No: 1306170114

Name: Enver Usta

Signature:

```
1.   SELECT student.*
     FROM student o
     JOIN take t ON o.sid = t.sid
     WHERE notu > SOME( SELECT grade
                 FROM take t1
                 JOIN course c ON t1.ders_kodu = c.ders_kodu
                 WHERE c.adi = 'Veri Yapıları');
```

```
2.  SELECT *
    FROM teacher
    WHERE tid in(
          SELECT tid
          FROM teach
          WHERE cid in(
               SELECT cid
               FROM course
               WHERE title != "Operating Systems"
                 AND title = "Database Systems"
          )
    );
```

No: 1306170114
Name: Enver Usta
Signature:

3.
```
SELECT cid, COUNT(sid), avg(grade)
FROM take
GROUP BY cid
MAVING avg(grade) > 50;
```

__ilişkisel Cebir__ : $\pi_{cid,\ AVG(grade),\ COUNT(sid)}(\sigma_{AVG(grade)>50}(\gamma_{cid,\ AVG(grade),\ COUNT(sid)}))(take)$

5.
```
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(*)
                        FROM teach t
                        WHERE t.tid = d.tid)
        WHERE d.tid = n.tid;
    IF(DELETING OR UPDATING)
        UPDATE teacher d
        SET courseCount = (
                        SELECT COUNT(*)
                        FROM teach t
                        WHERE t.did = d.did)
        WHERE d.did = o.did;
 END
```

No: 130617011 4
Name: Enver Usta
Signature:

6.
```
CREATE FUNCTION
dersnotOgr(ogr_noArgument integer)
    RETURNS table(
        ders_kodu1 Integer,
        adi1 varchar(30),
        notu1 integer
    )
  AS $my-stored-function$
  BEGIN
    RETURN QUERY
    SELECT ders_al.ders_kodu, ders.adi, ders_al.notu
    FROM ders_al
    JOIN ogrenci ON ders_al.ogr-no = ogrenci.ogr_no
    JOIN ders    ON ders_al.ders_kodu = ders.ders_kodu
    WHERE ogr-no = ogr_noArgument;
  END;
  $my-stored-function$
  LANGUAGE plpgsql
```

NO: 1306170114

Name: Enver Usta

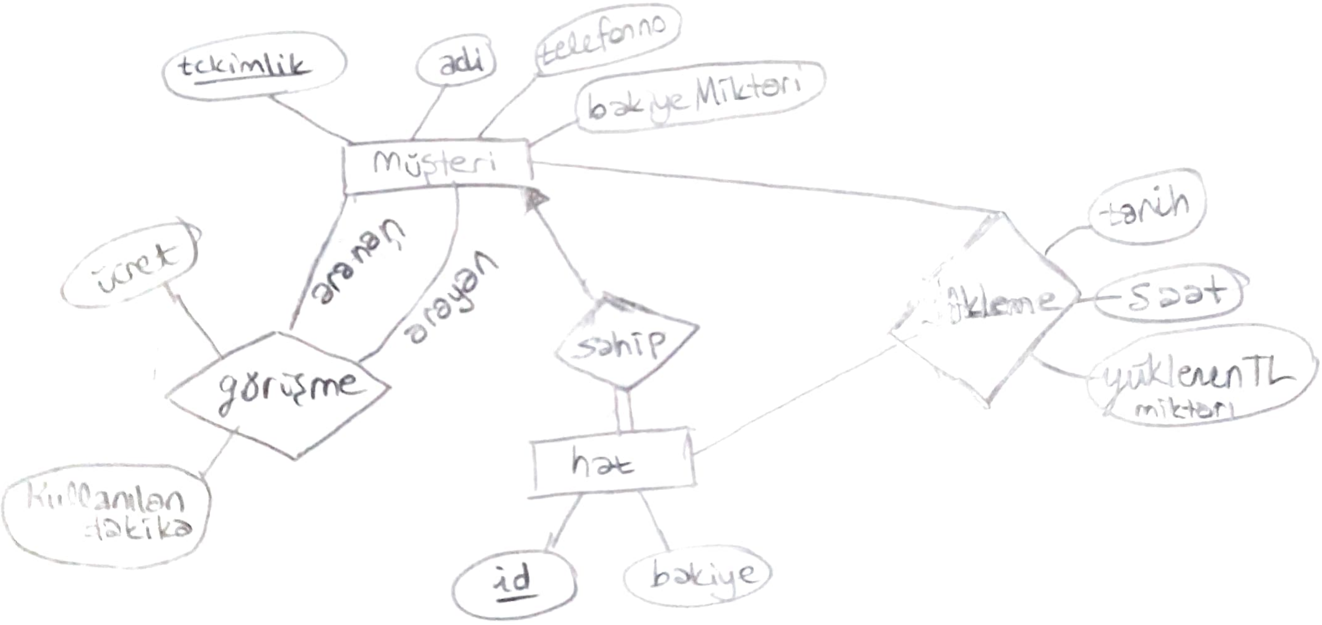
Signature:

4.

4.a.
```
db.Student.aggregate([{"$group":
        {_id:"$name", count:{$sum:1}}}])
```

4.b.
```
db.Student.find($and:[{"courses.cid":101},
        {"courses.grade":{$gt:70}}])
```

7.



İlişkisel veritabanı şeması:

müşteri (tckimlik, adi, telefon-no, bakiye)

yükleme(tckimlik, id, tarih, saat, miktar)

hat (id, bakiye)

görüşme(tckimlik, id, ücret, dakikasayisi)

sahip (tckimlik, id)