HW3

1.

```
CREATE TABLE students
        (sid VARCHAR(10),
        name VARCHAR(20),
        age INTEGER,
        gpa FLOAT,
        PRIMARY KEY (sid));
CREATE TABLE courses
        (cid VARCHAR(10),
        deptid VARCHAR(15),
        name   VARCHAR(20),
        PRIMARY KEY (cid));
CREATE TABLE professors
        (ssn INTEGER,
        name VARCHAR(20),
        address VARCHAR(30),
        phone VARCHAR(10),
        deptid VARCHAR(15),
        PRIMARY KEY (ssn));
CREATE TABLE enrollment
        (sid VARCHAR(10),
        cid VARCHAR(10),
        section INTEGER,
        grade CHAR(2),
        PRIMARY KEY (sid, cid),
        FOREIGN KEY (sid) REFERENCES students,
        FOREIGN KEY (cid) REFERENCES courses,
        FOREIGN KEY (cid,section) REFERENCES teaches(cid, section);
CREATE TABLE teaches
        (cid VARCHAR(10),
        section INTEGER,
        ssn INTEGER,
        PRIMARY KEY (cid, section),
        FOREIGN KEY (cid) REFERENCES courses,
        FOREIGN KEY (ssn) REFERENCES professors(ssn));
```

2.

```
SELECT name FROM professors WHERE deptid = 'cs';
```

3.

```
SELECT s.sid FROM enrollm e, courses c, students s WHERE s.sid = e.sid AND e.sid = c.cid AND cdeptid = ‘CS’;
```

4.

```
SELECT p.ssn, p.name FROM professors p WHERE p.deptid = ‘CS’ AND p.ssn NOT IN (SELECT
p.ssn, FROM professors p, teaches t, courses c, WHERE p.ssn = t.ssn AND t.cid = c.cid AND c.deptid =
‘CS’ );
```

5.

```
SELECT Count(*), FROM courses c, GROUP BY by c.deptid;
```

6.

```
SELECT deptid, COUNT(cid)FROM courses GROUP BY deptid HAVING COUNT(cid) > 10 ;
```

7.

```
SELECT DISTINCT s.name FROM students AS s
        INNER JOIN teaches t ON e.cid = t.cid
        INNER JOIN  enrollm e ON s.sid=e.sid
        INNER JOIN professors p ON p.ssn = t.ssn
WHERE p.name LIKE 'M%'
```

8.

SELECT c.deptid, COUNT (e.sid)=30 AND COUNT (e.sid)=80 AS large FROM enroll e, courses c, WHERE e.cid = c.cid, GROUP BY e.sections, c.cid;

9.

```
SELECT p.ssn
FROM professors p,
        (SELECT t1 .deptid,
                SUM(if (t1 sectionsize < 30, 1, 0) small,
                SUM(if (t1. sectionsize >= 30 AND t1.sectionsize < 80, 1, 0) medium,
                SUM(if (t1. sectionsize >= 80, 1, 0) large
        FROM
                SELECT c.deptid deptid,
                        e cid.
                        e section,
                        COUNT(*) sectionsize
                FROM enrollment e,
                courses C
                WHERE e cid = c.cid
                GROUP BY c.deptid, e.cid, e. section
                ) t1
        GROUP BY t1. deptid
        ) t2
WHERE p.deptid IN (
                SELECT deptid
                FROM (
                        SELECT deptid, COUNT(*)
                        FROM professors
                        GROUP BY deptid
                        HAVING COUNT(*) > 20
                    ) t3
                ) t4
        AND p.deptid = t2. deptid
        AND t2.large > (t2.medium + t2.small);
```

10.

CREATE TEMPORARY TABLE Fails SELECT sid, FROM enroll e, students s WHERE s.sid = e.sid and e.grades I ('D', 'F');

CREATE Temporary Table AllEnrolledStudents SELECT sid, FROM enroll e, students s, WHERE s.sid = e.sid; SELECT c.cid, COUNT(Fails.sid) / COUNT(AllEnrolledStudents.sid) * 100 FailedPercent FROM enroll e, courses c, Fails, AllEnrolledStudent, WHERE c.cid = e.cid, GROUP BY c.cid;

11.

Assume only one professor has the max. & of students that failed his/her course

```
SELECT p.name
FROM professors p,
        teaches t,
        (
        SELECT t1.course course, t1 .section section, (12.fail/t1 total*100) coursefails
        FROM (
                SELECT e.cid course,
                        e section section,
                        COUNT(*) total
                FROM enrollment e
                GROUP BY e. cid, e. section
                )t1
        LEFT JOIN (
                SELECT e cid course
                        e.section section
                        COUNT(*) fail
                FROM enrollment e
                WHERE e grade = 'D' OR e grade = 'F'
                GROUP BY e. cid, e section
                ) t2
        ON t1 course = t2.course AND t1 section = t2. section;


        ) temp
WHERE t.id = temp.cid
        AND t.section = temp section
        AND t.ssn = p.ssn
        AND temp. coursefails = (
                                        SELECT MAX((t2. fail/+1.total*100))
                                        FROM (
                                                SELECT e. cid course
                                                        e section section,
                                                        COUNT(*) total
                                                 FROM enrollment e
                                                GROUP BY e.cid, e, section
                                                ) t3
                                        LEFT JOIN (
                                                SELECT e.cid course
```

```
                    e.section section,
                    COUNT(*) fail
                FROM enrollment e
                WHERE e.grade = 'D' OR e grade = 'F'
                GROUP BY e.cid, e section
                ) t4
        ON t3.course = t4 course AND t3, section = t4. section;
        );
```

12.

SELECT e.cid, (COUNT(e.sid)/(COUNT(Distinct sid) FROM enrollment)) * 100 AS Average

13.

```
SELECT e.sections FROM enroll e, courses c WHERE e.grades IN('D','F') GROUP BY
e.sections Having COUNT(e.grades) > ( SELECT Avg( sections) FROM enrollm );
```

14.

```
CREATE TEMPORARY TABLE T1
        SELECT DISTINCT e.sid, c.dept,
                e.sections, IF( e.grades IN (‘A’), 1, 0) NumA,
                IF(e.grades IN (‘B’), 1, 0) NumB,
                IF(e.grades IN (‘C’), 1, 0) NumC,
                IF(e.grades IN (‘D’), 1, 0) NumD,
                IF(e.grades IN (‘F’), 1, 0) NumF
         FROM enrollment e, courses c;
Select c.deptid,
        (SUM(T1.NumA)/(Select Count(s.sid)) as A%,
        (SUM(T1.NumB)/(Select Count(s.sid)) as B%,
         (SUM(T1.NumC)/(Select Count( s.sid)) as C%,
        (SUM(T1.NumD)/(Select Count(s.sid)) as D%,
        (SUM(T1.NumF)/(Select Count(s.sid)) as F% From T1 inner join courses c On c.cid =
e.cid, students s;
```