# Exercises about Entity FrameWork Core

* Solve them in Visual Studio.

## Exercise 23.01

|  |  |
| --- | --- |
|  |  |
|  | A good starting point is 12001-a DemoEntityFrameWorkCore.  Study the code good.   * This removes the database and recreates it, without using migrations. |
|  |  |

* You have Products.
* You have Categories.
* A Product has a specific Category.
* A Category can have more than one Product.
* Create Code First Classes.
* Use those Classes to create a new SQL Server Database.
* Write some code to add data to the tables.
* Write some code to query the data from the database.

## Exercise 23.02

|  |  |
| --- | --- |
|  |  |
|  | A good starting point is for this exercise the code in  12002-a MigrationEntityFrameWorkCore.  Study the code good.   * The goal is to use migrations to set up the database. |
|  |  |

* You have a Book.
* You have a Genre.
* You have a Publisher.
* A Book has a specific Genre.
* A Book has a specific Publisher.
* A Genre can have more than one Book.
* A Publisher can have more than one Book.
* Create Code First the class for Book.
* Use Package Manager Console for creating the first migration.
* Create Code First the class for Genre.
* Use Package Manager Console for creating the second migration.
* Create Code First the class for Publisher.
* Use Package Manager Console for creating the third migration.
* Write some code to add data to the tables.
* Write some code to query the data from the database.

### Variant 1

* Do the same thing, but add data to the tables after every migration.

## Exercise 23.03

|  |  |
| --- | --- |
|  |  |
|  | A good starting point is for this exercise the code in  12002-a MigrationEntityFrameWorkCore.  Study the code good.   * The goal is to use migrations to set up the database. |
|  |  |

* You have a Client, an Order and a Payment.
* An Order is for one Client.
* A Payment is for one Order.
* A Client can have multiple Orders.
* An Order can have multiple Payments.
* Create Code First the class for Payment.
* Use Package Manager Console for creating the first migration.
* Create Code First the class for Order.
* Use Package Manager Console for creating the second migration.
* Create Code First the class for Client.
* Use Package Manager Console for creating the third migration.
* Write some code to add data to the tables.
* Write some code to query the data from the database.

### Variant 1

* Do the same thing, but add data to the tables after every migration.

### Variant 2

* Work the other way around.
* First the class Client.
* Then the class Order
* Then the class Payment.

## Exercise 23.04

|  |  |
| --- | --- |
|  |  |
|  | A good starting point is for this exercise the code in  12002-a MigrationEntityFrameWorkCore.  Study the code good.   * The goal is to use migrations to set up the database. |
|  |  |

* Invent an example for yourself.
* Make sure that in the comments it is clear what will be made.
* You have list of migrations or only one.
* Write some code to add data to the tables.
* Write some code to query the data from the database.
* For testing, the database will be created using the set of migrations.
* The application must show all the data that is in the database.