1. Create Schema:

CREATE TABLE students( sid CHAR(10),

sname VARCHAR(40),

age INT,

gpa REAL,

PRIMARY KEY(sid));

CREATE TABLE courses( cid CHAR(10),

deptid VARCHAR(10),

description VARCHAR(10),

PRIMARY KEY(cid));

CREATE TABLE professors( ssn CHAR(10),

pname VARCHAR(40),

address VARCHAR(50),

phone CHAR(15),

deptid VARCHAR(10),

PRIMARY KEY(ssn));

CREATE TABLE enrollment( sid CHAR(10),

cid CHAR(10),

section CHAR(5),

grade REAL,

PRIMARY KEY(sid, cid, section),

FOREIGN KEY(sid) REFERENCES students(sid),

FOREIGN KEY(cid) REFERENCES courses(cid));

CREATE TABLE teaches( cid CHAR(10),

section CHAR(5),

ssn CHAR(10),

PRIMARY KEY(cid, section),

FOREIGN KEY(cid) REFERENCES enrollment(cid),

FOREIGN KEY(section) REFERENCES enrollment(section));

2. Name of professors that work in the CS department

SELECT P.pname

FROM professors P

WHERE P.deptid = ‘cs’;

3. Students(sid) enrolled in courses in the CS department

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. SSN and name of professors that work for cs department, but are not teaching any cs courses.

SELECT P.ssn, P.pname

FROM professors P, courses C

WHERE P.deptid = ‘cs’

EXCEPT

SELECT P.ssn, P.pname

FROM professors P2, courses C2, teaches T

WHERE P2.deptid = ‘cs’

AND C2.deptid = ’cs’

AND T.cid = C2.cid

AND T.ssn = P2.ssn;

5. List of number of courses offered by each department.

SELECT C.deptid, count(\*)

FROM courses C

GROUP BY C.deptid;

6. Departments that offer more than 10 courses

SELECT DISTINCT C.deptid

FROM courses C

GROUP BY C.deptid

HAVING 10 < (SELECT count(\*)

FROM courses C2

WHERE C2.deptid = C.deptid);

7. Names of students whose professor’s name starts with an M (no duplicates).

SELECT DISTINCT S.name

FROM students S, professors P, teaches T, enrollment E

WHERE S.sid = E.sid AND P.ssn = T.ssn AND E.cid = T.cid AND E.section = T.section AND P.pname LIKE ‘M%’

8.

a = (SELECT DISTINCT E.section, E.cid, count(\*) S.sid AS selection\_count

FROM enrollment E, students S

WHERE E.sid = S.sid)

b = (SELECT a.deptid, SUM( if( selection\_count < 30, 1, 0)) AS ‘small’,

SUM( if( selection\_count>30 AND selection\_count < 80, 1, 0)) AS ‘medium’,

SUM( if( selection\_count>80, 1, 0)) AS ‘large’

FROM a

GROUP BYa.deptid);

9. SELECT \*

FROM Professors P

WHERE P.deptid IN (SELECT b. deptid

FROM b, professors P2

WHERE b.large > (b.small + b.medium)

AND b.deptid = P2.deptid

HAVING COUNT(\*) >20);

10. f = (SELECT E.cid, E.section, AVG( if( E.grade IN (‘D’,’F’), 1, 0) AS percent\_failed

FROM enrollement E

GROUP BY E.cid, E.section

11. SELECT P.name

FROM f a, teaches T, professors P

WHERE a.cid = t.cid AND a.section = t.section AND t.ssn = p.ssn AND a.percent\_failed = (SELECT MAX(b.percent\_failed)

FROM f b);

12. SELECT AVG(pf.failed

FROM (SELECT EXISTS

(SELECT \*

FROM students S, enrollment E

WHERE S.sid = E.sid AND (E.grade IN (‘D’,’F’))) AS ‘Failed’

) AS ‘pf’

)

13. SELECT a.cid, a.section

FROM f a

WHERE a.percent\_failed > (SELECT AVG(b.percent\_failed) FROM f b);

14. SELECT C.deptid, AVG(num.students) AS ‘SPS’,

((100.0 \* SUM(num.gradeA))/SUM(num.s\_total) AS ‘p.A’,

((100.0 \* SUM(num.gradeB))/SUM(num.s\_total) AS ‘p.B’,

((100.0 \* SUM(num.gradeC))/SUM(num.s\_total) AS ‘p.C’,

((100.0 \* SUM(num.gradeD))/SUM(num.s\_total) AS ‘p.D’,

((100.0 \* SUM(num.gradeF))/SUM(num.s\_total) AS ‘p.F’,

FROM courses C,

(SELECT E.cid, E.section, COUNT(\*)AS ‘s\_total’,

SUM(if (e.grade = ‘A’,1,0)) AS ‘gradeA’,

SUM(if (e.grade = ‘B’,1,0)) AS ‘gradeB’,

SUM(if (e.grade = ‘C’,1,0)) AS ‘gradeC’,

SUM(if (e.grade = ‘D’,1,0)) AS ‘gradeD’,

SUM(if (e.grade = ‘F’,1,0)) AS ‘gradeF’)