# Exercises: Entity Framework Code First

This document defines the **exercise assignments** for the ["Databases Advanced – Entity Framework" course @ Software University.](https://softuni.bg/courses/databases-advanced-entity-framework)

## Local Store

Imagine that you are hired by baba Gicka to create a database for her local store. The database should only have one table – it’s called **Products**. Products should have name, distributor name, description and price. Insert some sample data in the products table (at least **3 rows**). Create the database using the Code First approach.

## Local Store Improvement

Baba Gicka was satisfied with the **database** you have made, but she wants to make a **change** – now every product should have **weight** and **quantity**.

Everything was chill until you said that you have to make a migration in order to apply the new changes. Baba Gicka said that she hates birds and she is terrified from migrations. She **forbids** you to use any sort of **migrations**. Your task is to create a new up-to-date database. Keep in mind that you should not lose any data as well!

### Constraints/Hints

You could delete (with **MSSMS**) and re-create the same database (with the updated table).

You could **change** the database **initializer** to more suitable (you could even implement your own one).

For saving the data you could **script** it to **SQL** file or **save** it in a **CSV** file (yes, you can [import](https://support.discountasp.net/kb/a1179/how-to-import-a-csv-file-into-a-database-using-sql-server-management-studio.aspx)/[export](http://stackoverflow.com/questions/17471994/export-from-sql-server-2012-to-csv-through-management-studio) data from [file](http://www.computerhope.com/jargon/c/csv.htm)) and re-import it.

The above statements are just **suggestions** and you may not follow them but you are **not** **allowed** you use **migrations** in any form!

## Sales Database

Create database for storing data about sales using the Code First approach. The database should have the following tables:

* **Product** (Id, Name, Quantity, Price)
* **Customer** (Id, Name, Email, CreditCardNumber)
* **StoreLocation** (Id, LocationName)
* **Sale** (Id, ProductId, CustomerId, StoreLocationId, Date)

The relationships between tables are as follows:

* Sale has one product and a product can be sold in many sales
* Sale has one customer and a customer can participate in many sales
* Sale has one store location and one store location can have many sales

Write a **seed method** that fills the database with sample data (randomly generated). Configure Entity Framework to run the **Seed()** method after the database is created for the first time.

Run the application several times to ensure that it seeds sample data **only** **once**. Let the Seed() method insert 5 rows in every table.

### Hint

You can use the following format to design your model classes

* **Product**
  + int Id
  + string Name
  + double Quantity
  + decimal Price
  + ICollection<Sale> SalesOfProduct
* **Customer**
  + int Id
  + string Name
  + string Email
  + string CreditCardNumber
  + ICollection<Sale> SalesForCustomer
* **StoreLocation**
  + int Id
  + LocationName
  + ICollection<Sale> SalesInStore
* **Sale**
  + int Id
  + Product Product
  + Customer Customer
  + StoreLocation StoreLocation
  + DateTime Date

Remember to initialize ICollection<Sale> in the constructor of the class to avoid null pointer exceptions.   
You could use **“**CreateDatabaseIfNotExists**”** as database initializing strategy.

## Products Migration

For table Products add string column Description. You should use migrations. Current migration should be called: “ProductsAddColumnDescription”. Add default value for the description property:   
“No description”.

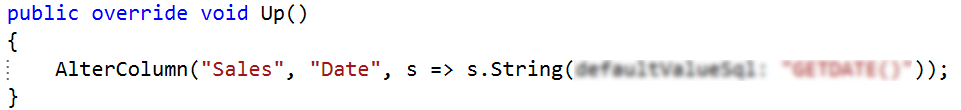
## Sales Migration

For table Sales make Date column with default value GETDATE() function, called from the database – not the application. Use explicit Migration. Do **not** use DateTime.Now(). Name the migration: “SalesAddDateDefault”.

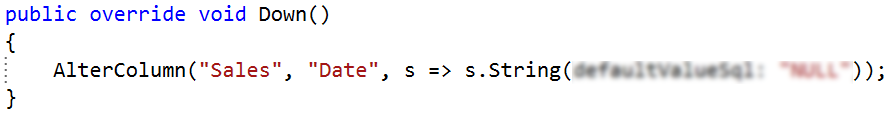
After that open your table data and see if the default value is applied or not.

### Hints

Add new empty migration. In the Up() method we can paste the logic for the altering the table:



Make sure the Down() method to be adequate too:



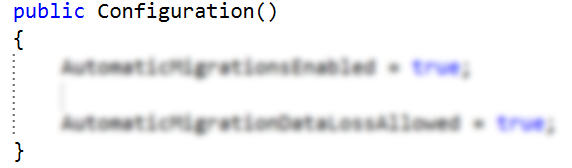
Then simply update the database and check if new constraint is added (you can see in MSSMS).

## Customers Migration

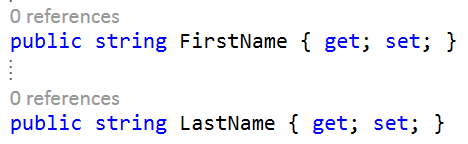
Enable automatic migrations and also enable data loss when migration is happening. Modify table Customers: remove Name column and add another two properties: FistName and LastName. Modify the Seed() method too in order to insert proper data. Then simply update database (without the Add-Migration command).

### Hints

To enable automatic migrations change the constructor of the Configuration class:



Simply add the new properties in the Customer class:



After that run the update database command.

## Add Default Age

In table Customers add new property Age. Set default value of 20. Do not change the Seed() method.

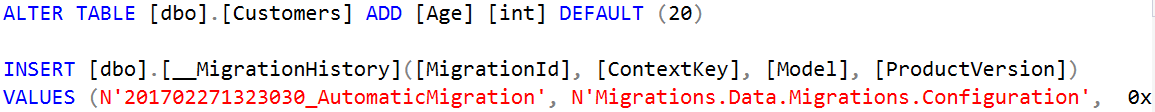
After that open your table data and see if the default value is applied or not.

## \*Script Migration

Your DBA asked you to present the SQL script (“ALTER TABLE…”) executed by **EntityFramework** from the last migration.

### Example

In the end you should have script similar to this one:



The “**ALTER TABLE …**” statement makes the actual changes to the table while the “**INSERT …**” preserves migrations history.

Note that the query data in the second query is most likely to be **different**!

### Hints

First rollback the changes made with the last migration:

* Find the name of the second last migration (you can see it in the MigrationsTable)
* Then run **“**Update-Database -TargetMigration <migrationName>”

Second update the database again but this time use the “-Script” command within. Query window should popup displaying script almost identical to the above one (in the picture).

Lastly see in your database if the migration has been applied (it should not). So what you need is again to run update database command, this time do not add anything specific.