**Entity Framework - Code First CRUD Exercise (4 Marks)**

[#](https://dmit-2018.github.io/exercises/ef.html#private-github-repo-only)**Private GitHub Repo Only**

Be aware that this is the first exercise in a series of cumulative exercises - you must be prepared to complete each exercise in light of the possibility that any given exercise may depend on the **correct** implementation of previous exercises. Place this work in your classroom private Exercise repository.

[#](https://dmit-2018.github.io/exercises/ef.html#setup)**Setup**

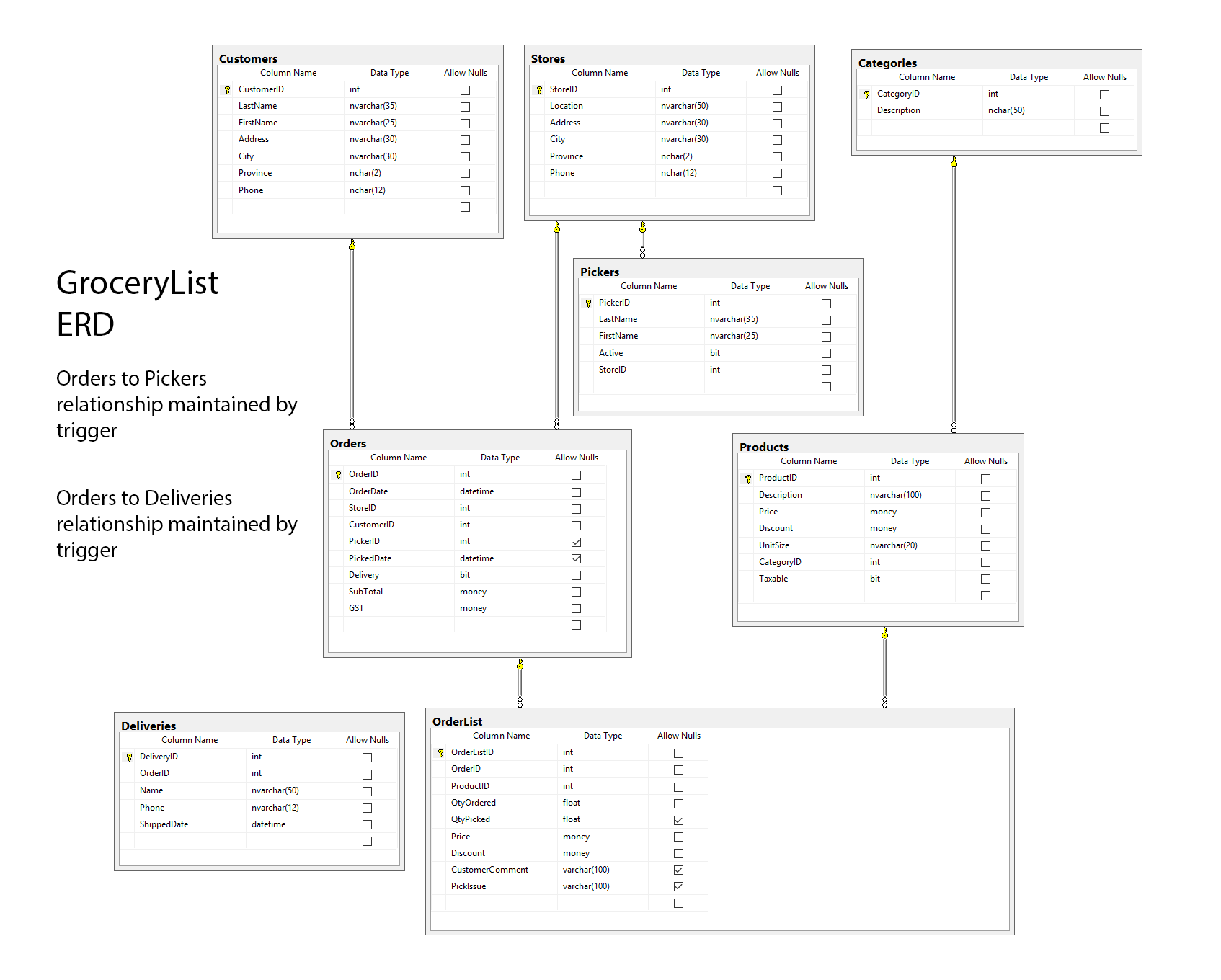
Create a new VS Solution called GroceryListSolution. Place two class projects into this solution called GroceryListSystem and GroceryList.Data, along with a web application called WebApp. GroceryList.Data will have 3 folder called Entities, DTOs and POCOs. GroceryListSystem will have 2 folders DAL and BLL. Add the NuGet package, EntityFramework, to your class libraries. Add references to your solution where needed/appropriate.

[#](https://dmit-2018.github.io/exercises/ef.html#entities)**Entities**

Create Entity classes using **reverse engineering**. Create the entity classes in the Entity folder. Name the context class GroceryListContext. Use GroceryListDB as the connection string name. [Required], and [StringLength] notations in your entities will not have error messages. You must place an ErrorMessage parameter within each notation such as that the message will be specific to the entity attribute.

Add the following not mapping properties:

* Customer: FullName: returns the customer full name as LastName, Firstname
* Customer: CityCustomer: returns the city and customer full name as City: LastName, Firstname
* Picker: FullName: returns the picker full name as LastName, Firstname
* Product: DescriptionUnitSize: returns the description and unitsize as Description (UnitSize)
* Store: CityLocation: returns the city and location as City:Location



[#](https://dmit-2018.github.io/exercises/ef.html#context-class)**Context Class**

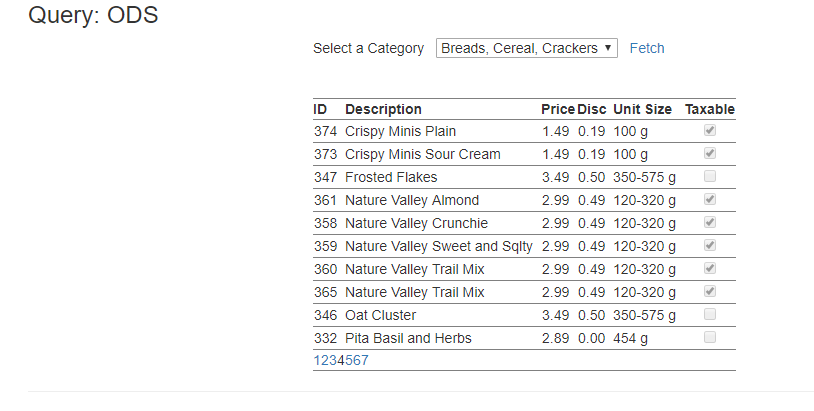
Move the **Context class** to the DAL folder from the Entities folder; alter the namespace to GroceryListSystem.DAL; and include any necessary using statements. Remove the Context class from the entity folder. Create an appropriate web connection string file pointing to the database GroceryList. Add the <context> tag in the web.config file under the <entityframe> tag. The context class is to be restricted to internal project access only.

[#](https://dmit-2018.github.io/exercises/ef.html#bll-controller-classes)**BLL Controller Classes**

Create **BLL controller classes** for the two entities: Category and Product. The classes will allow ODS access to methods. Both classses will have an appropriate List() and Get(int keyvalue) methods (exposed). The ProductController will have a method which will recieve an integer parameter representing a category and return all products within that category. The ProductController will implement Add(Product), Update(Product), Delete(Product) and Delete(int productid) methods. The overloaded Delete(int productid) will do the actual delete of the product record of the database.

[#](https://dmit-2018.github.io/exercises/ef.html#web-page-query)**Web Page: Query**

Create a web page that will display the product data related to a supplied category. Use **only ODS wired controls** to handle the data. Categories should be displayed in a ordered drop down list. Display the ordered product data in a customized data control which obtains its input parameter value from the drop down list. Your display requires appropriate page title, labels and layout/formatting. You will need a button on your page to force a post back. This page **must** be accessiable by the site master menu.



[#](https://dmit-2018.github.io/exercises/ef.html#web-page-crud)**Web Page: CRUD**

Create a web page that will maintain the Product entity. You will be requried to insert, update or delete records to this entity. The Category data will be handled via an appropriate dropdownlist. **Use a ODS wired ListView control** to maintain the data. Appropriate formatting is required on the display lines. This page **must** be accessiable by the site master menu.

