04.03 Creating Database with Entity Framework Core - Part 3

Purchase, Review and Favorite tables.

WE still have three tables remianing in our database.

Purchase table

A movie can be purchased by one User only once, that means a User will have many purchases but one purchase belongs to one user. Here we have **One-to-many** relationship.

Lets create the Purchase entity first and add Purchases collection to User entity

```
public class Purchase
   {
       public int Id { get; set; }
       public int UserId { get; set; }
       public Guid PurchaseNumber { get; set; }
       public decimal TotalPrice { get; set; }
       public DateTime PurchaseDateTime { get; set; }
       public int MovieId { get; set; }
       public Movie Movie { get; set; }
      public User Customer { get; set; }
   }
```

Also add Navigation property of Purchases in User Entity

```
public ICollection<Purchase> Purchases { get; set; }
```

Add DbSet of Purchases in MovieShopDbContext and Configuration in OnModelCreating

```
        public DbSet<Purchase> Purchases { get; set; }

        // this one is inside OnModelCreating method
        modelBuilder.Entity<Purchase>(ConfigurePurchase);
```

Here is the actual Configuration

```
private void ConfigurePurchase(EntityTypeBuilder<Purchase> builder)
        {
            builder.ToTable("Purchase");
```

```
            builder.HasKey(p => p.Id);
            builder.Property(p => p.Id).ValueGeneratedOnAdd();
            builder.Property(p => p.PurchaseNumber).ValueGeneratedOnAdd();

            // Combination of UserId and MovieId is Unique, that means a Movie can
be purchased by a user only once.
            builder.HasIndex(p => new { p.UserId, p.MovieId }).IsUnique();
        }
```

Now, add Migration and Update database

```
Add-Migration PurchaseTable
Update-Database
```

Favorite table

Similary, we are going to create Favorite table where A movie can be Favorited by one User only once, that means a User will have many diffeenr Movie Favorites but one Favorite belongs to one user. Here also we have **One-to-many** relationship.

Lets create the Favorite entity first and add Purchases collection to User entity

```
  public class Favorite
    {
        public int Id { get; set; }
        public int MovieId { get; set; }
        public int UserId { get; set; }
        public User User { get; set; }
        public Movie Movie { get; set; }
    }
```

Also add Navigation property of Favorites in User Entity

```
public ICollection<Favorite> Favorites { get; set; }
```

Add DbSet of Favorite in MovieShopDbContext and Configuration in OnModelCreating

```
        public DbSet<Favorite> Favorites { get; set; }

        // this one is inside OnModelCreating method
        modelBuilder.Entity<Favorite>(ConfigureFavorites);
```

Here is the actual Configuration

```
private void ConfigureFavorites(EntityTypeBuilder<Favorite> builder)
        {
            builder.ToTable("Favorite");
            builder.HasKey(p => p.Id);
            builder.HasIndex(f => new { f.MovieId, f.UserId }).IsUnique();
        }
```

Now, add Migration and Update database

```
Add-Migration FavoriteTable
Update-Database
```

Review table

Finally, we have Review entity where one Movie can have multiple reviews and one User can review Multiple Movies but he can review one particular Movie only Once.

Lets create Review Entity

```
public class Review
    {
        public int MovieId { get; set; }
        public int UserId { get; set; }
        public decimal Rating { get; set; }
        public string ReviewText { get; set; }
        public virtual User User { get; set; }
        public Movie Movie { get; set; }

    }

// Add Collection of reviews to Movie and User entities

 public ICollection<Review> Reviews { get; set; }
```

Add DbSet of Review in MovieShopDbContext and Configuration in OnModelCreating

```
        public DbSet<Review> Reviews { get; set; }

        // this one is inside OnModelCreating method
        modelBuilder.Entity<Review>(ConfigureReview);
```

Here is the actual Configuration

```
private void ConfigureReview(EntityTypeBuilder<Review> builder)
        {
            builder.ToTable("Review");
            builder.HasKey(r => new { r.MovieId, r.UserId });
            builder.Property(r => r.ReviewText).HasMaxLength(20000);
            builder.Property(r => r.Rating).HasColumnType("decimal(3, 2)");
        }
```

Now, add Migration and Update database

```
Add-Migration ReviewTable
Update-Database
```

With that we finished our Database creation with Entity Framework Code-First Approach.