

Kütüphane Veri Tabanı

Projede çalışanlar:

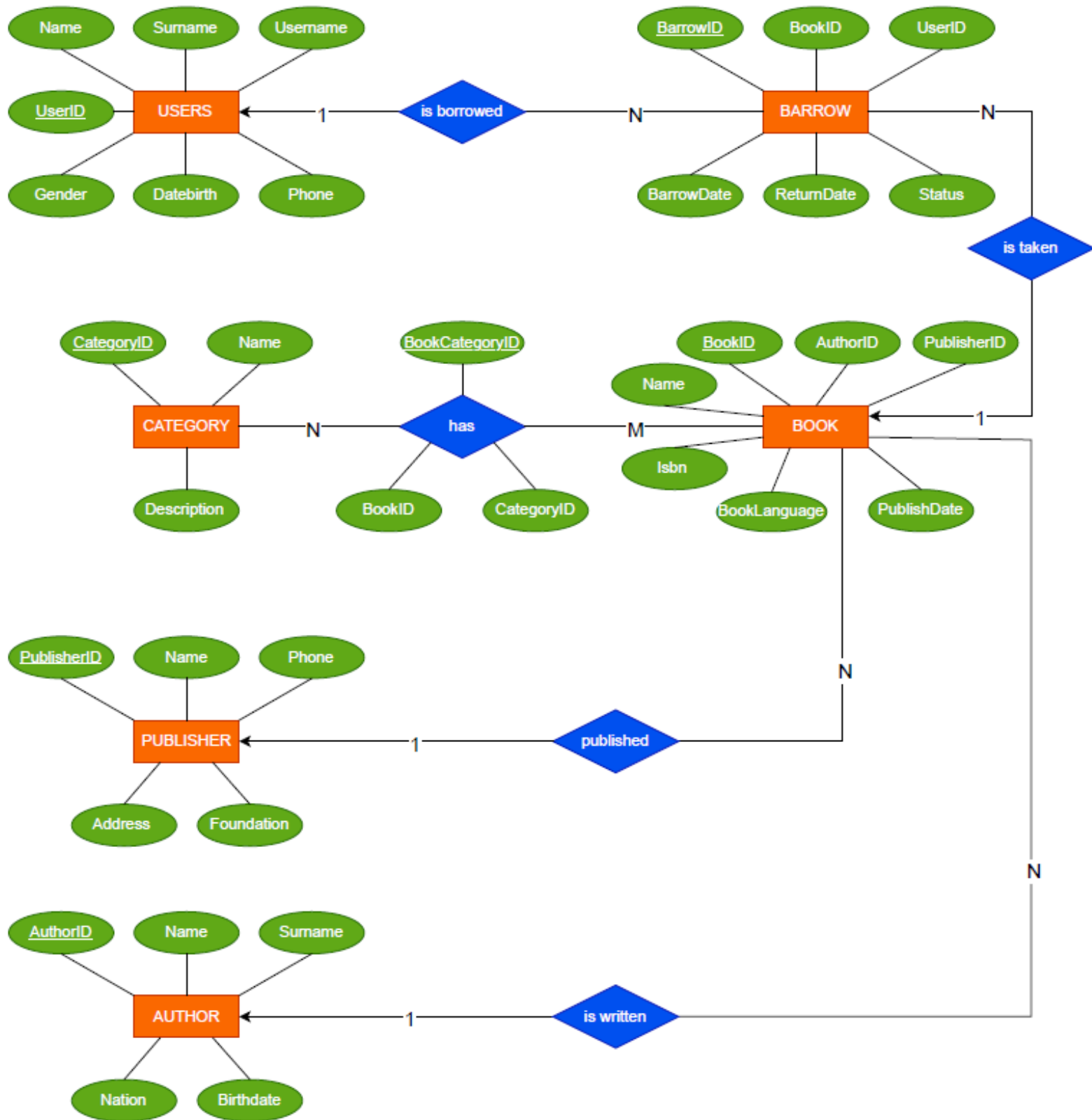
- Berkant Karaca - 2111502056 - 2.Sınıf (TR)
- Ahmet Caner Aybek - 2111502057 - 2.Sınıf (TR)
- Ulaş Kartal - 201502049 - 2.Sınıf (TR)

Projenin Amaçları:

Bir kütüphanenin yönetimi ve bilgi erişimi ile ilgili çeşitli problemleri çözmek için tasarlanır. Örnekler:

- Veri tabanındaki bilgilerin etkili bir şekilde yöneterek kütüphanenin envanteri ve kaynaklarının takibi daha kolay hale gelir.
- Kitapları kataloglandırarak kullanıcıların kitapları isimlerine, yazarlarına veya diğer kriterlere göre daha kolay bulmalarını sağlar.
- Kullanıcıların kitapları ödünç alabilmesi, iade tarihlerini takip edebilmesi ve gecikmeleri önleyebilmesi için kullanılabilir.
- Kütüphane kullanımıyla ilgili istatistiksel verileri toplayabilir ve raporlar oluşturabilir. En çok ödünç alınan kitaplar, popüler yazarlar veya kütüphane kaynaklarının kullanımı gibi bilgilere erişebilir.

ER DİYAGRAMI



NORMALİZASYON

- Normal Form (1NF): Her tablodaki tüm alanlar atomik değere sahip olmalı ve her satır benzersiz bir tanımlayıcıya sahip birincil anahtarla tanımlanır.
- Normal Form (2NF): Tabloda herhangi bir kısmi bağımlılık olmaması gerekir. Kısmi bağımlılık, birincil anahtarın yalnızca bir kısmına bağlı olan bir alanın bulunması durumunda ortaya çıkar.
- Normal Form (3NF): Tabloda herhangi bir geçişken bağımlılığı olmamalıdır. Geçişken bağımlılık, birincil anahtarın olmayan bir alan tarafından belirlenen bir bağımlılık durumudur.

AUTHOR TABLOSUNUN NORMALİZASYONU

- 1NF kuralına tüm veri değerleri atomik olduğu için uygundur.
- 2NF kuralına tabloda kısmi bağımlılık olmadığı için uygundur.
- 3NF kuralına tabloda geçişken bağımlılık olmadığı için uygundur.

BARROW TABLOSUNUN NORMALİZASYONU

- 1NF kuralına tüm veri değerleri atomik olduğu için uygundur.
- 2NF kuralına tabloda kısmi bağımlılık olmadığı için uygundur.
- 3NF kuralına tabloda geçişken bağımlılık olmadığı için uygundur.

BOOK TABLOSUNUN NORMALİZASYONU

- 1NF kuralına tüm veri değerleri atomik olduğu için uygundur.
- 2NF kuralına tabloda kısmi bağımlılık olmadığı için uygundur.
- 3NF kuralına tabloda geçişken bağımlılık olmadığı için uygundur.

BOOKCATEGORY TABLOSUNUN NORMALİZASYONU

- 1NF kuralına tüm veri değerleri atomik olduğu için uygundur.
- 2NF kuralına tabloda kısmi bağımlılık olmadığı için uygundur.
- 3NF kuralına tabloda geçişken bağımlılık olmadığı için uygundur.

CATEGORY TABLOSUNUN NORMALİZASYONU

- 1NF kuralına tüm veri değerleri atomik olduğu için uygundur.
- 2NF kuralına tabloda kısmi bağımlılık olmadığı için uygundur.
- 3NF kuralına tabloda geçişken bağımlılık olmadığı için uygundur.

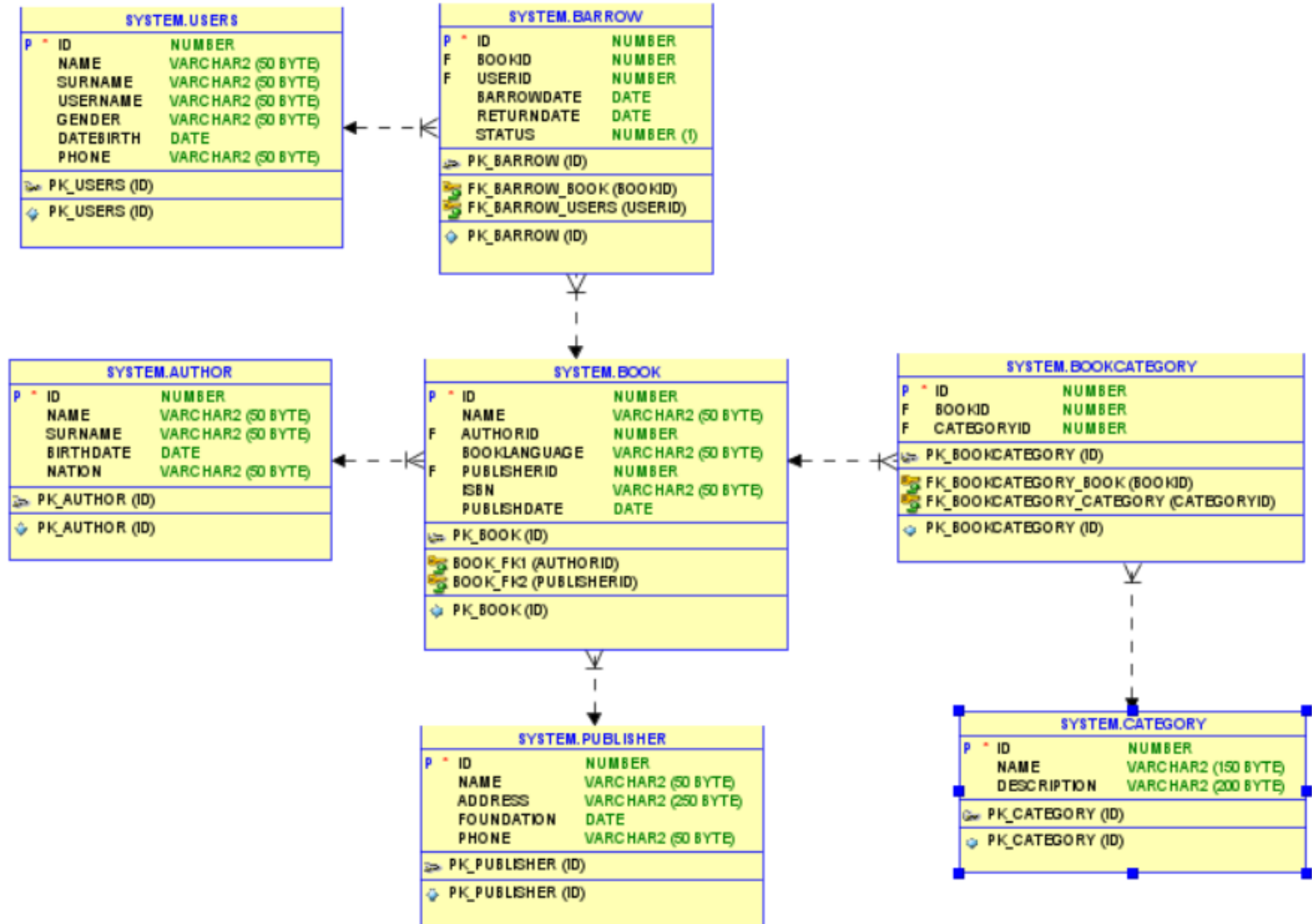
PUBLISHER TABLOSUNUN NORMALİZASYONU

- 1NF kuralına tüm veri değerleri atomik olduğu için uygundur.
- 2NF kuralına tabloda kısmi bağımlılık olmadığı için uygundur.
- 3NF kuralına tabloda geçişken bağımlılık olmadığı için uygundur.

USERS TABLOSUNUN NORMALİZASYONU

- 1NF kuralına tüm veri değerleri atomik olduğu için uygundur.
- 2NF kuralına tabloda kısmi bağımlılık olmadığı için uygundur.
- 3NF kuralına tabloda geçişken bağımlılık olmadığı için uygundur.

İLİŞKİ ŞEMASI



ÖRNEK SORGULAR

1.SORGU: Bir kitabın hangi kategoriye ait olduğunu veren sorgu.

```
SELECT b.Name as BookName, c.Name as CategoryName
FROM BookCategory n
JOIN Book b ON n.BookID = b.ID
JOIN Category c ON n.CategoryID= c.ID
WHERE b.Name = 'Pride and Prejudice';
```

AÇIKLAMA:

Sorgunun temel yapısı, "BookCategory" adlı bir tablo ile "Book" ve "Category" adlı diğer iki tabloyu birleştirmektir. "BookCategory" tablosu, kitapların kategorilerini eşleştiren bir ara tablo olarak kullanılmaktadır.

İlk JOIN ifadesi, "BookCategory" tablosunu "Book" tablosuyla "BookID" sütunu üzerinden birleştirir. Bu birleştirme, her bir kitap kaydını ilgili kitabın kimliği ile eşleştirir.

İkinci JOIN ifadesi, "BookCategory" tablosunu "Category" tablosuyla "CategoryID" sütunu üzerinden birleştirir. Bu birleştirme, her bir kategori kaydını ilgili kategorinin kimliği ile eşleştirir.

WHERE ifadesi, "Book" tablosunda "Name" sütununun "Pride and Prejudice" olduğu kitabı filtrelemek için kullanılır.

2.SORGU: Bir kategoriye ait tüm kitapları listeleyen sorgu.

```
SELECT b.Name as BookName
FROM BookCategory ba
JOIN Book b ON ba.BookID = b.ID
JOIN Category c ON ba.CategoryID = c. ID
WHERE c.Name = 'Mystery';
```

AÇIKLAMA:

Sorgunun temel yapısı, "BookCategory" adlı bir tabloyu "Book" ve "Category" adlı diğer iki tablo ile birleştirmektir. "BookCategory" tablosu, kitapların kategorilerini eşleştiren bir ara tablo olarak kullanılmaktadır.

İlk JOIN ifadesi, "BookCategory" tablosunu "Book" tablosuyla "BookID" sütunu üzerinden birleştirir. Bu birleştirme, her bir kitap kaydını ilgili kitabın kimliği ile eşleştirir.

İkinci JOIN ifadesi, "BookCategory" tablosunu "Category" tablosuyla "CategoryID" sütunu üzerinden birleştirir. Bu birleştirme, her bir kategori kaydını ilgili kategorinin kimliği ile eşleştirir.

WHERE ifadesi, "Category" tablosunda "Name" sütununun "Mystery" olduğu kategoriye filtrelemek için kullanılır.

3.SORGU: En çok kitap içeren kategori ve bu kategorideki kitap sayısını veren sorgu.

```
SELECT c.Name AS Category, COUNT(b.ID) AS BookCount
FROM Category c
JOIN BookCategory n ON c.ID = n.CategoryID
JOIN Book b ON n.BookID = b.ID
GROUP BY c.Name
HAVING COUNT(b.ID) = (
    SELECT MAX(book_count)
    FROM(
        SELECT COUNT(b.ID) AS book_count
        FROM Category c
        JOIN BookCategory n ON c.ID = n.CategoryID
        JOIN Book b ON n.BookID = b.ID
        GROUP BY c.ID)
    );
```

AÇIKLAMA:

Sorgunun temel yapısı, "Category" adlı bir tabloyu "BookCategory" ve "Book" tablolarıyla birleştirmektedir. İlk JOIN ifadesi, "Category" tablosunu "BookCategory" tablosuyla "CategoryID" sütunu üzerinden birleştirir. Bu birleştirme, her bir kategori kaydını ilgili kategorinin kimliği ile eşleştirir.

İkinci JOIN ifadesi, "BookCategory" tablosunu "Book" tablosuyla "BookID" sütunu üzerinden birleştirir. Bu birleştirme, her bir kitap kaydını ilgili kitabın kimliği ile eşleştirir. GROUP BY ifadesi, sonuçları kategori adına göre gruplandırır. COUNT(b.ID) ifadesi, her bir kategori için kitap sayısını hesaplar.

HAVING ifadesi, gruplanmış sonuçları filtrelemek için kullanılır. COUNT(b.ID) = (SELECT MAX(book_count)...) ifadesi, kitap sayısı en fazla olan kategoriyi bulmak için bir alt sorgu kullanır. Alt sorgu, her kategori için kitap sayısını hesaplar ve bunların içinden en yüksek değeri seçer.

4.SORGU: Bir kullanıcının ödünç aldığı kitapları veren sorgu.

```
SELECT b.Name, bc.BorrowDate, bc.ReturnDate
FROM BORROW bc
JOIN Book b ON bc.BookID = b.ID
JOIN Users u ON bc.UserID = u.ID
WHERE u.Username = 'sofiascott22';
```

AÇIKLAMA:

Sorgunun temel yapısı, "BORROW" adlı bir tabloyu "Book" ve "Users" tablolarıyla birleştirmektedir. İlk JOIN ifadesi, "BORROW" tablosunu "Book" tablosuyla "BookID" sütunu üzerinden birleştirir. Bu birleştirme, her bir ödünç alma kaydını ilgili kitabın kimliği ile eşleştirir.

İkinci JOIN ifadesi, "BARROW" tablosunu "Users" tablosuyla "UserID" sütunu üzerinden birleştirir. Bu birleştirme, her bir ödünç alma kaydını ilgili kullanıcının kimliği ile eşleştirir. WHERE ifadesi, "Users" tablosunda "Username" sütununun "sofiascott22" olduğu kullanıcıyı filtrelemek için kullanılır.

5.SORGU: Ödünç alınan kitapların cinsiyete göre dağılımını veren sorgu.

```
SELECT u.Gender, COUNT(*) AS BorrowCount
FROM User u
JOIN Barrow br ON u.ID = br.UserID
GROUP BY u.Gender;
```

AÇIKLAMA:

Sorgunun temel yapısı, "User" adlı bir tabloyu "Barrow" tablosuyla birleştirmektedir. JOIN ifadesi, "User" tablosunu "Barrow" tablosuyla "ID" sütunu üzerinden birleştirir. Bu birleştirme, her bir ödünç alma kaydını ilgili kullanıcının kimliği ile eşleştirir.

GROUP BY ifadesi, sonuçları cinsiyetlerine göre gruplandırır. COUNT(*) ifadesi, her bir cinsiyet için ödünç alma sayısını hesaplar.

6.SORGU: Bir yazarın yazdığı kitapları ve yayın tarihlerini getiren sorgu.

```
SELECT a.Name, a.Surname, b.Name AS BookTitle, b.PublishDate
FROM Author a
JOIN Book b ON a.ID = b.AuthorID;
```

AÇIKLAMA:

Sorgunun temel yapısı, "Author" adlı bir tabloyu "Book" tablosuyla birleştirmektedir. JOIN ifadesi, "Author" tablosunu "Book" tablosuyla "ID" sütunu üzerinden birleştirir. Bu birleştirme, her bir yazar kaydını ilgili kitabın yazar kimliği ile eşleştirir.

7.SORGU: Kullanıcının adı, ödünç aldığı kitapları ve ödünç alma tarihini gösteren sorgu.

```
SELECT u.Name AS UserName, b.Name AS BookName, br.BorrowDate
FROM User u
JOIN Barrow br ON u.ID = br.UserID
JOIN Book b ON br.BookID = b.ID;
```

AÇIKLAMA:

Sorgunun temel yapısı, "User" adlı bir tabloyu "Barrow" tablosuyla ve "Book" tablosuyla birleştirmektedir.

İlk JOIN ifadesi, "User" tablosunu "Barrow" tablosuyla "ID" sütunu üzerinden birleştirir. Bu birleştirme, her bir ödünç alma kaydını ilgili kullanıcının kimliği ile eşleştirir.

İkinci JOIN ifadesi, "Barrow" tablosunu "Book" tablosuyla "BookID" sütunu üzerinden birleştirir. Bu birleştirme, her bir ödünç alma kaydını ilgili kitabın kimliği ile eşleştirir.

Sonuç olarak, sorgu kullanıcının adını, ödünç alınan kitabın adını ve ödünç alınma tarihini döndürür. Sonuçlar "UserName", "BookName" ve "BarrowDate" olarak adlandırılan üç sütunda listelenir.

8.SORGU: Bir kitabın adı, kategorisini ve açıklamasını getiren sorgu.

```
SELECT b.Name AS BookName, c.Name AS CategoryName, c.Description
FROM BOOK b
INNER JOIN BookCategory bc ON b.ID = bc.BookID
INNER JOIN CATEGORY c ON bc.CategoryID = c.ID
WHERE b.Name = '1984';
```

AÇIKLAMA:

Sorgunun temel yapısı, "Book" adlı bir tabloyu "BookCategory" ve "Category" adlı diğer iki tablo ile birleştirmektedir.

İlk INNER JOIN ifadesi, "Book" tablosunu "BookCategory" tablosuyla "ID" sütunu üzerinden birleştirir. Bu birleştirme, her bir kitap kaydını ilgili kitabın kimliği ile eşleştirir.

İkinci INNER JOIN ifadesi, "BookCategory" tablosunu "Category" tablosuyla "CategoryID" sütunu üzerinden birleştirir. Bu birleştirme, her bir kategori kaydını ilgili kategorinin kimliği ile eşleştirir.

WHERE ifadesi, "Book" tablosunda "Name" sütununun "1984" olduğu kitabı filtrelemek için kullanılır.

Sonuç olarak, sorgu "1984" adlı kitabın adını, kategorisini ve kategori açıklamasını döndürür. Sonuçlar "BookName", "CategoryName" ve "Description" olarak adlandırılan üç sütunda listelenir.

9.SORGU: Bir yayınevinin kaç tane kitap yayınladığını getiren sorgu.

```
SELECT p.Name AS PublisherName, COUNT(b.ID) AS PublishedBooksCount
FROM PUBLISHER p
LEFT JOIN BOOK b ON p.ID = b.PublisherID
GROUP BY p.Name;
```

AÇIKLAMA:

Sorgunun temel yapısı, "Publisher" adlı bir tabloyu "Book" tablosuyla left join kullanarak birleştirmektedir.

Left join, "Publisher" tablosundaki her bir yayınevi kaydını ilgili kitapları içeren "Book" tablosuyla yayınevi kimliği "PublisherID" üzerinden eşleştirir.

GROUP BY ifadesi, sonuçları yayınevi adına göre gruplandırır. COUNT(b.ID) ifadesi, her bir yayınevi için yayımlanan kitap sayısını hesaplar.

Sonuç olarak, sorgu her bir yayınevinin adını ve yayınladığı kitap sayısını döndürür. Sonuçlar "PublisherName" ve "PublishedBooksCount" olarak adlandırılan iki sütunda listelenir.

10.SORGU: Bir kategoride toplam kaç farklı kitap olduğunu gösteren sorgu.

```
SELECT c.Name AS CategoryName, COUNT(DISTINCT b.ID) AS TotalBooks
FROM CATEGORY c
LEFT JOIN BookCategory bc ON c.ID = bc.CategoryID
LEFT JOIN Book b ON bc.BookID = b.ID
GROUP BY c.ID, c.Name;
```

AÇIKLAMA:

Sorgunun temel yapısı, "Category" adlı bir tabloyu "BookCategory" ve "Book" adlı diğer iki tablo ile sol birleştirme (LEFT JOIN) kullanarak birleştirmektedir.

Sol birleştirme, "Category" tablosundaki her bir kategori kaydını ilgili kitap kategorilerini içeren "BookCategory" tablosuyla kategori kimliği (CategoryID) üzerinden eşleştirir.

Ardından, "BookCategory" tablosunu "Book" tablosuyla sol birleştirme kullanarak kitapları ekler. Bu sayede, her bir kategoriye ait kitapları içeren tüm kayıtları alır.

GROUP BY ifadesi, sonuçları kategori kimliği (ID) ve kategori adına göre gruplandırır. COUNT(DISTINCT b.ID) ifadesi, her bir kategori için benzersiz kitap sayısını hesaplar.

Sonuç olarak, sorgu her bir kategori için kategori adını ve o kategoriye ait benzersiz kitap sayısını döndürür. Sonuçlar "CategoryName" ve "TotalBooks" olarak adlandırılan iki sütunda listelenir.