```
★ CREATE ~~Teaches~~ Teaches(cid:int, section:int, ssn:varchar(9),
   TABLE              PRIMARY KEY(cid, section)
                      FOREIGNKEY (sid) REFERENCES Students,
                      FOREIGNKEY(cid) REFERENCES Courses);
```

Kristina Poon
Homework 3

```
1) CREATE TABLE Students (sid: varchar (8), name: varchar(20), age:int,
                          gpa: float,
                          PRIMARY KEY (sid));
   CREATE TABLE Courses (cid: int, deptid: int, description: varchar (100),
                        PRIMARY KEY (cid, deptid));
   CREATE TABLE Professors (ssn: varchar (9), name: varchar(20),
                           address: varchar (20), phone: varchar (10),
                           deptid: int,
                           PRIMARY KEY (ssn));
   CREATE TABLE Enrollment (sid: varchar(8), cid: int, section: int,
                            grade: int,
                            PRIMARY KEY (sid, cid, section),
                            FOREIGN KEY (sid) REFERENCES Students,
                            FOREIGN KEY (cid) REFERENCES Courses);
```

★ (see margin)

```
2) SELECT  p.Name
   FROM    Professors p
   WHERE   p.deptId = "CS";

3) SELECT  s.sid
   FROM    Students s, Courses c, Enrollment E
   WHERE   s.sid = e.sid  AND  e.cid = c.cid  AND  c.deptid = "CS";

4) SELECT  p.ssn, p.name
   FROM    Professors p
   WHERE   p.deptid = "CS"  AND
           NOT EXIST (select c.cid
                      FROM  Courses c, Teaches t,
                      WHERE c.deptid = "CS" AND
                            c. cid = t.cid AND
                            p.ssn = t.ssn);
```

```
5) SELECT (Count (*)), c.deptid
   FROM Courses C,
   GROUP BY c.deptid ;

6) SELECT ( Count(*)), c.deptid
   FROM Courses C
   GROUP BY c.deptid
   HAVING COUNT(*)>10;

7) SELECT DISTINCT S.name
   FROM Students S, Enrollment E, Professor p, Teaches t,
   WHERE e.sid =S.sid AND e.cid = t.cid AND p.ssn = t.ssn
         AND t.section = c.section AND p.name LIKE 'M%';

8) t1 = (SELECT DISTINCT (e.section, e.cid (count(*) ssid) AS
               selection cnt)
          FROM Enrollment e, Students s,
          WHERE e.cid = s.sid)
   t2 = ( SELECT t1.deptid
               ~~SUM~~ if( selection cnt < 30 [illegible], 'small'
               ~~SUM~~ if (selection cnt < 80 ,'medium'
                   if (selectioncnt > 80, 'large'))) selection
          FROM t1  t1
          GROUP BY t2.deptid);

9) SELECT *
   FROM Professors p
   WHERE p.deptid IN (SELECT t2.deptid
                      FROM t2, Professors pp
                      WHERE t2. large > (t2.small + t2.medium)
                            AND t2.deptid = pp.deptid
                      HAVING COUNT (*) >20);
```

```
10) fail (SELECT e.cid, e.section,
          AVG (if(e.grade IN ('D','F'),1.0) > AS 'percent_failed'
          FROM Enrollment e
          GROUP BY e.cid, e.section);
```

```
11) SELECT p.name
    FROM fail f, Teaches t, Professors p,
    WHERE f.cid =t.cid AND f.section = t.section AND t.ssn = p.ssn
          AND f.percent_failed = (SELECT MAX(tt percent_failed
                                  FROM Teaches tt);
```

```
12) SELECT AVG (percFail)
    FROM (SELECT EXISTS
                (SELECT *
                 FROM Students s, Enrollment e
                 WHERE s.sid = e.sid AND (e.grade IN ('D','F'))
                ) AS 'Failed'
         ) AS 'percFail';
```

```
13) SELECT f. cid, f. section
    FROM fail, f
    WHERE f. percent_failed > (SELECT AVG( ff. percent_failed)
                               FROM fail ff);
```

```
14) SELECT c.deptid. AVG( count(*) Students) AS 'SPS',
      ((100.0 * SUM(count(*) grade = 'A')/SUM((count(*) Students) AS 'pA',
      ((100 * SUM(count(*) grade = 'B')/SUM (count (*) Students) AS 'pB',
      ((100 * SUM (count(*)grade = 'C' )/ SUM((cant(*) Students) AS 'pC',
      ((100* SUM (count (*) grade ='D)/ SUM ((count(*) Students) AS 'pD',
      ((100* SUM (count(*)grade= 'F'/SUM (count(*) Students) AS 'pF'
    FROM Courses c
    GROUP BY c.deptid;
```