Migrations

Code First Migrations

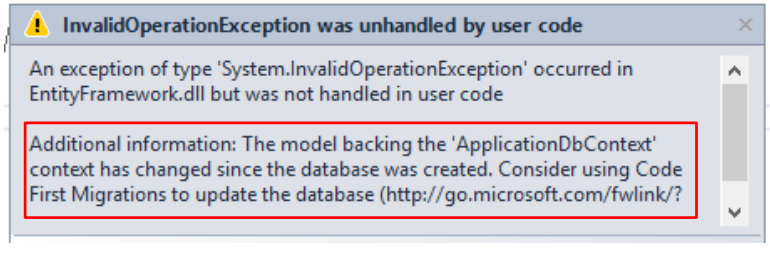
After we get our application to run, we might decide that we want to **change** one or more of our models.

Let's continue with our Book Demo App and see what we would have to do to make changes to a model at this point.

Let's add an ISBN property to the Book class.

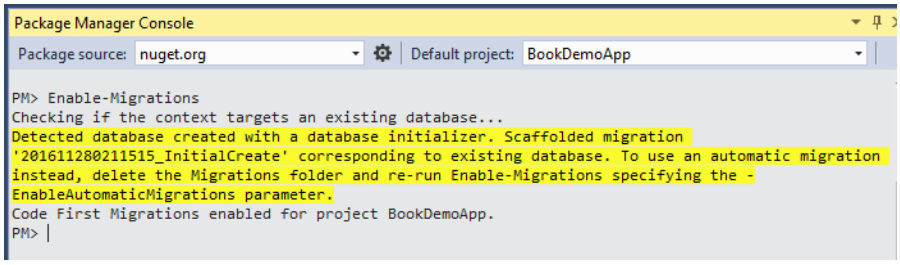
```
public string ISBN { get; set; }
```

Now when we run the application, we get the following exception.

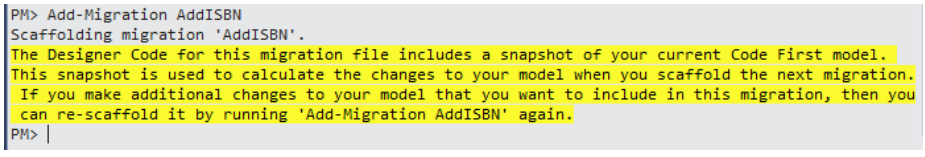


To avoid having to start over and create the application all over again, we will use **Code First Migrations**.

Run the **Enable-Migrations** command in the NuGet Package Manager Console. If there is more than one context class in your application, use Enable-Migrations -ContextTypeName BookDemoApp.Models.BookDemoAppContext

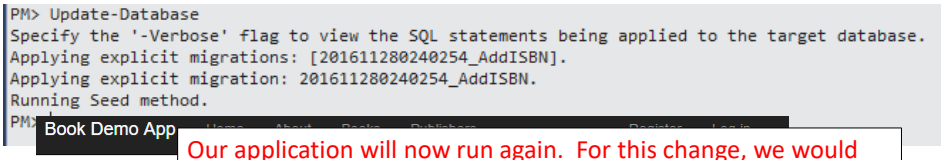


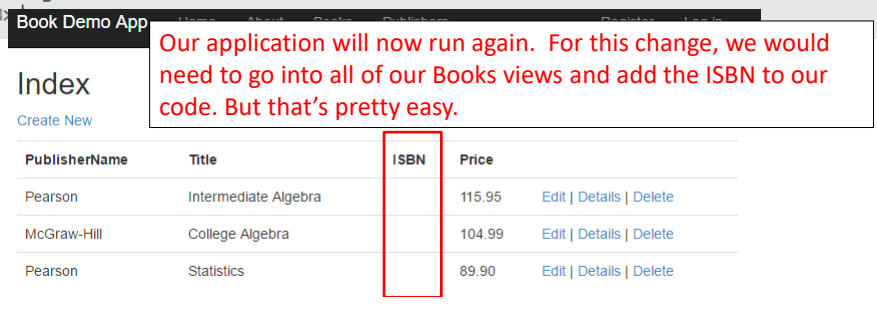
Now run the **Add-Migration AddISBN.** We name our migrations and we are calling this one AddISBN.



Run the **Update-Database** command.

Code First Migrations will compare the migrations in the **Migrations** folder with the ones that have been applied to the database. It will see that the **AddISBN** migration needs to be applied.





Our application will now run again. For this change, we would need to go into all of our Books views and add the ISBN to our code. But that's pretty easy.

Notice that our books are still in the database but an empty field called ISBN was added for each record.

For more information, visit https://msdn.microsoft.com/en-us/library/jj591621(v=vs.113).aspx

We now need to change all of the other Books views so that ISBN will be included.

One more thing: Very Important!

In the Edit and Create methods in the Books controller, add ISBN to the Bind attribute.

```
([Bind(Include = "BookID,Title,ISBN,PublisherID,Price")]
```