İstanbul Un. Bilgisayar Müh No: Name: Signature: 80 dk
Veritabanı Sistemleri, 2017 Güz, bütünleme, ÖRGÜN, AKURT
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, courseCount)// hoca(hoca-no, adi, dogum-yeri, bolum-no, ogrenciSayisi)
Teach (tid, cid) // ders-ver(hoca-no, ders-kodu)

1. Aşağıdaki işlemler için SQL komutlarını veriniz (Give SQL ommand for the followings)
a. (5) Tüm kullanıcılara course tablosu üzerinde DELETE ve INSERT hakkı ver (Onlar da bu hakları başkasına verebilsinler) (Give all users DELETE ve INSERT on course table. They should be able give this to others)
GRANT DELETE, INSERT ON course TO PUBLIC WITH GRANT OPTION
b. (5) 'Ahmet'ten course tablosundaki DELETE hakkını geri al (Give DELETE on course table to the 'sekreter' role)
REVOKE DELETE ON course FROM Ahmet;
2. (10) Bir öğrenci en fazla 40 ders alabilir kısıtını (assertion) SQLle yazınız (A stdent can take maximum 40 classes. Write an assertion using SQL for this constraint)

```
CREATE ASSERTION enFazlaKirk AS
        NOT EXISTS(SELECT t.sid, COUNT(cid)
                FROM take t
                GROUP BY t.sid
                HAVING COUNT(cid) > 40)
```

3. (5) course collection'indan 'ad' özelliği 'ali' olan dokümanları döndüren MongoDB sorgusu yazınız

```
db.mycol.find({"ad":"ali"})
```

4. (5) course collection'ından sid'si 25 olan dokümanı silen mongoDB sorgusunu yazınız

```
db.course.remove({'sid':25})
```

5. (20) Teacher tablosundaki courseCount (verdiği ders sayısı) alanını gerektiğinde güncelleyen triggerları yazınız (Give triggers that updates the courseCount value -which represents the number of courses taught by the teacher- whenever necessasy in the Teacher table.)

Teacher tablosundaki courseCount alanı (i) Teach tablosuna yeni kayıt eklendiğinde (ii) Teach tablosundan bir kayıt silindiğinde (iii) (i) Teach tablosunaki bir kaydın tid (hoca-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 courseCount AFTER INSERT OR DELETE OR UPDATE OF tid ON teach
REFERRING OLD ROW AS o NEW ROW AS n
BEGIN
  IF(INSERTING OR UPDATING)
    UPDATE teacher d SET courseCount =
        (SELECT COUNT(*)      // veya COUNT(t.did)
        FROM teach t
        WHERE t.did=d.did)    // veya t.did=n.did
    WHERE d.did = n.did;
  IF(DELETING OR UPDATING)
    UPDATE teacher d SET courseCount =
        (SELECT COUNT(*)      // veya COUNT(t.did)
        FROM teach t
        WHERE t.did=d.did)    // veya t.did=o.did
    WHERE d.did = o.did;
END;
```

6. (15) sid'si verilen bir öğrencinin ağırlıklı not ortalamasını, ve aldığı derslerin sayısını döndüren stored function'ı veriniz (Give the stored function that returns "the sid, the average grade and the number of courses taken" of the student whose sid is given as a parameter to the function)

10 puan SELECT komutu
5 puan geri kalanlar

Fonksiyon tek değer döndürür. Soruda 2 değer döndürülmesi istenmiş. O zaman geriye bir RECORD, ROW veya STRUCT, OBJECT gibi kompleks bir nesne döndürebiliriz. (SELECT'te GRUP BY yok!)

```
CREATE FUNCTION dersler(sid INT) RETURNS RECORD(notOrtalama INT, dersSayisi INT)
BEGIN
        RETURN RECORD(SELECT SUM(c.credits * t.grade) / SUM(c.credits), COUNT(t.cid)
                        FROM course c , take t
                        WHERE c.cid = t.cid AND t.sid = dersler.sid);
END;
```

VEYA 2 değer istendiği için sonuçları argümanlarda geri döndüren bir procedure yazılabilir

```
CREATE PROCEDURE dersler(IN sid INT, OUT notOrtalama INT, OUT dersSayisi INT)
BEGIN
        SELECT SUM(c.credits * t.grade) / SUM(c.credits), COUNT(t.cid)
        INTO dersler.notOrtalama, dersler.dersSayisi
        FROM course c , take t
        WHERE c.cid = t.cid AND t.sid = dersler.sid;

        RETURN;
END;
```

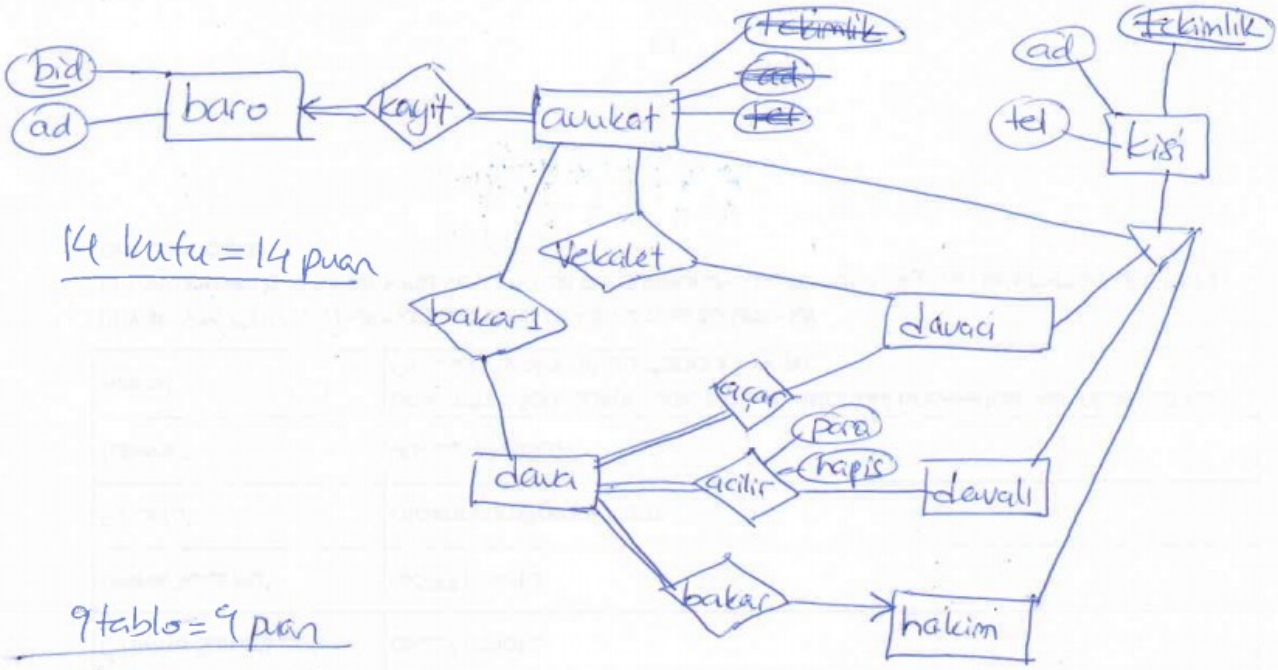
7. (15) Konsole ekranından bölümno (did) verilen bölümdeki derslerin cid'si ve adını listeleyen Java JDBC programını bütün olarak veriniz. (Give the full Java JDBC program that list the cid and names of courses in the department whose id is given from the console)

```
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("did?");
        int did = in.nextInt();
        try {
                Class.forName("org.postgresql.Driver");
                Connection c = DriverManager.getConnection("jdbc:postgresql://127.0.0.1:5432/school",
"user", "password");
                // buraya kadar 5 puan
                Statement s = c.createStatement();
                Recordset r = s.executeQuery("SELECT * FROM course WHERE did="+did);
                // buraya kadar 5 puan
                // gerisi 5 puan
                while(r.next())
                        System.out.println("Cid:" + r.getInt('cid') + " Name:" + r.getString('title'))
                s.close();
                c.close();
        } catch(SQLException e){
                System.out.println("SQLException : " + sqle);
        }
}
}
```

8. (25) Avukatlar için ER diagramı çiziniz ve ER diagramını ilişkisel veritabanı şemasına çeviriniz. (Give en ER diagram for the following lawyer site and convert that ER into a relational database schema.)

a. Avukatların tckimler, ad, telefonları ve kayıtlı oldukşarı baroları vardır
b. Baroların adı, ve hangi şehire ait oldukları bilgisi vardır
c. Davacıların, davalıların ve hakimlerin TCkimlikno, adı, telefonu vardır
d. Davacılar avukatlara vekalet verirler ve davalılara dava açarlar
e. Davaların tarihi, konusu, türü (idari, ceza), davacılar ve davalılar ve 1 adet hakimi vardır. Davanın olumlu veya olumsuz sonucu vardır
f. Bir havada birçok davalı ve davacı bulunabilir. Dava sonucunda davalılar para ve/veya hapis cezası alabilir.
g. Bir avukat birçok kişinin (davalı veya davacı) vekili olabilir ve birçok davaya bakabilir.
h. Bir hakim birçok davaya bakabilir.



14 kutu = 14 puan

9 tablo = 9 puan

baro(id, ad)
avukat(id, ad, tel)
davaci(id, ad, tel)
davali(id, ad, tel)
vekalet(avukatid, davaciid)
dava(id, tarih, konu, tutu, hakimid)
acar(davaid, davaciid)
acilir(davaid, davaliid)
bakar1(avukatid, davaid)