

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, courseCount)// ogrenci(ogrenci-no, adi, dogum-yeri, bolum-no,dersSayisi)

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, courseCount) // bolum(bolum-no, adi, hocaSayisi)

Teacher (tid, name, birthPlace, did, courseCount) // hoca(hoca-no, adi, dogum-yeri, bolum-no, dersSayisi)

Teach (tid, cid) // ders-ver(hoca-no, ders-kodu)

1. Aşağıdaki işlemler için SQL komutlarını veriniz:
 - a. (5) Tüm kullanıcılara teacher tablosunda INSERT ve DELETE hakkı ver
GRANT INSERT, DELETE ON teacher TO PUBLIC
 - b. (5) Ahmet'ten teacher tablosundaki tüm hakları geri al.
REVOKE ALL ON teacher FROM Ahmet
2. (10) Bir bölümde en fazla 10 ders olabilir kısıtını (assertion) SQL'le yazınız

```
CREATE ASSERTION enFazla10 AS
NOT EXISTS(SELECT did, COUNT(cid)
FROM course
GROUP BY did
HAVING COUNT(cid) > 10)
```

3. Aşağıdaki işlemleri mongoDB ile yazınız
 - a. (5) Course collection'ına cid özelliği 101, dersAdi özelliği 'Programlama' ve kredi özelliği 5 olan dokümanı ekleyen MongoDB komutu
db.course.insert({"cid":101, "dersAdi":"Programlama", "kredi":5})
 - b. (5) Course collection'ından sid'si 25 olan dokümanın dersAdi özelliğini 'Veritabanı' olarak değiştiren MongoDB komutu
db.course.update({"sid":25}, {\$set:{"dersAdi":"Veritabanı"}})
1. (20) Department tablosundaki courseCount (bölümdeki ders sayısı) alanını gerektiğinde güncelleyen triggerları yazınız

Department tablosundaki courseCount alanı (i) Course tablosuna yeni kayıt eklendiğinde (ii) Course tablosundan bir kayıt silindiğinde (iii) (i) Course tablosundaki 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 courseCount AFTER INSERT OR DELETE OR UPDATE OF did ON Course
REFERRING OLD ROW AS o NEW ROW AS n
BEGIN
IF(INSERTING OR UPDATING)
UPDATE department d SET courseCount =
```

```

        (SELECT COUNT(*)    // veya COUNT(t.did)
        FROM Course t
        WHERE t.did=d.did) // veya t.did=n.did
    WHERE d.did = n.did;
IF(DELETING OR UPDATING)
    UPDATE department d SET courseCount =
        (SELECT COUNT(*)    // veya COUNT(t.did)
        FROM Course t
        WHERE t.did=d.did) // veya t.did=o.did
    WHERE d.did = o.did;
END;

```

2. (15) tid'si verilen hocaların verdiği dersler için dersin cid'si, dersin not ortalaması ve dersi alan öğrenci sayılarını döndüren stored function'ı veriniz

10 puan SELECT komutu

5 puan geri kalanlar

```

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

```

3. (15) Konsole ekranından dersin kodumu (cid) okuyup, dersi alan öğrencilerin sid, name, birthPlace ve notunu (grade) listeleyen Java JDBC programını bütün olarak veriniz.

```

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

```

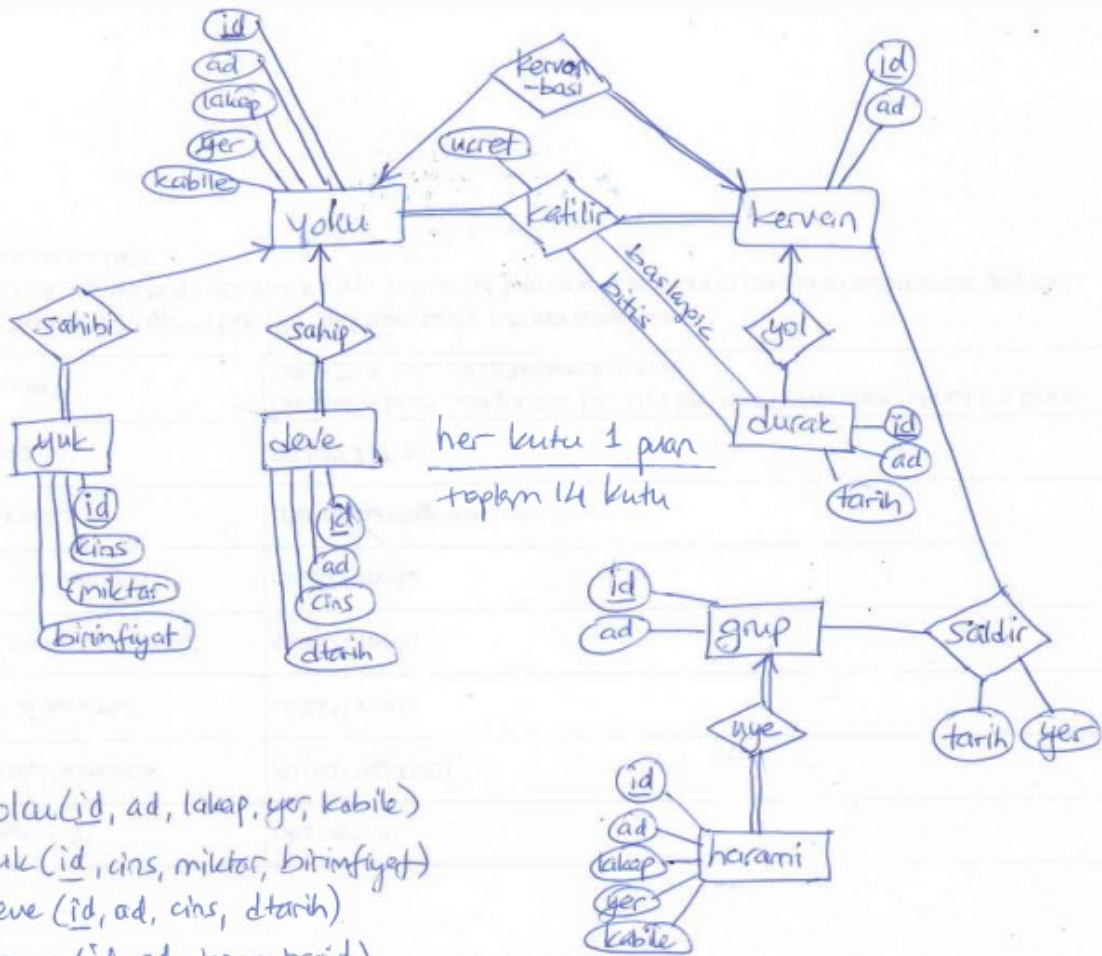
```

public class JDBCExample {
public static void main(String[] argv) {
    Scanner in = new Scanner(System.in);
    System.out.println("cid?");
    int cid = 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 s.sid, s.name, s.birthPlace, t.grade
            FROM take t, student s
            WHERE t.sid=s.sid AND t.cid="+cid);
        // buraya kadar 5 puan, gerisi 5 puan
        while(r.next())

```

```
        System.out.println("Cid:" + r.getInt('sid') + " Name:" + r.getString('title')+ " Birth  
Place:" + r.getString('birthPlace') + " Grade:" + r.getInt("grade"));  
        s.close();  
        c.close();  
    } catch(SQLException e){  
        System.out.println("SQLException : " + sqle);  
    }  
}  
}
```

4. (25) Kervan başları için aşağıdaki şekilde bir Kervan Bilgi Sistemi ER diagramı çizin ve ER diagramını ilişkisel veritabanı şemasına çeviriniz.
- Kervanlar yük ve yolcu taşıyan deve katarlarıdır.
 - Her kervanın yolcuları ve yolcuların da deve ve yükleri vardır.
 - Her yolcunun adı, lakabı, kabile ve memleketi (yaşadığı yer) vardır
 - Her devenin adı cinsi ve yaşı vardır. Yükün cinsi, miktarı ve birim fiyatı vardır.
 - Yolcunun sıfır yada daha fazla devesi vardır. Bir yolcunun sıfır yada daha fazla yükü olabilir
 - Kervan 1 yada daha fazla yolcudan oluşur. Kervanı işleten (organizatör) yolculardan birisi kervan başıdır. Kervana katılan diğer yolcular bunu için kervan başına ücret öderler. Kervanın bir rotası, yani bir başlangıç konaklama yeri bir bitiş konaklama yeri ve birçok ara konaklama yerleri vardır.
 - Bir yolcu birçok kervana katılabilir. Kervana katılığında hangi konaklama (şehir) alanından katılıp hangi konaklama alanında ayrıldığı ve ne kadar ücret ödediği önemlidir.
 - Haramiler kervanlara saldıran haydut gruplarıdır. Haramiler yolcularla aynı özelliklere sahiptir.
 - Haramiler grup halinde dolaşırlar. Grupların isimleri vardır, Kırk Haramiler gibi. Gruptan bir harami grup lideridir.
 - Haramiler kervan basarak geçinirler. Baskında amaç hırsızlık yapmaktır. Gerekmedikçe adam öldürölüp insan kaçırlmaz. "Kervan basma"da önemli bilgiler basan grup, basılan kervan, tarih ve yer'dir.



9 puan

- yolcu(id, ad, lakap, yer, kabile)
- yolk(id, cins, miktar, birimfiyat)
- deve(id, ad, cins, dtarih)
- kervan(id, ad, kervanbasid)
- durak(id, ad, kervanid)
- katilir(kervanid, yolcu, baslangicid, bitisid, ucret)
- grup(id, ad)
- harami(id, ad, lakap, yer, kabile, grupid)
- saldir(grupid, kervanid, tarih, yer)