

## Ödev 1: Kütüphane Yönetim Sistemi (Library Management System)

### Senaryo:

Basit bir kütüphane sistemi tasarlayacaksınız. Sistemde **Yazarlar** (Author) ve **Kitaplar** (Book) olacak. **Bir yazarın birden fazla kitabı olabilir, ancak her kitap sadece bir yazara aittir.** Bu bire-çok ilişkiyi EF Core'da modellemeniz beklenmektedir.

### İstenen Bilgiler:

- Her **yazarın** bir Id'si, bir tam adı ve bir doğum yılı olmalıdır.
- Her **kitabın** bir Id'si, bir başlığı ve sayfa sayısı bilgisi tutulmalıdır.

### Görevler:

- Yukarıdaki bilgileri ve senaryodaki ilişkiyi modelleyen Author ve Book entity sınıflarını oluşturun. Ardından LibraryContext adında bir DbContext sınıfı hazırlayın.
- add-migration ve update-database komutlarını kullanarak veritabanını oluşturun.
- Veri Ekleme:** En az 3 yazar ve her yazara ait en az 2'şer kitap ekleyin. SaveChanges() metodunu kullanarak bu verileri veritabanına kaydedin.
- Veri Listeleme (Projeksiyon ile):** Sadece kitapların başlıklarını (Title) ve o kitabın yazarının tam adını (FullName) gösterecek şekilde bir sorgu yazın. Bu sorgu sonucunu yeni bir anonim tipe yansıtarak konsola yazdırın. (Bu görev, ilişkiyi doğru kurup Include kullanmanızı gerektirir).
- Veri Güncelleme:** Belirli bir Id'ye sahip kitabın sayfa sayısı bilgisini (PageCount) güncelleyin ve değişikliği veritabanına kaydedin.

## Ödev 2: Şirket Departman ve Çalışan Yönetimi (Company Department and Employee Management)

### Senaryo:

Bir şirketin organizasyon yapısını modelleyeceksiniz. Şirkette **Departmanlar** (Department) ve bu departmanlara bağlı **Çalışanlar** (Employee) olacak. **Bir departmanda birden fazla çalışan olabilir, ancak her çalışan yalnızca bir departmana bağlıdır.**

### İstenen Bilgiler:

- Her **departmanın** bir Id'si ve bir adı olmalıdır.
- Her **çalışanın** bir Id'si, tam adı, pozisyonu ve maaş bilgisi saklanmalıdır.

### Görevler:

- Senaryoya uygun Department ve Employee entity'lerini ve CompanyContext adında bir DbContext'i oluşturun.
- Migration'ları kullanarak veritabanını oluşturun.
- Veri Ekleme:** "Software", "Accounting" ve "Human Resources" adında üç departman oluşturun. Her departmana en az 2 çalışan ekleyin.

4. **Veri Sorgulama (Filtreleme):** "Software" departmanında çalışan ve maaşı (Salary) belirli bir miktardan yüksek olan çalışanların listesini konsola yazdırın.
5. **Veri Silme:** Bir çalışanı Id'sine göre bulup işten çıkarın (veritabanından silin). SaveChanges() ile değışiklięi onaylayın.

## Ödev 3: Müşteri Sipariş Sistemi (Customer Order System)

### Senaryo:

Basit bir e-ticaret senaryosu canlandırılacak. Sistemde **Müşteriler** (Customer) ve bu müşterilerin verdiği **Siparişler** (Order) tutulacak. **Bir müşterinin birden fazla siparişı olabilir, ancak her sipariş sadece bir müşteriye aittir.**

### İstenen Bilgiler:

- Her **müşterinin** bir Id'si, adı, soyadı ve e-posta adresi olmalıdır.
- Her **siparişin** bir Id'si, siparişin verildięi tarih ve siparişin toplam tutarı bilgisi bulunmalıdır.

### Görevler:

1. Yukarıdaki senaryoyu modelleyen Customer ve Order entity sınıflarını ve ECommerceContext adında bir DbContext'i oluşturun.
2. Veritabanını migration'lar ile oluşturun.
3. **Veri Ekleme:** En az 2 müşteri ve her müşteriye ait en az 3'er sipariş ekleyin.
4. **İlişkili Veri Çekme:** Belirli bir müşterinin (Id ile bulunacak) tüm siparişlerini, sipariş tarihine (OrderDate) göre en yeniden en eskiye doğru sıralayarak konsola yazdırın (Include ve OrderByDescending kullanmalısınız).
5. **Veri Güncelleme:** Bir müşterinin e-posta (Email) adresini deęiştirin ve SaveChanges() ile veritabanını güncelleyin.

## Ödev 4: Film ve Kategori Sistemi (Movie and Category System)

### Senaryo:

Bir film veritabanı oluşturacaksınız. Sistemde **Kategoriler** (Category) ve bu kategorilere ait **Filmler** (Movie) olacak. **Bir kategoride birden çok film olabilir, ancak her film yalnızca bir kategoriye aittir.**

### İstenen Bilgiler:

- Her film **kategorisinin** bir Id'si ve bir adı olmalıdır.
- Her **filmin** bir Id'si, başlığı, yönetmeni ve yapım yılı bilgileri tutulmalıdır.

### Görevler:

1. Senaryoyu ve istenen bilgileri dikkate alarak Category ve Movie entity'lerini ve MovieContext adında bir DbContext sınıfını oluşturun.
2. Migration'lar ile veritabanını ayaęa kaldırın.
3. **Veri Ekleme:** En az 3 kategori ve her kategoriye ait en az 2 film ekleyin.

4. **Veri Listeleme (Projeksiyon):** Her filmin adını (Title), yönetmenini (Director) ve ait olduğu kategorinin adını (CategoryName) içeren bir liste oluşturup konsola yazdırın. (Bu görev için Select ve Include kullanmanız gerekecektir).
5. **Veri Silme:** Adı "Comedy" olan kategoriye ve bu kategoriye bağlı **tüm filmleri** silin. (Bu görev, ilişkili verileri silerken dikkatli olmanız gerektiğini anlamanızı sağlayacaktır).

## Ödev 5: Blog Sistemi (Blog System with Posts, Categories, and Comments)

### Senaryo:

Bu ödevde daha katmanlı bir ilişki yapısı kuracaksınız. Sistemde **Kategoriler** (Category), **Blog Yazıları** (Post) ve **Yorumlar** (Comment) olacak.

İlişkiler şu şekildedir:

- Her Post yalnızca bir Category'ye aittir, ancak bir Category'nin birden fazla Post'u olabilir.
- Benzer şekilde, her Post'a birden fazla Comment yapılabilir, ancak her Comment yalnızca bir Post'a aittir.

### İstenen Bilgiler:

- Her **kategorinin** bir Id'si ve bir adı olmalıdır.
- Her **blog yazısının** (Post) bir Id'si, başlığı, içeriği ve yayınlanma tarihi olmalıdır.
- Her **yorumun** (Comment) bir Id'si, yorumu yapan kişinin adı ve yorum mesajı tutulmalıdır.

### Görevler:

1. Bu üç katmanlı ilişkiyi modelleyecek Category, Post ve Comment entity'lerini ve BlogContext'i oluşturun.
2. Migration'ları kullanarak veritabanını oluşturun.
3. **Veri Ekleme:** Birkaç kategori, bu kategorilere ait postlar ve bu postlara ait yorumlar ekleyin.
4. **İleri Düzey İlişkili Veri Çekme:** Belirli bir kategorideki (Id ile bulunacak) tüm postları, bu postlara yapılmış yorumlarla birlikte getirin. Sonucu konsola hiyerarşik bir yapıda yazdırın (Kategori Adı -> Post Başlığı -> Yorumlar). Bu görev, Include ve ThenInclude metodlarını doğru bir şekilde kullanmanızı gerektirir.
5. **Veri Güncelleme:** Belirli bir yoruma (Id ile bularak) ait mesaj (Message) içeriğini güncelleyin.