

E-KİTAP OTOMASYONU

E-Kitap Sistemi ER Diyagramı

Bu proje, bir e-kitap sistemi için tasarlanmış veritabanı yapısını temsil eden ER (Entity-Relationship) diyagramını içermektedir. Aşağıda, diyagramdaki tabloların açıklamaları ve aralarındaki ilişkiler detaylı bir şekilde açıklanmıştır.

Genel Bakış

ER diyagramı, bir e-kitap platformundaki kullanıcılar, kitaplar, yazarlar, yayınevleri, kategoriler ve kullanıcıların kişisel kütüphaneleri gibi bileşenlerin arasındaki ilişkileri modellemektedir. Veritabanı, kullanıcıların kitaplara erişimi, kitap değerlendirmeleri ve sipariş işlemleri gibi işlemleri yönetmek üzere tasarlanmıştır.

Tablolar ve Alanlar

1. kullanıcılar

idkullanıcılar (INT): Kullanıcıların benzersiz kimliği.

kullanıcad (VARCHAR): Kullanıcının adı.

kullanıcisoyad (VARCHAR): Kullanıcının soyadı.

eposta (VARCHAR): Kullanıcının e-posta adresi.

sifre (VARCHAR): Kullanıcının şifresi.

2. kisiselkutuphane

idkisiselkutuphane (INT): Kişisel kütüphanenin benzersiz kimliği.

kitapdurumu (VARCHAR): Kitap durumu (okundu, okunacak vb.).

kisiselkutuphaneol (VARCHAR): Kütüphaneye eklenme durumu.

yorum/gorus_idyorum/gorusid (INT): İlgili yorum veya görüş.

3. kitap

kitapid (INT): Kitapların benzersiz kimliği.

kitapad (VARCHAR): Kitap adı.

isbn (VARCHAR): ISBN numarası.

yayintarihi (VARCHAR): Yayın tarihi.

kitapdili (VARCHAR): Kitap dili.

4. yazarlar

idyazarlar (INT): Yazarların benzersiz kimliği.

yazarad (VARCHAR): Yazar adı.

yazarsoyad (VARCHAR): Yazar soyadı.

5. yayınevi

idyayinevi (INT): Yayınevlerinin benzersiz kimliği.

yayineviad (VARCHAR): Yayınevi adı.

yayinevikurulusyili (VARCHAR): Yayınevinin kuruluş yılı.

6. kategori

idkategori (INT): Kategorilerin benzersiz kimliği.

kategoriad (VARCHAR): Kategori adı.

7. siparis/indirme

idsiparis/indirmeid (INT): Sipariş veya indirme işleminin benzersiz kimliği.

tarih (DATETIME): Sipariş/indirme tarihi.

8. yorum/gorus

idyorum/gorusid (INT): Yorumların veya görüşlerin benzersiz kimliği.

metin (VARCHAR): Yorum metni.

puan (INT): Yorum puanı.

yayintarihi (DATETIME): Yorumun yayın tarihi.

Tablolar Arasındaki İlişkiler

kullanıcılar ve kisiselkutuphane arasında bire çok ilişki bulunmaktadır. Bir kullanıcı birden fazla kişisel kütüphaneye sahip olabilir. Bunlara bazı örnekler vermek gerekirse.

*kitap, *yazarlar ve yayinevi tabloları arasında ilişkiler:

Bir kitap bir veya birden fazla yazara ait olabilir (kitap ve yazarlar arasındaki ilişki).

Bir kitap bir yayinevi tarafından yayımlanır (kitap ve yayinevi arasındaki ilişki).

kitap ve kategori arasında bire çok ilişki vardır. Bir kitap bir kategoriye ait olabilir.

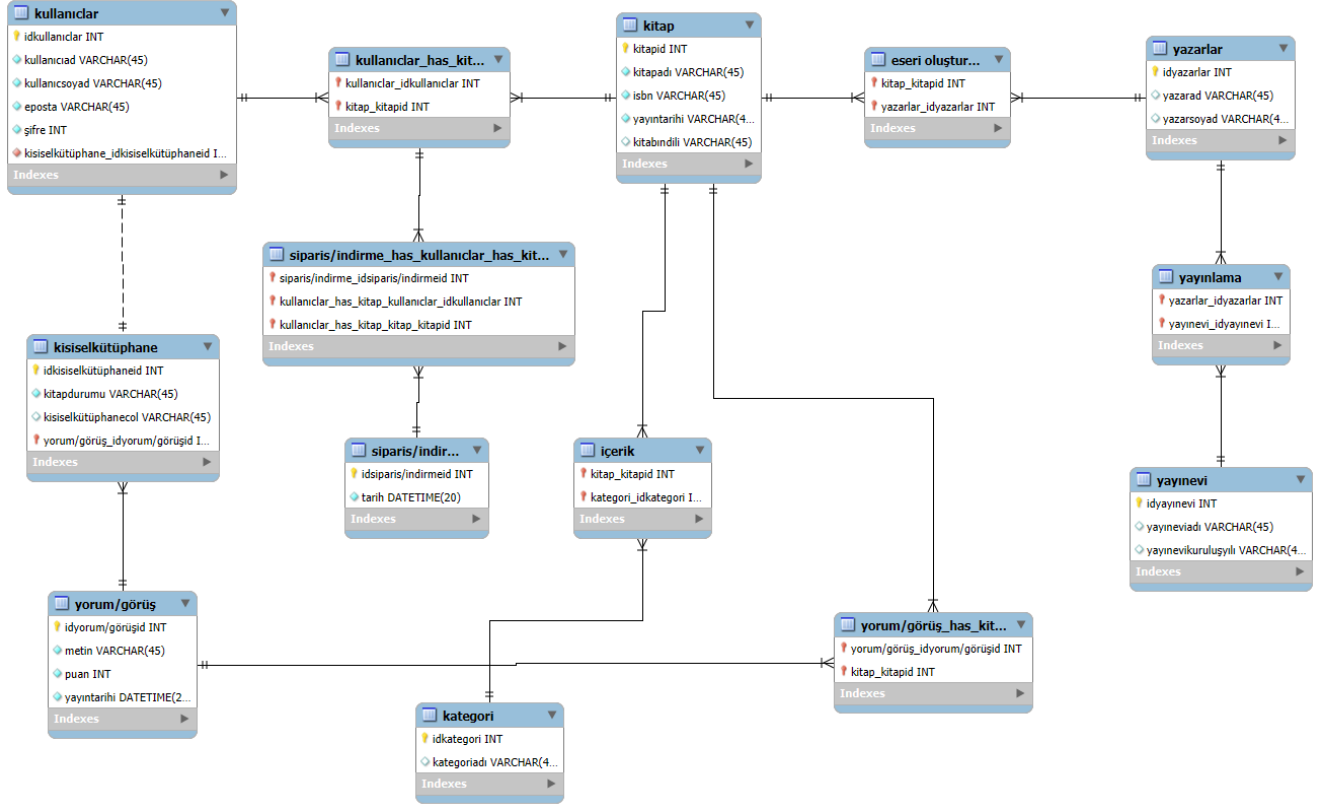
*kullanıcılar, **kitap, ve *siparis/indirme arasındaki ilişki, kullanıcıların kitap siparişleri ve indirmeleri üzerinden modellenmiştir.

yorum/gorus ve kitap arasında bire çok ilişki bulunmaktadır. Bir kitap birden fazla yorum alabilir.

Kullanım

Bu ER diyagramı, e-kitap sisteminin veritabanı tasarımını belgelemek ve uygulama geliştiricilere rehberlik etmek amacıyla oluşturulmuştur. Diyagram, kullanıcıların e-kitap platformundaki işlemlerini ve ilişkilerini anlamayı kolaylaştırır.

E-R DİYAGRAMI



Ana tablolar ve açıklamaları:

1. Kullanıcılar (kullanıcılar)

Attributes:

idkullanıcılar birincil anahtar.

Kullanıcı bilgileri (kullanıcıad, kullanıcısoyad, eposta, şifre)

İlişkiler:

Kullanıcılar, kişisel kütüphane ve sipariş/indirme tablosuyla ilişkili.

Kullanıcıların kitapla ilişkisi kullanıcılar_has_kitap ara tablosuyla sağlandı.

2. Kitap (kitap)

Attributes:

kitapid birincil anahtar.

İlişkiler:

Kitaplar:

Kategori ile ilişkilendirilmiş (icerik tablosu üzerinden).

Yazarlar ile eseri_olusturur tablosu üzerinden bağlandı.

Kullanıcıların yorumları ve kişisel kütüphaneleriyle ilişkisi yapıldı.

3. Yazarlar (yazarlar)

Attributes:

idyazarlar birincil anahtar.

yazar, yazarsoyad net ve işlevsel.

İlişkiler:

Yazarlar:

Yayınevi ile Many-to-Many (çoktan çokla) ilişki kurmuş (yayınlama tablosu üzerinden).

Kitaplar ile eseri_olusturur tablosu üzerinden bağlandı.

4. Yayınevi (yayınevi)

Attributes:

idyayınevi birincil anahtar.

Yayınevi adı ve kuruluş yılı bilgileri mantıklı.

İlişkiler:

Yayınevleri, yazarlarla ilişkili (yayınlama tablosu üzerinden).

5. Kategori (kategori)

Attributes:

idkategori birincil anahtar.

Kategori adı doğru şekilde eklendi.

İlişkiler:

Kitaplarla icerik ara tablosu üzerinden Many-to-Many ilişki kuruldu.

6. Sipariş/İndirme (siparis/indirme)

Attributes:

idsiparis/indirme birincil anahtar.

Tarih bilgisi yerinde.

İlişkiler:

Kullanıcılar ve kitaplar arasındaki ilişkileri takip ediyor.

7. Yorum/Görüş (yorum/görüş)

Attributes:

idyorum/görüş birincil anahtar.

Puanlama ve metin bilgisi yerinde.

Yayın tarihi eklendi.

İlişkiler:

Kitaplarla ve kullanıcılarla ilişkilidir.

A.Veri Tabanı Şemasının SQL Server’da Oluşturulması

Aşağıda diyagramdaki ilişkileri göz önünde bulundurarak, tablolar ve ilişkileri oluşturan ilgili SQL kodları:

-- Kullanıcılar tablosu

```
CREATE TABLE kullanicilar (  
    idkullanicilar INT PRIMARY KEY,  
    kullaniciad VARCHAR(45) NOT NULL,  
    kullanicisoyad VARCHAR(45) NOT NULL,  
    eposta VARCHAR(100) UNIQUE NOT NULL,  
    sifre VARCHAR(100) NOT NULL  
);
```

-- Kişisel Kütüphane tablosu

```
CREATE TABLE kisiselkutuphane (  
    idkisiselkutuphane INT PRIMARY KEY,  
    kitapdurumu VARCHAR(45) NOT NULL,  
    kisiselkutuphaneol VARCHAR(45) NOT NULL,  
    yorum_gorus_idyorum_gorus_id INT,  
    FOREIGN KEY (yorum_gorus_idyorum_gorus_id) REFERENCES yorum_gorus  
(idyorum_gorus)  
);
```

-- Kitap tablosu

```
CREATE TABLE kitap (  
    kitapid INT PRIMARY KEY,  
    kitapadi VARCHAR(45) NOT NULL,  
    isbn VARCHAR(45) NOT NULL,  
    yayintarihi VARCHAR(45),  
    kitapdili VARCHAR(45)  
);
```

-- Yazarlar tablosu

```
CREATE TABLE yazarlar (  
    idyazarlar INT PRIMARY KEY,  
    yazarad VARCHAR(45) NOT NULL,  
    yazarisoyad VARCHAR(45) NOT NULL  
);
```

-- Yayın Evi tablosu

```
CREATE TABLE yayinevi (  
    idyayinevi INT PRIMARY KEY,  
    yayineviad VARCHAR(45) NOT NULL,  
    yayinevikurulusyili VARCHAR(45)  
);
```

-- Yayınlama tablosu

```
CREATE TABLE yayinlama (  
    idyayinlama INT PRIMARY KEY,  
    yazarlar_idyazarlar INT,  
    yayinevi_idyayinevi INT,  
    FOREIGN KEY (yazarlar_idyazarlar) REFERENCES yazarlar (idyazarlar),  
    FOREIGN KEY (yayinevi_idyayinevi) REFERENCES yayinevi (idyayinevi)  
);
```

-- İçerik tablosu

```
CREATE TABLE icerik (  
    kitap_kitapid INT,  
    kategori_idkategori INT,  
    PRIMARY KEY (kitap_kitapid, kategori_idkategori),  
    FOREIGN KEY (kitap_kitapid) REFERENCES kitap (kitapid),
```



```
FOREIGN KEY (kategori_idkategori) REFERENCES kategori (idkategori)
);
```

-- Kategori tablosu

```
CREATE TABLE kategori (
    idkategori INT PRIMARY KEY,
    kategoriadi VARCHAR(45) NOT NULL
);
```

-- Sipariş / İndirme tablosu

```
CREATE TABLE siparis_indirme (
    idsiparis_indirmeid INT PRIMARY KEY,
    tarih DATETIME NOT NULL
);
```

-- Kullanıcılar ve Kitaplar İlişkisi tablosu

```
CREATE TABLE kullanicilar_has_kitap (
    kullanicilar_idkullanicilar INT,
    kitap_kitapid INT,
    PRIMARY KEY (kullanicilar_idkullanicilar, kitap_kitapid),
    FOREIGN KEY (kullanicilar_idkullanicilar) REFERENCES kullanicilar
(idkullanicilar),
    FOREIGN KEY (kitap_kitapid) REFERENCES kitap (kitapid)
);
```

B.Örnek Veri Ekleme

Her tabloya örnek veri ekliyorum. Örnek olarak birkaç veri ekleyelim.

-- Kullanıcılar tablosu için veri ekleme

INSERT INTO kullanicilar VALUES

(1, 'Ahmet', 'Yılmaz', 'ahmet@gmail.com', 'password123'),

(2, 'Ayşe', 'Kaya', 'ayse@gmail.com', 'pass456');

-- Kitap tablosu için veri ekleme

INSERT INTO kitap VALUES

(1, 'Sefiller', '9781234567890', '1862', 'Türkçe'),

(2, 'Suç ve Ceza', '9780987654321', '1866', 'Türkçe');

-- Yazarlar tablosu için veri ekleme

INSERT INTO yazarlar VALUES

(1, 'Victor', 'Hugo'),

(2, 'Fyodor', 'Dostoyevski');

-- Yayın Evi tablosu için veri ekleme

INSERT INTO yayinevi VALUES

(1, 'Can Yayınları', '1920'),

(2, 'İş Bankası Kültür Yayınları', '1956');

C. SQL İşlemleri ve Komutları

Örnek bazı SQL sorguları şunlardır:

-- 1. Bir kullanıcının kişisel kütüphanesindeki tüm kitapları listele

```
SELECT k.kitapadi, kk.kitapdurumu
```

```
FROM kitap k
```

```
JOIN kisiselkutuphane kk ON k.kitapid = kk.idkisiselkutuphane;
```

-- 2. Belirli bir kategorideki tüm kitapları listele

```
SELECT k.kitapadi
```

```
FROM kitap k
```

```
JOIN icerik i ON k.kitapid = i.kitap_kitapid
```

```
JOIN kategori c ON i.kategori_idkategori = c.idkategori
```

```
WHERE c.kategoriadi = 'Roman';
```

D. Saklı Yordam ve Tetikleyici

Bir saklı yordam ve bir tetikleyici örneği:

-- Saklı Yordam: Yeni kitap ekleme

CREATE PROCEDURE YeniKitapEkle

 @kitapadi VARCHAR(45),

 @isbn VARCHAR(45),

 @yayintarihi VARCHAR(45),

 @kitapdili VARCHAR(45)

AS

BEGIN

 INSERT INTO kitap (kitapadi, isbn, yayintarihi, kitapdili)

 VALUES (@kitapadi, @isbn, @yayintarihi, @kitapdili);

END;

-- Tetikleyici: Kitap silindiğinde log kaydı tutma

CREATE TRIGGER KitapSilindi

ON kitap

AFTER DELETE

AS

BEGIN

 INSERT INTO kitap_log (kitapadi, silinme_tarihi)

 SELECT kitapadi, GETDATE()

 FROM DELETED;

END;

E. Transaction Yönetimi

Örnek transaction yönetimi:

```
BEGIN TRANSACTION;
```

```
-- Sipariş işlemi
```

```
INSERT INTO siparis_indirme (idsiparis_indirmeid, tarih) VALUES (1, GETDATE());
```

```
-- Kullanıcıya kitap ekleme
```

```
INSERT INTO kullanicilar_has_kitap (kullanicilar_idkullanicilar, kitap_kitapid)
```

```
VALUES (1, 2);
```

```
-- Hata durumunda geri al
```

```
IF @@ERROR <> 0
```

```
BEGIN
```

```
    ROLLBACK TRANSACTION;
```

```
    PRINT 'Transaction geri alındı.';
```

```
END
```

```
ELSE
```

```
BEGIN
```

```
    COMMIT TRANSACTION;
```

```
    PRINT 'Transaction başarıyla tamamlandı.';
```

```
END;
```

HAZIRLAYANLAR

200260058 BARIŞ TAŞKIRAN

200260034 MUSTAFA KOPARAL