

Please put your name and number on both sides of all sheets. Adınızı ve numaranızı tüm kağıtların her iki yüzüne yazınız. Use the following schema for the questions in this exam/Soruları cevaplamak için aşağıdaki şemayı kullanınız.

Student (sid, name, birthPlace, did) // ogrenci(ogrenci-no, adi, dogum-yeri, bolum-no)
 Take (sid, cid, grade) // ders-al(ogrenci-no, ders-kodu, notu)
 Course (cid, title, credits, did) // ders(ders-kodu, adi, kredisi, bolum-no)
 Department (did, name, teacherCount) // bolum(bolum-no, adi, hocaSayisi)
 Teacher (tid, name, birthPlace, did) // hoca(hoca-no, adi, dogum-yeri, bolum-no)
 Teach (tid, cid) // ders-ver(hoca-no, ders-kodu)

1
2
3
4
5

- (20) Aşağıdaki işlemler için SQL komutlarını veriniz (Give SQL ommand for the followings)
 - 'Ahmet'e student tablosu üzerindeki tüm hakları ver (Ahmet bu hakkı başkasına verebilsin) (Give Ahmet all priveledges on student table. Ahmet should be able give this to other users)
(5 puan) **GRANT ALL ON student TO Ahmet WITH GRANT OPTION**
 - 'sekreter' rolüne course tablosunda DELETE hakkı ver. (Give DELETE on course table to the 'sekreter' role)
(5 puan) **GRANT DELETE ON course TO sekreter**
 - 'sekreter' rolünü Ahmet'e ver. (Give sekreter' role to Ahmet)
(5 puan) **GRANT sekreter TO ahmet**
 - Ahmet'ten student tablosundaki UPDATE hakkını geri al. (Take UPDATE priveledge on student table from Ahmet)
(5 puan) **REVOKE UPDATE ON student FROM ahmet**
- (20) Department tablosundaki teacherCount (bölümdeki hocaların sayısı) alanını gerektiğinde güncelleyen triggerları yazınız (Give triggers that updates the teacherCount value -which represents the number of techers in the department- whenever necessasy in the Department table.)

Department tablosundaki teacherCount alanı (i) Teacher tablosuna yeni kayıt eklendiğinde (ii) Teacher tablosundan bir kayıt silindiğinde (iii) (i) Teacher tablosunaki bir kaydın did(bolum-no) alanı değiştirildiğinde yeniden hesaplanmalıdır (3 ayrı trigger veya birleşik tek bir trigger şeklinde yazılabilir, trigger standart SQLle kitaptaki gibi veya PostgreSQLle yazılabilir):

DELETE trigger 5 puan
INSERT trigger 5 puan
UPDATE trigger 10 puan

**CREATE TRIGGER teacherCount AFTER INSERT OR DELETE OR UPDATE OF did ON teacher
 REFERRING OLD ROW AS o NEW ROW AS n**

BEGIN

IF(INSERTING OR UPDATING)

**UPDATE department d SET teacherCount =
 (SELECT COUNT(*) // veya COUNT(t.did)
 FROM teacher t
 WHERE t.did=d.did) // veya t.did=n.did**

WHERE d.did = n.did;

IF(DELETING OR UPDATING)

**UPDATE department d SET teacherCount =
 (SELECT COUNT(*) // veya COUNT(t.did)**

```

FROM teacher t
WHERE t.did=d.did) // veya t.did=o.did
WHERE d.did = o.did;
END;

```

3. (20) did'si verilen bir bölümdeki derslerin cid'si ve dersin not ortalaması ve dersi alan öğrenci sayılarını döndüren stored function'ı veriniz (Give the stored function that returns "the cid, the average grade in the course and the number of students taking the course" of the courses in a department whose did is given as a parameter to the function)

10 puan SELECT komutu

10 puan geri kalanlar

```

CREATE FUNCTION dersler(cid INT) RETURNS TABLE(cid INT, notOrtalama INT, ogrenciSayisi INT)
BEGIN
    RETURN TABLE(SELECT c.cid (veya t.cid), AVG(t.grade), COUNT(t.sid)
        FROM course c , take t
        WHERE c.cid = t.cid AND c.did = dersler.cid
        GROUP BY c.cid (veya t.cid) );
END;

```

4. (15) Konsole ekranından öğrencinin numarasını(sid) ve yeni/güncel/doğru adını(name) okuyup, veritabanında öğrencinin numarasını kullanarak adını güncelleyen Java JDBC programını bütün olarak veriniz. (Give the full Java JDBC program that updates the name of student whose sid and new name is given from the console)

5 puan UPDATE komutu

5 puan try-catch içindeki bolum

5 puan geri kalan kod

```

import java.sql.*;
import java.util.Scanner;

```

```

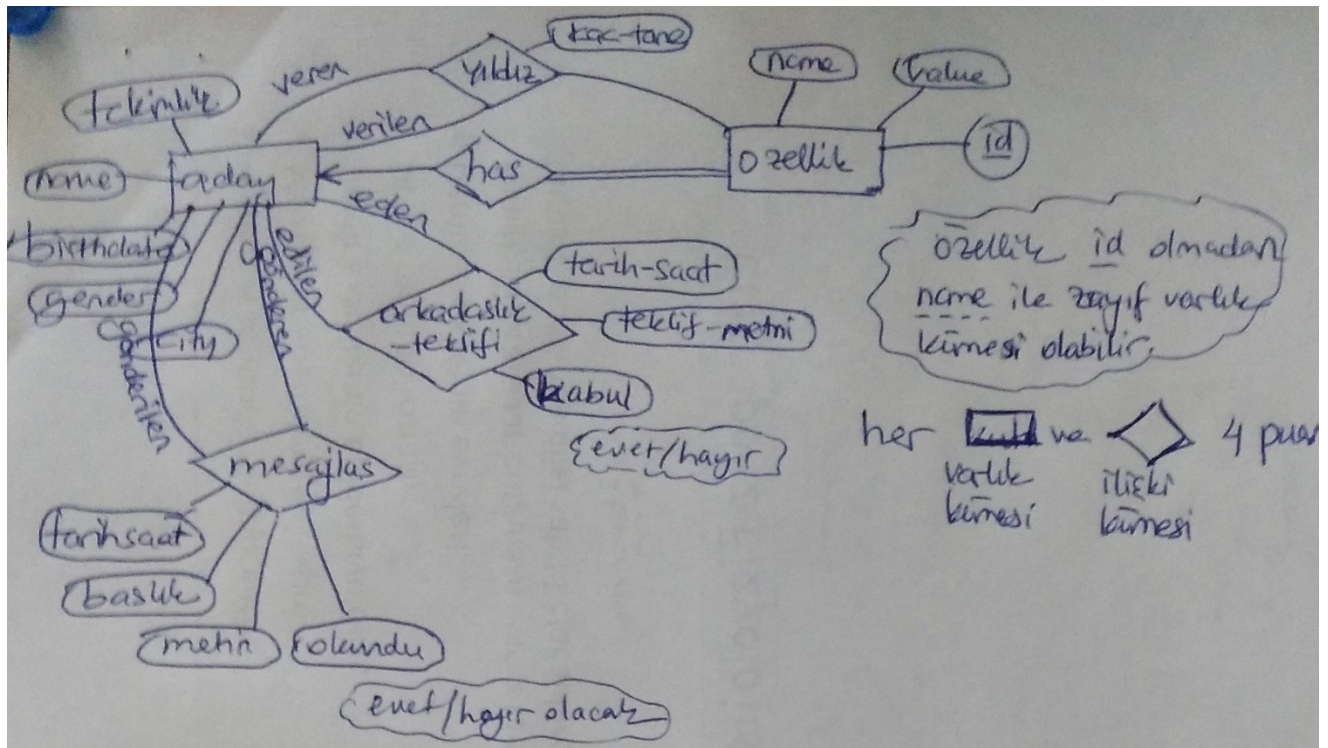
public class JDBCExample {
public static void main(String[] argv) {
    Scanner in = new Scanner(System.in);
    System.out.println("Sid?");
    int ogrno = in.nextInt();
    System.out.println("New name?");
    int ogrname = in.next();
    try {
        Class.forName("org.postgresql.Driver");
        Connection c = DriverManager.getConnection("jdbc:postgresql://127.0.0.1:5432/school",
"user", "password");
        Statement s = c.createStatement();
        s.executeUpdate("UPDATE student SET name='"+ogrname+"' WHERE sid="+ogrno);
        s.close();
        c.close();
    } catch(SQLException e){

```

```
        System.out.println("SQLException : " + sqle);
    }
}
}
```

5. (25) Aşağıda anlatıldığı şekilde bir çöpçatan sitesi için ER diagramı çizin ve ER diagramını ilişkisel veritabanı şemasına çeviriniz. (Give an ER diagram for the following matchmaker site and convert that ER into a relational database schema.)
- Adayların tckimlik, adı, doğum tarihi, cinsiyeti, yaşadığı şehir bilgisi vardır. (Candidates have tckimlik, name, birth date, gender, city of living attributes)
 - Adaylar istedikleri kadar özelliği profillerine (kişisel sayfalarına) ekleyebilir. Her özelliğin bir adı ve değeri vardır. Örnek: Sac:Kumral, Boy:1,75, gibi. (Candidates can add as many characteristics as they like to their profiles/pages such as Height:1,7cm, Hair:Blonde etc)
 - Adaylar başka adayların özelliklerini 1-5 arası yıldız vererek beğenebilirler. Beğenme sırasında kimin kime ait hangi özelliğe kaç yıldız verdiği tutulmalıdır. (candidates can like each others characteristics by giving 1-5 stars. Who is liking whose what characteristics, and the number of stars should be stored in the database)
 - Adaylar birbirlerine arkadaşlık teklifi yapar veya geri alabilirler. Arkadaşlık teklifleri kabul ya da reddedilir. Arkadaşlık teklifi yapılırken kimin, kime yaptığı, tarih-saat, arkadaşlık teklif metni (teklif metni kişinin niçin arkadaşlık teklif ettiğini romantik bir şekilde açıklayan bir metindir) bilgisi tutulmalıdır. (Candidates send and revoke friendship requests to other. These requests are accepted or rejected. Whose is sending whom, the date-time of request and the message text of request should be stored in the database)
 - Adaylar birbirlerine sistemden mesaj gönderebilirler. Mesajların tarih-saati, başlığı, metni, kimden kime gönderildiği ve okunup okunmadığı tutulmalıdır. (Candidate can send text messages from the system. Tee date-time, heading, and text of messages should be stored)

ÇÖZÜM ve PUANLAMA AŞAĞIDA



aday(tc kimlik, ad, doğum tarihi, cinsiyet, şehir)
 özellik(id, name, value, tc kimlik)
 yıldız(verenid, verilenid, kaç-tane)
 arkadaşlık-teklifi(edenid, edilenid, tarih-saat, teklif-metni, kabul)
 mesajlar(gönderenid, gönderilenid, tarih-saat, başlık, metin, olundu)

herbiri 4 puan