Select \* from STUDENT;

SELECT \* FROM FACULTY;

SELECT \* FROM ENROLLMENT;

SELECT \* FROM OFFERING;

SELECT \* FROM COURSE;

/\* Get the faculty numbers of those faculty members who have ever taught a Monday

morning class (classes starting before noon on Mondays).

\*Note that your query must work for any future course offerings too. Not just for the

data you see in the database. \*/

SELECT DISTINCT F.FACNO FROM FACULTY F, OFFERING O

WHERE F.FACNO = O.FACNO

AND OFFDAYS LIKE '%M%'

AND OFFTIME < '12:00:00';

/\* Find the student id numbers and names of students who did not take a course

offered by a full professor (PROF). \*/

SELECT DISTINCT S.STDNO, S.STDFIRSTNAME, S.STDLASTNAME FROM STUDENT S

WHERE S.STDNO NOT IN (SELECT S.STDNO FROM STUDENT S, ENROLLMENT E, OFFERING O, FACULTY F

WHERE S.STDNO = E.STDNO AND E.OFFERNO = O.OFFERNO AND F.FACNO = O.FACNO

AND F.FACRANK = 'PROF')

ORDER BY S.STDNO;

/\* Get the faculty numbers, names, ranks, and departments of faculty who taught any

course in any SUMMER semester and order them by ascending order of ranks and

departments. \*/

SELECT F.FACNO, F.FACFIRSTNAME, F.FACLASTNAME, F.FACRANK, F.FACDEPT FROM FACULTY F, OFFERING O

WHERE F.FACNO = O.FACNO

AND O.OFFTERM = 'SUMMER'

ORDER BY (

CASE

WHEN F.FACRANK ='ASST' THEN 0

WHEN F.FACRANK ='ASSC' THEN 1

WHEN F.FACRANK ='PROF' THEN 2

END), F.FACDEPT ASC;

/\* Get the attending student numbers, course numbers, and course descriptions of

all summer courses that were not on Mondays.

\*Note that your query must work for any future course offerings too. Not just for the

data you see in the database. \*/

SELECT /\*DISTINCT\*/E.STDNO, C.COURSENO, C.CRSDESC FROM OFFERING O ,COURSE C, ENROLLMENT E

WHERE C.COURSENO = O.COURSENO

AND O.OFFERNO = E.OFFERNO

AND O.OFFTERM = 'SUMMER'

AND O.OFFDAYS NOT IN (SELECT OFFDAYS FROM OFFERING WHERE OFFDAYS LIKE '%M%');

/\* Did any student take more than two courses in any given semester? If so display

the student numbers, number of courses taken, and the term details. \*/

SELECT E.STDNO, COUNT(O.COURSENO) AS NUMCOURSES, O.OFFTERM, O.OFFYEAR FROM ENROLLMENT E, OFFERING O

WHERE O.OFFERNO = E.OFFERNO

GROUP BY E.STDNO, O.OFFTERM, O.OFFYEAR

HAVING COUNT(O.COURSENO) > 2;

/\* How many students have GPAs greater than 3.0? Report per major. \*/

SELECT S.STDMAJOR, COUNT(S.STDNO) AS NUMSTUDENTS FROM STUDENT S

WHERE S.STDGPA > 3

GROUP BY S.STDMAJOR;

/\* For each supervisor display the supervisor's number, name, rank and the

number of subordinates he/she has at each rank as given below. \*/

SELECT F.FACSUPERVISOR AS FACNO, (F1.FACFIRSTNAME||' '||F1.FACLASTNAME||'-'||F1.FACRANK) AS "SUPERVISORNAME - RANK", F.FACRANK AS SUBRANK, COUNT(F.FACNO) AS NUMOFSUBR FROM FACULTY F, FACULTY F1

WHERE F.FACSUPERVISOR = F1.FACNO

GROUP BY F.FACSUPERVISOR, (F1.FACFIRSTNAME||' '||F1.FACLASTNAME||'-'||F1.FACRANK), F.FACRANK;

/\* For each faculty display the faculty's number, name, rank and the number of

subordinates he/she has at each rank as given below. \*/

SELECT F1.FACNO, (F1.FACFIRSTNAME||' '||F1.FACLASTNAME||'-'||F1.FACRANK) AS "SUPERVISORNAME - RANK", F2.FACRANK AS SUBRANK, COUNT(F1.FACSUPERVISOR) AS NUMOFSUBR

FROM FACULTY F1 LEFT OUTER JOIN FACULTY F2

ON F1.FACNO = F2.FACSUPERVISOR

GROUP BY F1.FACNO, (F1.FACFIRSTNAME||' '||F1.FACLASTNAME||'-'||F1.FACRANK), F2.FACRANK;

/\* Find the number of students in each student class (level) and count the number of

different courses these students have taken so far? Make sure the order the results by student class as shown.

(Note: This query may require the use of COUNT(DISTINCT...), which does not work in MS Access. Even if you use MS Access,

write a standard SQL script for this question.) \*/

SELECT S.STDCLASS,COUNT(DISTINCT S.STDNO) AS NUMSTUDENTS,COUNT(DISTINCT C.COURSENO) AS NUMCOURSES FROM STUDENT S, ENROLLMENT E, OFFERING O, COURSE C

WHERE S.STDNO = E.STDNO

AND O.OFFERNO = E.OFFERNO

AND O.COURSENO = C.COURSENO

GROUP BY S.STDCLASS

ORDER BY (

CASE

WHEN S.STDCLASS ='FR' THEN 0

WHEN S.STDCLASS ='SO' THEN 1

WHEN S.STDCLASS ='JR' THEN 2

WHEN S.STDCLASS ='SR' THEN 3

END);

/\* For each course find the number of different student classes who have taken it.

List the results from the most variation to the least variation in student classes. \*/

SELECT C.COURSENO, COUNT(DISTINCT S.STDCLASS) AS NUMLEVELS FROM STUDENT S,OFFERING O,ENROLLMENT E, COURSE C

WHERE S.STDNO = E.STDNO

AND O.OFFERNO = E.OFFERNO

AND O.COURSENO = C.COURSENO

GROUP BY C.COURSENO

ORDER BY NUMLEVELS DESC;

/\* List the course number, offering number, and the average grade of students

enrolled in FALL IS course offerings \*/

SELECT O.COURSENO, O.OFFERNO, TO\_CHAR(AVG(E.ENRGRADE),9.99) AS AVGGRADE FROM OFFERING O, ENROLLMENT E

WHERE O.OFFERNO = E.OFFERNO

AND O.OFFTERM = 'FALL'

GROUP BY O.COURSENO, O.OFFERNO;

/\* Report the faculty number and name of the professor, course number(s),

number of offerings of that course taught, number of total students, average grade;

for all courses offered. Order by faculty names. \*/

SELECT F.FACNO, F.FACFIRSTNAME, F.FACLASTNAME, C.COURSENO, C.CRSDESC, COUNT(DISTINCT O.OFFERNO) AS NUMSECTION,COUNT(E.STDNO) AS STDCOUNT, TO\_CHAR(AVG(E.ENRGRADE),9.99) AS OVERALLAVERAGEGPA

FROM OFFERING O LEFT OUTER JOIN COURSE C ON C.COURSENO = O.COURSENO

LEFT OUTER JOIN FACULTY F ON F.FACNO = O.FACNO

LEFT OUTER JOIN ENROLLMENT E ON O.OFFERNO = E.OFFERNO

GROUP BY F.FACNO, F.FACFIRSTNAME, F.FACLASTNAME, C.COURSENO, C.CRSDESC

ORDER BY F.FACNO;

/\* Find the faculty number, name, salary of each supervisor compared to the

average salaries of his/her subordinates. \*/

SELECT F1.FACSUPERVISOR, (F2.FACFIRSTNAME||' '||F2.FACLASTNAME) AS FACNAME, F2.FACSALARY, AVG(F1.FACSALARY) AS AVGSUBSALARY

FROM FACULTY F1, FACULTY F2

WHERE F1.FACSUPERVISOR = F2.FACNO

GROUP BY F1.FACSUPERVISOR, (F2.FACFIRSTNAME||' '||F2.FACLASTNAME), F2.FACSALARY;

/\* Find the supervisor whose salary is the most different (higher) compared to the

average salary of his/her subordinates in dollar terms. \*/

SELECT F1.FACSUPERVISOR, (F2.FACFIRSTNAME||' '||F2.FACLASTNAME) AS FACNAME, F2.FACSALARY AS SUPRSALARY, (AVG(F1.FACSALARY)) AS AVGSUBRSALRY, (F2.FACSALARY-(AVG(F1.FACSALARY))) AS DOLLARDIFFERENCE

FROM FACULTY F1, FACULTY F2

WHERE F1.FACSUPERVISOR = F2.FACNO

GROUP BY F1.FACSUPERVISOR, (F2.FACFIRSTNAME||' '||F2.FACLASTNAME), F2.FACSALARY

HAVING (F2.FACSALARY-(AVG(F1.FACSALARY))) = (SELECT MAX ((F4.FACSALARY-(AVG(F3.FACSALARY))))

FROM FACULTY F3, FACULTY F4 WHERE F3.FACSUPERVISOR = F4.FACNO GROUP BY F4.FACSALARY);

/\* Find the supervisor whose salary is the most different (higher) than the average

salary of his/her subordinates in percentage terms. \*/

SELECT F1.FACSUPERVISOR,(F2.FACFIRSTNAME||' '||F2.FACLASTNAME) AS FACNAME, F2.FACSALARY AS SUPRSALARY,(AVG(F1.FACSALARY)) AS AVGSUBRSALRY,

TO\_CHAR(((F2.FACSALARY - AVG(F1.FACSALARY))/AVG(F1.FACSALARY)) \* 100,99.99) || '%' AS PERCENTDIFFERNCE FROM FACULTY F1, FACULTY F2

WHERE F1.FACSUPERVISOR = F2.FACNO

GROUP BY F1.FACSUPERVISOR, (F2.FACFIRSTNAME||' '||F2.FACLASTNAME), F2.FACSALARY

HAVING (((F2.FACSALARY - AVG(F1.FACSALARY))/AVG(F1.FACSALARY)) \* 100) = (SELECT MAX (((F4.FACSALARY - AVG(F3.FACSALARY))/AVG(F3.FACSALARY)) \* 100)

FROM FACULTY F3, FACULTY F4 WHERE F3.FACSUPERVISOR = F4.FACNO GROUP BY F4.FACSALARY);

/\* List the names of the faculty members who have not taught the same course as

any of his or her subordinates or supervisors in 2013.

Do not include the faculty members who do not have a supervisor. \*/

SELECT F.FACNO

FROM FACULTY F

WHERE F.FACNO NOT IN

(SELECT F1.FACNO

FROM FACULTY F1, OFFERING O1, OFFERING O2

WHERE F1.FACNO = O1.FACNO

AND F1.FACSUPERVISOR = O2.FACNO

AND O1.COURSENO = O2.COURSENO

AND F1.FACSUPERVISOR IS NOT NULL

AND O1.OFFYEAR = 2013

AND O2.OFFYEAR = 2013)

AND F.FACNO NOT IN

(SELECT F2.FACSUPERVISOR

FROM FACULTY F2, OFFERING O1, OFFERING O2

WHERE F2.FACNO = O1.FACNO

AND F2.FACSUPERVISOR = O2.FACNO

AND O1.CourseNo = O2.COURSENO

AND F2.FACSUPERVISOR IS NOT NULL

AND O1.OFFYEAR = 2013

AND O2.OFFYEAR = 2013);

/\* Get the student numbers and names of students who took all 2013 IS courses

that were offered and had at least one enrollment. If an offering did not have any

enrollments, you may ignore it as no student had a chance to enroll in that course \*/

SELECT S.STDNO, S.STDFIRSTNAME, S.STDLASTNAME

FROM STUDENT S

WHERE NOT EXISTS

(SELECT COURSE.COURSE.NO

FROM COURSE, OFFERING

WHERE COURSE.COURSENO LIKE 'IS%'

AND COURSE.COURSENO =

OFFERING.COURSENO

AND OFFERING.OFFYEAR = 2013

AND NOT EXISTS (SELECT OFFERING.OFFERNO

FROM OFFERING, ENROLLMENT

WHERE ENROLLMENT.OfferNo = OFFERING.OFFERNO

AND ENROLLMENT.STDNO = STUDENT.STDNO

AND OFFERING.COURSENO = COURSE.COURSENO));

/\* Get the course numbers of courses that have NOT been taken by a single student

from Tacoma or Redmond(cities)so far.\*/

SELECT C.COURSENO FROM COURSE C WHERE C.COURSENO NOT IN

(SELECT C.COURSENO FROM COURSE C, OFFERING O, ENROLLMENT E, STUDENT S

WHERE C.COURSENO = O.COURSENO AND O.OFFERNO = E.OFFERNO AND S.STDNO = E.STDNO

AND (S.STDCITY = 'TACOMA' OR S.STDCITY = 'REDMOND'));

/\* For each room of the BLM building (OffLocation), report the number of students

who have attended at least a section of a course (offering) in that room. \*/

SELECT SUBSTR(O.OFFLOCATION,4,3) AS ROOM, COUNT(DISTINCT E.STDNO) FROM OFFERING O, ENROLLMENT E

WHERE E.OFFERNO = O.OFFERNO

GROUP BY SUBSTR(O.OFFLOCATION,4,3)

HAVING COUNT(E.OFFERNO) >=1;

/\* For each room of the BLM building (OffLocation), report the number of students

who have attended multiple sections(at least two offerings of a course) in that same

room. \*/

SELECT SUBSTR(OFFLOCATION,4,3) AS ROOM, COUNT(\*) AS NUMOFSTUDENTS FROM

(SELECT E.STDNO, OFFERING.OFFLOCATION, COUNT(\*) AS S

FROM ENROLLMENT E, OFFERING

WHERE E.OFFERNO = OFFERING.OFFERNO

GROUP BY E.STDNO, OFFERING.OFFLOCATION

HAVING COUNT(\*)>1) T GROUP BY SUBSTR(OFFLOCATION,4,3) ;