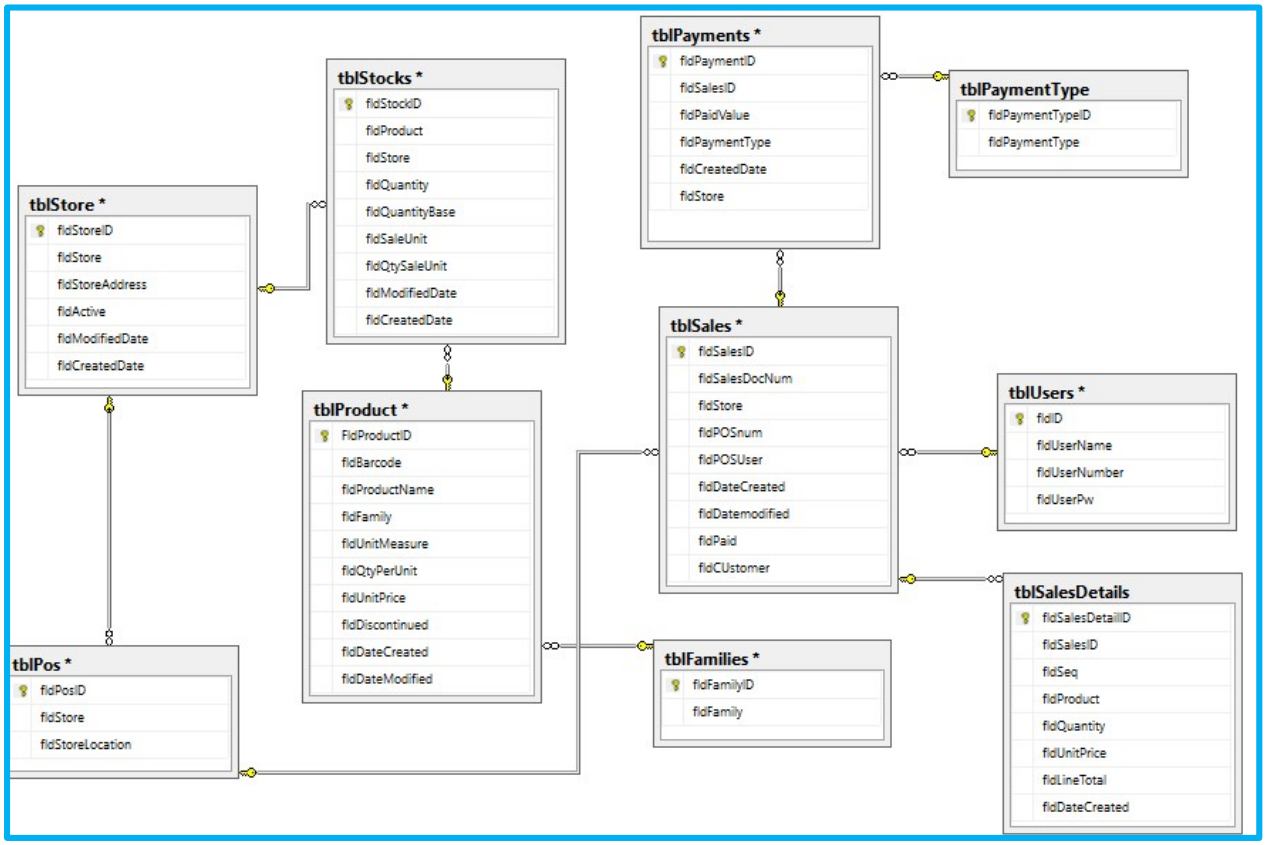
PROJECT – CREATE ENTITY FRAMEWORK CODE FIRST - PART I

Objective: Create a data access layer using Code First approach.

The project must be a Class Library to be reused in different types of projects.

1. Create a new Class Library Project and name it: POS_DataLayer.
2. Add to the class files provided to you. These files contain almost all the entities necessary to create an Entity Framework code first model for the POS store solution;
3. Add to the project a class file and add an abstract class named person:
    a. **Note:** Create an Entity named Person as an abstract class:
        i. Must have the following properties:
        ii. ID type: int
        iii. Name type: String
        iv. Address type: String
        v. City type: string
        vi. Email: String
    b. Using the class Person, create a class Customer Entity:
        i. CustomerID type: int and Primary Key and auto-number, Display Name: CustomerID;
        ii. Name string max length: 60 required with error message, Display name: Customer Name
        iii. Address type: string max length:120
        iv. City type: string max length:50
        v. Email type: string max length
        vi. NIF type: string max length 9, required with error message
        vii. Consider that a customer can have one or more Sales;
    c. Using the class Person, modify the class user:
        i. ID type: int and Primary Key and auto-number, display name: POSUserID;
        ii. Name string max length: 60, required with error message, Display name: POS User Name
        iii. Address type: string maxlength:120
        iv. City type: string max length:50
        v. Email type: string max length
        vi. Phone type: string max length 9, required with error message
        vii. Consider that a User can have one or more sales.

4. Create the Database context class with the name PosContext.
5. Configure the App.Config with the connection string. The database should be created on your Sql Server and be named POS_Store;
6. Enable automatic migrations;
7. Compile and test your component:
    a. Create a console application and add the following store:
        i. Store id- > H01
        ii. Store - > Hemsedal 01
        iii. Store Address -> Fanitullen Postboks 80, 3561, Norway
        iv. Check in your database if the record was created.
    b. Change the tblPayments entity, and replace the field fldSalesNumDoc with fldSalesID and data type to int;
    c. Update the migrations and test the component, and add the following data to tblPaymentType entity:
        i. PaymentType - > Cash
    d. Change your context class and add relations: Between Payments and Sales.
    e. Update the database.
    f. Create a SQL Diagram and verify if you get all the relations:



8. Use the code below to add some data to your database.

```
  PosContext context = new PosContext();
  var Families = new List<tblFamily>
  {
          new   tblFamily {fldFamilyID="SECAL", fldFamily = "SECO ALIMENTAR"},
          new   tblFamily {fldFamilyID="FRESC", fldFamily="FRESCOS"},
          new   tblFamily { fldFamilyID="BEBID",fldFamily="BEBIDAS" },
  };
Families.ForEach(f => context.tblFamilies.Add(f));

var paymentType = new List<tblPaymentType>
 {
```