**1. Create a new ASP.NET Core project**

Open terminal and run:

dotnet new webapi -n dotnetapp

cd dotnetapp

**2. Install EF Core packages**

dotnet add package Microsoft.EntityFrameworkCore.SqlServer --version 6.0.0

dotnet add package Microsoft.EntityFrameworkCore.Tools --version 6.0.0

Install EF CLI locally:

dotnet new tool-manifest

dotnet tool install --local dotnet-ef --version 6.0.6

Verify installation:

dotnet dotnet-ef

**3. Create Models**

Create a folder named Models and add the following classes:

**Category.cs**

public class Category

{

public int CategoryID { get; set; }

public string CategoryName { get; set; }

public string Description { get; set; }

public byte[] Picture { get; set; }

// Navigation property

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

}

**Customer.cs**

public class Customer

{

public int CustomerID { get; set; }

public string CompanyName { get; set; }

public string ContactName { get; set; }

public string ContactTitle { get; set; }

public string Address { get; set; }

public string City { get; set; }

public string Region { get; set; }

public string PostalCode { get; set; }

public string Country { get; set; }

public string Phone { get; set; }

public string Fax { get; set; }

// Navigation property

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

}

**Product.cs**

public class Product

{

public int ProductID { get; set; }

public string ProductName { get; set; }

public int SupplierID { get; set; }

public int CategoryID { get; set; }

public string QuantityPerUnit { get; set; }

public decimal UnitPrice { get; set; }

public short UnitsInStock { get; set; }

public short UnitsOnOrder { get; set; }

public short ReorderLevel { get; set; }

public bool Discontinued { get; set; }

// Navigation properties

public Category Category { get; set; }

public ICollection<OrderDetail> OrderDetails { get; set; }

}

**Order.cs**

public class Order

{

public int OrderID { get; set; }

public int CustomerID { get; set; }

public DateTime OrderDate { get; set; }

public DateTime RequiredDate { get; set; }

public int ShipVia { get; set; }

public decimal Freight { get; set; }

public string ShipName { get; set; }

public string ShipAddress { get; set; }

public string ShipCity { get; set; }

public string ShipRegion { get; set; }

public string ShipPostalCode { get; set; }

public string ShipCountry { get; set; }

// Navigation properties

public Customer Customer { get; set; }

public ICollection<OrderDetail> OrderDetails { get; set; }

}

**OrderDetail.cs**

public class OrderDetail

{

public int OrderDetailID { get; set; }

public int OrderID { get; set; }

public int ProductID { get; set; }

public decimal UnitPrice { get; set; }

public short Quantity { get; set; }

public float Discount { get; set; }

// Navigation properties

public Order Order { get; set; }

public Product Product { get; set; }

}

**4. Create DbContext**

Create **OrdersDbContext.cs** in the Models folder:

using Microsoft.EntityFrameworkCore;

public class OrdersDbContext : DbContext

{

private readonly string connectionString =

"User ID=sa;password=examlyMssql@123;server=localhost;Database=appdb;trusted\_connection=false;Persist Security Info=False;Encrypt=False";

public OrdersDbContext(DbContextOptions<OrdersDbContext> options) : base(options) { }

public DbSet<Category> Categories { get; set; }

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

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

public DbSet<OrderDetail> OrderDetails { get; set; }

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

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseSqlServer(connectionString);

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

// Composite key for OrderDetail if needed

modelBuilder.Entity<OrderDetail>()

.HasKey(od => od.OrderDetailID);

// Relationships

modelBuilder.Entity<Order>()

.HasOne(o => o.Customer)

.WithMany(c => c.Orders)

.HasForeignKey(o => o.CustomerID);

modelBuilder.Entity<Product>()

.HasOne(p => p.Category)

.WithMany(c => c.Products)

.HasForeignKey(p => p.CategoryID);

modelBuilder.Entity<OrderDetail>()

.HasOne(od => od.Order)

.WithMany(o => o.OrderDetails)

.HasForeignKey(od => od.OrderID);

modelBuilder.Entity<OrderDetail>()

.HasOne(od => od.Product)

.WithMany(p => p.OrderDetails)

.HasForeignKey(od => od.ProductID);

}

}

**5. Apply Migrations**

dotnet dotnet-ef migrations add InitialSetup

dotnet dotnet-ef database update

This will create the appdb database with all tables and relationships.