# Southwind Exercises

## 1.0 Open EF\_ModelFirst Solution

Make sure that your EF\_ModelFirst solution (i.e. your Southwind Project) is correctly setup (install packages, update git repo, etc.) from the lesson.

## 1.1 Experiment with Entity Framework by writing queries to create, update, read and delete to your new database.

Your database will be empty, so make sure you add before you try to update read or delete.

Display to the console the content of your database.

Add a customer, find that customer, display their data.

Update that customer and display the updated information.

Delete a customer and display the remaining customers.

## 1.2 Create an interactive Console where a user can perform CRUD operations on the customer table purely through the console.

If you have extra time, extend that functionality to your entire database.

## 1.3 Create a self-contained Seed() function

Remember all methods (including Main()!) should obey SRP and be DRY.

To that end, write a Seed() function so every table has at least 3 entries. Be careful of the order you input data, as for foreign keys to be properly filled you will need the primary key data (which may be auto incremented).

## 1.4 Add a column (such as Country) to the customer table, add-migration, update-database

Note you may need to update your Seed() function!

## 1.5 Encapsulate your CRUD functionality for the Customer Class in a CustomerManager class

Write self-contained CRUD methods in a manager where:

* the Create method takes in an object parameter
* the Read method returns a list of all objects
* the Update method takes in an object that represents an updated version of an existing entity in the database (i.e. the ID is the same)
* the Delete method takes in a parameter for the ID of the entity to be deleted

## 1.6 Add a new table to your database that has a one to many or many to one relationship with an existing table (such as Supplier, where one Supplier can have many associated OrderDetails)

Remember to add both a property for the foreign key and an ICollection for the model. Refer to the existing classes as a template.

Write CRUD functionality for the new table in its own Manager. Again you may need to update your Seed() function, as well as your CRUD functionality in other tables.

## 1.7 Database Testing, write tests for your CRUD functionality of the Customer table

Write tests that will manipulate a customer called “MANDA”. Note that the SetUp and TearDown methods, depending on your migrations, may need updating.

Check your CRUD functionality:

* When a Customer is added, the number of customers increases by 1
* When a Customer is added, the details in the table are correct
* When Reading Customers, a list of Customers is returned of the right length
* When Reading a Customer, that customer has the correct data
* When a Customer is updated, the details in the table are updated
* When a Customer not in the database is attempted to be updated, the method returns false
* When a Customer is removed, the number of customers decreases by 1
* When a Customer not in the database is attempted to be removed, the method returns false