# **Entity Framework Demo 1 - Create Models using Code First**

"Server=localhost;Database=OnlineShopDB;User Id=sa;Password=Work6ten"

"Server=DESKTOP-V782QN0;Database=OnlineShopDB;Trusted\_Connection=True;"

* Create a new solution *01.EntityFramework*
* Create new project *01.EntityFramework-CodeFirst*
* Add the following packages using NuGet

Microsoft.EntityFrameworkCore.SqlServer

Microsoft.EntityFrameworkCore.Design

Microsoft.EntityFrameworkCore.Tools

* Create a new folder Models
* Create a new Product class under Models

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace EntityFrameworkDemo.Models

{

public class Product

{

// Convention is Id or TableNameId

[Key]

public int Id { get; set; }

[Required]

public string Name { get; set; }

[Column(TypeName = "decimal(18, 2)")]

public decimal Price { get; set; }

}

}

* Create a new Customer class under Models and generate Order class (non-existing)

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Collections.Generic;

namespace EntityFrameworkDemo.Models

{

public class Customer

{

public int Id { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public string? Address { get; set; }

public string? Phone { get; set; }

// Navigation Property. One-to-Many

public ICollection<Order> Orders { get; set; }

}

}

}

}

* Order class code.

namespace EntityFrameworkDemo.Models

{

public class Order

{

public int Id { get; set; }

public DateTime OrderPlaced { get; set; }

public DateTime? OrderFulfilled { get; set; }

// Foreign Key

public int CustomerId { get; set; }

public Customer Customer { get; set; }

// Intersection table. Many-to-Many relationship

public ICollection<ProductOrder> ProductOrders { get; set; }

}

}

* ProductOrder class

public class ProductOrder

{

public int Id { get; set; }

public int Quantity { get; set; }

public int ProductId { get; set; }

public int OrderId { get; set; }

public Order Order { get; set; }

public Product Product { get; set; }

}

* Create a new database context class under data called OnlineShopContext

using EntityFrameworkDemo.Models;

using Microsoft.EntityFrameworkCore;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace EntityFrameworkDemo.Data

{

public class OnlineShopContext: DbContext

{

public DbSet<Customer> Customers { get; set; }

public DbSet<Order> Orders { get; set; }

public DbSet<Product> Products { get; set; }

public DbSet<ProductOrder> ProductOrders { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

if(!optionsBuilder.IsConfigured)

{

optionsBuilder.UseSqlServer("Server=localhost;Database=ToyUniverse;User Id=sa;Password=Work6ten");

}

}

}

}

* Open Package Manager Console (PMC) *Tools > NuGET Package Manager > Package Manager Console*
* Add new database migration execute Add-Migration command

PM > Add-Migration InitialCreate

* Run database migration

// ONLY RUN THIS

PM > Update-Database

// Unapply all migrations

PM > Update-Database -Migration 0

// Remove specific migration

Update-Database -Migration LastGoodMigrationName

// Unapply last migration

PM > Remove-Migration

PM > Remove-Migration -Force

* Check tables generated by Update-Database command
* Add a a new Email column to Customer class and run Add-Migration and Update-Database commands

PM > Add-Migration AddEmail

PM > Update-Database