

# LAB\_3

```
RetailInventoryApp > Program.cs > ...
1  using RetailInventoryApp;
2  using RetailInventoryApp.Models;
3  using System;
4  using System.Linq;
5
6  using System;
7
8  0 references
9  class Program
10 {
11     0 references
12     static void Main()
13     {
14         using var context = new AppDbContext();
15
16         if (!context.Categories.Any())
17         {
18             var electronics = new Category { Name = "Electronics" };
19             var phone = new Product { Name = "Smartphone", Price = 699.99M, Category = electronics };
20
21             context.Categories.Add(electronics);
22             context.Products.Add(phone);
23             context.SaveChanges();
24
25             Console.WriteLine("Seed data added.");
26         }
27         else
28         {
29             Console.WriteLine("Data already exists.");
30         }
31     }
32 }
```

```
RetailInventoryApp > RetailInventoryApp.csproj
1  <Project Sdk="Microsoft.NET.Sdk">
2      <PropertyGroup>
3          <OutputType>Exe</OutputType>
4          <TargetFramework>net9.0</TargetFramework>
5      </PropertyGroup>
6
7      <ItemGroup>
8          <PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer" Version="7.0.0" />
9          <PackageReference Include="Microsoft.EntityFrameworkCore.Design" Version="7.0.0" />
10      </ItemGroup>
11  </Project>
```

```

1  using Microsoft.EntityFrameworkCore;
2  using RetailInventoryApp.Models;
3
4  namespace RetailInventoryApp
5  {
6      1 reference
7      public class AppDbContext : DbContext
8      {
9          2 references
10         public DbSet<Category> Categories { get; set; }
11         1 reference
12         public DbSet<Product> Products { get; set; }
13
14         0 references
15         protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
16         {
17             optionsBuilder.UseSqlServer(
18                 "Server=localhost;Database=RetailDB;Trusted_Connection=True;Encrypt=False;");
19         }
20     }
21 }

```

OUTPUT:

```

PS C:\Users\Lenovo\OneDrive\Desktop\RetailInventoryApp\RetailInventoryApp> dotnet ef database update
>> dotnet run
>>
Build started...
Build succeeded.
No migrations were applied. The database is already up to date.
Done.
Data already exists.

```

## Lab-4

```
AppDbContext.cs > ...
1  using Microsoft.EntityFrameworkCore;
2  using Core.Models;
3
4  3 references
5  public class AppDbContext : DbContext
6  {
7      1 reference
8      public DbSet<Category> Categories { get; set; }
9      1 reference
10     public DbSet<Product> Products { get; set; }
11
12     0 references
13     protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
14     {
15         optionsBuilder.UseSqlite("Data Source=retail.db");
16     }
17 }
```

```
Program.cs > ...
1  using System;
2  using System.Threading.Tasks;
3  using Core.Models;
4
5  0 references
6  class Program
7  {
8      0 references
9      static async Task Main(string[] args)
10     {
11         using var context = new AppDbContext();
12
13         var electronics = new Category { Name = "Electronics" };
14         var groceries = new Category { Name = "Groceries" };
15
16         await context.Categories.AddRangeAsync(electronics, groceries);
17
18         var product1 = new Product { Name = "Laptop", Price = 75000M, Category = electronics };
19         var product2 = new Product { Name = "Rice Bag", Price = 1200M, Category = groceries };
20
21         await context.Products.AddRangeAsync(product1, product2);
22         await context.SaveChangesAsync();
23
24         Console.WriteLine("Initial data inserted successfully.");
25     }
26 }
```

```
1  namespace Core.Models;
2
3  4 references
4  public class Product
5  {
6      0 references
7      public int Id { get; set; }
8      2 references
9      public string Name { get; set; }
10     2 references
11     public decimal Price { get; set; }
12     2 references
13     public Category Category { get; set; }
14 }
15
```

```

AppDbContext.cs > ...
1  using Microsoft.EntityFrameworkCore;
2  using Core.Models;
3
3 references
4  public class ApplicationDbContext : DbContext
5  {
6      1 reference
        public DbSet<Category> Categories { get; set; }
7      1 reference
        public DbSet<Product> Products { get; set; }
8
0 references
9  protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
10 {
11     optionsBuilder.UseSqlite("Data Source=retail.db");
12 }
13 }

```

OUTPUT:

```

Build started...
Build succeeded.
Acquiring an exclusive lock for migration application. See https://aka.ms/efcore-docs-migrations-lock for more information if this takes too long.
Applying migration '20250705153242_InitialCreate'.
Done.
Initial data inserted successfully.
PS C:\Users\Lenovo\core> dotnet run
Initial data inserted successfully.

```

## LAB-5

```

Program.cs > Program
1  using System;
2  using System.Threading.Tasks;
3  using Microsoft.EntityFrameworkCore;
4  using LAB5.Models;
5
6  class Program
7  {
8      0 references
9      static async Task Main(string[] args)
10     {
11         using var context = new AppDbContext();
12         if (!await context.Products.AnyAsync())
13         {
14             var category = new Category { Name = "Electronics" };
15
16             context.Categories.Add(category);
17
18             context.Products.AddRange(
19                 new Product { Name = "Smartphone", Price = 69999, Category = category },
20                 new Product { Name = "Laptop", Price = 189999, Category = category },
21                 new Product { Name = "Headphones", Price = 2999, Category = category }
22             );
23
24             await context.SaveChangesAsync();
25             Console.WriteLine("Seeded sample data.");
26         }
27
28         var products = await context.Products.ToListAsync();
29         foreach (var p in products)
30             Console.WriteLine($"{p.Name} - ₹{p.Price}");
31
32         var product = await context.Products.FindAsync(1);
33         Console.WriteLine($"Found: {product?.Name}");
34
35         var expensive = await context.Products.FirstOrDefaultAsync(p => p.Price > 50000);
36         Console.WriteLine($"Expensive: {expensive?.Name}");
37     }
38 }

```

```

Models > Category.cs > ...
1  using System.Collections.Generic;
2
3  namespace LAB5.Models
4  {
5      3 references
6      public class Category
7      {
8          0 references
9          public int Id { get; set; }
10         1 reference
11         public string Name { get; set; } = "";
12         0 references
13         public List<Product> Products { get; set; } = new();
14     }
15 }

```

```

Models > Products > ...
1  namespace LAB5.Models
2  {
3      5 references
4      public class Product
5      {
6          0 references
7          public int Id { get; set; }
8          6 references
9          public string Name { get; set; } = "";
10         5 references
11         public decimal Price { get; set; }
12         3 references
13         public Category Category { get; set; }
14     }
15 }

```

```

LAB5.csproj
1  <Project Sdk="Microsoft.NET.Sdk">
2
3      <PropertyGroup>
4          <OutputType>Exe</OutputType>
5          <TargetFramework>net9.0</TargetFramework>
6          <ImplicitUsings>enable</ImplicitUsings>
7          <Nullable>enable</Nullable>
8      </PropertyGroup>
9
10     <ItemGroup>
11         <PackageReference Include="Microsoft.EntityFrameworkCore" Version="9.0.6" />
12         <PackageReference Include="Microsoft.EntityFrameworkCore.Design" Version="9.0.6">
13             <IncludeAssets>runtime; build; native; contentfiles; analyzers; buildtransitive</IncludeAssets>
14             <PrivateAssets>all</PrivateAssets>
15         </PackageReference>
16         <PackageReference Include="Microsoft.EntityFrameworkCore.Sqlite" Version="9.0.6" />
17         <PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer" Version="9.0.6" />
18         <PackageReference Include="Microsoft.EntityFrameworkCore.Tools" Version="9.0.6">
19             <IncludeAssets>runtime; build; native; contentfiles; analyzers; buildtransitive</IncludeAssets>
20             <PrivateAssets>all</PrivateAssets>
21         </PackageReference>
22     </ItemGroup>
23
24 </Project>

```

OUTPUT:

```
PS C:\Users\Lenovo\LAB5> dotnet run
>>
C:\Users\Lenovo\LAB5\Models\Product.cs(8,25): warning CS8618: Non-nullable property 'Category' must contain a non-null value when exiting constructor. Consider adding the 'required' modifier or declaring the property as nullable.
Seeded sample data.
Smartphone - ₹69999
Laptop - ₹109999
Headphones - ₹2999
Found: Smartphone
Expensive: Smartphone
```

## Retail Inventory

```
Models > Category.cs > Category
1  using System.Collections.Generic;
2  using System.ComponentModel.DataAnnotations;
3
4  namespace RetailInventory.Models
5  {
6      4 references
7      public class Category
8      {
9          [Key]
10         0 references
11         public int Id { get; set; }
12
13         [Required]
14         3 references
15         public string? Name { get; set; }
16
17         0 references
18         public List<Product>? Products { get; set; }
19     }
20
21 }
```

```
Models > Product.cs > Product
1  using System.ComponentModel.DataAnnotations;
2  using System.ComponentModel.DataAnnotations.Schema;
3
4  namespace RetailInventory.Models
5  {
6      4 references
7      public class Product
8      {
9          [Key]
10         0 references
11         public int Id { get; set; }
12
13         [Required]
14         3 references
15         public string? Name { get; set; }
16
17         3 references
18         public int Stock { get; set; }
19
20         [ForeignKey("Category")]
21         0 references
22         public int CategoryId { get; set; }
23
24         4 references
25         public Category? Category { get; set; }
26     }
27 }
```



```

AppDbContext.cs > AppDbContext > OnConfiguring
1  using Microsoft.EntityFrameworkCore;
2  using RetailInventory.Models;
3
3 references
4  public class AppDbContext : DbContext
5  {
6      2 references
7      public DbSet<Product> Products { get; set; }
8      2 references
9      public DbSet<Category> Categories { get; set; }
10
0 references
11  protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
12  {
13      optionsBuilder.UseSqlServer("Server=localhost;Database=RetailDB;Trusted_Connection=True;TrustServerCertificate=True");
14  }

```

```

Program.cs > Program > Main
1  using RetailInventory.Models;
2  using Microsoft.EntityFrameworkCore;
3  using System;
4  using System.Linq;
5
0 references
6  class Program
7  {
8      0 references
9      static void Main()
10     {
11         using var context = new AppDbContext();
12
13         if (!context.Categories.Any())
14         {
15             var electronics = new Category { Name = "Electronics" };
16             var grocery = new Category { Name = "Grocery" };
17
18             context.Categories.AddRange(electronics, grocery);
19
20             context.Products.AddRange(
21                 new Product { Name = "Laptop", Stock = 5, Category = electronics },
22                 new Product { Name = "Banana", Stock = 20, Category = grocery }
23             );
24
25             context.SaveChanges();
26
27             Console.WriteLine("Products in Inventory:");
28             foreach (var p in context.Products.Include(p => p.Category))
29             {
30                 Console.WriteLine($"{p.Name} ({p.Category.Name}) - Stock: {p.Stock}");
31             }
32         }
33     }

```

OUTPUT:

```

Build succeeded in 0.7s
PS C:\Users\Lenovo\RetailInventory> dotnet build
>>
Restore complete (0.7s)
RetailInventory succeeded with 1 warning(s) (1.2s) -> bin\Debug\net9.0\RetailInventory.dll
C:\Users\Lenovo\RetailInventory\Program.cs(30,44): warning CS8602: Dereference of a possibly null reference.

Build succeeded with 1 warning(s) in 2.3s
PS C:\Users\Lenovo\RetailInventory> dotnet run
Products in Inventory:
Laptop (Electronics) - Stock: 5
Banana (Grocery) - Stock: 20

```



# Retail Inventory

```
Program.cs > Program > Main
1  using System;
2  using System.Linq;
3  using Microsoft.EntityFrameworkCore;
4  using RetailInventory;
5  using RetailInventory.Models;
6
7  0 references
8  class Program
9  {
10     0 references
11     static void Main(string[] args)
12     {
13         using var context = new AppDbContext();
14         if (!context.Categories.Any())
15         {
16             var electronics = new Category { Name = "Electronics" };
17             var phone = new Product { Name = "Smartphone", Price = 699.99M, Category = electronics };
18
19             context.Categories.Add(electronics);
20             context.Products.Add(phone);
21             context.SaveChanges();
22             Console.WriteLine("Data added successfully!");
23         }
24         else
25         {
26             var products = context.Products.Include(p => p.Category).ToList();
27             foreach (var product in products)
28             {
29                 Console.WriteLine($"{product.Name} - {product.Price:C} - Category: {product.Category.Name}");
30             }
31         }
32         Console.WriteLine("Press any key to exit...");
33         Console.ReadKey();
34     }
35 }
```

```
AppDbContext.cs > AppDbContext
1  using Microsoft.EntityFrameworkCore;
2  using RetailInventory.Models;
3
4  namespace RetailInventory
5  {
6     4 references
7     public class AppDbContext : DbContext
8     {
9         2 references
10         public DbSet<Product> Products { get; set; }
11         2 references
12         public DbSet<Category> Categories { get; set; }
13
14         0 references
15         protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
16         {
17             optionsBuilder.UseSqlServer("Server=localhost;Database=RetailDb;Trusted_Connection=True;TrustServerCertificate=True;");
18         }
19
20         0 references
21         protected override void OnModelCreating(ModelBuilder modelBuilder)
22         {
23             modelBuilder.Entity<Product>()
24                 .Property(p => p.Price)
25                 .HasColumnType("decimal(18,2)");
26         }
27     }
28 }
```

Output:

```
PS C:\Users\Lenovo\OneDrive\Desktop\RetailInventory> dotnet run
>>
Smartphone - ₹ 699.99 - Category: Electronics
Press any key to exit...
```