

# CS 353 HW 3



**Section: 1**

**Utku Kurtulmuş - 21903025**

**Q1)**

**student (s-id, s-name, department, cgpa)**

**club (c-id, c-name)**

**membership (s-id, c-id, since)**

**s-id is a foreign key to student, c-id is a foreign key to club**

**\*Only answers are provided for Q1.**

**(a)**    SELECT s-id, s-name  
         FROM student  
         WHERE department = "CS" and cgpa > 3.0

**(b)**    SELECT c-id, c-name  
         FROM student  
         NATURAL JOIN membership  
         NATURAL JOIN club  
         WHERE cgpa > 2.0;

**(c)**    SELECT c-id, c-name, COUNT(\*) as cnt  
         FROM club  
         NATURAL JOIN membership  
         GROUP BY c-id, c-name  
         HAVING COUNT(\*) > 100

**(d)**    WITH temp1 AS (  
             SELECT c-id, COUNT(\*) as cnt  
             FROM membership  
             GROUP BY c-id  
         ), temp2 AS (  
             SELECT MAX(cnt) as cnt  
             FROM temp1  
         )  
         SELECT c-id, c-name  
         FROM temp2  
         NATURAL JOIN temp1  
         NATURAL JOIN club

**Q2)**

**Students(TCK, sname, sdept, cgpa)**

**Courses(cid, cname, cdept, credit)**

**Grades(TCK, cid, grade)** TCK is a foreign key to Students,  
cid is a foreign key to Courses

**(a) Find the students (TCK, sname) who have taken a grade higher than 90 in a 4 credit course from the CS department.**

```
WITH t1 AS (  
    SELECT * FROM Courses  
    WHERE credit = 4  
    AND cdept = "CS"  
)  
SELECT TCK, sname  
FROM students  
NATURAL JOIN t1  
NATURAL JOIN (SELECT * FROM Grades WHERE grade > 90)
```

**(b) Find the CS students (TCK, sname) whose cgpa is higher than 3.50, but have not taken any course from the EE department**

```
SELECT TCK sname  
FROM Students  
WHERE sdept = "CS" AND cgpa > 3.5  
AND TCK not in (  
    SELECT TCK  
    FROM Courses  
    NATURAL JOIN GRADES  
    WHERE Courses.cdept="EE"  
)
```

**c) Find the CS student(s) (TCK, sname) with the highest cgpa.**

```
SELECT TCK, sname  
FROM Students  
WHERE sdept = 'CS'  
AND cgpa = (SELECT MAX(cgpa) FROM Students WHERE sdept = 'CS');
```

**(d) Find the CS courses (cid, cname) which have been taken by the students of at least 5 different departments.**

```
WITH t1 AS (  
    SELECT * FROM Courses  
    NATURAL JOIN Grades  
    NATURAL JOIN Students  
    WHERE cdept = "CS"  
)  
SELECT cid, cname  
FROM t1  
GROUP BY cid  
HAVING COUNT(DISTINCT sdept) >= 5
```

**(e) Find the students (TCK, sname) who have received a grade of 100 in at least 5 courses.**

```
SELECT TCK, sname  
FROM Grades  
NATURAL JOIN Students  
WHERE grade = 100  
GROUP BY TCK  
HAVING COUNT(DISTINCT cid) >= 5
```

**(f) For each course of each department, find the average of student grades. Give the resulting list in increasing alphabetical order of department names.**

```
SELECT cdept, cid, cname, AVG(grade) as average  
FROM Courses  
NATURAL JOIN Grades  
GROUP BY cdept, cid, cname  
ORDER BY cdept ASC;
```

**(g) For each department, find the course(s) (cid) which has the highest of average of student grades. Give the resulting list in increasing alphabetical order of department names.**

```
WITH t1 AS ( SELECT cdept, cid, cname, AVG(grade) as average
              FROM Courses
              NATURAL JOIN Grades
              GROUP BY cdept, cid, cname
              ORDER BY cdept ASC
            )
SELECT * FROM t1 t_1
WHERE t_1.average = (
    SELECT MAX(average)
    FROM t1 t_2
    WHERE t_1.cdept = t_2.cdept
);
```

**(h) Find the student(s) (TCK, sname) whose cgpa is higher than the average cgpa of his/her department.**

```
SELECT TCK, sname
FROM Students s1
WHERE s1.cgpa > (
    SELECT AVG(s2.cgpa) as cgpa
    FROM Students s2
    WHERE s1.sdept = s2.sdept
    GROUP BY s2.sdept
);
```

**(i) For each department, find the student(s) (TCK, sname) with the highest cgpa. Give the resulting list in increasing alphabetical order of department names.**

```
SELECT sdept, TCK, sname
FROM Students s1
WHERE s1.cgpa = (
    SELECT MAX(s2.cgpa) as cgpa
    FROM Students s2
    WHERE s1.sdept = s2.sdept
    GROUP BY s2.dept
)
ORDER BY sdept ASC;
```

**(j) Find pairs of students (TCK) who are from different departments and have taken at least 10 common courses. Each pair should be listed only once**

```
WITH t1 as (
    SELECT TCK, cid, sdept
    FROM Grades
    NATURAL JOIN Students
)
SELECT s1.TCK as TCK1, s2.TCK as TCK2
FROM t1 s1
JOIN t1 s2 ON
s1.TCK < s2.TCK
AND s1.sdept != s2.sdept
AND s1.cid = s2.cid
GROUP BY s1.TCK, s2.TCK
HAVING COUNT(*) >= 10;
```