```csharp
1  using DatabaseConnect.Entities;
2  using Microsoft.EntityFrameworkCore;
3  using System;
4  using System.Collections.Generic;
5  using System.Linq;
6  using System.Text;
7  using System.Threading.Tasks;
8
9  namespace DatabaseConnect
10 {
11     public class Context : DbContext
12     {
13         public Context(DbContextOptions DbCfg) : base(DbCfg)
14         {
15         }
16         // Simple
17         public DbSet<Book> Books { get; set; }
18         public DbSet<Author> Authors { get; set; }
19         public DbSet<Cover> Covers { get; set; }
20         public DbSet<User> Users { get; set; }
21         public DbSet<UType> UTypes { get; set; }
22         public DbSet<Checkout> Checkouts { get; set; }
23         public DbSet<Reservation> Reservations { get; set; }
24         // Relationships
25         public DbSet<AuthorBook> AuthorBook_rel { get; set; }
26         public DbSet<UserUType> UserUType_rel { get; set; }
27
28
29         protected override void OnModelCreating(ModelBuilder modelBuilder)
30         {
31             modelBuilder.Entity<AuthorBook>()  // Composite Key for AuthorBooks
32                 .HasKey(ab => new { ab.BookID, ab.AuthorID });
33
34             modelBuilder.Entity<AuthorBook>()  // One to many AuthorBooks.Book to ⮡
                 Book
35                 .HasOne(ab => ab.Book)
36                 .WithMany(b => b.AuthorBooks)
37                 .HasForeignKey(ab => ab.BookID);
38
39             modelBuilder.Entity<AuthorBook>()  // One to many AuthorBooks.Author ⮡
                 to Author
40                 .HasOne(ab => ab.Author)
41                 .WithMany(a => a.AuthorBooks)
42                 .HasForeignKey(ab => ab.AuthorID);
43
44             modelBuilder.Entity<UserUType>()  // Composite Key
45                 .HasKey(ut => new { ut.UserID, ut.UTypeID });
46
47             modelBuilder.Entity<Book>()
48                 .Property(a => a.DeweyDecimal)
49                 .HasDefaultValue("n/a");
50
```

```
51              modelBuilder.Entity<Book>()
52                  .Property(a => a.FicID)
53                  .HasDefaultValue("n/a");
54
55              modelBuilder.Entity<Book>()
56                  .Property(a => a.DeweyDecimal)
57                  .HasDefaultValue("n/a");
58
59              modelBuilder.Entity<AuthorBook>()
60                  .Property(ab => ab.AuthorID)
61                  .HasDefaultValue(3);
62
63          }
64      }
65  }
66
```