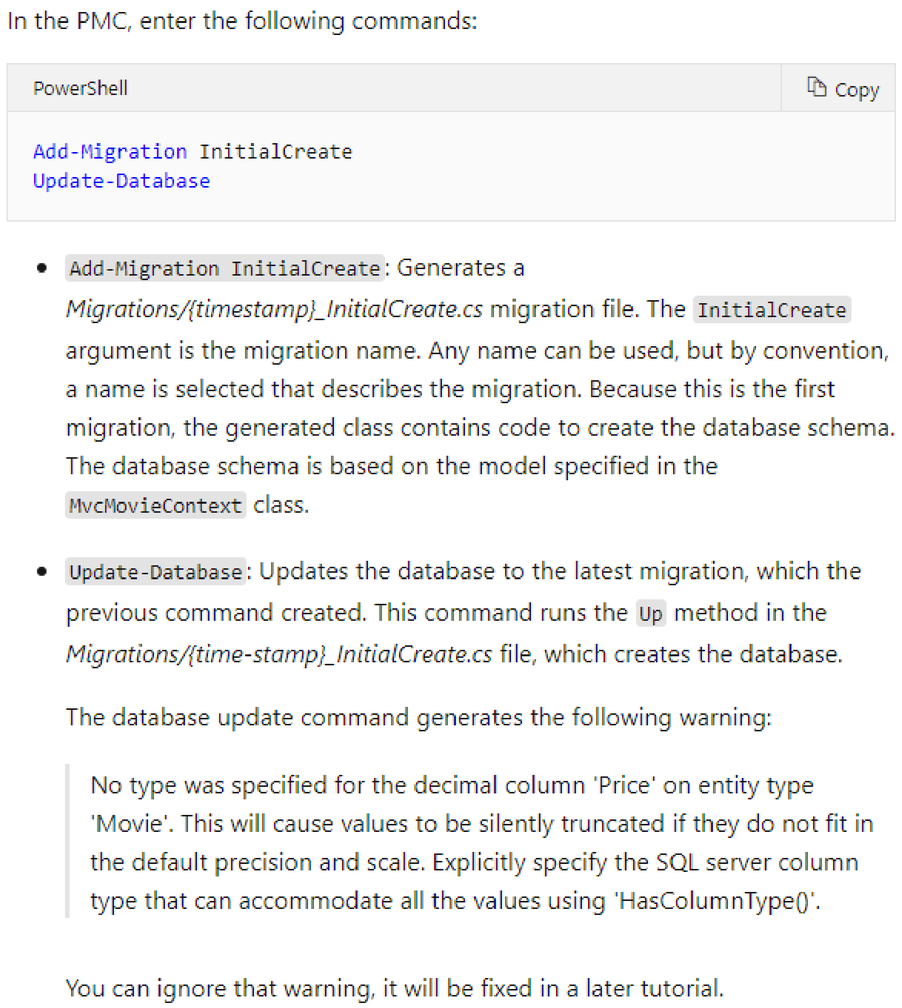
Lab Assignment 9

Jerry Belmonte
Keira Wong

ISSUES

Warnings reached after the following steps in Part IV:



In the PMC, enter the following commands:

```powershell
Add-Migration InitialCreate
Update-Database
```

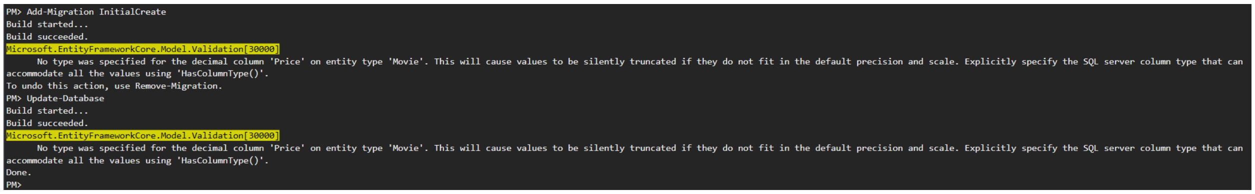
- `Add-Migration InitialCreate`: Generates a *Migrations/{timestamp}_InitialCreate.cs* migration file. The `InitialCreate` argument is the migration name. Any name can be used, but by convention, a name is selected that describes the migration. Because this is the first migration, the generated class contains code to create the database schema. The database schema is based on the model specified in the `MvcMovieContext` class.

- `Update-Database`: Updates the database to the latest migration, which the previous command created. This command runs the `Up` method in the *Migrations/{time-stamp}_InitialCreate.cs* file, which creates the database.

  The database update command generates the following warning:

  > No type was specified for the decimal column 'Price' on entity type 'Movie'. 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 using 'HasColumnType()'.

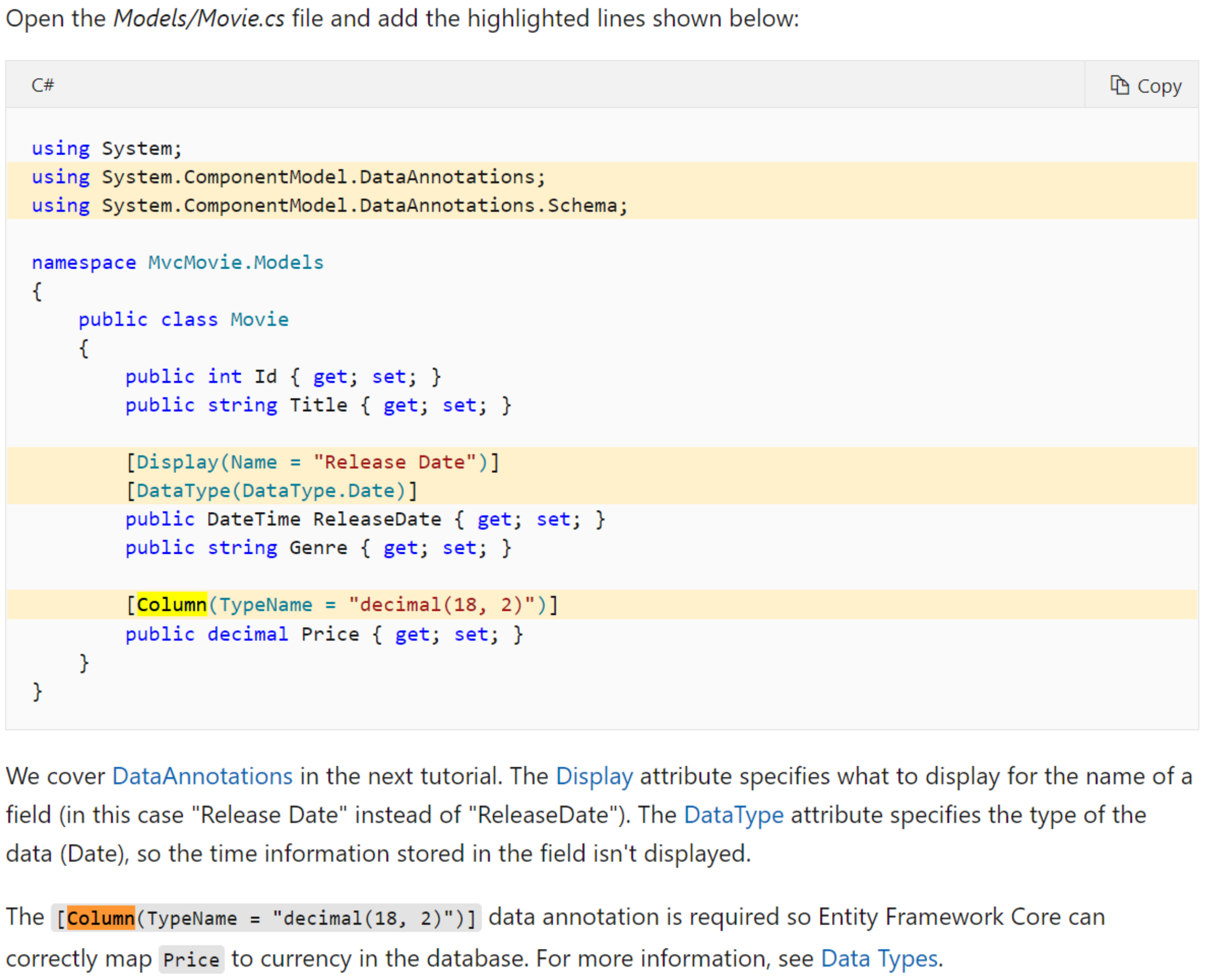
  You can ignore that warning, it will be fixed in a later tutorial.



```
PM> Add-Migration InitialCreate
Build started...
Build succeeded.
Microsoft.EntityFrameworkCore.Model.Validation[30000]
      No type was specified for the decimal column 'Price' on entity type 'Movie'. 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 using 'HasColumnType()'.
To undo this action, use Remove-Migration.
PM> Update-Database
Build started...
Build succeeded.
Microsoft.EntityFrameworkCore.Model.Validation[30000]
      No type was specified for the decimal column 'Price' on entity type 'Movie'. 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 using 'HasColumnType()'.
Done.
PM>
```

SOLUTION

Solution found in Part VI:



Open the *Models/Movie.cs* file and add the highlighted lines shown below:

```csharp
using System;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace MvcMovie.Models
{
    public class Movie
    {
        public int Id { get; set; }
        public string Title { get; set; }

        [Display(Name = "Release Date")]
        [DataType(DataType.Date)]
        public DateTime ReleaseDate { get; set; }
        public string Genre { get; set; }

        [Column(TypeName = "decimal(18, 2)")]
        public decimal Price { get; set; }
    }
}
```

We cover DataAnnotations in the next tutorial. The Display attribute specifies what to display for the name of a field (in this case "Release Date" instead of "ReleaseDate"). The DataType attribute specifies the type of the data (Date), so the time information stored in the field isn't displayed.

The `[Column(TypeName = "decimal(18, 2)")]` data annotation is required so Entity Framework Core can correctly map `Price` to currency in the database. For more information, see Data Types.