**CODE:**

Category.cs

public class Category

{

    public int Id { get; set; }

    public string Name { get; set; }

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

}

Product.cs

public class Product

{

    public int Id { get; set; }

    public string Name { get; set; }

    public decimal Price { get; set; }

    public int CategoryId { get; set; }

    public Category Category { get; set; }

}

Program.cs

using Microsoft.EntityFrameworkCore;

using var context = new AppDbContext();

var products = await context.Products.Include(p => p.Category).ToListAsync();

Console.WriteLine("\n--- All Products ---");

foreach (var p in products)

{

    Console.WriteLine($"{p.Name} - ₹{p.Price} - Category: {p.Category?.Name}");

}

var product = await context.Products.FindAsync(1);

Console.WriteLine($"\nFound by ID 1: {product?.Name}");

var expensive = await context.Products.FirstOrDefaultAsync(p => p.Price > 50000);

Console.WriteLine($"\nExpensive Product (> ₹50,000): {expensive?.Name}");

Inserting Data

using var context = new AppDbContext();

var electronics = new Category { Name = "Electronics" };

var groceries = new Category { Name = "Groceries" };

await context.Categories.AddRangeAsync(electronics, groceries);

var product1 = new Product { Name = "Laptop", Price = 75000, Category = electronics };

var product2 = new Product { Name = "Rice Bag", Price = 1200, Category = groceries };

await context.Products.AddRangeAsync(product1, product2);

await context.SaveChangesAsync();

Console.WriteLine("Initial data inserted");

appDb.cs

using Microsoft.EntityFrameworkCore;

using Npgsql.EntityFrameworkCore.PostgreSQL;

using System;

public class AppDbContext : DbContext

{

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

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

    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

    {

        optionsBuilder.UseNpgsql("Host=localhost;Port=5432;Database=retailinventorydb;Username=postgres;Password=postgres;");

    }

}

**OUTPUT:**



