Veri Modeli ■nceleme ve Öneriler Raporu

FEZAR SA■LIK YAZILIM DANI■MANLI■I

Kaynak: APPROVAL123.txt

Bu rapor, yüklenen TXT içeri■indeki veri modeli s■n■flar■n■, ili■kileri ve normalizasyon önerilerini bölümlere ay■rarak sunar. Kod bloklar■ monospace biçiminde, aç■klamalar metin olarak i■lenmi■tir.

■çindekiler

APPROVAL

```
using ilterisg.Models.Enums;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("Approvals", Schema = "dbo")]
    public class Approval
    {
        [Key]
        public int Id { get; set; }

[Required]
        public string RecordId { get; set; } // Onaylanacak kayd■n ID'si

[Required]
        [StringLength(100)]
        public string TableName { get; set; } // Hangi tabloya ait?

[StringLength(100)]
        public string? ApproverRole { get; set; }

[StringLength(450)]
        [ForeignKey("ApprovedByUser")]
        public string? ApprovedByUserId { get; set; }

[Required]
        public ApprovalStatus Status { get; set; } = ApprovalStatus.Pending;

[Required]
        public ApprovalType ApprovalType { get; set; } // Yeni eklenen alan: Hesap m■, ödeme mi?

public DateTime? ApprovedDate { get; set; }

public string? Notes { get; set; }

// Navigation Property
        public virtual ApplicationUser? ApprovedByUser { get; set; }
    }
}
```

Yapay Zeka Notu:Model iyi yap■land■r■lm■■ iyile■tirme olarak

```
public virtual ApplicationUser? ApprovedByUser { get; set; }
public virtual ApplicationUser? CreatedByUser { get; set; }
```

bu alanlar eklenebilir gereksiz sütun yok denildi.

B■LD■R■MLER

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("Bildirimler", Schema = "dbo")]
    public class Bildirimler
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int BildirimId { get; set; }

[Required]
        [MaxLength(450)]
        public string RefUserId { get; set; } // Bildirimin gönderildi■i kullan■c■ (ApplicationUse

[ForeignKey(nameof(RefUserId))]
```

```
        public virtual ApplicationUser ApplicationUser { get; set; }

[Required]
        [MaxLength(500)]
        public string Mesaj { get; set; } // Bildirim mesaj■ (örne■in, "Toplant■ X 15 Nisan 2025 t

public DateTime GonderimTarihi { get; set; } = DateTime.Now;

public bool OkunduMu { get; set; } = false; // Okundu/Okunmad■ durumu

[MaxLength(100)]
        public string BildirimTuru { get; set; } // Örne■in: "Etkinlik Olu■turma", "Etkinlik Silme

public int? RefEgitimId { get; set; } // ■lgili e■itim (opsiyonel, e■itimle ilgiliyse)

[ForeignKey(nameof(RefEgitimId))]
        public virtual Egitimler? Egitim { get; set; }

public int? RefEtkinlikId { get; set; } // ■lgili etkinlik (opsiyonel, etkinlik ile ilgiliyse)

[ForeignKey(nameof(RefEtkinlikId))]
        public virtual Etkinlikler? Etkinlik { get; set; }

[MaxLength(500)] // URL uzunlu■u için uygun bir s■n■r
        public string? RedirectUrl { get; set; } // Yönlendirme URL'si
    }
}

ReftableName string de■il enum yapmak ve her reftable için bir enum belirlemek stringden daha güven
        [Required]
        [MaxLength(450)]
        public string RefUserId { get; set; } // Bildirimin gönderildi■i kullan■c■ (ApplicationUse

// Opsiyonel audit eklentisi
        [MaxLength(450)]
        public string? CreatedByUserId { get; set; } // Kim gönderdi?
        [ForeignKey(nameof(CreatedByUserId))]
        public virtual ApplicationUser? CreatedByUser { get; set; }
```

Gereksiz sütun bulunamad■.

BLOGPOST

```
using Microsoft.AspNetCore.Mvc.ModelBinding;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    public class BlogPost
    {
        [Key]
        public int Id { get; set; }

[Required]
        [MaxLength(200)]
        public string Title { get; set; }

[Required]
        public string Content { get; set; }

public DateTime CreatedAt { get; set; } = DateTime.Now;

public bool IsPublished { get; set; } = true;

// ■ Görsel için
        [MaxLength(500)]
        public string? ImageUrl { get; set; }

// ■ Özet (opsiyonel ama önerilir)
        [MaxLength(500)]
        public string? Summary { get; set; }
```

```
// ■ Popülerlik takibi için
        public int ViewCount { get; set; } = 0;

// ■ Kullan■c■ bilgileri
        [BindNever]
        [MaxLength(450)]
        public string AuthorUserId { get; set; }

[BindNever]
        [ForeignKey(nameof(AuthorUserId))]
        public virtual ApplicationUser Author { get; set; }
    }
}
```

%90 iyi yap■land■r■lm■■ ve sorunsuz sadece IsPublished alan■ boolean de■ilde Enum olarak yap■land■r■lsayd■ ar■ivlendi, yay■nland■, kald■r■ld■ gibi çoklu i■lem opsiyonlar■ olabilirdi. Güncellemeler için UpdatedUserId ve UpdateDate gibi alanlar eklenebilir.

DestekFeedback

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using ilterisg.Models.Enums;

namespace ilterisg.Models
{
    [Table("DestekTalepleri", Schema = "dbo")]
    public class DestekTalepleri
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int TalepId { get; set; }

[Required]
        [MaxLength(200)]
        public string KonuBasligi { get; set; }

[Required]
        public string Icerik { get; set; }

[Required]
        public int RefKullaniciId { get; set; }

[ForeignKey(nameof(RefKullaniciId))]
        public virtual Kullanicilar Kullanici { get; set; }

public int? RefFirmaId { get; set; } // Nullable, firma opsiyonel

[ForeignKey(nameof(RefFirmaId))]
        public virtual Firmalar? Firma { get; set; }

[MaxLength(450)] // AspNetUsers.Id ile e■le■mesi için
        public string? CozumuYapanId { get; set; } // Nullable, çözüm yapan admin (ApplicationUser.

[ForeignKey(nameof(CozumuYapanId))]
        public virtual ApplicationUser? CozumuYapan { get; set; }

[MaxLength(450)] // AspNetUsers.Id ile e■le■mesi için
        public string? SahiplenenAdminId { get; set; } // Nullable, talebi sahiplenen admin (Applic

[ForeignKey(nameof(SahiplenenAdminId))]
        public virtual ApplicationUser? SahiplenenAdmin { get; set; }

[Required]
        public DateTime OlusturmaTarihi { get; set; } = DateTime.Now;

public DateTime? GuncellemeTarihi { get; set; } // Nullable, güncelleme opsiyonel

[Required]
        public DestekTalebiDurumu Durum { get; set; } = DestekTalebiDurumu.Acik;
```

```
[Required]
        public OnemDurumu OnemDurumu { get; set; } = OnemDurumu.Orta;

[Required]
        public KonuTuru KonuTuru { get; set; }

public virtual ICollection<DestekMesajlari> Mesajlar { get; set; } = new List<DestekMesajlari>();

public virtual ICollection<Dokumanlar> Dokumanlar { get; set; } = new List<Dokumanlar>();
        public virtual DestekFeedback Feedback { get; set; } // Tek bir feedback varsay■m■
    }
}
```

Sadele■tirme Önerisi: RefKullaniciId ve RefFirmaId'i kald■r – bunlar transitive (Talep üzerinden e
Tarih: DateTime.Now yerine DateTime.UtcNow kullan – timezone sorunlar■n■ önler.
Yildiz: Int yeterli, ama enum yap (e.g., Rating enum 1-5) veri tutarl■l■■■ için – opsiyonel.
Audit Geni■letme: E■er feedback editlenecekse UpdatedAt/Yorum ekle. Index'ler migration'da: (RefTa
Geni■letme: E■er foto■rafl■ feedback varsa ImageUrl ekle, ama mevcut haliyle minimal iyi.

DestekTalepler tablosundan Eri■im yap■labilir RefKullan■c■Id ve RefFirmaId'ye
Önerilen Yap■;

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("DestekFeedback", Schema = "dbo")]
    public class DestekFeedback
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int FeedbackId { get; set; }

[Required]
        public int RefTalepId { get; set; }
        [ForeignKey(nameof(RefTalepId))]
        public virtual DestekTalepleri Talep { get; set; } = null!; // Talep üzerinden Kullanici/Fi

[MaxLength(450)]
        public string? CozumuYapanId { get; set; } // Nullable, çözüm yapan admin
        [ForeignKey(nameof(CozumuYapanId))]
        public virtual ApplicationUser? CozumuYapan { get; set; }

[Required]
        [Range(1, 5)]
        public int Yildiz { get; set; } // Veya enum yap

[MaxLength(500)]
        public string? Yorum { get; set; } // Opsiyonel yorum

[Required]
        public DateTime Tarih { get; set; } = DateTime.UtcNow; // UtcNow'a çevir

// Opsiyonel audit eklentisi (e■er edit varsa)
        public DateTime? UpdatedAt { get; set; }
    }
}
```

DESTEKTALEPLER■

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using ilterisg.Models.Enums;

namespace ilterisg.Models
{
    [Table("DestekTalepleri", Schema = "dbo")]
    public class DestekTalepleri
    {
```

```
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int TalepId { get; set; }

[Required]
        [MaxLength(200)]
        public string KonuBasligi { get; set; }

[Required]
        public string Icerik { get; set; }

[Required]
        public int RefKullaniciId { get; set; }

[ForeignKey(nameof(RefKullaniciId))]
        public virtual Kullanicilar Kullanici { get; set; }

public int? RefFirmaId { get; set; } // Nullable, firma opsiyonel

[ForeignKey(nameof(RefFirmaId))]
        public virtual Firmalar? Firma { get; set; }

[MaxLength(450)] // AspNetUsers.Id ile e■le■mesi için
        public string? CozumuYapanId { get; set; } // Nullable, çözüm yapan admin (ApplicationUser.

[ForeignKey(nameof(CozumuYapanId))]
        public virtual ApplicationUser? CozumuYapan { get; set; }

[MaxLength(450)] // AspNetUsers.Id ile e■le■mesi için
        public string? SahiplenenAdminId { get; set; } // Nullable, talebi sahiplenen admin (Applic

[ForeignKey(nameof(SahiplenenAdminId))]
        public virtual ApplicationUser? SahiplenenAdmin { get; set; }

[Required]
        public DateTime OlusturmaTarihi { get; set; } = DateTime.Now;

public DateTime? GuncellemeTarihi { get; set; } // Nullable, güncelleme opsiyonel

[Required]
        public DestekTalebiDurumu Durum { get; set; } = DestekTalebiDurumu.Acik;

[Required]
        public OnemDurumu OnemDurumu { get; set; } = OnemDurumu.Orta;

[Required]
        public KonuTuru KonuTuru { get; set; }

public virtual ICollection<DestekMesajlari> Mesajlar { get; set; } = new List<DestekMesajlari>();

public virtual ICollection<Dokumanlar> Dokumanlar { get; set; } = new List<Dokumanlar>();
        public virtual DestekFeedback Feedback { get; set; } // Tek bir feedback varsay■m■
    }
}
```

Tarihler: DateTime.Now yerine DateTime.UtcNow kullan – timezone uyumsuzluklar■ önler (kullan■c■lar global olabilir).

■yi yap■land■r■lm■■ gereksiz alan mevcut de■il.

DÖKÜMANLAR

```
using ilterisg.Models.Enums;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("Dokumanlar", Schema = "dbo")]
    public class Dokumanlar
    {
        [Key]
```

```
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int DokumanId { get; set; }

[Required]
        public int RefId { get; set; } // Ba■lanaca■■ kayd■n ID’si (ödeme, kullan■c■, sözle■me, e

[Required]
        [MaxLength(100)]
        public string RefTableName { get; set; } // Hangi tabloya ba■l■? (Odemeler, Sozlesmeler, E

[Required]
        [MaxLength(255)]
        public string DosyaAdi { get; set; }

[Required]
        public string DosyaYolu { get; set; }
        [MaxLength(500)]
        public string? KlasorYolu { get; set; } // Yeni alan: Klasör hiyerar■isi (örne■in, "Egitim

[Required]
        public DateTime YuklemeTarihi { get; set; } = DateTime.Now;

[MaxLength(50)]
        public DokumanTuru DokumanTuru { get; set; } // "Dekont", "Kimlik", "Sözle■me", "EtkinlikDo

[MaxLength(500)]
        public string? Aciklama { get; set; }

[MaxLength(128)] // AspNetUsers.Id uzunlu■una uygun
        public string? UserId { get; set; } // Kullan■c■ ID'sini saklamak için yeni alan
    }
}
```

Yap■ zaten iyi, her modül için reusable – zorunlu de■i■iklik yok. Ama normalization ve maintainability için:

```
RefTableName: String yerine enum yap (e.g., DokumanRefType: Odemeler, Kullanicilar, Etkinlikler vb.
```

YuklemeTarihi: DateTime.Now yerine DateTime.UtcNow kullan – timezone sorunlar■n■ önler (global dosya upload'lar■ için).

```
UserId: FK yap ve navigation ekle (ApplicationUser) – EF ile dokuman.User diye eri■. [MaxLength(128
```

```
using ilterisg.Models.Enums;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("Dokumanlar", Schema = "dbo")]
    public class Dokumanlar
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int DokumanId { get; set; }

[Required]
        public int RefId { get; set; } // Ba■lanaca■■ kayd■n ID’si (polymorphic)

[Required]
        public DokumanRefType RefTableName { get; set; } // String'den enum'a (yeni enum tan■mla)

[Required]
        [MaxLength(255)]
        public string DosyaAdi { get; set; }

[Required]
        [MaxLength(1000)] // Uzun path'ler için geni■let
        public string DosyaYolu { get; set; }

[MaxLength(500)]
```

```
        public string? KlasorYolu { get; set; } // Opsiyonel hiyerar■i

[Required]
        public DateTime YuklemeTarihi { get; set; } = DateTime.UtcNow; // UtcNow'a çevir

[Required]
        public DokumanTuru DokumanTuru { get; set; } // Enum zaten iyi

[MaxLength(500)]
        public string? Aciklama { get; set; }

// FK ve navigation ekle
        [MaxLength(450)] // AspNetUsers.Id için do■ru uzunluk
        [ForeignKey(nameof(User))]
        public string? UserId { get; set; }
        public virtual ApplicationUser? User { get; set; }
    }

// Yeni enum örne■i (Enums klasöründe, mevcut modüllere göre geni■let)
    public enum DokumanRefType
    {
        Odemeler,
        Kullanicilar,
        Sozlesmeler,
        Etkinlikler,
        Egitimler
        // Yeni modül ekle: , YeniModul
    }
}
```

DOKUMAN PAYLA■IM

```
using System.ComponentModel.DataAnnotations;

namespace ilterisg.Models
{
    public class DokumanPaylasim
    {
        [Key]
        public int PaylasimId { get; set; } // Primary key
        public string Token { get; set; } // Unique token for sharing
        public List<int> DokumanIdList { get; set; } // List of document IDs, stored as JSON
        public string RefTableName { get; set; } // Reference table (e.g., Firmalar, Kullanicilar)
        public int RefId { get; set; } // Reference ID (e.g., FirmaId or KullaniciId)
        public string PaylasanKullaniciId { get; set; } // ID of the user who created the share
        public bool SureliMi { get; set; } // Is the share time-limited?
        public DateTime? GecerlilikTarihi { get; set; } // Expiry date for time-limited shares
        public bool SadeceGoruntulemeMi { get; set; } // View-only permission
        public DateTime OlusturmaTarihi { get; set; } // Creation date
    }
}
```

DokumanIdList alan■n■ kald■r■p bir Junction tabloda saklamak daha sa■lam bir yap■ sa■lar Id 1, 2, 3 gibi tutmak yer■ne payla■■m id ile junction, ara tabloda tutmak daha sa■lam

```
Önerilen yap■ alttad■r;

using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("DokumanPaylasim", Schema = "dbo")]
    public class DokumanPaylasim
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
```

```
        public int PaylasimId { get; set; } // Primary key

[Required]
        [MaxLength(450)] // Token uzunlu■u için (Guid string)
        public string Token { get; set; } // Unique token, service'te generate et

[Required]
        [MaxLength(100)]
        public string RefTableName { get; set; } // Reference table

[Required]
        public int RefId { get; set; } // Reference ID

[Required]
        [MaxLength(450)]
        [ForeignKey(nameof(PaylasanKullanici))]
        public string PaylasanKullaniciId { get; set; } // FK

public bool SureliMi { get; set; } = false; // Default false

public DateTime? GecerlilikTarihi { get; set; } // Validation: SureliMi ise required

public bool SadeceGoruntulemeMi { get; set; } = true; // Default view-only, güvenlik için

[Required]
        public DateTime OlusturmaTarihi { get; set; } = DateTime.UtcNow; // UtcNow default

// Navigation Properties
        public virtual ApplicationUser PaylasanKullanici { get; set; } = null!;

// Many-to-many: Doküman listesi için junction collection
        public virtual ICollection<DokumanPaylasimDokuman> DokumanPaylasimDokumanlar { get; set; }
    }

// Yeni junction model (ayr■ tablo: DokumanPaylasimDokumanlar)
    [Table("DokumanPaylasimDokumanlar", Schema = "dbo")]
    public class DokumanPaylasimDokuman
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Id { get; set; }

[Required]
        public int PaylasimId { get; set; }
        [ForeignKey(nameof(PaylasimId))]
        public virtual DokumanPaylasim Paylasim { get; set; } = null!;

[Required]
        public int DokumanId { get; set; }
        [ForeignKey(nameof(DokumanId))]
        public virtual Dokumanlar Dokuman { get; set; } = null!;

// Opsiyonel: Per-doküman permission (e.g., bu doküman view-only m■?)
        public bool SadeceGoruntulemeMi { get; set; } = true;
    }
}
```

EG■T■M GRUPLARI - EG■T■MGRUPKULLANICILAR - KULLAN■C■_EG■T■MGRUBU

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("EgitimGruplari", Schema = "dbo")]
    public class EgitimGruplari
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
```

```
        public int GrupId { get; set; }

[Required]
        [MaxLength(100)]
        public string GrupAdi { get; set; }

[MaxLength(500)]
        public string? Aciklama { get; set; }

public TehlikeSinifi TehlikeSinifi { get; set; }

public int? RefFirmaId { get; set; } // Firma ile ili■ki

public bool SilindiMi { get; set; } = false;

// Navigation properties
        [ForeignKey(nameof(RefFirmaId))]
        public virtual Firmalar Firma { get; set; }

public virtual ICollection<EgitimGrupKullanicilar> EgitimGrupKullanicilar { get; set; } = new List<

// E■itimlerle ili■ki
        public virtual ICollection<Egitimler> Egitimler { get; set; } = new List<Egitimler>();

// S■nav oturumlar■yla ili■ki
        public virtual ICollection<SinavOturumlari> SinavOturumlari { get; set; } = new List<SinavO
        public virtual ICollection<Kullanici_EgitimGrubu> KullaniciEgitimGruplari { get; set; } = n
    }
}

using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("EgitimGrupKullanicilar", Schema = "dbo")]
    public class EgitimGrupKullanicilar
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Id { get; set; }

[Required]
        public int RefGrupId { get; set; }

[Required]
        public int RefKullaniciId { get; set; }

[ForeignKey(nameof(RefGrupId))]
        public virtual EgitimGruplari Grup { get; set; }

[ForeignKey(nameof(RefKullaniciId))]
        public virtual Kullanicilar Kullanici { get; set; }
    }
}

using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("Kullanici_EgitimGrubu", Schema = "dbo")]
    public class Kullanici_EgitimGrubu
    {
        [Key, Column(Order = 0)]
        public int RefKullaniciId { get; set; }

[Key, Column(Order = 1)]
        public int RefGrupId { get; set; }

[ForeignKey(nameof(RefKullaniciId))]
        public virtual Kullanicilar Kullanici { get; set; }

[ForeignKey(nameof(RefGrupId))]
```

```
        public virtual EgitimGruplari Grup { get; set; }
    }
}
```

EgitimGruplari ve JuNCTion tablesi EgitimGrupKullanicilar iyi yap■land■r■lm■■ normal■zasyon için ekleme ve ç■karmaya gerek yok Kullanici_EgitimGrubu isimli duplicate bir junction tablo daha bulundu o kald■r■lacak.

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("Egitimler", Schema = "dbo")]
    public class Egitimler
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int EgitimId { get; set; }

[MaxLength(200)]
        public string Ad { get; set; } // E■itim ad■

public DateTime EgitimTarihi { get; set; } // E■itim tarihi

public int Sure { get; set; } // Süre (saat cinsinden)

public TehlikeSinifi TehlikeSinifi { get; set; } // Enum: AzTehlikeli, Tehlikeli, CokTehlikeli

public int? EgitimTuruId { get; set; } // E■itim türü (foreign key), nullable

public DateTime? YenilemeTarihi { get; set; } // Periyodik e■itimler için yenileme tarihi

public int? GrupId { get; set; } // E■itim grubu

public int? EgitmenId { get; set; } // Yeni alan: E■itmen

public bool TamamlandiMi { get; set; } = false; // E■itimin tamamlan■p tamamlanmad■■■

public bool SilindiMi { get; set; } = false; // Soft delete için silme durumu

// Navigation properties
        [ForeignKey(nameof(EgitimTuruId))]
        public virtual EgitimTurleri EgitimTuru { get; set; }

[ForeignKey(nameof(GrupId))]
        public virtual EgitimGruplari Grup { get; set; } // Yeni navigation property

// Bir e■itime birden fazla firma kat■labilir (Firma_Egitim tablosu üzerinden)
        public virtual ICollection<Firma_Egitim> FirmaEgitimler { get; set; } = new List<Firma_Egit

// Bir e■itime birden fazla kullan■c■/çal■■an kat■labilir (Kullanici_Egitim tablosu üzerinden)
        public virtual ICollection<Kullanici_Egitim> KullaniciEgitimler { get; set; } = new List<Ku
    }
}
```

E■itim sistemimizin eski kal■nt■ bir tablo yeni yap■land■r■lm■■ S■nav sistemimizde tablo hiçbir ■ekilde kullan■lm■yor, kald■r■labilir.

EG■T■MTURUMATERYALLER■

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("EgitimTuruMateryalleri", Schema = "dbo")]
    public class EgitimTuruMateryalleri
    {
```

```
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Id { get; set; }

[Required]
        public int EgitimTuruId { get; set; }

[Required]
        public int DokumanId { get; set; } // Materyalin Dokumanlar tablosundaki ID'si

[ForeignKey(nameof(EgitimTuruId))]
        public virtual EgitimTurleri EgitimTuru { get; set; }

[ForeignKey(nameof(DokumanId))]
        public virtual Dokumanlar Dokuman { get; set; }
    }
}
```

Gereksiz Alan?: Hay■r, minimal – ekstra metadata (e.g., s■ralama) yoksa bu haliyle yeterli. E■er materyalin "s■ras■" (order) önemliyse, int S■raNo ekle (ama zorunlu de■il).

Potansiyel ■yile■tirmeler (De■i■iklik Gerekir mi?):
Yap■ zaten production-ready, zorunlu de■i■iklik yok – ama long-term için ufak eklemeler öneririm (normalization'■ bozmaz, audit/performans art■r■r):

```
Audit Alanlar■: CreatedAt/UpdatedAt ekle (DateTime.UtcNow default) – materyal atamalar■n■n ne zama
Soft Delete: SilindiMi bool ekle (default false) – materyali silmek yerine gizle (veri kayb■ önler,
Index'ler: Migration'da composite unique index (EgitimTuruId, DokumanId) ekle – duplicate atamay■ ö
Geni■letme: E■er materyal-tür ili■kisinde ek info (e.g., zorunlu mu/opsiyonel mi) varsa, enum Duru
```

EG■T■MTURUSINAVLARI - EG■T■M TURLER■ - SINAVLAR

S■navlar sorular■n tutuldu■u taslak s■nav■ belirler, e■iti turu s■navlar■ s■nav içeri■indeki sorular■ belirler.

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("Sinavlar", Schema = "dbo")]
    public class Sinavlar
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int SinavId { get; set; }

[Required(ErrorMessage = "TrainingTypeRequired")]
        public int EgitimTuruId { get; set; }

[Required(ErrorMessage = "ExamNameRequired")]
        [MaxLength(100, ErrorMessage = "ExamNameMaxLength")]
        [MinLength(3, ErrorMessage = "ExamNameMinLength")]
        public string SinavAdi { get; set; }

[Required(ErrorMessage = "RepeatCountRequired")]
        [Range(1, 10, ErrorMessage = "RepeatCountRange")]
        public int TekrarlamaSayisi { get; set; } = 3;

[Required(ErrorMessage = "PassingScoreRequired")]
        [Range(0, 100, ErrorMessage = "PassingScoreRange")]
        public int GecmePuani { get; set; } = 70;

public bool SilindiMi { get; set; } = false;
```

```
// Navigation properties
        [ForeignKey(nameof(EgitimTuruId))]
        public virtual EgitimTurleri EgitimTuru { get; set; }

public virtual ICollection<EgitimTuruSinavlari> Sorular { get; set; } = new List<EgitimTuruSinavlar
    }
}

using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("EgitimTurleri", Schema = "dbo")]
    public class EgitimTurleri
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int EgitimTuruId { get; set; }

[Required]
        [MaxLength(100)]
        public string Ad { get; set; } // Tür ad■

[MaxLength(500)]
        public string? Aciklama { get; set; } // Tür aç■klamas■

[MaxLength(50)]
        public string? EgitimKodu { get; set; } // ■BYS e■itim kodu (örne■in, "140")

public bool SilindiMi { get; set; } = false; // Soft delete için

// Navigation properties
        public virtual ICollection<Egitimler> Egitimler { get; set; } = new List<Egitimler>();

public virtual ICollection<EgitimTuruMateryalleri> Materyaller { get; set; } = new List<EgitimTuruM

public virtual ICollection<EgitimTuruSinavlari> Sinavlar { get; set; } = new List<EgitimTuruSinavla
    }
}

using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("EgitimTuruSinavlari", Schema = "dbo")]
    public class EgitimTuruSinavlari
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Id { get; set; }

[Required]
        public int EgitimTuruId { get; set; }

public int? SinavId { get; set; }

[Required]
        [MaxLength(500)]
        public string SoruMetni { get; set; }

[Required]
        [MaxLength(100)]
        public string SecenekA { get; set; }

[Required]
        [MaxLength(100)]
        public string SecenekB { get; set; }

[Required]
        [MaxLength(100)]
        public string SecenekC { get; set; }
```

```
[Required]
        [MaxLength(100)]
        public string SecenekD { get; set; }

[Required]
        [MaxLength(50)]
        public string DogruCevap { get; set; }

// Navigation properties
        [ForeignKey(nameof(EgitimTuruId))]
        public virtual EgitimTurleri? EgitimTuru { get; set; }

[ForeignKey(nameof(SinavId))]
        public virtual Sinavlar Sinav { get; set; }
    }
}
```

Güçlü Yönler:

```
■li■kiler ve Esneklik: EgitimTuruId required FK ile her soru bir türe zorunlu ba■l■ (kategorile■t
Veri Bütünlü■ü: [Required]/[MaxLength] validation'lar■ sa■lam (SoruMetni geni■, seçenekler k■sa).
Mant■k Uyumu: Taslak sorular■ için ideal (SinavId null = taslak). Collections (Materyaller, Sinavl
Gereksiz Alan?: Hay■r, hepsi core – EgitimKodu gibi opsiyonel alanlar (IBYS entegrasyonu için) fayd
```

Potansiyel ■yile■tirmeler (De■i■iklik Gerekir mi?):

Yap■ zaten iyi, zorunlu de■i■iklik yok – ama normalization ve maintainability için:

```
DogruCevap: String yerine enum yap (e.g., CevapSecenegi: A, B, C, D) – veri tutarl■l■■■ artar, val
Audit Eklemeleri: CreatedAt/UpdatedAt ekle (DateTime.UtcNow default) – taslak/soru de■i■ikliklerin
```

EG■T■MTURUMATERYALLER■

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("EgitimTuruMateryalleri", Schema = "dbo")]
    public class EgitimTuruMateryalleri
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Id { get; set; }

[Required]
        public int EgitimTuruId { get; set; }

[Required]
        public int DokumanId { get; set; } // Materyalin Dokumanlar tablosundaki ID'si

[ForeignKey(nameof(EgitimTuruId))]
        public virtual EgitimTurleri EgitimTuru { get; set; }

[ForeignKey(nameof(DokumanId))]
        public virtual Dokumanlar Dokuman { get; set; }
    }
}
```

Güçlü Yönler:

```
Many-to-Many ■li■ki: EgitimTuruId ve DokumanId [Required] FK'ler ile zorunlu ba■, navigation'lar E
Veri Bütünlü■ü: [Key]/[DatabaseGenerated] otomatik ID, [Required] ile bo■ atama yok – validation s
Esneklik: E■itim materyallerini tür bazl■ kategorize etmek için ideal (Dokumanlar'daki polymorphic
Gereksiz Alan?: Hay■r, core – ekstra info (e.g., materyalin zorunlu mu oldu■u) yoksa bu hali yeter
```

Potansiyel ■yile■tirmeler (De■i■iklik Gerekir mi?):
Yap■ zaten production-ready, zorunlu de■i■iklik yok – ama scalability ve traceability için ufak ekler öneririm (normalization'■ bozmaz):

Audit Alanlar■: CreatedAt/UpdatedAt ekle (DateTime.UtcNow default) – materyal atamas■n■n ne zaman yap■ld■■■n■ track et (rapor/denetim için faydal■).
Soft Delete: SilindiMi bool ekle (default false) – materyali silmek yerine gizle, ili■kileri koru (veri bütünlü■ü için).
Index'ler: Migration'da unique composite index (EgitimTuruId, DokumanId) ekle – duplicate atamay■ önle, sorgu h■z■n■ art■r (e.g., bir materyalin ayn■ türe birden fazla atanmas■n■ engelle).
Geni■letme: E■er materyallerin s■ras■ önemliyse int S■raNo ekle (default 0) – UI'de s■ralama için (opsiyonel, overkill de■il).

S■navOTURUMLARI

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("SinavOturumlari", Schema = "dbo")]
    public class SinavOturumlari
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int OturumId { get; set; }

public int? RefEgitimId { get; set; }

[ForeignKey(nameof(RefEgitimId))]
        public virtual Egitimler Egitim { get; set; }

public int SinavId { get; set; }

[ForeignKey(nameof(SinavId))]
        public virtual Sinavlar Sinav { get; set; }

public int? KullaniciId { get; set; } // Nullable yap■ld■

[ForeignKey(nameof(KullaniciId))]
        public virtual Kullanicilar Kullanici { get; set; }

[MaxLength(500)]
        public string OturumToken { get; set; }

public SinavOturumDurumu Durum { get; set; }

public DateTime? BaslamaTarihi { get; set; }

public DateTime? BitisTarihi { get; set; }

public int? Puan { get; set; }

public bool TcDogrulandiMi { get; set; } = false;

public int? GrupId { get; set; }

[Column(TypeName = "bit")]
        public bool DavetGonderildiMi { get; set; } = false;

[ForeignKey(nameof(GrupId))]
        public virtual EgitimGruplari Grup { get; set; }

public int? RefEtkinlikId { get; set; }

[ForeignKey(nameof(RefEtkinlikId))]
        public virtual Etkinlikler Etkinlik { get; set; }

public virtual ICollection<KullaniciCevaplari> KullaniciCevaplari { get; set; } = new List<Kullanic
```

```
    }
public enum SinavOturumDurumu
    {
        [Display(Name = "NotStarted")]
        Baslamadi,
        [Display(Name = "Ongoing")]
        DevamEdiyor,
        [Display(Name = "Completed")]
        Tamamlandi
    }
}
```

Her personelin girdi■i s■nav oturumlar■n■ temsil eder her s■nava giren ve bitiren ki■i için ayr■ kay■t aç■l■r. Normalizasyona tamamen uygun SinavOuturumDurumu enumu ise 3NF normalizasyona uygun

EG■TMENLER

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("Egitmenler", Schema = "dbo")]
    public class Egitmenler
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int EgitmenId { get; set; }

public int KullaniciId { get; set; }

public int FirmaId { get; set; }

public bool SilindiMi { get; set; } = false;

[ForeignKey("KullaniciId")]
        public virtual Kullanicilar Kullanici { get; set; }

[ForeignKey("FirmaId")]
        public virtual Firmalar Firma { get; set; }
    }
}
```

De■erlendirme:

Egitmenler tablosu, kullan■c■ (Kullanicilar) ile firma (Firmalar) aras■nda ba■lant■ (ili■ki) görevi görüyor.

Yani asl■nda bu tablo bir ili■ki (junction) tablosu gibi davran■yor.

SilindiMi alan■ hariç, fazladan bilgi bar■nd■rm■yor.
Bu durumda tablo 3NF (Third Normal Form) kurallar■na uygundur.

Olas■ ■yile■tirmeler:

E■er bir e■itmen sadece bir firmaya ba■l■ olabiliyorsa, bu yap■ do■ru.

Ancak ayn■ e■itmen birden fazla firmada görev alabiliyorsa, Egitmenler tablosu zaten olmas■ gereken ili■ki tablosu konumundad■r.

SilindiMi yerine soft delete için genellikle ortak bir “BaseEntity” s■n■f■nda IsDeleted, CreatedDate, UpdatedDate gibi alanlar tan■mlanabilir.

Sonuç:■yi yap■land■r■lm■■ gereksiz alan yok iyile■tirme için alanlar eklenebilir.

■uan Kullan■lm■yor.

LanguageResoure

```
using System;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    public class LanguageResource
    {
        [Key]
        public long Id { get; set; }

[Required]
        [StringLength(200)]
        public string ResourceKey { get; set; }

[Required]
        [StringLength(10)]
        public string LanguageCode { get; set; }

[Required]
        [StringLength(1000)]
        public string Value { get; set; }

public DateTime LastUpdated { get; set; } = DateTime.UtcNow;
    }
}
```

Localization için ilgili kay■tlar■n tutuldu■u tablo, güncellemeye gerek yok.

MSLEKKODLARI

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{

public class MeslekKodlari
    {
        [Key]
        public int Id { get; set; }

[Required]
        [MaxLength(20)]
        public string IscoKodu { get; set; }  // ISCO-08 Kodu

[Required]
        [MaxLength(255)]
        public string Ad { get; set; }  // Meslek Ad■
    }
}
```

Veri seti tablosu güncellemeye gerek yok.

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
```

```
namespace ilterisg.Models
{
    [Table("Mevzuat", Schema = "dbo")]
    public class Mevzuat
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Id { get; set; }

[Required]
        [MaxLength(50)]
        public string MevzuatKodu { get; set; }

[Required]
        [MaxLength(255)]
        public string MevzuatAdi { get; set; }

public string Aciklama { get; set; } // LONGTEXT (MySQL) / nvarchar(max) (SQL Server)

public DateTime? YayinTarihi { get; set; }

public DateTime CreatedAt { get; set; } = DateTime.Now;

[InverseProperty("Mevzuat")]
        public virtual ICollection<RiskDegerlendirme> RiskDegerlendirmeler { get; set; } = new List
    }
}
```

Veri seti tablosu tsv dosyas■ set ediiyor güncellemeye gerek yok.

MUAYENE

```
using Microsoft.AspNetCore.Mvc.ModelBinding;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    public class Muayene
    {
        [NotMapped]
        public string HastaAdSoyad
           => Kullanici?.AdSoyad ?? string.Empty;
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int? MuayeneId { get; set; }

[Required]
        [Display(Name = "Personel")]
        public int? KullaniciId { get; set; }

[ForeignKey(nameof(KullaniciId))]
        [BindNever]
        public Kullanicilar? Kullanici { get; set; }

public DateTime? Tarih { get; set; } // DateTime'dan DateTime? olarak güncellendi

[Required, StringLength(500)]
        public string Sikayet { get; set; } = "";

[Required, StringLength(500)]
        public string Bulgular { get; set; } = "";

public string OnTani { get; set; } = "";

[Required, StringLength(250)]
        public string TetkikOnerileri { get; set; } = "";

[Required, StringLength(250)]
        public string TedaviOnerisi { get; set; } = "";
```

```
// Art■k DB'de saklanacak:
        [StringLength(1000)]
        public string Hikaye { get; set; } = "";

[StringLength(1000)]
        public string Sonuclar { get; set; } = "";

//[StringLength(1000)]
        //public string Recete { get; set; } = "";

[StringLength(1000)]
        public string Sonuc { get; set; } = "";

// Olu■turan kullan■c■y■ action içinde ata, formdan gelmesin:
        [BindNever]
        public string? OlusturanUserId { get; set; }

[BindNever]
        public ApplicationUser? OlusturanUser { get; set; }

// → Navigation property: bir muayeneye birden fazla te■his
        public List<MuayeneTeshis> MuayeneTeshisler { get; set; } = new();
        public List<MuayeneRecete> MuayeneReceteler { get; set; } = new();
        [NotMapped]
        public string? ManuelTc { get; set; } = KALKTI
    }
}
```

ManuelTC alan■ normalizasyona ayk■r■ Kullan■c■ tc kimlik nosu dahil her bilgi Kullan■c■Id ile kullan■c■lar tablosundan eri■ilebiliyor.

```
public class Ilac
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int IlacId { get; set; }

[Required]
        public string Ad { get; set; } = null!;

[Required]
        [StringLength(50)]
        public string Barkod { get; set; } = null!;

[Required]
        [StringLength(20)]
        public string ATCKodu { get; set; } = null!;

[Required]
        [StringLength(100)]
        public string ATCAdi { get; set; } = null!;

[Required]
        public KullanimSekliEnum KullanimSekli { get; set; } = KullanimSekliEnum.Belirtilmemis;
    }
```

Veri seti tablosu revizeye gerek yok.

MUAYENERECETE

```
public class MuayeneRecete
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        [BindNever]
        public int Id { get; set; }

// E-reçete için ana reçete kimli■i (DVO: protokolNo)
```

```
        [Required]
        [StringLength(50)]
        public string ReceteId { get; set; } = "";

// Muayene ile ili■ki
        public int? MuayeneId { get; set; }
        [ForeignKey(nameof(MuayeneId))]
        [ValidateNever]
        public Muayene? Muayene { get; set; }

// ■laç ile ili■ki
        [Required]
        public int IlacId { get; set; }
        [ForeignKey(nameof(IlacId))]
        [ValidateNever]
        public Ilac Ilac { get; set; } = null!;

// Hasta (Yaz■lan ki■i) - Opsiyonel (Medula için)
        public int? KullaniciId { get; set; } // [Required] kald■r■ld■, nullable yap■ld■
        [ForeignKey(nameof(KullaniciId))]
        [ValidateNever]
        public Kullanicilar? Kullanici { get; set; } // Nullable

// Yazan doktor (ApplicationUser ID'si için - her zaman bilinsin)
        [StringLength(450)]
        public string OlusturanUserId { get; set; } = string.Empty; // Doktor'un appUser.Id'si

// Genel reçete aç■klamas■ (DVO: ereceteAciklamaListesi)
        [StringLength(500)]
        public string? Aciklama { get; set; } = "";

// ■laç özel aç■klamas■ (DVO: ereceteIlacAciklamaListesi)
        [StringLength(500)]
        public string? IlacAciklama { get; set; } = "";

// Doz 1 x Doz 2 (DVO: kullanimDoz1, kullanimDoz2)
        [Required, Range(0.01, 9999)]
        [Column(TypeName = "decimal(8,2)")]
        public decimal Doz1 { get; set; }

[Required, Range(0.01, 9999)]
        [Column(TypeName = "decimal(8,2)")]
        public decimal Doz2 { get; set; }

// Kullan■m periyodu (DVO: kullanimPeriyot, kullanimPeriyotBirimi)
        [Required, Range(1, 3650)]
        public int KullanimPeriyot { get; set; } = 1;

[Required]
        public KullanimPeriyotBirimiEnum KullanimPeriyotBirimi { get; set; } = KullanimPeriyotBirim

// Adet (DVO: adet)
        [Required, Range(1, 999)] // Min 1 olarak güncellendi (0 mant■ks■z)
        public int Adet { get; set; } = 1;

// Kullan■m ■ekli (DVO: kullanimSekli)
        [Required]
        public KullanimSekliEnum KullanimSekli { get; set; } = KullanimSekliEnum.Belirtilmemis;

// E-reçete için ek alanlar
        [StringLength(20)]
        public string? Barkod { get; set; } // DVO: barkod (Ilac.Barkod ile yedekli)

// DVO'dan gelen reçete seviyesindeki ek alanlar
        [Required]
        public ReceteTuruEnum ReceteTuru { get; set; }

[Required]
        public DateTime ReceteTarihi { get; set; }

[StringLength(20)]
        public string? EreceteNo { get; set; }
```

```
[StringLength(20)]
        public string? TakipNo { get; set; }

[StringLength(10)]
        public string? SeriNo { get; set; }

public int? TesisKodu { get; set; }

public int? DoktorBransKodu { get; set; }

public long? DoktorSertifikaKodu { get; set; }

[StringLength(50)]
        public string? DoktorAdi { get; set; }

[StringLength(50)]
        public string? DoktorSoyadi { get; set; }

public long? DoktorTcKimlikNo { get; set; }

[Required]
        public long TcKimlikNo { get; set; } // Hasta TC'si (Medula'dan gelir, zorunlu)

// Yeni eklenen alan: Medula'dan silinip silinmedi■ini belirten bool
        public bool MedulaSilindiMi { get; set; } = false;
        public DateTime? MedulaSilmeTarihi { get; set; }

// Computed properties
        [NotMapped]
        public string DozOzet => $"{Doz1} x {Doz2} / {KullanimPeriyot} {KullanimPeriyotBirimi}";

[NotMapped]
        public string TamAciklama => string.IsNullOrEmpty(IlacAciklama) ? (Aciklama ?? "") : $"{Aci
    }
```

Sonuç: ■yi yap■land■r■lm■■ güncelleme gerekmiyor.

MuayeneRecete tablosu (ve ili■kili modeller) genel olarak çok iyi yap■land■r■lm■■—özellikle Medula entegrasyonu (e-reçete DVO standartlar■) için dü■ünülmü■, pratik bir tasar■m. Nullable'lar, foreign key'ler ve computed property'ler (DozOzet, TamAciklama) gibi detaylar, hem veritaban■ verimlili■ini hem de kod okunabilirli■ini art■r■yor. ■ki fark (KullaniciId'nin opsiyonel olmas■ ve TC'nin zorunlu fallback'i) d■■■nda, ak■■ tutarl■: Hasta detaylar■ opsiyonelken TC her zaman anchor noktas■, bu da ghost hasta senaryolar■n■ destekliyor.

Güçlü Yönler:

```
Standart Uyumu: E-reçete alanlar■ (EreceteNo, TakipNo, SeriNo, TesisKodu vb.) do■rudan DVO'dan uya
Esneklik: MuayeneId nullable (muayenesiz reçete için), KullaniciId opsiyonel (sistem d■■■ hasta iç
Navigation Property'ler: Ilac, Muayene, Kullanici gibi FK'lar EF Core ile sorunsuz join'lere izin v
Ek Alanlar: MedulaSilindiMi/SilmeTarihi gibi flag'ler, silme takibini entegre etmi■—pratik, audit i
Ilac Modeli: Basit ve yeterli; Barkod/ATC entegrasyonu için haz■r.
Kullanicilar & Kullanici_Firma: Standart user-firma ili■kisi, composite key ile many-to-many'yi ter
```

Gereksiz Alan Var m■?

Hay■r, aç■kça gereksiz alan yok—hepsi i■levsel ve Medula/reçete ak■■■na hizmet ediyor. Ama ufak optimizasyon önerileri (zorunlu de■il, mevcut haliyle bile mükemmel):

```
Doktor Alanlar■ (DoktorAdi, DoktorSoyadi, DoktorTcKimlikNo, BransKodu, SertifikaKodu): Bunlar denor
Barkod: Ilac.Barkod ile yedekli, ama fallback için mant■kl■—gereksiz de■il, silme.
Aciklama & IlacAciklama: StringLength 500, yeterli; birle■tirilebilir (TamAciklama gibi), ama ayr■
Genel: Column(TypeName) ve BindNever'lar gibi attribute'ler temiz—veri tipi tutarl■ (decimal doz ic
```

■ K■sa Cevap:

Hay■r, tamamen TC kimlik numaras■yla yönetmek do■ru ve sürdürülebilir de■ildir.
TC kimlik alan■ sadece kimlik do■rulama veya e■le■tirme amac■yla kullan■labilir,
ama sistem içi ili■ki yönetimi (firma-personel, yetkilendirme, muayene geçmi■i, reçete kay■tlar■) için
KullaniciId kullan■lmal■d■r.

■■ Detayl■ Aç■klama
■ 1. Kullanici_Firma yap■s■n■n amac■

Bu tablo, firma-personel ili■kisini tan■ml■yor:

```
public class Kullanici_Firma
{
    public int RefFirmaId { get; set; }
    public int RefKullaniciId { get; set; }
}
```

Bu model, sistemin içinde “bu kullan■c■ ■u firmada çal■■■yor” bilgisini ta■■r.
Yani ili■ki yönetimi KullaniciId üzerinden yap■l■r — çünkü:

Firma tablosunun anahtar■ FirmaId

Kullan■c■ tablosunun anahtar■ KullaniciId

Dolay■s■yla bu ili■ki, veritaban■ düzeyinde yabanc■ anahtar (FK) ba■■ ile tan■ml■d■r.

■■ 2. TC kimlik numaras■ üzerinden yönetim neden hatal■ olur?

E■er sadece TC ile kontrol etmeye çal■■■rsan, birkaç ciddi problem ortaya ç■kar:

■ a) Veri bütünlü■ü bozulur

TC bir metinsel de■erdir, ili■kisel bir FK de■ildir.
Yani:

Kullanici_Firma tablosunda TcKimlikNo ile ba■ kuramazs■n (FK atayamazs■n).

Bir ki■i ayn■ TC ile yanl■■l■kla iki kez eklenirse (duplicate kay■t),
sistem bunlar■ ay■rt edemez.
→ Reçete, muayene veya personel atamalar■ kar■■■r.

■ b) Performans dü■er

TC üzerinden sorgular (join, where) daima string kar■■la■t■rmas■ gerektirir.
Büyük veri setlerinde (JOIN on NVARCHAR) hem indeks kullan■lamaz,
hem sorgular yava■lar.

■ c) Yabanc■ uyruk / geçici kimlik senaryosu

Sisteme yabanc■ uyruklu çal■■an, geçici kimlik numaras■ olan ya da TC’si hatal■ girilmi■ ki■iler geldi■inde
KullaniciId tabanl■ sistem bunlar■ yönetebilir ama TC tabanl■ sistem yönetemez.

■ d) Uygulama mant■■■ tutars■zla■■r

Birçok servis (PopulatePersonelList, GetDoktorFirmaIdsAsync, MuayeneKaydet, ReceteOlustur) zaten KullaniciId ile çal■■■yor.
Bu kodlar■ TC'ye çevirmek demek, tüm ili■ki zincirini (FK navigation property) bozmak olur.

■ 3. Do■ru Yakla■■m:

■li■kisel olarak her zaman KullaniciId kullan■lmal■,
ama kimlik do■rulama veya harici sistemlerle entegrasyon (örne■in Medula) a■amas■nda TcKimlikNo kullan■labilir.

Yani:

Senaryo Kullan■lacak Alan
Firma-personel e■le■mesi Kullanici_Firma.RefKullaniciId
Muayene, Reçete, E■itim ili■kileri KullaniciId
Medula veya e-Reçete gönderimi TcKimlikNo
TC'siz d■■ hastalar KullaniciId = null, ManuelTc geçici de■er
■ 4. En ■deal Mimari (Önerilen)

Firma yönetimi, yetkilendirme, personel filtreleme → KullaniciId

E-reçete, d■■ hasta, Medula i■lemleri → TcKimlikNo

■li■kisel bütünlük → Kullanici_Firma tablosu üzerinden

■ Sonuç
Kriter De■erlendirme
Tamamen TC ile yönetim mümkün mü? ■ Hay■r — ili■kisel bütünlük ve performans sorunlar■ olur.
Kullanici_Firma ne i■e yar■yor? Firma-personel ba■lant■s■n■ KullaniciId ile tan■mlar.
TC kimlik nerede kullan■lmal■? Sadece Medula/e-Reçete gibi d■■ sistem i■lemlerinde veya kay■tl■ olmayan hastalarda.
Do■ru yönetim modeli Firma personeli KullaniciId ile, kay■t d■■■ hasta ManuelTc ile yönetilir.

■stersen bu yap■y■ ■ematik olarak (ili■ki diyagram■ ■eklinde) çizip gösterebilirim —
örne■in Kullanici, Firma, Kullanici_Firma, Muayene, MuayeneRecete ili■kilerini net ■ekilde.
■ster misin o diyagram■ da olu■turay■m?

EK2FORM

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using ilterisg.Models.Enums;

namespace ilterisg.Models
{
    [Table("Ek2Formlar", Schema = "dbo")]
```

```
    public class Ek2Form
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Ek2FormId { get; set; }

[Required(ErrorMessage = "Kullan■c■ ID zorunludur.")]
        public int KullaniciId { get; set; }

[Required(ErrorMessage = "Firma ID zorunludur.")]
        public int RefFirmaId { get; set; }

public int? RefOSGBId { get; set; }

[Required(ErrorMessage = "Olu■turan kullan■c■ ID zorunludur.")]
        public string OlusturanUserId { get; set; }

[Required(ErrorMessage = "Olu■turma tarihi zorunludur.")]
        public DateTime OlusturmaTarihi { get; set; }

[Required(ErrorMessage = "Form durumu zorunludur.")]
        public Ek2FormDurum Durum { get; set; } = Ek2FormDurum.AcildiBekliyor; // Varsay■lan: Aç■l

// ■■yeri Bilgileri
        [Required(ErrorMessage = "Firma ad■ zorunludur.")]
        public string FirmaAdi { get; set; }
        public string? SgkSicilNo { get; set; }
        public string? FirmaAdresi { get; set; }
        public string? TelefonFaks { get; set; }
        public string? EPosta { get; set; }
        public string? CalisanBeyani { get; set; }

// Çal■■an Bilgileri
        [Required(ErrorMessage = "Ad soyad zorunludur.")]
        public string AdSoyad { get; set; }
        [Required(ErrorMessage = "T.C. Kimlik No zorunludur.")]
        public string TcKimlikNo { get; set; }

public string? DogumYeri { get; set; }
        [Required(ErrorMessage = "Do■um tarihi zorunludur.")]
        public DateTime DogumTarihi { get; set; }
        [Required(ErrorMessage = "Cinsiyet zorunludur.")]
        public string Cinsiyet { get; set; }
        public string? EgitimDurumu { get; set; }
        public string? MedeniDurum { get; set; }
        public int? CocukSayisi { get; set; }
        public string? EvAdresi { get; set; }
        public string? TelefonNo { get; set; }
        public string? Meslek { get; set; }
        public string? YaptigiIs { get; set; }
        public string? CalistigiBolum { get; set; }
        public virtual ICollection<Ek2OncekiIsYeri> OncekiIsYerleri { get; set; } = new List<Ek2Onc
        public string? Ozgecmis { get; set; }
        public string? KanGrubu { get; set; }
        public string? KonjenitalKronikHastalik { get; set; }
        public bool TetanozBagisiklama { get; set; } = false;
        public bool HepatitBagisiklama { get; set; } = false;
        public string? DigerBagisiklama { get; set; }
        public string? SoygecmisAnne { get; set; }
        public string? SoygecmisBaba { get; set; }
        public string? SoygecmisKardes { get; set; }
        public string? SoygecmisCocuk { get; set; }

// T■bbi Anamnez
        public bool BalgamliOksuruk { get; set; } = false;
        public bool NefesDarligi { get; set; } = false;
        public bool GogusAgrisi { get; set; } = false;
        public bool Carpinti { get; set; } = false;
        public bool SirtAgrisi { get; set; } = false;
```

```
        public bool Ishalkabizlik { get; set; } = false;
        public bool EklemlerdeAgri { get; set; } = false;
        public bool KalpHastaligi { get; set; } = false;
        public bool SekerHastaligi { get; set; } = false;
        public bool BobrekRahatsizligi { get; set; } = false;
        public bool Sarilik { get; set; } = false;
        public bool MideUlseri { get; set; } = false;
        public bool IsitmeKaybi { get; set; } = false;
        public bool GormeBozuklugu { get; set; } = false;
        public bool SinirSistemiHastaligi { get; set; } = false;
        public bool DeriHastaligi { get; set; } = false;
        public bool BesinZehirlenmesi { get; set; } = false;
        public string? HastanedeYatisTani { get; set; }
        public string? AmeliyatNeden { get; set; }
        public string? IsKazaDetay { get; set; }
        public string? MeslekHastaligiSonuc { get; set; }
        public string? MaluliyetDetay { get; set; }
        public string? TedaviDetay { get; set; }
        public bool SigaraIciyorMu { get; set; } = false;
        public string? SigaraBirakmaZamani { get; set; }
        public string? SigaraIcmeSuresi { get; set; }
        public int? SigaraGunlukAdet { get; set; }
        public bool AlkolAliyorMu { get; set; } = false;
        public string? AlkolBirakmaZamani { get; set; }
        public string? AlkolIcmeSuresi { get; set; }
        public string? AlkolSiklik { get; set; }

// Fizik Muayene
        public string? GozMuayene { get; set; }
        public string? KbbMuayene { get; set; }
        public string? DeriMuayene { get; set; }
        public string? KardiyovaskulerMuayene { get; set; }
        public string? SolunumMuayene { get; set; }
        public string? SindirimMuayene { get; set; }
        public string? UrogenitalMuayene { get; set; }
        public string? KasIskeletMuayene { get; set; }
        public string? NorolojikMuayene { get; set; }
        public string? PsikiyatrikMuayene { get; set; }
        public string? Tansiyon { get; set; }
        public int? Nabiz { get; set; }
        public double? Boy { get; set; }
        public double? Kilo { get; set; }
        public double? VucutKitleIndeksi { get; set; }

// Laboratuvar Bulgular■
        public string? KanAnalizi { get; set; }
        public string? IdrarAnalizi { get; set; }
        public string? RadyolojikAnaliz { get; set; }
        public string? Odyometri { get; set; }
        public string? SFT { get; set; }
        public string? PsikolojikTestler { get; set; }

// Kanaat ve Sonuç
        public bool CokTehlikeliIslereUygun { get; set; } = false;
        public bool YuksekteCalismaUygun { get; set; } = false;
        public bool GeceVardiyaliCalismaUygun { get; set; } = false;
        public string? Kanaat { get; set; }
        [Required(ErrorMessage = "■■yeri hekimi ad soyad zorunludur.")]
        public string IsyeriHekimiAdSoyad { get; set; }
        public string? DiplomaBilgileri { get; set; }

// Navigation Properties (Nullable)
        [ForeignKey(nameof(KullaniciId))]
        public virtual Kullanicilar? Kullanici { get; set; }

[ForeignKey(nameof(RefFirmaId))]
        public virtual Firmalar? Firma { get; set; }
```

```
[ForeignKey(nameof(RefOSGBId))]
        public virtual OSGBler? OSGB { get; set; }

[ForeignKey(nameof(OlusturanUserId))]
        public virtual ApplicationUser? OlusturanUser { get; set; }
    }
}
```

Normalizasyona uygun form d■■■ alanlar sadece ba■lant■lar■ belli edebilmek için do■ru bir ■ekilde tan■mlanm■■.

EK2Oncek■IsYer■

EK2Formda b■rden fazla eski i■yeri olabilece■i için Junction Table olarak yap■land■r■ld■ o alan

```
using System;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("Ek2OncekiIsYerleri", Schema = "dbo")]
    public class Ek2OncekiIsYeri
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Id { get; set; }

public int? Ek2FormId { get; set; }

public string? IsKolu { get; set; }
        public string? YaptigiIs { get; set; }
        public DateTime? GirisTarihi { get; set; }
        public DateTime? CikisTarihi { get; set; }

[ForeignKey(nameof(Ek2FormId))]
        public virtual Ek2Form? Ek2Form { get; set; }
    }
}
```

Normalizasyona uygun sadece Ek2FormID nullable int yerine int olab■lilir her Ek2 Öncek ■ yeri alan■ bir forma ba■l■ olaca■■ için.

EMA■L SETT■NGS - OPS■YONEL - YERLE■■K EMA■L S■STEM■ ■Ç■N OLU■TURULDU DEVAM ED■LMEYECEKSE S■L■NEB■L■R

```
using System.ComponentModel.DataAnnotations;

namespace ilterisg.Models
{
    public class EmailSettings
    {
        [Key]
        public int Id { get; set; }

[Required]
        public string UserId { get; set; }

[EmailAddress]
        public string EmailAddress { get; set; }

public string Password { get; set; } // OAuth kullan■lmayan durumlarda
```

```
public string ImapServer { get; set; }
        public string ImapPort { get; set; }
        public bool ImapEnableSsl { get; set; }

public string PopServer { get; set; }
        public string PopPort { get; set; }
        public bool PopEnableSsl { get; set; }

public string SmtpServer { get; set; }
        public string SmtpPort { get; set; }
        public bool SmtpEnableSsl { get; set; }

public string Protocol { get; set; }

// OAuth için token'lar
        public string AccessToken { get; set; }
        public string RefreshToken { get; set; }
        public string TokenProvider { get; set; } // "Gmail", "Outlook", "Yahoo" vb.

public ApplicationUser User { get; set; }
    }
}
```

ETK■NL■K-ETKNL■K KULLANICI

```
using ilterisg.Models.Enums;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("Etkinlikler", Schema = "dbo")]
    public class Etkinlikler
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Id { get; set; }

[Required]
        [MaxLength(200)]
        public string Ad { get; set; } // Etkinlik ad■

[Required]
        public DateTime BaslangicTarihi { get; set; } // Etkinlik ba■lang■ç tarihi

public DateTime? BitisTarihi { get; set; } // Etkinlik biti■ tarihi (opsiyonel)

[MaxLength(1000)]
        public string Aciklama { get; set; } // Etkinlik aç■klamas■ (opsiyonel)

[Required]
        public string KullaniciId { get; set; } // Etkinli■i olu■turan kullan■c■ (foreign key)

public EtkinlikTuru EtkinlikTuru { get; set; } // Etkinlik türü (Etkinlik, Toplanti, Ziyaret, Diger

public int? RefFirmaId { get; set; } // Firma ba■lant■s■ (Toplanti, Ziyaret, Diger, Sinav için)

public bool SilindiMi { get; set; } = false; // Soft delete için
        public bool IsClosed { get; set; } = false; // Etkinlik kapatma durumu
        public bool FirmaOnay { get; set; } = false; // Firma onay■
        public bool PersonelOnay { get; set; } = false; // Personel onay■

[Required]
        public DateTime OlusturulmaTarihi { get; set; } = DateTime.Now; // Yeni alan

// Navigation property: Kullan■c■
        [ForeignKey(nameof(KullaniciId))]
        public virtual ApplicationUser Kullanici { get; set; }

// Navigation property: Firma
        [ForeignKey(nameof(RefFirmaId))]
        public virtual Firmalar Firma { get; set; }
```

```
// Navigation property: Etkinli■e atanan personeller
        public virtual ICollection<Etkinlik_Kullanici> EtkinlikKullanicilar { get; set; } = new Lis

// Navigation property: Etkinli■e ba■l■ s■nav oturumlar■ (Yeni)
        public virtual ICollection<SinavOturumlari> SinavOturumlari { get; set; } = new List<SinavO
    }
}
```

```
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("Etkinlik_Kullanici", Schema = "dbo")]
    public class Etkinlik_Kullanici
    {
        public int Id { get; set; } // Yeni birincil anahtar
        public int RefEtkinlikId { get; set; }
        public int RefKullaniciId { get; set; }

[ForeignKey(nameof(RefEtkinlikId))]
        public virtual Etkinlikler Etkinlik { get; set; }

[ForeignKey(nameof(RefKullaniciId))]
        public virtual Kullanicilar Kullanici { get; set; }
    }
}
```

Gereksiz alan yok güzel yap■aland■r■lm■■

Ufak Öneriler (Zorunlu De■il, Opsiyonel ■yile■tirmeler):

```
Etkinlikler'e Index Ekle: Migration'da [Index(nameof(BaslangicTarihi))] veya composite index (Etkir
Junction'a Ek Potansiyel: E■er kat■l■m rolü (kat■l■mc■/koordinatör) veya kat■l■m tarihi laz■msa,
Validation: Etkinlikler'e [Required]'leri geni■let (örne■in, BaslangicTarihi için), ama mevcut hal
Geni■leme: SinavOturumlari navigation'■ güzel—s■nav etkinlikleri için one-to-many haz■r.
```

FAAL■YETALANI

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("FaaliyetAlanlari", Schema = "dbo")]
    public class FaaliyetAlani
    {
        [Key]
        public int Id { get; set; }

[Required, MaxLength(200)]
        public string MeslekGrubu { get; set; }

[Required, MaxLength(50)]
        public string NaceKodu { get; set; }

[Required, MaxLength(500)]
        public string NaceFaaliyetAdi { get; set; }
    }
}
```

Veri seti tablosu tsv dosyas■ set ediliyor güncellemeye gerek yok

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using ilterisg.Models.Enums;

namespace ilterisg.Models
{
```

```
    [Table("FAQ", Schema = "dbo")]
    public class Faq
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int FaqId { get; set; }

[Required]
        [StringLength(200)]
        public string Baslik { get; set; }

[Required]
        public string Icerik { get; set; }

[Required]
        public KonuTuru KonuTuru { get; set; }

[Required]
        public DateTime OlusturmaTarihi { get; set; } = DateTime.Now;

public DateTime? GuncellemeTarihi { get; set; }

[Required]
        [MaxLength(128)]
        public string OlusturanId { get; set; } // Admin'in AspNetUsers.Id'si
    }

[Table("FaqFeedback", Schema = "dbo")]
    public class FaqFeedback
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int FeedbackId { get; set; }

[Required]
        public int FaqId { get; set; }

[Required]
        [MaxLength(128)]
        public string UserId { get; set; } // Oy veren kullan■c■n■n AspNetUsers.Id'si

[Required]
        public bool YararliOlduMu { get; set; }

public DateTime Tarih { get; set; } = DateTime.Now;

[ForeignKey("FaqId")]
        public virtual Faq Faq { get; set; }
    }
}
```

Gereksiz alan yok normalizasyon için bölüm eklemeye veya ç■karmaya gerek yok.

FATURA B■LG■LER■

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("FaturaBilgileri", Schema = "dbo")]
    public class FaturaBilgileri
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Id { get; set; }

[Required]
        public int FirmaId { get; set; } // Firma ID’si (Firmalar tablosu ile ili■kilendirme)
```

```
[Required]
        [MaxLength(500)]
        public string Adres { get; set; } // Fatura adresi

[Required]
        [MaxLength(100)]
        public string Sehir { get; set; } // ■ehir

[Required]
        [MaxLength(100)]
        public string Ulke { get; set; } // Ülke

[MaxLength(20)]
        public string? PostaKodu { get; set; } // Posta kodu (opsiyonel)

[MaxLength(255)]
        public string IletisimAdi { get; set; } // ■leti■im ad■ (örne■in, ad soyad)

[ForeignKey(nameof(FirmaId))]
        public virtual Firmalar? Firma { get; set; } // Nullable navigation property
    }
}
```

Iyz■co ihtiyac■ olan alanlar evcut ekstradan Kullan■c■lar içinde alan eklenebilir firma bilgisi tutuluyor ■uan yaln■zca.

FEATURED CONTENT

```
// Models/FeaturedContent.cs
namespace ilterisg.Models
{
    public class FeaturedContent
    {
        public int Id { get; set; }
        public string Section { get; set; } // "LatestPosts", "RecommendedPosts", "PopularPosts"
        public int BlogPostId { get; set; }
        public BlogPost BlogPost { get; set; }
        public int DisplayOrder { get; set; } // S■ralama için
    }
}
```

```
Blog postlar■ s■ralamas■ için Popüler Önerilen postlar gibi alanlar için tablo
Yap■ genel olarak çok iyi.
Sadece Section alan■ için enum kullan■m■ ve PublishedAt gibi bir alan eklemen, uzun vadede yönetim
```

F■RMA DEPARTMAN

Proje aç■l■rken olu■turuldu henüz yap■land■r■lmas■ yok

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("Firma_Departman", Schema = "dbo")]
    public class Firma_Departman
    {
        [Key, Column(Order = 0)]
        public int RefFirmaId { get; set; }

[Key, Column(Order = 1)]
        public int RefDepartmanId { get; set; }

// Firma tablosuna ait FK
        [ForeignKey(nameof(RefFirmaId))]
        public virtual Firmalar Firma { get; set; }
```

```
// Departmanlar tablosuna ait FK
        [ForeignKey(nameof(RefDepartmanId))]
        public virtual Departmanlar Departman { get; set; }
    }
}
```

GEREKS■Z ESKI EGITIM SISTEMINDEN KALMA BIR TABLO KALDIRILABILIR

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("Firma_Egitim", Schema = "dbo")]
    public class Firma_Egitim
    {
        [Key, Column(Order = 0)]
        public int RefFirmaId { get; set; } // Firmalar tablosuna foreign key

[Key, Column(Order = 1)]
        public int RefEgitimId { get; set; } // Egitimler tablosuna foreign key

// Navigation properties
        [ForeignKey(nameof(RefFirmaId))]
        public virtual Firmalar Firma { get; set; }

[ForeignKey(nameof(RefEgitimId))]
        public virtual Egitimler Egitim { get; set; }
    }
}
```

F■RMA SEKTOR

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("Firma_Sektor", Schema = "dbo")]
    public class Firma_Sektor
    {
        [Key, Column(Order = 0)]
        public int RefFirmaId { get; set; }

[Key, Column(Order = 1)]
        public int RefSektorId { get; set; }

// Navigation Property: Firmalar tablosuyla ili■ki
        [ForeignKey(nameof(RefFirmaId))]
        public virtual Firmalar Firma { get; set; }

// Navigation Property: Sektorler tablosuyla ili■ki
        [ForeignKey(nameof(RefSektorId))]
        public virtual Sektorler Sektor { get; set; }
    }
}
```

Proje ba■■nda aç■ld■ suan kullan■m■ yok

F■RMA DAVETLER

```
using System.ComponentModel.DataAnnotations;
```

```
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    public class FirmaDavetler
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int DavetId { get; set; }

[Required]
        [MaxLength(256)]
        public string Email { get; set; }

[Required]
        public int RefOSGBId { get; set; }

[MaxLength(450)]
        public string? RefUserId { get; set; }

[Required]
        public DateTime DavetTarihi { get; set; } = DateTime.Now;

public bool? OnayDurumu { get; set; }

[MaxLength(450)]
        public string? InvitationToken { get; set; }

[ForeignKey(nameof(RefOSGBId))]
        public virtual OSGBler OSGB { get; set; }

[ForeignKey(nameof(RefUserId))]
        public virtual ApplicationUser? User { get; set; }
    }
}

Normalizasyon■■ 3NF’ye uygun
FK yap■s■■■ Do■ru tan■mlanm■■
Navigation property■■ Eksiksiz
Gereksiz alan■■ Yok
Geni■letilebilirlik■■ ■yi düzeyde (enum / index eklenebilir)
Migration uyumu■■ Sorunsuz
```

Öneri

Email alan■ için do■rulama

[EmailAddress] attribute’u eklenirse form bazl■ validasyon kolayla■■r:

```
[Required, MaxLength(256), EmailAddress]
public string Email { get; set; }
```

F■RMAKATILIMTALEB■

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using ilterisg.Models.Enums;

namespace ilterisg.Models
{
    [Table("FirmaKatilimTalebi", Schema = "dbo")]
    public class FirmaKatilimTalebi
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int IstekId { get; set; }

[Required]
        public int RefFirmaId { get; set; }
```

```
[Required]
        [MaxLength(450)]
        public string RefUserId { get; set; }

[Required]
        public DateTime IstekTarihi { get; set; } = DateTime.Now;

public ApprovalStatus? OnayDurumu { get; set; }

[MaxLength(500)]
        public string? BasvuruNotu { get; set; }

[ForeignKey(nameof(RefFirmaId))]
        public virtual Firmalar Firma { get; set; }

[ForeignKey(nameof(RefUserId))]
        public virtual ApplicationUser Kullanici { get; set; }
    }
}
```

Bu tablo tam anlam■yla do■ru normalize edilmi■,

hiç gereksiz alan bar■nd■rm■yor,

FK ili■kileri do■ru kurgulanm■■,

ve Entity Framework taraf■nda sorunsuz çal■■■r.

KULLANICILAR-F■RMALAR-OSGBLER

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("Kullanicilar", Schema = "dbo")]
    public class Kullanicilar
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int KullaniciId { get; set; }

[Required]
        [MaxLength(100)]
        public string AdSoyad { get; set; }

[Required]
        [MaxLength(11)]
        public string TcKimlikNo { get; set; }

[Required]
        public DateTime DogumTarihi { get; set; }

[Required]
        [MaxLength(50)]
        public string? Cinsiyet { get; set; }

[MaxLength(250)]
        public string? Adres { get; set; }

public DateTime? KayitTarihi { get; set; }

[MaxLength(450)]
        public string RefUserId { get; set; }

[ForeignKey(nameof(RefUserId))]
        public virtual ApplicationUser ApplicationUser { get; set; }

public virtual ICollection<Kullanici_Egitim> KullaniciEgitimler { get; set; } = new List<Kullanici_

public virtual ICollection<Kullanici_Firma> KullaniciFirmalar { get; set; } = new List<Kullanici_Fi
        public virtual ICollection<Ek2Form> Ek2Formlar { get; set; } = new List<Ek2Form>();
```

```
        public ICollection<Muayene> Muayeneler { get; set; } = new List<Muayene>();

/// <summary>
        /// Kullan■c■n■n çevrimiçi olup olmad■■■n■ belirtir.
        /// </summary>
        public bool IsActive { get; set; }
        public DateTime? LastLoginTime { get; set; }
        /// Kullan■c■n■n ■ifre de■i■tirmesi gerekip gerekmedi■ini belirtir.
        /// </summary>
        public bool ForcePasswordChange { get; set; } // Yeni alan

// Yeni alan: Profil foto■raf■ için Dokumanlar tablosuna referans
        public int? ProfilFotografiDokumanId { get; set; }

[ForeignKey(nameof(ProfilFotografiDokumanId))]
        public virtual Dokumanlar? ProfilFotografi { get; set; }
        // Yeni alan: Meslek Kodu
        public int? MeslekKoduId { get; set; }
        [ForeignKey(nameof(MeslekKoduId))]
        public virtual MeslekKodlari? MeslekKodu { get; set; }
    }
}

using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("Firmalar", Schema = "dbo")]
    public class Firmalar
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int FirmaId { get; set; }

[Required]
        [MaxLength(150)]
        public string FirmaAdi { get; set; }

[Required]
        [MaxLength(100)]
        public string VergiDairesi { get; set; }

[Required]
        [MaxLength(20)]
        public string VergiNumarasi { get; set; }

[MaxLength(50)]
        public string? SskSicilNo { get; set; }

[MaxLength(250)]
        public string? Adres { get; set; }

public int? CalisanSayisi { get; set; }

public int? NaceKoduId { get; set; } // Yeni foreign key sütunu, NaceKodlari tablosunun Id sütununa

[ForeignKey(nameof(NaceKoduId))]
        public virtual NaceKodlari? NaceKodu { get; set; }

public int? FaaliyetAlaniId { get; set; } // Yeni foreign key
        [ForeignKey(nameof(FaaliyetAlaniId))]
        public virtual FaaliyetAlani? FaaliyetAlani { get; set; } // Yeni navigation property

public bool? IsActive { get; set; }
        public bool? IsApproved { get; set; }

[Column("Kay■tTarihi")]
        public DateTime? KayitTarihi { get; set; }

// TehlikeSinifi sütununu kald■rd■k, bu bilgi NaceTehlikeSiniflari tablosunda tutulacak
        public bool? IsOSGB { get; set; }
```

```
public string? ApplicationUserId { get; set; }

[ForeignKey(nameof(ApplicationUserId))]
        public virtual ApplicationUser? ApplicationUser { get; set; }

public virtual ICollection<Firma_Departman> Firma_Departman { get; set; }
            = new HashSet<Firma_Departman>();

public virtual ICollection<Firma_Sektor> Firma_Sektor { get; set; }
            = new HashSet<Firma_Sektor>();
        public virtual ICollection<OSGB_Firmalar> OSGB_Firmalar { get; set; }
            = new HashSet<OSGB_Firmalar>();

public virtual ICollection<Firma_Egitim> FirmaEgitimler { get; set; }
            = new HashSet<Firma_Egitim>();

public virtual ICollection<Ek2Form> Ek2Formlar { get; set; } = new HashSet<Ek2Form>();
    }
}

using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    public class OSGBler
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int OSGBId { get; set; }

[Required]
        [MaxLength(255)]
        public string OSGBAdi { get; set; }

[Required]
        public string Adres { get; set; }

[Required]
        [MaxLength(20)]
        public string Telefon { get; set; }

[Required]
        [Column("Kay■tTarihi")]
        public DateTime KayitTarihi { get; set; }

[Required]
        public int RefFirmaId { get; set; }

[Required]
        public string ApplicationUserId { get; set; }

[MaxLength(100)]
        public string? SubMerchantKey { get; set; } // Nullable SubMerchantKey

[MaxLength(34)] // IBAN uzunlu■u genelde 34 karakter
        public string? Iban { get; set; } // Nullable IBAN
        public string? SubMerchantExternalId { get; set; } // Yeni alan
        public string? HesapAdi { get; set; } // Yeni eklenen nullable HesapAdi alan■

// Navigation property'ler
        [ForeignKey(nameof(RefFirmaId))]
        public virtual Firmalar Firma { get; set; }

[ForeignKey(nameof(ApplicationUserId))]
        public virtual ApplicationUser ApplicationUser { get; set; }

public virtual ICollection<OSGB_Firmalar> OSGB_Firmalar { get; set; } = new List<OSGB_Firmalar>();
        public virtual ICollection<Ek2Form> Ek2Formlar { get; set; } = new List<Ek2Form>();


}

}
```

Çok güzel — bu üç tablo (Firmalar, OSGBler, Kullanicilar) sisteminin en temel “çekirdek” modelleri,
yani senin domain’inde i■veren, OSGB ve kullan■c■ varl■klar■n■ temsil ediyorlar.
Bu yüzden normalizasyonun burada do■ru olmas■, tüm sistemin veri bütünlü■ünü etkiliyor.
Kodlar■n■ inceledim, üçü de çok iyi normalize edilmi■.
Ancak profesyonel olarak inceleyip art■lar■n■ ve küçük geli■tirme f■rsatlar■n■ da a■a■■da özetledim
■

■ 1. Firmalar Tablosu
■ Genel De■erlendirme
Alan Durum Aç■klama
FirmaId ■ Otomatik PK.
FirmaAdi, VergiDairesi, VergiNumarasi ■ Temel kimlik bilgileri, do■ru tür ve k■s■tlar.
SskSicilNo, Adres ■ Opsiyonel, atomik de■erler.
CalisanSayisi ■ Say■sal de■er, do■ru tip.
NaceKoduId, FaaliyetAlaniId ■ D■■ anahtarlar, ili■kisel bütünlük sa■lanm■■.
IsActive, IsApproved, IsOSGB ■ Durum alanlar■, mant■kl■.
KayitTarihi ■ Tarih alan■, nullable ve uygun.
ApplicationUserId ■ Firma sahibini/hesab■n■ ba■lar, FK do■ru.
ICollection navigation’lar ■ 1:N ili■kiler do■ru modellenmi■.
■■ Normalizasyon Analizi

1NF: ■ Her alan atomik (örne■in adres, tek string).

2NF: ■ Bütün alanlar sadece FirmaId’ye ba■■ml■.

3NF: ■ Transitif ba■■ml■l■k yok (örne■in NACE kodu ba■ka tabloya ta■■nm■■ — çok do■ru).

■■ Küçük ■yile■tirme Fikirleri

```
IsActive, IsApproved gibi boolean çiftleri yerine
bir enum daha aç■klay■c■ olur:
```

```
public enum FirmaDurumu { Pasif, Beklemede, Aktif, Onayli }
```

```
[Column("Kay■tTarihi")] alan ad■n■ Türkçe yazman gerekmez; veritaban■nda kar■■■kl■k yapabilir.
■ngilizce (KayitTarihi) olarak b■rakmak genelde daha temiz olur.
```

■ Sonuç

3NF düzeyinde, tamamen do■ru yap■.
Her alan do■ru ba■■ml■l■■a sahip, hiçbir tekrar veya gereksizlik yok.

1NF: ■ Tüm alanlar atomik.

2NF: ■ Her alan KullaniciId’ye tam ba■■ml■.

3NF: ■ Transitif ba■■ml■l■k yok — örne■in “meslek kodu” ili■kisel tabloya ta■■nm■■, mükemmel.

■■ Küçük ■yile■tirme Fikirleri

```
Cinsiyet string olarak tutuluyor; bu ileride “Erkek”, “ERKEK”, “e” gibi da■■labilir.
Bunun yerine küçük bir enum kullanmak daha do■ru olur:
```

```
public enum Cinsiyet { Erkek, Kadin, Diger }
```

TcKimlikNo benzersiz olmal■:

[Index(nameof(TcKimlikNo), IsUnique = true)]

```
RefUserId [Required] olarak tan■mlanabilir – aksi halde orphan user (ApplicationUser’s■z) olu■abil
```

■ 4. ■li■kiler Aras■ Normalizasyon Kontrolü
■li■ki Tür Aç■klama
Firma ↔ OSGB 1:N Her firma birden fazla OSGB’ye sahip olabilir.
Firma ↔ Kullanici N:N (Kullanici_Firma üzerinden) Do■ru, normalizasyonu sa■l■yor.
OSGB ↔ Kullanici Dolayl■ (Firmalar üzerinden) ■li■ki zinciri do■ru.
Firma ↔ E■itim / Departman / Sektör N:N Do■ru ■ekilde ara tablolarla kurulmu■.

Tüm bu ba■lant■lar, EF Core’un ili■ki çözümleme algoritmas■na tamamen uygun.
Hiçbir “many-to-many” do■rudan tutulmuyor — aratablolarla ayr■lm■■.
Bu da tam normalize edilmi■ anlam■na gelir.

■ Sonuç: Üç Tablo Aras■ndaki Normalizasyon Özeti
Tablo Normalizasyon Düzeyi Gereksiz Alan Öneri / ■yile■tirme
Firmalar 3NF ■ Yok Enum ile durum yönetimi yap■labilir.
OSGBler 3NF ■ Yok Default tarih eklenebilir.
Kullanicilar 3NF ■ Yok Cinsiyet enum’a ta■■nmal■, TC kimlik unique olmal■.
■ Genel De■erlendirme

Bu üç tablo birlikte çal■■t■■■nda:

Veri tekrar■ hiç yok,

FK yap■lar■ do■ru ve mant■kl■,

Alan ba■■ml■l■klar■ do■ru anahtarlara ba■l■,

Sistem tam 3. normal form’da.

■ K■sacas■: Tam normalize, kurumsal seviyede güçlü bir veri modeli.
■ Sadece küçük “kalite iyile■tirmeleri” (enum’lar, unique index) önerilebilir.

```
■stersen bu üç tabloyu birlikte (Firmalar ↔ OSGBler ↔ Kullanicilar) içeren bir ER diyagram■ (ili■
ba■lant■lar■n nas■l kuruldu■unu net ■ekilde görebilirsin.
■ster misin o diyagram■ da olu■turay■m?
```

```
using Microsoft.AspNetCore.Mvc.ModelBinding;
using Microsoft.AspNetCore.Mvc.ModelBinding.Validation;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    public class MuayeneTeshis
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        [BindNever]
```

```
        public int Id { get; set; }

public int? MuayeneId { get; set; }
        [ForeignKey(nameof(MuayeneId))]
        [ValidateNever]
        public Muayene? Muayene { get; set; }

// SADECE ReceteId string → navigation YOK
        [StringLength(50)]
        public string? ReceteId { get; set; }

// KALDIR: public MuayeneRecete? Recete { get; set; }

[Required, StringLength(50)]
        public string Kod { get; set; } = "";

[Required, StringLength(1000)]
        public string Adi { get; set; } = "";

public long? TcKimlikNo { get; set; } // Hasta TC'si (Medula'dan gelir, zorunlu)
    }
}
```

Tamamen normalize gereksiz alan yok Muayene var ise MuayeneID ile ba■lan■yor her reçete ReceteID ile ba■lan■yor bu ileride ProtokolNo ile de■i■ebilir Reçetede birden fazla tan olaca■■ için bu senaryoyada uygun

TcKimlikNo alan■n■ ayr■ca tutmak, kullan■c■ olmayan ki■iler için te■hisleri ay■rt etmek — tamamen do■ru, profesyonel ve sa■l■k sistemleri için standart bir uygulamad■r. ■

Yani bu tasar■m■ kesinlikle korumal■s■n.

NACEKODLARI

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("NaceKodlari", Schema = "dbo")]
    public class NaceKodlari
    {
        [Key]
        public int Id { get; set; }

[Required]
        public string SektorKodu { get; set; } // LONGTEXT (MySQL) / nvarchar(max) (SQL Server)

[Required]
        public string SektorTanim { get; set; } // LONGTEXT (MySQL) / nvarchar(max) (SQL Server)

[Required]
        public string MeslekKodu { get; set; } // LONGTEXT (MySQL) / nvarchar(max) (SQL Server)

[Required]
        public string MeslekTanim { get; set; } // LONGTEXT (MySQL) / nvarchar(max) (SQL Server)

[MaxLength(50)]
        public string NaceRev21Kodu { get; set; }

[Required]
        public string NaceRev21Tanim { get; set; } // LONGTEXT (MySQL) / nvarchar(max) (SQL Server)

public virtual NaceTehlikeSiniflari NaceTehlikeSinifi { get; set; }
        public virtual ICollection<RiskDegerlendirme> RiskDegerlendirmeler { get; set; } = new Hash
        public virtual ICollection<Firmalar> Firmalar { get; set; } = new HashSet<Firmalar>();
    }
```

```
}
```

Veri Seti tablosu de■i■ie gerek yok

NaceTehlikeS■n■flar■

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("NaceTehlikeSiniflari", Schema = "dbo")]
    public class NaceTehlikeSiniflari
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Id { get; set; } // Birincil anahtar olarak Id ekledik

public int NaceKoduId { get; set; } // NaceRev21Kodu yerine NaceKoduId kullan■yoruz

public TehlikeSinifi TehlikeSinifi { get; set; } // Enum: AzTehlikeli, Tehlikeli, CokTehlikeli

[ForeignKey(nameof(NaceKoduId))]
        public virtual NaceKodlari NaceKodu { get; set; } // NaceKodlari ile 1:1 ili■ki
    }
}
```

Veri Seti tablosu de■i■ime gerek yok

ÖDEMELER

Iyz■co ■■lemleri için Ödemeler tablosu

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("Odemeler", Schema = "dbo")]
    public class Odemeler
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int OdemeId { get; set; }

[Required]
        public int RefSozlesmeId { get; set; }

[Column(TypeName = "decimal(18, 2)")]
        public decimal Tutar { get; set; }

[Required]
        public DateTime OdemeTarihi { get; set; }

[Required]
        public string OdemeYontemi { get; set; }

[Required]
        public bool YapildiMi { get; set; }

[Required]
        public string OnayDurumu { get; set; }

public DateTime? OnayTarihi { get; set; }
        // iyzico'dan dönen PaymentId'yi saklamak için yeni alan
        public string? IyzicoPaymentId { get; set; }

// Navigation Property: Sozlesmeler tablosuyla FK ili■kisi
        [ForeignKey(nameof(RefSozlesmeId))]
```

```
        public virtual Sozlesmeler Sozlesme { get; set; }
    }
}
```

OSGB_Firmalar

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("OSGB_Firma", Schema = "dbo")]
    public class OSGB_Firmalar
    {
        [Key, Column(Order = 0)]
        public int RefOSGBId { get; set; } // OSGBler tablosuna foreign key

[Key, Column(Order = 1)]
        public int RefFirmaId { get; set; } // Firmalar tablosuna foreign key

// Navigation property: OSGBler tablosu
        [ForeignKey(nameof(RefOSGBId))]
        public virtual OSGBler OSGB { get; set; }

// Navigation property: Firmalar tablosu
        [ForeignKey(nameof(RefFirmaId))]
        public virtual Firmalar Firma { get; set; }
    }
}
```

Tamamen Normalize OSGB Firmalar■n■ tutmak için bir junction table

PERSONEL DAVETLER

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    public class PersonelDavetler
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int DavetId { get; set; }

[MaxLength(450)]
        public string? RefUserId { get; set; }

[Required]
        public int RefFirmaId { get; set; }

[Required]
        [MaxLength(50)]
        public string Rol { get; set; }

[MaxLength(256)]
        public string? Email { get; set; } // Yeni e-posta alan■

public DateTime DavetTarihi { get; set; } = DateTime.Now;

public bool? OnayDurumu { get; set; }

[MaxLength(450)]
        public string? InvitationToken { get; set; }
        public int? MeslekKoduId { get; set; } // Yeni nullable alan

[ForeignKey(nameof(RefUserId))]
        public virtual ApplicationUser? Kullanici { get; set; }

[ForeignKey(nameof(RefFirmaId))]
```

```
        public virtual Firmalar Firma { get; set; }
    }
}
```

Normalize mi, Eksik varm■?
Hay■r, kritik eksik yok—temel davet sistemi (firma + rol + token + onay) tam kapsanm■■. Ama ufak audit/esneklik için ■u alanlar eklenebilir (zorunlu de■il, mevcut haliyle de i■ görür):

OlusturanUserId: Davet eden kullan■c■y■ track etmek için (string, max 450, FK to AspNetUsers). Neden? Audit/raporlama için (kim kimi davet etti)—■u an RefUserId davet edilen, ama olu■turan ayr■ olmal■.
ExpiryDate: DateTime? (davet token süresi için, default DateTime.Now.AddDays(7)). Neden? Token'lar sonsuz kalmas■n, güvenlik için.
DavetNotu: string? (max 500, opsiyonel mesaj). Neden? Email'de ki■iselle■tirme için, ama basit tutmak istersen gereksiz.

Öneriler (Opsiyonel ■yile■tirmeler):

Enum Kullan■m■:

```
Rol: String yerine enum (örne■in, public RolEnum Rol { get; set; }—Doktor, Personel, Admin vb.). Ne
OnayDurumu: Bool? yerine enum (örne■in, public DavetDurumu? OnayDurumu { get; set; }—Beklemede, Kab
```

POZ■SYONLAR

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("Pozisyonlar", Schema = "dbo")]
    public class Pozisyonlar
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Id { get; set; }

[Required]
        [MaxLength(100)]
        public string PozisyonAdi { get; set; }
    }
}
```

Aç■ld■ ama yap■alnd■rma uygulanmad■ planlama a■amas■nda kuruldu

RANDEVU

```
// Dosya: Models/Randevu.cs
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    public enum RandevuDurum
    {
        Beklemede,
        Onaylandi,
        Reddedildi,
```

```
        Iptal
    }
public class Randevu
    {
        [Key]
        public int RandevuId { get; set; }
[Required]
        [Display(Name = "Hasta")]
        public int HastaId { get; set; }
[ForeignKey(nameof(HastaId))]
        public Kullanicilar Hasta { get; set; }
[Required]
        [Display(Name = "Doktor")]
        public string DoktorId { get; set; }
// E■er ApplicationUser kullan■yorsan■z
        [ForeignKey(nameof(DoktorId))]
        public ApplicationUser Doktor { get; set; }
[Required]
        [Display(Name = "Ba■lang■ç Zaman■")]
        public DateTime BaslangicZamani { get; set; }
[Required]
        [Display(Name = "Biti■ Zaman■")]
        public DateTime BitisZamani { get; set; }
[Required]
        [Display(Name = "Durum")]
        public RandevuDurum Durum { get; set; }
    }
}
```

Ertugrulun Kurdugu andevu sistemi

Sonuç:

Evet, normalizasyon iyi – randevu yönetimi için sa■lam temel (redundansi yok, ili■kiler sa■l■kl■). Bu haliyle deploy-ready, ama audit/validation tweak'leriyle (özellikle zaman çak■■mas■ için) daha robust olur.

RiskANAL■ZDAVETLER

```
using System;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
namespace ilterisg.Models
{
    [Table("RiskAnalizDavetler", Schema = "dbo")]
    public class RiskAnalizDavet
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Id { get; set; }
[Required]
        public Guid AnalizGrupId { get; set; } // Davet edilen risk analizi grubu
[Required]
        [MaxLength(450)]
        public string DoktorUserId { get; set; } // Davet edilen i■yeri hekiminin UserId'si
        [ForeignKey(nameof(DoktorUserId))]
        public virtual ApplicationUser? DoktorUser { get; set; } // ■li■kili kullan■c■
```

```
public int? FirmaId { get; set; } // Opsiyonel: Firma ba■lant■s■
        [ForeignKey(nameof(FirmaId))]
        public virtual Firmalar? Firma { get; set; } // Nullable firma ili■kisi

[Required]
        public Guid Token { get; set; } // Benzersiz eri■im token'■ (Guid'e çevir, otomatik genera

[Required]
        public DateTime DavetTarihi { get; set; } // Davet olu■turulma tarihi

public DateTime? SonGecerlilikTarihi { get; set; } // Token'■n geçerlilik süresi

public bool KullanildiMi { get; set; } = false; // Token kullan■ld■ m■?

[Required]
        [MaxLength(450)]
        public string DavetEdenUserId { get; set; } // Davet eden kullan■c■n■n UserId'si
        [ForeignKey(nameof(DavetEdenUserId))]
        public virtual ApplicationUser? DavetEdenUser { get; set; } // Davet eden kullan■c■

// Audit ekleri
        public DateTime CreatedAt { get; set; } = DateTime.UtcNow;
        public DateTime? UpdatedAt { get; set; }

public bool SilindiMi { get; set; } = false; // Soft delete

// Opsiyonel: Notlar
        [MaxLength(500)]
        public string? Notlar { get; set; }
    }
}
```

Normalizasyona uygun Davet eden ki■i isg uzman■ de■il Firma ise FirmaID Doluyor ama DavetEdenUSERID'de mevcut firma sorumlusu ■d atan■p firma ■d alan■ silinerek revize edilebilir.

R■SKANALZ KAYDI

Risk Analizlerinin Bulundugu Tablo

```
using ilterisg.Models.Enums; // SahaEnum için
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
namespace ilterisg.Models
{
    [Table("RiskAnalizKayitlari", Schema = "dbo")]
    public class RiskAnalizKaydi
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Id { get; set; }
        public int? FirmaId { get; set; } // Nullable yap■ld■
        [ForeignKey(nameof(FirmaId))]
        public Firmalar? Firma { get; set; } // Nullable yap■ld
        public string? UserId { get; set; }
        [ForeignKey(nameof(UserId))]
        public ApplicationUser? User { get; set; }
        [MaxLength(200)]
        public string RiskAdi { get; set; } = string.Empty;
        [MaxLength(500)]
        public string RiskAciklamasi { get; set; } = string.Empty;
        [MaxLength(500)]
        public string? OnerilenOnlem { get; set; }
        [Range(0.2, 10, ErrorMessage = "Olas■l■k 0.2 ile 10 aras■nda olmal■d■r.")]
        public double? Olasilik { get; set; }
        [Range(1, 5, ErrorMessage = "■iddet 1 ile 5 aras■nda olmal■d■r.")]
        public int? Siddet { get; set; }
```

```
        [NotMapped]
        public int? RiskSkoru => Olasilik.HasValue && Siddet.HasValue ? (int?)(Olasilik.Value * Sid
        [MaxLength(50)]
        public string AnalizMetodu { get; set; } = "5x5";
        [Range(0.5, 10)]
        public double? Maruziyet { get; set; }
        [Range(1, 100)]
        public double? FinneySiddet { get; set; }
        [NotMapped]
        public double? FinneySkor => Olasilik.HasValue && Maruziyet.HasValue && FinneySiddet.HasVal
            ? Olasilik.Value * Maruziyet.Value * FinneySiddet.Value : null;
        public int? RiskDegerlendirmeId { get; set; }
        [ForeignKey(nameof(RiskDegerlendirmeId))]
        public RiskDegerlendirme? KaynakRisk { get; set; }
        public SahaEnum Saha { get; set; } = SahaEnum.GenelSaha;
        [MaxLength(50)]
        public string TeminSuresi { get; set; } = "3 ay";
        [MaxLength(500)]
        public string? Notlar { get; set; }
        public int? MevzuatId { get; set; }
        [ForeignKey(nameof(MevzuatId))]
        public Mevzuat? Mevzuat { get; set; }
        public DateTime Tarih { get; set; } = DateTime.Now;
        public Guid? AnalizGrupId { get; set; }
        [NotMapped]
        public int Adim { get; set; }
        [ForeignKey(nameof(RefBildirimId))]
        public RiskBildirimleri? Bildirim { get; set; }
        public int? RefBildirimId { get; set; }
        [MaxLength(50, ErrorMessage = "Revizyon numaras■ 50 karakterden fazla olamaz.")]
        public string? RevizyonNo { get; set; } // Yeni eklenen nullable alan
        public DateTime? RevizyonTarihi { get; set; } // Yeni eklenen nullable alanmigr
        public string? RiskZarari { get; set; }
        public string? OneriYapanDoktorId { get; set; } // Öneriyi yapan ■■yeri Hekimi'nin Applica
        [ForeignKey(nameof(OneriYapanDoktorId))]
        public ApplicationUser? OneriYapanDoktor { get; set; } // Nullable foreign key

public bool IsApproved { get; set; } = true; // Varsay■lan olarak true (normal analiz için), öneril
    }
}
```

RevizyonNo ve Revizyon tariji JUNCT■ON Tableye ta■■nmal■ 100 adet risk varsa yüzüde guncellenmemel■ revizyon için. Önerilen yap■ revize k■sm■n■n junction table ■le yönetilmesi.

R■SK B■LD■R■MLER■ - R■SK B■LD■R■M MESJALARI

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using ilterisg.Models.Enums;

namespace ilterisg.Models
{
    [Table("RiskBildirimleri", Schema = "dbo")]
    public class RiskBildirimleri
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int BildirimId { get; set; }

[Required]
        [MaxLength(200)]
        public string KonuBasligi { get; set; } = "Risk Bildirimi";
```

```
[Required]
        public string Icerik { get; set; }

[Required]
        [MaxLength(450)] // AspNetUsers.Id ile e■le■iyor
        public string GonderenId { get; set; } // ■SG Uzman■ UserId

[ForeignKey(nameof(GonderenId))]
        public virtual ApplicationUser Gonderen { get; set; }

[Required]
        [MaxLength(450)] // AspNetUsers.Id ile e■le■iyor
        public string AliciId { get; set; } // ■■veren UserId

[ForeignKey(nameof(AliciId))]
        public virtual ApplicationUser Alici { get; set; }

[Required]
        public int RefFirmaId { get; set; } // ■lgili firma

[ForeignKey(nameof(RefFirmaId))]
        public virtual Firmalar Firma { get; set; }

public int? OsgbId { get; set; } // ■lgili OSGB (nullable, opsiyonel)

[ForeignKey(nameof(OsgbId))]
        public virtual OSGBler OSGB { get; set; }

[Required]
        public DateTime OlusturmaTarihi { get; set; } = DateTime.Now;

public DateTime? GuncellemeTarihi { get; set; } // Nullable, güncelleme opsiyonel

[Required]
        public DestekTalebiDurumu Durum { get; set; } = DestekTalebiDurumu.Acik;

public virtual ICollection<RiskBildirimMesajlari> Mesajlar { get; set; } = new List<RiskBildirimMes
        public virtual ICollection<Dokumanlar> Dokumanlar { get; set; } = new List<Dokumanlar>();
    }
}

using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("RiskBildirimMesajlari", Schema = "dbo")]
    public class RiskBildirimMesajlari
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int MesajId { get; set; }

[Required]
        public int RefBildirimId { get; set; } // RiskBildirimleri.BildirimId ile ba■lanacak

[ForeignKey(nameof(RefBildirimId))]
        public virtual RiskBildirimleri RiskBildirimi { get; set; }

[Required]
        public string MesajIcerigi { get; set; }

[Required]
        [MaxLength(450)] // AspNetUsers.Id ile e■le■iyor
        public string GonderenId { get; set; } // ApplicationUser Id

[ForeignKey(nameof(GonderenId))]
        public virtual ApplicationUser Gonderen { get; set; }

[Required]
        public DateTime GonderimTarihi { get; set; } = DateTime.Now;
    }
}
```

■ 4. Gereksiz Alan Var m■?

Hay■r.
Her alan■n bir amac■ var:

AliciId ve GonderenId birlikte gerekli (mesaj yönü için).

OsgbId opsiyonel ama farkl■ kurumsal türleri destekler.

FirmaId zorunlu — kurumsal ba■lam.

Durum, Tarih, Icerik hepsi atomik ve i■levsel.

Hiçbir alan “fazlal■k” ya da “türev” de■il.
Hiçbir bilgi ba■ka alandan türetilemiyor — bu da 3NF’nin özüdür.

■ Sonuç
Tablo Normalizasyon Gereksiz Alan Not
RiskBildirimleri ■ 3NF ■ Yok Tüm alanlar anlaml■ ve ayr■ i■levde.
RiskBildirimMesajlari ■ 3NF ■ Yok Sade ve efektif.
■ K■sa Özet:

Modeller tamamen normalize (3. normal forma kadar).

OsgbId’nin nullable olmas■ do■ru, çünkü opsiyonel.

AliciId ve GonderenId ayr■ FK olarak tasarlanmas■ do■ru modelleme yakla■■m■.

Gereksiz veya silinmesi gereken hiçbir alan yok.

Sadece “performans” amaçl■ birkaç index önerisi yap■labilir.

■ Yani bu iki tablo üretim düzeyinde normalize,
veri bütünlü■ü sa■lam,
i■ modeli aç■s■ndan mükemmel orant■l■ bir tasar■ma sahip.

RiskDe■erlenDirme
Veri seti tablosu revizeye gerek yok

```
using ilterisg.Models.Enums;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    // Veritaban■nda dbo ■emas■nda RiskDegerlendirme tablosuna kar■■l■k gelir
    [Table("RiskDegerlendirme", Schema = "dbo")]
    public class RiskDegerlendirme
    {
        // Birincil anahtar (primary key), otomatik artan (identity)
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Id { get; set; }
```

```
// NaceKodlari tablosuna foreign key, zorunlu alan
        public int? NaceKoduId { get; set; }

// Riskin tan■m■ (örn. "Yüksekten dü■me"), zorunlu alan, maksimum 200 karakter
        [Required]
        [MaxLength(200)]
        public string RiskTanimi { get; set; }

// Riskin olu■mas■na neden olan faktörler (örn. "Kaygan zemin, yetersiz ayd■nlatma"), opsiyonel, m
        [MaxLength(500)]
        public string RiskFaktorleri { get; set; }

// Riski azaltmak için al■nacak önlemler (örn. "Kaymaz zemin, uyar■ levhalar■"), opsiyonel, maksim
        [MaxLength(500)]
        public string KontrolOnlemleri { get; set; }

// Fine-Kinney metodolojisi için olas■l■k (P), 0.2-10 aras■, opsiyonel
        [Range(0.2, 10)]
        public double? OlasilikFineKinney { get; set; } // double olarak de■i■tirildi

// Fine-Kinney metodolojisi için ■iddet (S), 1-100 aras■, opsiyonel
        [Range(1, 100)]
        public double? SiddetFineKinney { get; set; } // double olarak de■i■tirildi

// Fine-Kinney metodolojisi için frekans (F), 0.5-10 aras■, opsiyonel
        [Range(0.5, 10)]
        public double? FrekansFineKinney { get; set; } // double olarak de■i■tirildi

// Fine-Kinney risk skoru (P × S × F), opsiyonel
        public double? RiskSkoruFineKinney { get; set; } // double olarak de■i■tirildi

// Fine-Kinney risk seviyesi (örn. "Dü■ük Risk", "Yüksek Risk"), opsiyonel, maksimum 50 karakter
        [MaxLength(50)]
        public string RiskSeviyesiFineKinney { get; set; }

// 5x5 metodolojisi için olas■l■k (L), 1-5 aras■, opsiyonel
        [Range(1, 5)]
        public int? Olasilik5x5 { get; set; }

// 5x5 metodolojisi için ■iddet (I), 1-5 aras■, opsiyonel
        [Range(1, 5)]
        public int? Siddet5x5 { get; set; }

// 5x5 risk skoru (L × I), opsiyonel
        public int? RiskSkoru5x5 { get; set; }

// 5x5 risk seviyesi (örn. "Dü■ük Risk", "Çok Yüksek Risk"), opsiyonel, maksimum 50 karakter
        [MaxLength(50)]
        public string RiskSeviyesi5x5 { get; set; }

// Mevzuat tablosuna foreign key, opsiyonel
        public int? MevzuatId { get; set; }

// NaceKodlari tablosu ile ili■ki (navigation property)
        [ForeignKey(nameof(NaceKoduId))]
        public virtual NaceKodlari NaceKodu { get; set; }

// Mevzuat tablosu ile ili■ki (navigation property)
        [ForeignKey(nameof(MevzuatId))]
        public virtual Mevzuat Mevzuat { get; set; }
        public SahaEnum? Saha { get; set; } // Nullable SahaEnum alan■
        public bool OnaylandiMi { get; set; } = false;
        // Mevcut modele RiskZarari ekle
        [MaxLength(500)]
        public string? RiskZarari { get; set; }
    }
}
```

R■skOnlemSonrasiDurum

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
```

```
namespace ilterisg.Models
{
    [Table("RiskOnlemSonrasiDurum", Schema = "dbo")]
    public class RiskOnlemSonrasiDurum
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Id { get; set; }

[Required(ErrorMessage = "Risk analizi kayd■ zorunludur.")]
        public int RiskAnalizKaydiId { get; set; }

[ForeignKey(nameof(RiskAnalizKaydiId))]
        public RiskAnalizKaydi RiskAnalizKaydi { get; set; } = null!;

[Required(ErrorMessage = "Analiz grup ID zorunludur.")]
        public Guid AnalizGrupId { get; set; }

[MaxLength(500, ErrorMessage = "Mevcut durum 500 karakterden fazla olamaz.")]
        public string? MevcutDurum { get; set; }

[MaxLength(500, ErrorMessage = "Önlem sonras■ durum 500 karakterden fazla olamaz.")]
        public string? OnlemSonrasiDurum { get; set; } = string.Empty;

[Range(0.2, 10, ErrorMessage = "Olas■l■k 0.2 ile 10 aras■nda olmal■d■r (Fine-Kinney için). 5x5 iç
        public double? Olasilik { get; set; } // Fine-Kinney: 0.2-10, 5x5: 1-5

[Range(0.5, 10, ErrorMessage = "Maruziyet 0.5 ile 10 aras■nda olmal■d■r (Fine-Kinney için).")]
        public double? Maruziyet { get; set; } // Fine-Kinney için, 5x5'te null

[Range(1, 100, ErrorMessage = "Finney ■iddet 1 ile 100 aras■nda olmal■d■r (Fine-Kinney için).")]
        public double? FinneySiddet { get; set; } // Fine-Kinney için, 5x5'te null

[MaxLength(500, ErrorMessage = "Ek notlar 500 karakterden fazla olamaz.")]
        public string? EkNotlar { get; set; }

[MaxLength(250, ErrorMessage = "Sorumlu ki■i 250 karakterden fazla olamaz.")]
        public string? SorumluKisi { get; set; }

[MaxLength(500, ErrorMessage = "Etkilenen ki■iler 500 karakterden fazla olamaz.")]
        public string? EtkilenenKisiler { get; set; }

public DateTime GuncellemeTarihi { get; set; } = DateTime.UtcNow;
    }
}
```

tamamen normalize sadece Guncelleme tarihi junction table ile yonet■leb■l■r r■skanal■z kayd■ gibi

SeçiliHekimBilgileri

Tamamen normalize Medula için ilgili varsay■lan bilgilerin kaydedildi■i bir junction table

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    public enum SecimTuruEnum
    {
        [Display(Name = "TesisKoduSecimi")]
        TesisKodu = 1,
        [Display(Name = "BransKoduSecimi")]
        BransKodu = 2,
        [Display(Name = "SertifikaSecimi")]
        Sertifika = 3,
        [Display(Name = "MedulaSifresiSecimi")]
        MedulaSifresi = 4
    }

[Table("SeciliHekimBilgileri", Schema = "dbo")]
```

```
    public class SeciliHekimBilgileri
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Id { get; set; }

[Required]
        public int KullaniciId { get; set; }

[Required]
        public int IsyeriHekimiBilgileriId { get; set; }

[ForeignKey("IsyeriHekimiBilgileriId")]
        public virtual IsyeriHekimiBilgileri IsyeriHekimiBilgileri { get; set; }

[Required]
        public SecimTuruEnum SecimTuru { get; set; } // Hangi alan seçildi? (TesisKodu, BransKodu,

public DateTime SecimTarihi { get; set; } = DateTime.Now; // Seçim zaman■
    }
}
```

Sektorler

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("Sektorler", Schema = "dbo")]
    public class Sektorler
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int SektorId { get; set; }

[Required]
        [MaxLength(50)]
        public string SektorAdi { get; set; }

// Navigation Property: Firma_Sektor tablosuyla ili■ki
        public virtual ICollection<Firma_Sektor> Firma_Sektor { get; set; } = new HashSet<Firma_Sek
    }
}
```

Planlamada yap■land■r■ld■ hiçbir yere ba■l■ de■il kullan■lm■yor.

SektorRiski

```
using ilterisg.Models; // Sektorler s■n■f■ bu namespace'te
using System.ComponentModel.DataAnnotations.Schema;

public class SektorRiski
{
    public int Id { get; set; }

public int SektorId { get; set; }
    [ForeignKey("SektorId")]
    public Sektorler Sektor { get; set; }

public string RiskAdi { get; set; } = string.Empty;
    public string RiskAciklamasi { get; set; } = string.Empty;
    public string OnerilenOnlem { get; set; } = string.Empty;

// 5x5 matrise özel alanlar
    public int Olasilik { get; set; }  // 1-5
    public int Siddet { get; set; }    // 1-5
}
```

Planlamada yap■land■r■ld■ hiçbir yere ba■l■ de■il kullan■lm■yor.

Yanl■■BilgiBildirimi

```
namespace ilterisg.Models
{
    public class YanlisBilgiBildirimi
    {
        public int Id { get; set; }

// Yanl■■ bilgi oldu■u belirtilen alan■n ad■ ve de■eri
        public string? AlanAdi { get; set; } // Hata yoksa null olabilir
        public string? MevcutDeger { get; set; } // Hata yoksa null olabilir
        public string? DogruDeger { get; set; } // Hata yoksa null olabilir

// Kullan■c■n■n bilgilerin do■ru olup olmad■■■n■ onaylamas■
        public bool IsBilgilerDogru { get; set; } = true; // Varsay■lan true, nullable de■il

// Bildirim tarihi ve i■lem durumu
        public DateTime? TalepTarihi { get; set; } // Hata yoksa null olabilir, varsay■lan■ kald■r
        public bool? IsProcessed { get; set; } // Hata yoksa null olabilir, yönetici i■lemi yoksa a

// Yönetici i■lemleri için ek alanlar
        public string? YoneticiNotu { get; set; } // Yönetici devreye girmezse null
        public DateTime? IslemTarihi { get; set; } // Yönetici devreye girmezse null

// ApplicationUser ile ili■ki
        public string ApplicationUserId { get; set; } // Foreign Key, nullable de■il
        public ApplicationUser? ApplicationUser { get; set; } // Navigation Property
    }
}
```

Tamemen normalizeye uygun gereksiz alan yok, gereksiz navigation proerty yok.

Öneri

```
// Kullan■c■n■n bilgilerin do■ru olup olmad■■■n■ onaylamas■
        public bool IsBilgilerDogru { get; set; } = true; // Varsay■lan true, nullable de■il
```

Bu alan kald■r■labilir bilgiler do■ruysa hiçbir veri set edilmez de■ilse edilir tek.

Subeler

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("Subeler", Schema = "dbo")]
    public class Subeler
    {
        [Key]
        public int SubeId { get; set; }

[Required]
        [MaxLength(50)]
        public string SubeAdi { get; set; }

[Required]
        public int SicilNo { get; set; }

[Required]
        public int RefFirmaId { get; set; }

// Navigation Property: Firmalar tablosuyla FK ili■kisi
        [ForeignKey(nameof(RefFirmaId))]
        public virtual Firmalar Firma { get; set; }
    }
```

```
}
```

Planama taraf■nda aç■ld■. Kullan■lm■yor.

SÖZLESMELER

```
using ilterisg.Models.Enums;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("Sozlesmeler", Schema = "dbo")]
    public class Sozlesmeler
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Sozlesmeld { get; set; }

[Required]
        public int RefFirmaId { get; set; }

[Required]
        public int RefOSGBld { get; set; }

[Required]
        public DateTime BaslangicTarihi { get; set; }

public DateTime? BitisTarihi { get; set; }

[Required]
        public bool IsActive { get; set; } = false;

[Required]
        public DateTime OlusturulmaTarihi { get; set; }

[Required]
        public SozlesmeOnayDurumu OnayDurumu { get; set; } = SozlesmeOnayDurumu.Taslak;

public string SozlesmeMetni { get; set; } // LONGTEXT (MySQL) / nvarchar(max) (SQL Server)

[ForeignKey(nameof(RefFirmaId))]
        public virtual Firmalar Firma { get; set; }

[ForeignKey(nameof(RefOSGBld))]
        public virtual OSGBler OSGB { get; set; }
    }
}
```

Firmalar■n Taslak sonucu firmalarla yapt■■■ kar■■l■kl■ otomatik sözle■meyi temsil ediyor gereksiz alan yok Normalize.

SozlesmeTaslaklar■

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace ilterisg.Models
{
    [Table("SozlesmeTaslaklari", Schema = "dbo")]
    public class SozlesmeTaslaklari
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int TaslakId { get; set; }

[Required]
        public int RefOSGBId { get; set; }
```

```
public int? RefFirmaId { get; set; }

[Required]
        public string TaslakMetni { get; set; } // LONGTEXT (MySQL) / nvarchar(max) (SQL Server)

[Required]
        [Column(TypeName = "decimal(18, 2)")]
        public decimal TaslakTutar { get; set; }

[Required]
        public DateTime OlusturulmaTarihi { get; set; } = DateTime.Now;

[ForeignKey(nameof(RefOSGBId))]
        public virtual OSGBler OSGB { get; set; }

[ForeignKey(nameof(RefFirmaId))]
        public virtual Firmalar Firma { get; set; }

public bool IsVarsayilan { get; set; } = false;
    }
}
```

Taslak tutulan bir junction table sadece Normalize.