

ÖĞRENCİ DERS SEÇİM SİSTEMİ VERİ TABANI PROJESİ

Bu proje kapsamında bir Öğrenci Ders Seçim Sistemi oluşturulmuştur. Proje beş adımda incelenmiştir:

- 1) Veri Tabanı Oluşturma (Tablo Oluşturma)
- 2) Veri Tabanına Giriş Verilmesi
- 3) Procedure
- 4) Trigger
- 5) View
- 6) Sorgular

1. VERİ TABANI OLUŞTURMA

Bu adımda, veri tabanı kapsamında beş farklı tablo oluşturulmuştur. Bu tabloların başında "Students" tablosu gelmektedir. Bu tablo, öğrencilerin temel bilgilerini saklamak için kullanılır.

a) Students Tablosu:

Amaç: Öğrencilerin kimlik bilgilerini ve iletişim bilgilerini saklamak.

Sütunlar:

- **StudentID (Primary Key):** Öğrenci ID'leri, her bir öğrenciyi benzersiz bir şekilde tanımlayan anahtar sütundur. Bu sütun sıralı ve benzersiz olacak şekilde yapılandırılmıştır.
- **FirstName:** Öğrencinin adı.
- **LastName:** Öğrencinin soyadı.
- **Email:** Öğrencinin email adresi.
- **DateOfBirth:** Öğrencinin doğum tarihi.

Bu tablo sayesinde, öğrenci bilgilerini düzenli bir şekilde saklamamıza ve ihtiyaç duyulduğunda kolayca erişebilmemize olanak tanır.

StudentID	FirstName	LastName	Email	BirthDate
1	Rabia	Kaşıkcı	rabia.kaşıkcı...	1998-03-25
2	Ömer	Özütemiz	ömer.özüte...	1997-01-15
3	Gülnaz	Temiz	gülnaz.temi...	1996-06-08
4	Uğur	Yirmibeş	uğur.yirmib...	1996-04-12
5	Zeynep	Karay	zeynep.kara...	2001-05-27
6	Ali	Yılmaz	ali.yilmaz@...	2000-01-01
7	Ayşe	Kara	ayse.kara@...	1999-12-02
8	Tahir	Onur	tahir.onur@...	1999-09-05
9	Selma	Ay	selma.ay@e...	2000-10-18
10	Süha	Kaşıkcı	süha.kaşıkcı...	2001-11-06
11	Tuğba	Sayat	tugba.sayat...	2000-07-05

Resim 1. Öğrenciler Tablosu

b) Courses Tablosu:

Amaç: Derslerin kimlik bilgilerini ve kontenjan bilgilerini saklamak.

Sütunlar:

- **CourseID (Primary Key):** Her bir dersi benzersiz bir şekilde tanımlayan anahtar sütundur. Bu sütun sıralı ve benzersiz olacak şekilde yapılandırılmıştır.
- **CourseName:** Dersin adı.
- **Credits:** Dersin kredi değeri.
- **AvailableSeats:** Dersin mevcut kontenjanı.

CourseID	CourseName	Credits	AvailableS...
1	Matematik	4	1741
2	Fizik	3	1802
3	Biyoloji	3	1783
4	Tarih	3	1795
5	Kimya	3	1689
6	İngilizce	3	1797
7	Müzik	2	1768
8	Edebiyat	3	1758
9	Coğrafya	3	1792
10	Sanat	2	1784
11	Sosyal Bilgil...	2	1738
12	Bilgisayar Bi...	3	1728
13	Psikoloji	3	1770
14	Ekonomi	3	1824
15	Felsefe	2	1725
16	Sosyoloji	2	1870
17	Astronomi	2	1736

Resim 2. Dersler Tablosu

c) Student Courses Tablosu:

Amaç: Öğrencilerin hangi dersleri aldığını ve kayıt tarihlerini saklamak.

Sütunlar:

- **StudentCourseID (Primary Key):** Her bir öğrenci-ders kaydını benzersiz bir şekilde tanımlayan anahtar sütundur. Bu sütun sıralı ve benzersiz olacak şekilde yapılandırılmıştır.
- **StudentID:** Öğrencinin kimlik numarasını belirten alandır. "Students" tablosundaki StudentID sütununa yabancı anahtar (foreign key) olarak bağlanmıştır.
- **CourseID:** Dersin kimlik numarasını belirten alandır. "Courses" tablosundaki CourseID sütununa yabancı anahtar (foreign key) olarak bağlanmıştır.
- **EnrollmentDate:** Öğrencinin derse kaydolduğu tarihi belirten alandır.

StudentCo...	StudentID	CourselD	Enrollment...
1	21063	17	2023-02-24
2	42484	1	2022-09-02
3	27516	17	2022-04-12
4	53020	14	2022-10-06
5	30203	4	2021-11-15
6	37204	8	2023-09-28
7	63440	2	2020-02-06
8	13804	13	2021-02-23
9	69693	5	2022-04-20
10	65106	5	2023-04-17
11	19502	5	2022-10-06
12	50679	4	2021-10-30
13	16401	4	2021-04-23
14	21856	12	2023-03-29
15	47717	13	2022-10-22
16	48803	12	2021-03-11
17	14060	14	2020-10-12
18	69520	11	2023-08-15
19	56532	16	2022-07-24
20	52410	9	2020-11-08

Resim 3. Öğrencilerin Dersleri Tablosu

d) Instructors Tablosu:

Amaç: Öğretmenlerin kimlik bilgilerini ve iletişim bilgilerini saklamak.

Sütunlar:

- **InstructorID (Primary Key):** Her bir öğretmeni benzersiz bir şekilde tanımlayan anahtar sütundur. Bu sütun sıralı ve benzersiz olacak şekilde yapılandırılmıştır.
- **FirstName:** Öğretmenin adı.
- **LastName:** Öğretmenin soyadı.
- **Email:** Öğretmenin email adresi.

InstructorID	FirstName	LastName	Email
1	Mehmet	Demir	mehmet.demir@example.com
2	Fatma	Çelik	fatma.celik@example.com
3	Ahmet	Yılmaz	ahmet.yilmaz@example.com
4	Ayşe	Kaya	ayse.kaya@example.com
5	Mustafa	Yıldız	mustafa.yildiz@example.com
6	Zeynep	Şahin	zeynep.sahin@example.com
7	Ali	Koç	ali.koc@example.com
8	Aylin	Öztürk	aylin.ozturk@example.com
9	Hasan	Sarı	hasan.sari@example.com
10	Gülây	Erdoğan	gulay.erdogan@example.com
11	Osman	Aydın	osman.aydin@example.com
12	Sema	Turan	sema.turan@example.com
13	Emre	Kara	emre.kara@example.com
14	Zehra	Yılmaz	zehra.yilmaz@example.com
15	Deniz	Tekin	deniz.tekin@example.com

Resim 3. Instructors Tablosu

e) Course Assignments Tablosu:

Amaç: Derslerin hangi öğretmenler tarafından verildiğini ve atamaların takibini sağlamak.

Sütunlar:

- **CourseAssignmentID (Primary Key):** Her bir ders atamasını benzersiz bir şekilde tanımlayan anahtar sütundur. Bu sütun sıralı ve benzersiz olacak şekilde yapılandırılmıştır.
- **CourseID:** Dersin kimlik numarasını belirten alandır. "Courses" tablosundaki CourseID sütununa yabancı anahtar (foreign key) olarak bağlanmıştır.
- **InstructorID:** Öğretmenin kimlik numarasını belirten alandır. "Instructors" tablosundaki InstructorID sütununa yabancı anahtar (foreign key) olarak bağlanmıştır.

CourseAssi...	CourseID	InstructorID
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13
14	14	14
15	15	15
16	16	16
17	17	17
18	1	18
19	2	19
20	3	20

Resim 4. Course Assignments Tablosu

2. VERİ TABANINA GİRİŞ VERİLMESİ

a) Student Tablosuna Veri Ekleme

Öğrenci verilerinin tablosuna rastgele veri ekleyerek, veri tabanı üzerinde yapılacak işlemlerin gerçek dünya senaryolarına uygunluğunu sağlamak.

Öncelikle 11 adet önceden tanımlanmış öğrenci verisi "Students" tablosuna eklenmiştir. Daha sonra 70,000 adet rastgele öğrenci verisi oluşturulup "Students" tablosuna eklenmiştir

- b) Courses Tablosuna Veri Ekleme
Öğrenci Ders Seçim Sistemi'nde yer alacak derslerin temel bilgilerini veri tabanında tutmak.
Insert into kullanılarak 17 farklı ders sisteme eklenmiştir.
- c) Student Courses Tablosuna Veri Ekleme
Öğrencilerin derslere rastgele atanarak, ders kayıt ve yönetim süreçlerinin simüle edilmesi amaçlanmıştır.
While döngüsü kullanarak 30000 öğrenci derslere rastgele atanmıştır.
- d) Instructors Tablosuna Veri Ekleme
Öğrenci Ders Seçim Sistemi'nde yer alacak öğretmenlerin temel bilgilerini veritabanına işlenmiştir.
Bunun için insert into kullanılmıştır ve 27 farklı öğretmen sisteme eklenmiştir.
- e) Course Assignments Tablosuna Veri Ekleme
Öğretmenler derslere while döngüsü ile sırası ile atanmıştır.

3. PROCEDURE

Proje kapsamında geliştirilen prosedür, bir öğrenciyi belirli bir ders için kaydetmektir. Projenin genel amacına hizmet ederken, veri tabanında öğrenci-ders ilişkilerini oluşturur ve günceller.

Örnek kullanım;

```
exec sp_EnrollStudent @StudentID = 1, @CourseID = 5;
```

Yukarıdaki örnek, öğrenci kimliği 1 olan bir öğrenciyi 5 numaralı bir ders için kaydeder ve öğrencinin toplam kayıt sayısını döndürür.

Bu prosedür, ders kayıtlarının veri tabanında güvenli ve düzenli bir şekilde yapılmasını sağlar. Aynı zamanda, öğrenci kaydı yapıldıktan sonra ilgili bilgileri geri döndürerek işlemin başarılı olup olmadığını kontrol etmek için kullanılabilir.

Örnek çıktı ;

TotalEnrollmentsforStudent	StudentIDNumber
4	1

Resim 5. Procedure Çıktısı

Öğrenci ID'si 1 olan öğrenci, ID'si 5 olan derse kaydolmuştur ve toplam 4 derse kaydı vardır.

4. TRIGGER

Bu triggerın amacı bir öğrencinin ders kaydı yapıldıktan sonra belirli bir dersin mevcut kontenjanını kontrol etmektir. Eğer kontenjan dolmuşsa, kaydı geri alır ve bir hata mesajı gönderir. Eğer kontenjan dolu değilse, dersin mevcut kontenjanı bir azaltılır ve işlem devam eder.

Örnek çıktı;

CourseID	CourseName	Credits	AvailableSeats
5	Kimya	3	1687

TotalEnrollmentsforStudent	StudentIDNumber
1	2

CourseID	CourseName	Credits	AvailableSeats
5	Kimya	3	1686

Resim 6. Trigger Çıktısı

Burada da görüldüğü gibi ders kaydı yapıldıktan sonra kontenjan 1 azalmıştır.

5. VIEW

Bu view amacı, her bir dersin toplam kayıt sayısını göstermektir. Böylece, herhangi bir dersin kaç öğrenci tarafından alındığını görselleştirmek ve analiz etmek mümkün olur.

Örnek çıktı;

CourseName	TotalEnrollmentforCourses
Astronomi	1736
Bilgisayar Bilimleri	1728
Biyoloji	1783
Coğrafya	1793
Edebiyat	1758
Ekonomi	1824
Felsefe	1725
Fizik	1803

Resim 7. View Çıktısı

6. SORGULAR

- Belirli bir dersi alan öğrencilerin listesini veren sorgu

Örneğin burada ID'si 6 olan dersi alan öğrencilerin listesini, veren sorgu ve çıktısı bu şekildedir.

```
select s.StudentID,s.FirstName, s.LastName
from Students s
join StudentCourses sc on s.StudentID = sc.StudentID
where sc.CourseID = 6;
```

StudentID	FirstName	LastName
112	Tahir	Ay
122	Tuğba	Temiz
126	Zeynep	Sayat
130	Ayşe	Kaşıkcı
135	Tahir	Yılmaz
235	Rabia	Yılmaz
254	Rabia	Yılmaz
345	Ömer	Yılmaz
382	Uğur	Ay
427	Uğur	Kaşıkcı
460	Tahir	Kaşıkcı
498	Tahir	Temiz

Resim 8.Sorgu çıktısı

- Belirli bir öğrencinin aldığı dersleri gösteren sorgu

Örneğin burada ID'si 11 olan öğrencinin aldığı dersleri göstermesi istenmiştir.

```
select c.CourseID , c.CourseName from Courses c
join StudentCourses sc on c.CourseID = sc.CourseID
where sc.StudentID = 11
```

CourseID	CourseName
2	Fizik
5	Kimya

Resim 9.Sorgu çıktısı

- Bir öğretmenin verdiği dersi gösteren sorgu

Örneği burada ID'si 18 olan öğretmenin verdiği dersler gösterilmiştir.

```
select c.CourseName, in.FirstName, in.LastName
from Courses c
JOIN CourseAssignments ca on c.CourseID = ca.CourseID
JOIN Instructors in ON in.InstructorID = ca.InstructorID
where ca.InstructorID = 18;
```

CourseName	FirstName	LastName
Matematik	Burak	Aydın

Resim 10.Sorgu çıktısı

- Derslerin boş kontenjan bilgisini gösteren sorgu

```
select CourseName, AvailableSeats
from Courses
where AvailableSeats > 0;
```

CourseName	AvailableSeats
Fizik	1802
Biyoloji	1783
Tarih	1795
Kimya	1682
İngilizce	1797
Müzik	1768
Edebiyat	1758
Coğrafya	1791
Sanat	1784
Sosyal Bilgiler	1738
Bilgisayar Bilimleri	1728

Resim 11.Sorgu çıktısı

- Derslere kayıtlı öğrenci sayısını gösteren sorgu

```
select c.CourseName, COUNT(sc.StudentCourseID) as TotalEnrollments
from StudentCourses sc
JOIN Courses c ON sc.CourseID = c.CourseID
group by c.CourseName;
```

CourseName	TotalEnrollments
Astronomi	1736
Bilgisayar Bilimleri	1728
Biyoloji	1783
Coğrafya	1793
Edebiyat	1758
Ekonomi	1824
Felsefe	1725
Fizik	1803
İngilizce	1797
Kimya	1697
Matematik	1742

Resim 12.Sorgu çıktısı

- Belirli tarih arasında kayıt yapan öğrenciler ve kayıt yaptıkları dersi gösteren sorgu

Örneğin burada 2023.11.01 ve 2023.11.02 tarihi arasındaki kayıt listesini verecek bir sorgu.

```
select s.StudentID, s.FirstName, s.LastName , c.CourseName
from Students s
join StudentCourses sc on s.StudentID = sc.StudentID
join Courses c on sc.CourseID = c.CourseID
where EnrollmentDate between '2023-11-01' AND '2023-11-02'
```


StudentID	FirstName	LastName	CourseName
12790	Selma	Temiz	Psikoloji
22460	Uğur	Karay	Biyoloji
57093	Zeynep	Onur	Felsefe
29619	Selma	Temiz	İngilizce
21699	Tuğba	Ay	Coğrafya
37098	Tuğba	Temiz	Bilgisayar Bilimleri
39888	Selma	Sayat	Coğrafya
2745	Uğur	Karay	İngilizce
17360	Tuğba	Temiz	Sanat
7191	Uğur	Yılmaz	Sanat
59376	Tuğba	Karay	Sanat
21109	Uğur	Yirmibeş	Edebiyat
30101	Rabia	Ay	Biyoloji
38000	Ömer	Ay	Coğrafya

Resim 13.Sorgu çıktısı

- Hiç ders kaydı olmayan öğrenciyi bulmak için sorgu

```
select s.StudentID, s.FirstName, s.LastName, COUNT(sc.CourseID) AS TotalCourses
from Students s
LEFT JOIN StudentCourses sc ON s.StudentID = sc.StudentID
group by s.StudentID, s.FirstName, s.LastName
having COUNT(sc.CourseID) = 0;
```

StudentID	FirstName	LastName	TotalCourses
3	Gülnaz	Temiz	0
4	Uğur	Yirmibeş	0
5	Zeynep	Karay	0
8	Tahir	Onur	0
9	Selma	Ay	0
10	Süha	Kaşıkcı	0
15	Tahir	Ay	0
16	Ali	Kara	0
18	Rabia	Temiz	0
19	Rabia	Karay	0
20	Tahir	Yılmaz	0
21	Zeynep	Yılmaz	0
25	Rabia	Sayat	0
26	Ali	Kaşıkcı	0
28	Rabia	Yılmaz	0

Resim 14.Sorgu çıktısı