Kutuphane Veritabani Management

MOUAD ASSILA - 191502214 - Bilgisayar Mühendisliği (TR) 4. Sınıf

ILYES SAIDI - 201502203 - Bilgisayar Mühendisliği (TR) 2. Sınıf

ABDULLAHI MOHAMED JEYLANI – 2111502215 - Bilgisayar Mühendisliği (TR) 2. Sınıf

**Projenin Amacı:**

1. Kütüphane kaynaklarını etkin bir şekilde yönetmek: Projenin temel amacı, kütüphanede bulunan kitaplar, yazarlar, türler, yayınevleri ve diğer kaynaklar gibi bilgileri etkin bir şekilde yönetmektir. Bu, kitapların eklenmesi, düzenlenmesi ve silinmesi gibi işlemleri içerir.

2. Öğrencilerin kitap ödünç alma sürecini kolaylaştırmak: Proje, öğrencilerin kütüphaneden kitap ödünç almasını kolaylaştırmak için bir ödünç alma mekanizması sağlar. Öğrencilerin kitapları arama, ödünç alma, iade etme ve ödünç alma sürelerini takip etme gibi işlemleri gerçekleştirebilirler.

3. Personelin ve öğrencilerin yönetimini sağlamak: Proje, kütüphanedeki personelin ve öğrencilerin bilgilerini yönetmek için bir sistem sağlar. Personel, kaynakları yönetme, öğrencilerin taleplerini işleme ve raporlama gibi işlevlere erişebilir. Öğrenciler ise kitap ödünç alma, geçmiş ödünçleri görüntüleme ve iletişim bilgilerini güncelleme gibi işlemleri gerçekleştirebilir.

Projenin Kapsamı:

1. Kitap yönetimi: Proje, kitapların ekleme, düzenleme, silme ve arama gibi işlemlerini kapsar. Kitapların özelliklerini (başlık, yazar, yayın yılı, ISBN vb.) depolamak ve bu bilgilere erişim sağlamak mümkündür.

2. Öğrenci yönetimi: Proje, öğrencilerin kayıt bilgilerini (ad, soyad, sınıf, doğum tarihi, iletişim bilgileri vb.) depolar. Öğrenciler, kitap ödünç alma süreçlerini yönetebilir ve iletişim bilgilerini güncelleyebilir.

3. Personel yönetimi: Proje, kütüphanede çalışan personelin bilgilerini (ad, e-posta, uzmanlık alanı vb.) depolar. Personel, kitapların yönetimi, öğrenci taleplerinin işlenmesi ve diğer ilgili görevleri gerçekleştirir.

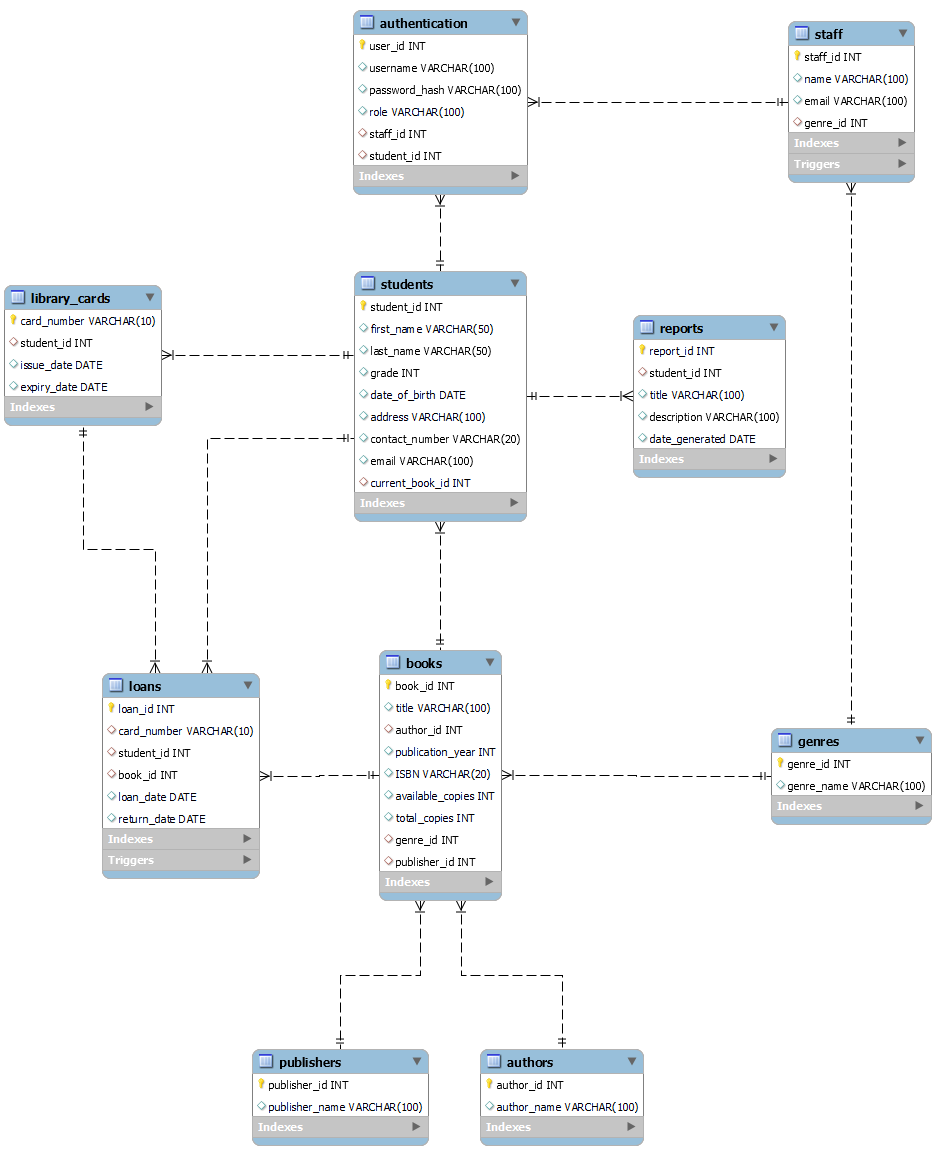
4. Ödünç alma ve iade işlemleri: Proje, öğrencilerin kitap ödünç alma ve iade etme işlemlerini takip eder. Öğrenciler, kütüphane kartları aracılığıyla kitapları ödünç alabilir, ödünç alma sürelerini kontrol edebilir ve kitapları iade edebilir.

5. Raporlama: Proje, öğrenci raporlarını oluşturma ve görüntüleme yeteneği sağlar. Raporlar, öğrenci ödünç alma geçmişi, iade süreleri, gecikme durumları vb. gibi bilgileri içerebilir.

6. Kimlik doğrulama ve kullanıcı rolleri: Proje, personel ve öğrencilerin sisteme giriş yapabilmesi için kimlik doğrulama mekanizması sağlar. Kullanıcılar, kullanıcı adı ve parola kullanarak giriş yaparlar ve kullanıcı rollerine bağlı olarak yetkilere erişebilirler.

Bu, genel bir kütüphane yönetim sistemi projesinin amaçlarını ve kapsamını açıklar. Projenin gerçek uygulaması, belirli gereksinimlere, özelleştirmelere ve kuruluşun ihtiyaçlarına bağlı olarak değişebilir.

**ER Diyagramı:**



**Normalizasyon Kurallarının Uygulanması:**

Oluşturulan tablolar, normalizasyon kurallarına uygun olarak tasarlanmalıdır. Genellikle, normalizasyon süreci veritabanının gereksinimlerine ve iş mantığına bağlı olarak değişir. Örneğin, tabloların birincil anahtarlarla tanımlanan tek bir temel bilgi birimi içermesi gerekmektedir. İkinci normal formda ise veri tekrarını önlemek için gereken düzeltmeler yapılmalıdır. Ayrıca, ilişkisel yapıda tutulan verilerin bütünlüğünü sağlamak için yabancı anahtar ilişkileri kullanılmalıdır.

**Tablolar Arası İlişki Şeması:**

Aşağıda, oluşturulan tablolar arasındaki ilişki şemasını özetlemek için basit bir metin tablosu sağlıyorum:

Authors (Yazarlar) 1:n Books (Kitaplar)

Genres (Türler) 1:n Books (Kitaplar)

Publishers (Yayınevleri) 1:n Books (Kitaplar)

Books (Kitaplar) 1:n Loans (Ödünç Verme)

Students (Öğrenciler) 1:n Loans (Ödünç Verme)

Students (Öğrenciler) 1:n Reports (Raporlar)

Staff (Personel) 1:n Reports (Raporlar)

Genres (Türler) 1:n Staff (Personel)

Staff (Personel) 1:n Authentication (Yetkilendirme)

Students (Öğrenciler) 1:n Authentication (Yetkilendirme)

Students (Öğrenciler) 1:n Library\_Cards (Kütüphane Kartları)

Library\_Cards (Kütüphane Kartları) 1:n Loans (Ödünç Verme)

Bu ilişki şeması, tablolar arasındaki

ilişkileri göstermektedir. Örneğin, "Books" tablosu "Authors", "Genres" ve "Publishers" tablolarına bağlıdır. "Loans" tablosu ise "Books" ve "Students" tablolarına bağlıdır. Aynı şekilde, diğer tablolar arasında da ilişkiler bulunmaktadır. İlişki şeması, veritabanının yapısını anlamak ve tablolar arasındaki ilişkileri görselleştirmek için kullanılabilir.

**Tabloların yapısına dayanarak sorguların detaylı açıklamalarını aşağıda bulabilirsiniz:**

1. Tüm kitapları kütüphaneden alın:

- `SELECT \* FROM Books;`

- Bu sorgu, Books tablosundaki tüm verileri döndürür.

2. Mevcut kopyası olan tüm kitapları alın:

- `SELECT \* FROM Books WHERE available\_copies > 5;`

- Bu sorgu, available\_copies sütunu değeri 5'dan büyük olan tüm kitapları döndürür.

3. Belirli bir yılda yayımlanan tüm kitapları alın:

- `SELECT \* FROM Books WHERE publication\_year = <year>;`

- Bu sorgu, belirli bir yılda (year) yayımlanan tüm kitapları döndürür.

4. Belirli bir türdeki tüm kitapları alın:

- `SELECT \* FROM Books WHERE genre\_id = <genre\_id>;`

- Bu sorgu, belirli bir genre\_id'ye sahip olan tüm kitapları döndürür.

5. Belirli bir öğrenci tarafından ödünç alınan tüm kitapları alın:

- `SELECT \* FROM Books JOIN Loans ON Books.book\_id = Loans.book\_id WHERE Loans.student\_id = <student\_id>;`

- Bu sorgu, belirli bir öğrenci tarafından ödünç alınan tüm kitapları döndürür. Kitaplar tablosu ile Loans tablosu arasında book\_id alanı kullanılarak birleştirme (JOIN) yapılmıştır.

6. Şu anda bir kitap ödünç alan tüm öğrencileri alın:

- `SELECT \* FROM Students JOIN Loans ON Students.student\_id = Loans.student\_id WHERE Loans.return\_date IS NULL;`

- Bu sorgu, şu anda bir kitap ödünç alan tüm öğrencileri döndürür. Students tablosu ile Loans tablosu arasında student\_id alanı kullanılarak birleştirme (JOIN) yapılmıştır. Return\_date alanının NULL olması, kitabın henüz iade edilmediğini gösterir.

7. Belirli bir kitabın mevcut kopya sayısını alın:

- `SELECT available\_copies FROM Books WHERE book\_id = <book\_id>;`

- Bu sorgu, belirli bir book\_id'ye sahip olan kitabın available\_copies değerini döndürür.

8. Belirli bir öğrenci tarafından ödünç alınan kitap sayısını alın:

- `SELECT COUNT(\*) FROM Loans WHERE student\_id = <student\_id>;`

- Bu sorgu, belirli bir öğrenci tarafından ödünç alınan kitapların sayısını döndürür. Loans tablosunda student\_id alanı <student\_id>'ye eşit olan kayıtların sayısı hesaplanır.

9. Belirli bir tarih aralığındaki ödünç alınan kitap sayısını alın:

- `SELECT COUNT(\*) FROM Loans WHERE loan\_date BETWEEN <start\_date> AND <end\_date>;`

- Bu sorgu, belirli bir tarih aralığında (start\_date ve end\_date arasında) ödünç alınan kitapların sayısını döndürür.

10. Belirli bir öğrenci için oluşturulan tüm raporları alın:

- `SELECT \* FROM Reports WHERE student\_id = <student\_id>;`

- Bu sorgu, belirli bir öğrenci için oluşturulan tüm raporları döndürür. Reports tablosunda student\_id alanı <student\_id>'ye eşit olan kayıtlar seçilir.

11. Belirli bir türü yöneten tüm personeli alın:

- `SELECT \* FROM Staff WHERE genre\_id = <genre\_id>;`

- Bu sorgu, belirli bir genre\_id'ye sahip olan türü yöneten tüm personeli döndürür. Staff tablosunda genre\_id alanı <genre\_id>'ye eşit olan kayıtlar seçilir.