DB TEST 2 STUDY GUIDE

/\*(10A)

Find the ssn and last name of every employee who doesn't have a  supervisor, or his last name contains at least two occurences of the letter 'a'. Sort the results by ssn.

\*/

SELECT E.ssn, E.lname

FROM   employee E

WHERE  E.super\_ssn IS NULL OR E.lname LIKE '%a%a%'

ORDER BY E.ssn;

--

-- JOINING 3 TABLES ------------------------------

--

/\*(11A)

For every employee who works more than 30 hours on any project: Find the ssn, lname, project number, project name, and numer of hours. Sort the results by ssn.

\*/

SELECT E.ssn, E.lname, P.pnumber, P.pname, W.hours

FROM   employee E, works\_on W, project P

WHERE  E.ssn = W.essn AND

       W.hours > 30 AND

       W.pno = P.pnumber

ORDER BY E.ssn;

--

-- JOINING 3 TABLES ---------------------------

--

/\*(12A)

Write a query that consists of one block only.

For every employee who works on a project that is not controlled by the department he works for: Find the employee's lname, the department he works for, the project number that he works on, and the number of the department that controls that project. Sort the results by lname.

\*/

SELECT E.lname, E.dno, P.pnumber, P.dnum

FROM   employee E, works\_on W, project P

WHERE  E.ssn = W.essn AND

       W.pno = P.pnumber AND

       E.dno != P.dnum

ORDER BY E.lname;

--

-- JOINING 4 TABLES -------------------------

--

/\*(13A)

For every employee who works for more than 20 hours on any project that is located in the same location as his department: Find the ssn, lname, project number, project location, department number, and department location.Sort the results by lname

\*/

SELECT DISTINCT E.ssn, E.lname, W.pno, P.plocation, E.dno, L.dlocation

FROM   employee E, dept\_locations L, project P, works\_on W

WHERE  E.ssn = W.essn AND

       E.dno = L.dnumber AND

       W.pno = P.pnumber AND

       P.plocation = L.dlocation AND

       W.hours > 20

ORDER BY E.lname;

--

-- SELF JOIN -------------------------------------------

--

/\*(14A)

Write a query that consists of one block only.

For every employee whose salary is less than 70% of his immediate supervisor's salary: Find his ssn, lname, salary; and his supervisor's ssn, lname, and salary. Sort the results by ssn.

\*/

SELECT E.ssn, E.lname, E.salary, S.ssn, S.lname, S.salary

FROM   employee E, employee S

WHERE  E.super\_ssn = S.ssn AND

       E.salary < 0.7 \* S.salary

ORDER BY E.ssn;

--

-- USING MORE THAN ONE RANGE VARIABLE ON ONE TABLE -------------------

--

/\*(15A)

For projects located in Houston: Find pairs of last names such that the two employees in the pair work on the same project. Remove duplicates. Sort the result by the lname in the left column in the result.

\*/

SELECT E1.lname, E2.lname

FROM   employee E1, employee E2, works\_on W1, works\_on W2, project P

WHERE  E1.ssn = W1.essn AND E2.ssn = W2.essn AND

       W1.pno = W2.pno AND W1.pno = P.pnumber AND

       p.plocation = 'Houston' AND

       E1.ssn > E2.ssn

ORDER BY E1.lname;

--

/\*(16A) Hint: A NULL in the hours column should be considered as zero hours.

Find the ssn, lname, and the total number of hours worked on projects for every employee whose total is less than 40 hours. Sort the result by lname

\*/

SELECT E.ssn, E.lname, SUM(W.hours)

FROM   employee E, works\_on W

WHERE  E.ssn = W.essn

GROUP BY E.ssn, E.lname

HAVING SUM(W.hours) < 40 OR SUM(W.hours) IS NULL

ORDER BY E.lname;

--

-- GROUP BY 2 ----------------------------------

--

/\*(17A)

For every project that has more than 2 employees working on it: Find the project number, project name, number of employees working on it, and the total number of hours worked by all employees on that project. Sort the results by project number.

\*/

SELECT P.pnumber, P.pname, COUNT(\*), SUM (W.hours)

FROM   works\_on W, project P

WHERE  W.pno = P.pnumber

GROUP BY P.pnumber, P.pname

HAVING COUNT(\*) > 2

ORDER BY P.pnumber;

--

-- CORRELATED SUBQUERY --------------------------------

--

/\*(18A)

For every employee who has the highest salary in his department: Find the dno, ssn, lname, and salary . Sort the results by department number.

\*/

SELECT E1.dno, E1.ssn, E1.lname, E1.salary

FROM   employee E1

WHERE  E1.salary =

       (SELECT MAX(E2.salary)

        FROM   Employee E2

        WHERE  E1.dno = E2.dno)

ORDER BY E1.dno;

--

-- NON-CORRELATED SUBQUERY -------------------------------

--

/\*(19A)

For every employee who does not work on any project that is located in Houston: Find the ssn and lname. Sort the results by lname

\*/

SELECT E.ssn, E.lname

FROM   employee E

WHERE  E.ssn NOT IN

       (SELECT W.essn

        FROM   works\_on W, project P

        WHERE  W.pno = P.pnumber AND P.plocation = 'Houston')

ORDER BY E.lname;

--

-- DIVISION ---------------------------------------------

--

/\*(20A) Hint: This is a DIVISION query

For every employee who works on every project that is located in Stafford: Find the ssn and lname. Sort the results by lname

\*/

SELECT E.ssn, E.lname

FROM   employee E

WHERE  NOT EXISTS (

       (SELECT P.pnumber

        FROM   project P

        WHERE  P.plocation = 'Stafford')

       MINUS

       (SELECT W.pno

        FROM works\_on W

        WHERE  E.ssn = W.essn))

ORDER BY E.lname;

/\*(110) Using ROWNUM to limit the size of the result. (Notice that SQL and some systems use the LIMIT or TOP clauses. Oracle uses ROENUM to accomplish similar tasks.)

Find the ssn, lname, and salary of only four employees.

\*/

-- << write your sql code here >>>

SELECT ssn, lname, salary

FROM   Employee

WHERE  ROWNUM < 5;

--

/\*(115) TOP-N query.

Find the ssn, lname, and salary of the four highest paid employees.

\*/

-- << write your sql code here >>>

SELECT  ssn, lname, salary

FROM   (SELECT \* FROM Employee ORDER BY salary DESC)

WHERE   ROWNUM < 5;

--

/\*(120) TOP-N query.

Find the ssn, lname, and salary of the four lowest paid employees

\*/

-- << write your sql code here >>>

SELECT  ssn, lname, salary

FROM   (SELECT \* FROM Employee ORDER BY salary)

WHERE   ROWNUM < 5;

--

/\*(125) TOP-N query.

Find the lowest two salaries in the company.(Notice that in our database, the two lowest salaries are 25K and 30K.)

\*/

-- << write your sql code here >>>

SELECT salary

FROM   (SELECT DISTINCT salary FROM Employee ORDER BY salary)

WHERE ROWNUM < 3;

/\*(130) TOP-N query.

For every employee whose salary is equal to one of the two lowest salaries, Find the ssn, lname, and salary.

\*/

-- << write your sql code here >>>

SELECT ssn, lname, salary

FROM   employee

WHERE salary in

      (SELECT salary

       FROM (SELECT DISTINCT salary FROM Employee ORDER BY salary)

       WHERE ROWNUM < 3)

ORDER BY salary;

--

/\*(135) RANK query

Find the rank of the salary 30000 among all salaries. (HINT: The ranks in our database are 1 for 25000, 4 for 30000, 5 for 38000, and so on.)

\*/

-- << write your sql code here >>>

SELECT RANK (30000) WITHIN GROUP

       (ORDER BY salary) "Rank of salary 30000"

FROM employee;

--

/\*(140) RANK query ... compare with the previous query.

Find the rank of the salary 31000 among all salaries.

\*/

-- << write your sql code here >>>

SELECT RANK (31000) WITHIN GROUP

       (ORDER BY salary) "Rank of salary 31000"

FROM employee;

--

/\*(145) DENSE RANK query

Find the dense rank of the salary 30000 among all salaries. Hint: The dense ranks in our database are 1 for 25000, 2 for 30000, 3 for 38000, and so on.

\*/

-- << write your sql code here >>>

SELECT DENSE\_RANK (30000) WITHIN GROUP

       (ORDER BY salary) "Rank of salary 30000"

FROM employee;

--

/\*(150) DENSE RANK query ... compare with the previous query.

Find the dense rank of the salary 31000 among all salaries. Hint: The dense ranks in our database are 1 for 25000, 2 for 30000, 3 for 38000, and so on.

\*/

-- << write your sql code here >>>

SELECT DENSE\_RANK (31000) WITHIN GROUP

       (ORDER BY salary) "Rank of salary 31000"

FROM employee;

--

/\*(155)HIERARCHICAL query (uses START WITH and CONNECT BY PRIOR)

Find pairs of SSN's such that the first SSN in the pair is that of an employee while the second SSN in the pair is that of his/her supervisor. Start with SSN 453453453.

Hint: The output of your query should be:

453453453     333445555

333445555     888665555

888665555     -

\*/

-- << write your sql code here >>>

SELECT ssn, super\_ssn

FROM Employee

START WITH ssn = 453453453

CONNECT BY PRIOR super\_ssn = ssn;