

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



SESLİ KİTAP UYGULAMASI
BİLGİ SİSTEMİ

Hazırlayanlar

Beyda Güler - 19011010

beyda.guler@std.yildiz.edu.tr

Şeymanur Korkmaz - 20011055

seymanur.korkmaz@std.yildiz.edu.tr

Elif Sena Yılmaz - 20011040

sena.yilmaz4@std.yildiz.edu.tr

Ders / Grup: BLM3041 - Veritabanı Yönetimi / Grup:2

Ders Yürütücüsü:

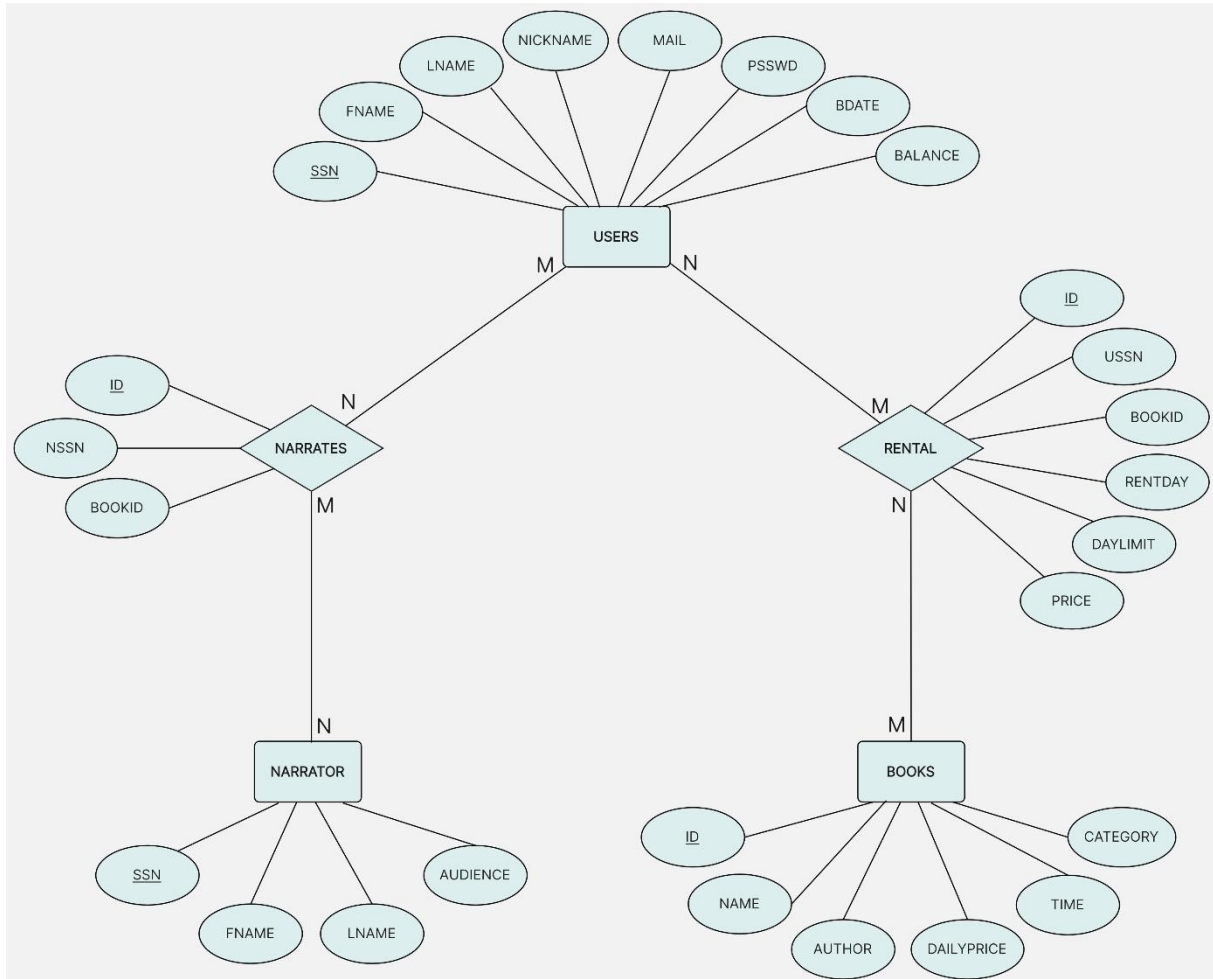
Serkan AYVAZ

Ocak, 2023

İçindekiler

Tasarlanan Veri Tabanına Ait ER Diyagramı	3
Tabloların Ekran Görüntüleri	4
users Tablosu.....	4
books Tablosu.....	4
narrator Tablosu.....	5
narrates Tablosu.....	5
rental Tablosu.....	6
Kod Blokları.....	6

Tasarlanan Veri Tabanına Ait ER Diyagramı



Tabloların Ekran Görüntüleri

users Tablosu

	ssn [PK] character varying (10)	fname character varying (15)	lname character varying (15)	nickname character varying (15)	mail character varying (30)	psswd character varying (20)	bdate date	balance integer
1	4000	Beyda	Güler	beydak	bg@gmail.com	bg123	2001-12-12	0
2	4001	Elif Sena	Yılmaz	elifsena	elylmz@gmail.com	12345	2001-02-01	0
3	4002	Şeymanur	Korkmaz	xseymal	skorkmaz@gmail.com	seyma123	2002-10-04	0
4	4003	Serkan	Ayvaz	serqan	sayvaz@gmail.com	1sayvaz1	1975-10-07	0
5	4004	Ali	Çetin	thisisceto	ceto@gmail.com	bela123	1986-07-15	0
6	4005	Merve	Öz	mervedeyim	merveler@gmail.com	gurill	1999-11-07	0
7	4006	Gupse	Cici	cicikiz	cici@gmail.com	g6h7	2000-11-28	0
8	4007	Halis	Kaya	halis1mi	kaya@gmail.com	sym33	2000-06-08	0
9	4008	Fatma	Yılmaz	fatmahnimm06	fatmylm@gmail.com	kizim	1970-10-19	0
10	4009	Özlem	Korkmaz	catlover	seyma@gmail.com	remix	1975-11-30	0

books Tablosu

	id [PK] character varying (10)	name character varying (30)	author character varying (30)	dailyprice integer	time integer	category character varying (20)
1	1000	İcatlarım	Nikola Tesla	8	179	Biography
2	1001	Denemeler	Montaigne	13	447	Nonfiction
3	1002	Bir Aşk Masalı	Ahmet Ümit	15	528	Novel
4	1003	Babamın Bavulu	Orhan Pamuk	7	103	Biography
5	1004	Gece Yarısı Kütüphanesi	Matt Haig	14	491	Novel
6	1005	Huzursuzluk	Zülfü Livaneli	10	242	Novel
7	1006	Suç ve Ceza	Fyodor Dostoyevski	20	1363	Classic
8	1007	Bir İdam Mahkumunun Son Günü	Victor Hugo	8	166	Classic
9	1008	Doğu Ekspresinde Cinayet	Agatha Cristie	14	392	Detective
10	1009	Morgue Sokağı Cinayeti	Edgar Allan Poe	10	188	Detective
11	1010	Sultanı Öldürmek	Ahmet Ümit	19	1139	Detective
12	1011	Dahi Diktatör	Celal Şengör	8	252	History
13	1012	Bunu Herkes Bilir	Emrah Safa Gürkan	15	486	History
14	1013	Uğultulu Tepeler	Emily Bronte	16	746	History
15	1014	Fatih Harbiye	Peyami Safa	7	236	Novel
16	1015	Aylak Adam	Yusuf Atılgan	6	362	Novel
17	1016	Dönüşüm	Franz Kafka	5	133	Classic
18	1017	Milena'ya Mektuplar	Franz Kafka	10	300	Nonfiction
19	1018	Yer Altından Notlar	Fyodor Dostoyevski	4	77	Drama
20	1019	Gazi Mustafa Kemal Atatürk	İlber Ortaylı	9	777	Biography

narrator Tablosu

	ssn [PK] character varying (10)	fname character varying (15)	lname character varying (15)	audience integer
1	2000	Hasan	Tanılmış	2
2	2001	Başı	Özcan	2
3	2002	Toprak	Sergen	1
4	2003	Orhan	Pamuk	2
5	2004	Sezin	Akbaşoğulları	1
6	2005	Murat	Sarı	1
7	2006	Şerif	Erol	1
8	2007	Burak	Sergen	1
9	2008	Murat	Eken	1
10	2009	Kerim	Öztürk	1
11	2010	Tolga	Korkut	1
12	2011	Serhat	Yiğit	2
13	2012	Cüneyt	Kıran	1
14	2013	Özlem	Dinsel	1
15	2014	Mazlum	Kiper	1
16	2015	Göker	Ersivri	1
17	2016	Okan	Bayülgen	1
18	2017	Mehmet	Konu	0

narrates Tablosu

	id [PK] character varying (10)	nssn character varying (10)	bookid character varying (10)
1	3000	2000	1000
2	3001	2001	1001
3	3002	2002	1002
4	3003	2003	1003
5	3004	2004	1004
6	3005	2005	1005
7	3006	2006	1006
8	3007	2007	1007
9	3008	2008	1008
10	3009	2009	1009
11	3010	2010	1010
12	3011	2011	1011
13	3012	2012	1012
14	3013	2013	1013
15	3014	2014	1014
16	3015	2015	1015
17	3016	2016	1016
18	3017	2016	1017
19	3018	2017	1018
20	3019	2014	1019

rental Tablosu

	id [PK] character varying (10)	ssn character varying (10)	bookid character varying (10)	rentday date	daylimit integer	price integer
1	5000	4000	1000	2023-01-17	1	[null]
2	5001	4000	1003	2023-01-17	8	[null]
3	5002	4000	1001	2023-01-17	6	[null]
4	5003	4000	1011	2023-01-17	6	[null]
5	5004	4001	1006	2023-01-17	5	[null]
6	5005	4001	1007	2023-01-17	5	[null]
7	5006	4001	1008	2023-01-17	3	[null]
8	5007	4001	1009	2023-01-17	7	[null]
9	5008	4001	1010	2023-01-17	4	[null]
10	5009	4001	1011	2023-01-17	2	[null]
11	5010	4002	1012	2023-01-17	7	[null]
12	5011	4002	1013	2023-01-17	5	[null]
13	5012	4002	1014	2023-01-17	4	[null]
14	5013	4002	1015	2023-01-17	2	[null]
15	5014	4002	1016	2023-01-17	2	[null]
16	5015	4003	1000	2023-01-17	3	[null]
17	5016	4003	1001	2023-01-17	4	[null]
18	5017	4003	1002	2023-01-17	5	[null]
19	5018	4003	1003	2023-01-17	4	[null]
20	5019	4003	1004	2023-01-17	5	[null]
21	5020	4003	1005	2023-01-17	7	[null]

Kod Blokları

1. Oluşturacağınız veri tabanı en az 4 tablo içermelidir. Her tabloda en az 10 kayıt bulunmalıdır.
2. Tablolarınızda primary key ve foreign key kısıtlarını kullanmalısınız.

```
CREATE TABLE users(  
    ssn varchar(10) not null primary key,  
    fname varchar(15) not null,  
    lname varchar(15) not null,  
    nickname varchar(15) not null,  
    mail varchar(30) not null,  
    psswd varchar(20) not null,  
    bdate date,  
    balance int default 0  
);  
  
CREATE TABLE narrator(  
    ssn varchar(10) not null primary key,  
    fname varchar(15) not null,  
    lname varchar(15) not null,  
    audience int default 0  
);
```

3. En az bir tabloda silme kısıtı ve sayı kısıtı olmalıdır.

```
CREATE TABLE books(  
    id varchar(10) not null primary key,  
    name varchar(30) not null,  
    author varchar(30) not null,  
    dailyPrice int not null,  
    time int not null,  
    category varchar(20) not null  
    check(time != 0);  
);  
  
CREATE TABLE narrates(  
    id varchar(10) not null primary key,  
    nssn varchar(10) not null,  
    bookid varchar(10) not null,  
    foreign key (nssn) references narrator(ssn) on delete cascade,  
    foreign key (bookid) references books(id) on delete cascade  
);  
  
CREATE TABLE rental(  
    id varchar(10) not null primary key,  
    ussn varchar(10) not null,  
    bookid varchar(10) not null,  
    rentDay date not null,  
    dayLimit int not null,  
    price int,  
    foreign key (ussn) references users(ssn) on delete cascade,  
    foreign key (bookid) references books(id) on delete cascade  
);
```

4. Arayüzden en az birer tane insert, update ve delete işlemi gerçekleştirilebilmelidir.

```
CREATE OR REPLACE FUNCTION increaseBalance(amount INTEGER, userssn VARCHAR(10))  
RETURNS void AS $$  
BEGIN  
    UPDATE users  
    SET balance = balance + amount  
    WHERE ssn = userssn;  
END;  
$$LANGUAGE 'plpgsql';
```

Update İşlemi

```
public void deleteAccount(String user_ssn) throws Exception{  
    String query = "delete from users where ssn = ?";  
    PreparedStatement pstmt = connection.prepareStatement(query);  
    pstmt.setString(1, user_ssn);  
    pstmt.executeUpdate();  
}
```

Delete İşlemi

```

public String userRegister(String fname,String lname,String userName,String mail,String psswd, java.sql.Date date1) throws Exception {

    if(searchUser(userName) == null) {
        String query2 = "INSERT INTO users (ssn, fname, lname, nickname, mail, psswd, bdate) VALUES (nextval('userseq'), ?, ?, ?, ?, ?, ?)";
        PreparedStatement stmt = connection.prepareStatement(query2);

        stmt.setString(1,fname);
        stmt.setString(2,lname);
        stmt.setString(3,userName);
        stmt.setString(4,mail);
        stmt.setString(5,psswd);
        stmt.setDate(6, date1);
        stmt.executeUpdate();

        return searchUser(userName);
    }else {
        return null;
    }
}

```

Insert işlemi

5. Arayüzden girilecek bir değere göre ekrana sonuçların listelendiği bir sorgu yazmalısınız.

```

CREATE OR REPLACE FUNCTION filterBooks(Category VARCHAR, hour INTEGER)
RETURNS book[] AS $$
DECLARE
    books book[];
    i INTEGER;
    book_cursor CURSOR FOR SELECT name, author, fname, lname, time, b.category
                           FROM books b, narrates n, narrator
                           WHERE b.id = bookid AND nssn = ssn;
BEGIN
    i := 1;
    IF Category = 'All' AND hour != 0 THEN
        FOR b IN book_cursor LOOP
            IF b.time >= (hour-1)*120 AND b.time <= hour*120 THEN
                books[i] = b;
                i = i+1;
            END IF;
        END LOOP;
    ELSIF Category != 'All' AND hour = 0 THEN
        FOR b IN book_cursor LOOP
            IF b.category = Category THEN
                books[i] = b;
                i = i+1;
            END IF;
        END LOOP;
    ELSIF Category != 'All' AND hour != 0 THEN
        FOR b IN book_cursor LOOP
            IF b.time >= (hour-1)*120 AND b.time <= hour*120 AND b.category = Category THEN
                books[i] = b;
                i = i+1;
            END IF;
        END LOOP;
    ELSIF Category = 'All' AND hour = 0 THEN
        FOR b IN book_cursor LOOP
            books[i] = b;
            i = i+1;
        END LOOP;
    END IF;
    RETURN books;
END;
$$LANGUAGE 'plpgsql';

```


6. Arayüzden çağrılan sorgulardan en az biri “view” olarak tanımlanmış olmalıdır.

```
CREATE OR REPLACE VIEW userLibrary AS
SELECT name,author,n.fname,n.lname,category,time,u.ssn
FROM books,users u,rental,narrates r,narrator n
WHERE books.id = rental.bookid AND rental.ussn = u.ssn AND books.id = r.bookid AND n.ssn=n.ssn
```

7. En az bir adet “sequence” oluşturmalı ve arayüzden yapılacak insert sırasında ilgili sütundaki değerlerin otomatik olarak atanmasını sağlamalısınız.

```
--DROP SEQUENCE userseq;
CREATE SEQUENCE userseq
INCREMENT 1
START 4000;

--DROP SEQUENCE rentalseq;
CREATE SEQUENCE rentalseq
INCREMENT 1
START 5000;
```

8. Arayüzden çağrılan sorgulardan en az birinde union veya intersect veya except kullanmış olmalısınız.

```
public ArrayList<Record> callView(String user_ssn) throws SQLException{
    ArrayList<Record> myLibrary = new ArrayList<>();
    String query = "select * from userLibrary where ssn = ? "
        + "except select name,author,fname,lname,category,time,ssn,u.id "
        + "from userLibrary u, rental where bookid = u.id AND isexpired = ?";
    PreparedStatement pstmt = connection.prepareStatement(query);
    pstmt.setString(1, user_ssn);
    pstmt.setInt(2, 1);
    ResultSet result = pstmt.executeQuery();

    while(result.next()) {
        Record rec = new Record();
        rec.bookName = result.getString(1);
        rec.author = result.getString(2);
        String fname = result.getString(3);
        String fullname = fname + " " + result.getString(4);
        rec.narrator = fullname;
        rec.category = result.getString(5);
        rec.time = result.getInt(6);
        myLibrary.add(rec);
    }

    return myLibrary;
}
```

9. Sorgularınızın en az biri aggregate fonksiyonlar içermeli, having ifadesi kullanılmalıdır.

```
CREATE OR REPLACE FUNCTION favCategory(user_ssn varchar(15),OUT num int,OUT outCat varchar(15))
AS $$
DECLARE
i int := 0;
BEGIN
    SELECT count(category),category INTO num,outCat
    FROM users,books,rental
    WHERE users.ssn = rental.usn AND rental.bookid = books.id
    GROUP BY category,ssn
    HAVING ssn = user_ssn
    ORDER BY count(category) DESC
    LIMIT 1;
END;
$$LANGUAGE 'plpgsql';
```

10. Arayüzden girilen değerleri parametre olarak alıp ekrana sonuç döndüren 3 farklı SQL fonksiyonu tanımlamış olmalısınız. Bu fonksiyonların en az birinde “record” ve “cursor” tanımı-kullanımı olmalıdır.

```
CREATE TYPE book AS (bookName VARCHAR(30), author VARCHAR(30), narratorFname VARCHAR(15),
narratorLname VARCHAR(15), time INTEGER, category VARCHAR(30));
--DROP FUNCTION searchBook(VARCHAR)
CREATE OR REPLACE FUNCTION SearchBook(bookName VARCHAR(30))
RETURNS book[] AS $$
DECLARE
    books book[];
    book_cursor CURSOR FOR SELECT name, author, fname, lname, time, category
    FROM books b, narrates n, narrator
    WHERE b.id = bookid AND nssn = ssn;
    i INTEGER;
BEGIN
    i := 1;
    FOR b IN book_cursor LOOP
        IF b.name LIKE Concat('%',bookName,'%') THEN
            books[i] = b;
            i = i+1;
        END IF;
    END LOOP;
    RETURN books;
END;
$$LANGUAGE 'plpgsql';
```

** filterBooks fonksiyonunun tamamı 5. Maddede verildiği için burada sadece cursor kullanılan kısım gösterilmiştir.

```
CREATE OR REPLACE FUNCTION filterBooks(Category VARCHAR, hour INTEGER)
RETURNS book[] AS $$
DECLARE
    books book[];
    i INTEGER;
    book_cursor CURSOR FOR SELECT name, author, fname, lname, time, b.category
    FROM books b, narrates n, narrator
    WHERE b.id = bookid AND nssn = ssn;
BEGIN
```

```

CREATE OR REPLACE FUNCTION calculatePrice(dayLimit INTEGER, book_id VARCHAR)
RETURNS INTEGER AS $$
DECLARE
    total_price INTEGER;
    book_price INTEGER;
BEGIN

    SELECT dailyPrice INTO book_price
    FROM books
    WHERE id = book_id;

    total_price = book_price * 0.5;
    RETURN total_price;
END;
$$LANGUAGE 'plpgsql';

```

11. 2 adet trigger tanımlamalı ve arayüzden girilecek değerlerle tetiklemelisiniz. Trigger'ın çalıştığına dair arayüze bilgilendirme mesajı döndürülmelidir.

```

CREATE OR REPLACE FUNCTION IncreaseNumberListeners( )
RETURNS TRIGGER AS $$
DECLARE
    x INTEGER;
BEGIN
    UPDATE narrator
    SET audience = audience + 1
    WHERE ssn = (SELECT ssn AS x
                  FROM narrator,narrates n
                  WHERE n.bookid = new.bookid AND n.nssn = ssn);

    RETURN NEW;
END;
$$ LANGUAGE 'plpgsql';

CREATE OR REPLACE TRIGGER trigger1 AFTER INSERT ON rental FOR EACH ROW EXECUTE PROCEDURE IncreaseNumberListeners();

```

Trigger 1

```

CREATE OR REPLACE FUNCTION ReduceBalance( )
RETURNS TRIGGER AS $$
DECLARE
    b INTEGER;
BEGIN

    SELECT balance INTO b
    FROM users
    WHERE ssn = new.ussn;

    IF b >= new.price THEN
        UPDATE users
        SET balance = balance - new.price
        WHERE ssn = new.ussn;
        RETURN NEW;
    ELSE
        RAISE WARNING 'Your balance is not enough to rent this book';
        RETURN OLD;
    END IF;
END;
$$ LANGUAGE 'plpgsql';

CREATE OR REPLACE TRIGGER trigger2 BEFORE INSERT ON rental FOR EACH ROW EXECUTE PROCEDURE ReduceBalance();

```

Trigger 2

```

CREATE OR REPLACE FUNCTION controlRent( )
RETURNS TRIGGER AS $$
DECLARE
    cur CURSOR FOR SELECT bookid
                    FROM rental
                    WHERE ussn = new.ussn;
BEGIN
    FOR c in cur LOOP
        IF c.bookid = new.bookid THEN
            RAISE WARNING 'You have already rented this book!';
            RETURN OLD;
        END IF;
    END LOOP;
    RETURN NEW;
END;
$$ LANGUAGE 'plpgsql';

CREATE OR REPLACE TRIGGER trigger3 BEFORE INSERT ON rental FOR EACH ROW EXECUTE PROCEDURE controlRent();

```

Trigger 3