SQL – ORACLE 11G XE ON SHELL

Content

Datasets to work with.

```
CREATE TABLE EMP
(EMPNO NUMBER(6) PRIMARY KEY,
ENAME VARCHAR2(10),
JOB VARCHAR2(9),
MGR NUMBER(9),
HIREDATE DATE,
SAL NUMBER(9),
COMM NUMBER(9),
DEPTNO NUMBER(9) REFERENCES DEPT ON DELETE CASCADE);
```

```
SQL> DESC EMP;
 Name                                                      Null?    Type
 --------------------------------------------------------- -------- ------------
 EMPNO                                                     NOT NULL NUMBER(6)
 ENAME                                                              VARCHAR2(10)
 JOB                                                                VARCHAR2(9)
 MGR                                                                NUMBER(9)
 HIREDATE                                                           DATE
 SAL                                                                NUMBER(9)
 COMM                                                               NUMBER(9)
 DEPTNO                                                             NUMBER(9)
```

```
INSERT INTO EMP VALUES(
7369,'SMITH','CLERK',7902,'17/12/80',800,NULL,20);
INSERT INTO EMP VALUES(
7499,'ALLEN','SALESMAN',7698,'20/02/81',1600,300,30);
INSERT INTO EMP VALUES(
7521,'WARD','SALESMAN',7698,'22/02/81',1250,500,30);
INSERT INTO EMP VALUES(
7566,'JONES','MANAGER',7839,'02/04/81',2975,NULL,20);
INSERT INTO EMP VALUES(
7654,'MARTIN','SALESMAN',7698,'28/09/81',1250,1400,30);
INSERT INTO EMP VALUES(
7698,'BLAKE','MANAGER',7839,'01/05/81',2850,NULL,30);
INSERT INTO EMP VALUES(
7782,'CLARK','MANAGER',7839,'09/06/81',2450,NULL,10);
INSERT INTO EMP VALUES(
7788,'SCOTT','ANALYST',7566,'19/04/87',3000,NULL,20);
INSERT INTO EMP VALUES(
7839,'KING','PRESIDENT',NULL,'17/11/81',5000,NULL,10);
INSERT INTO EMP VALUES(
7844,'TURNER','SALESMAN',7698,'08/09/81',1500,0,30);
INSERT INTO EMP VALUES(
7876,'ADAMS','CLERK',7788,'23/05/87',1100,NULL,20);
INSERT INTO EMP VALUES(
7900,'JAMES','CLERK',7698,'03/12/81',950,NULL,30);
INSERT INTO EMP VALUES(
```

7902,'FORD','ANALYST',7566,'03/12/81',3000,NULL,20);
INSERT INTO EMP VALUES(
7934,'MILLER','CLERK',7782,'23/01/82',1300,NULL,10);

```
SQL> select * FROM EMP;

     EMPNO ENAME      JOB              MGR HIREDATE        SAL       COMM     DEPTNO
---------- ---------- --------- ---------- -------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17/12/80        800                    20
      7499 ALLEN      SALESMAN        7698 20/02/81       1600        300         30
      7521 WARD       SALESMAN        7698 22/02/81       1250        500         30
      7566 JONES      MANAGER         7839 02/04/81       2975                    20
      7654 MARTIN     SALESMAN        7698 28/09/81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01/05/81       2850                    30
      7782 CLARK      MANAGER         7839 09/06/81       2450                    10
      7788 SCOTT      ANALYST         7566 19/04/87       3000                    20
      7876 ADAMS      CLERK           7788 23/05/87       1100                    20
      7900 JAMES      CLERK           7698 03/12/81        950                    30
      7902 FORD       ANALYST         7566 03/12/81       3000                    20
      7934 MILLER     CLERK           7782 23/01/82       1300                    10
      7839 KING       PRESIDENT            17/11/81       5000                    10
      7844 TURNER     SALESMAN        7698 08/09/81       1500          0         30
```

CREATE TABLE DEPT
(DEPTNO NUMBER(9) PRIMARY KEY,
DNAME VARCHAR2(14),
LOC VARCHAR2(12));

```
SQL> DESC DEPT;
 Name                                                      Null?    Type
 --------------------------------------------------------- -------- ------------
 DEPTNO                                                    NOT NULL NUMBER(9)
 DNAME                                                              VARCHAR2(14)
 LOC                                                                VARCHAR2(12)
```

INSERT INTO DEPT VALUES (
10, 'ACCOUNTING', 'NEW YORK');
INSERT INTO DEPT VALUES (
20, 'RESEARCH', 'DALLAS');
INSERT INTO DEPT VALUES (
30, 'SALES', 'CHICAGO');
INSERT INTO DEPT VALUES (
40, 'OPERATIONS', 'BOSTON');

```
SQL> SELECT * FROM DEPT;

    DEPTNO DNAME          LOC
---------- -------------- ------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
```

```
CREATE TABLE OFICIO(JOB VARCHAR2(9) PRIMARY KEY,
FUNCIONES VARCHAR2(20),
CATEGORIA NUMBER(1),
EMPLEADOS NUMBER(1));
```

```
SQL> DESC OFICIO;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ------------
 JOB                                                   NOT NULL VARCHAR2(15)
 FUNCIONES                                                      VARCHAR2(20)
 CATEGORIA                                                      NUMBER(1)
 EMPLEADOS                                                      NUMBER(1)
```

```
INSERT INTO OFICIO VALUES(
'CLERK','DEPENDIENTE',1,4);
```

```
SQL> SELECT * FROM OFICIO;

JOB             FUNCIONES            CATEGORIA  EMPLEADOS
--------------- -------------------- ---------- ----------
CLERK           DEPENDIENTE                   1          4
SALESMAN        VENDEDOR                      2          4
ANALYST         ECONOMISTA                    3          2
MANAGER         RESPONSABLE                   4          3
PRESIDENT       FUNDADOR                      5          1
```

```
ALTER TABLE EMP
ADD CONSTRAINT FK_OFICIO FOREIGN KEY(JOB)
REFERENCES OFICIO(JOB) ON DELETE CASCADE;
```

DQL EXERCISES

1. Indicates the employee code and the name of those earning a salary between 1000 and 2000, sorted by name from A to Z.

```
SELECT EMPNO,ENAME FROM EMP WHERE SAL BETWEEN 1000 AND 2000 ORDER BY ENAME ASC;
```

```
SQL> SELECT EMPNO,ENAME FROM EMP WHERE SAL BETWEEN 1000 AND 2000 ORDER BY ENAME ASC;

     EMPNO ENAME
---------- ----------
      7876 ADAMS
      7499 ALLEN
      7654 MARTIN
      7934 MILLER
      7844 TURNER
      7521 WARD
```

2. Indicates the code of the employees who have a commission.

```
SELECT EMPNO FROM EMP WHERE COMM <>0;
```

```
SQL> SELECT EMPNO FROM EMP WHERE COMM <>0;

     EMPNO
----------
      7499
      7521
      7654
```

3. Indicate the date of entry, name and commission of those whose salary is above 500, have non-zero commission and head.

```
SELECT ENAME,HIREDATE,COMM FROM EMP WHERE SAL>500 AND
COMM<>0 AND MGR IS NOT NULL;
```

```
SQL> SELECT ENAME,HIREDATE,COMM FROM EMP WHERE SAL>500 AND COMM<>0 AND MGR IS NOT NULL;

ENAME      HIREDATE       COMM
---------- -------- ----------
ALLEN      20/02/81        300
WARD       22/02/81        500
MARTIN     28/09/81       1400
```

4. Indicate the employees with their names who joined the company before 1/05/1981.

```
SELECT ENAME FROM EMP WHERE HIREDATE < '1/05/1981';
```

```
SQL> SELECT ENAME FROM EMP WHERE HIREDATE < '01/05/1981';

ENAME
----------
SMITH
ALLEN
WARD
JONES
```

5. Indicates the employee code in ascending order for employees hired between 1/1/1980 and 1/12/1981 as long as they are from department 10 or 20, have 7698 as their boss and are MANAGER or SALESMAN.

```
SELECT EMPNO FROM EMP WHERE HIREDATE BETWEEN '1/1/1980'
AND '1/12/1981' AND DEPTNO IN(10,20) AND JOB IN
('SALESMAN','MANAGER') AND MGR=7698 ORDER BY EMPNO;
```

```
SQL> SELECT EMPNO FROM EMP
  2  WHERE HIREDATE BETWEEN '1/1/1980' AND '1/12/1981' AND DEPTNO IN(10,20) AND JOB IN ('SALESMAN','MANAGER')
  3  AND MGR=7698 ORDER BY EMPNO;

no rows selected
```

6. List all employees, together with their bosses, as long as the boss is not KING.

SELECT * FROM EMP WHERE MGR<>(SELECT EMPNO FROM EMP WHERE ENAME='KING');

```
SQL> SELECT *
  2  FROM EMP
  3  WHERE MGR<>(SELECT EMPNO FROM EMP WHERE ENAME='KING');

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- -------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17/12/80        800                    20
      7499 ALLEN      SALESMAN        7698 20/02/81       1600        300         30
      7521 WARD       SALESMAN        7698 22/02/81       1250        500         30
      7654 MARTIN     SALESMAN        7698 28/09/81       1250       1400         30
      7788 SCOTT      ANALYST         7566 19/04/87       3000                    20
      7844 TURNER     SALESMAN        7698 08/09/81       1500          0         30
      7876 ADAMS      CLERK           7788 23/05/87       1100                    20
      7900 JAMES      CLERK           7698 03/12/81        950                    30
      7902 FORD       ANALYST         7566 03/12/81       3000                    20
      7934 MILLER     CLERK           7782 23/01/82       1300                    10
```

7. List the name, code, and salary of those whose occupation has an A in its second and penultimate character, or who are from department 10 or 20.

SELECT ENAME NOMBRE,EMPNO CODIGO_EMPLEADO,SAL SALARIO FROM EMP WHERE JOB LIKE '_A%A_' OR DEPTNO IN(10,20);

```
SQL> SELECT ENAME NOMBRE,EMPNO CODIGO_EMPLEADO,SAL SALARIO FROM EMP WHERE JOB LIKE '_A%A_' OR DEPTNO IN(10,20);

NOMBRE     CODIGO_EMPLEADO    SALARIO
---------- --------------- ----------
SMITH                 7369        800
ALLEN                 7499       1600
WARD                  7521       1250
JONES                 7566       2975
MARTIN                7654       1250
CLARK                 7782       2450
SCOTT                 7788       3000
ADAMS                 7876       1100
FORD                  7902       3000
MILLER                7934       1300
KING                  7839       5000
TURNER                7844       1500

12 rows selected.
```

8. Average the salary per department and order it from highest to lowest.

SELECT ROUND(AVG(SAL),2) SALARIO_MEDIO, DEPTNO DEPARTAMENTO FROM EMP GROUP BY DEPTNO ORDER BY AVG(SAL) DESC;

```
SQL> SELECT ROUND(AVG(SAL),2) SALARIO_MEDIO, DEPTNO DEPARTAMENTO FROM EMP GROUP BY DEPTNO ORDER BY AVG(SAL) DESC;

SALARIO_MEDIO DEPARTAMENTO
------------- ------------
      2916,67           10
         2175           20
      1566,67           30
```

9. Show the employee with his annualised salary and with the column name SAL YEAR. Taking the salary from the table as monthly. Then add the same, but with the commission.

SELECT ENAME, SAL*12 AS SAL_ANUAL, COMM*12 AS COM_ANUAL FROM EMP;

```
SQL> SELECT ENAME, SAL*12 AS SAL_ANUAL, COMM*12 AS COM_ANUAL FROM EMP;

ENAME      SAL_ANUAL  COM_ANUAL
---------- ---------- ----------
SMITH            9600
ALLEN           19200       3600
WARD            15000       6000
JONES           35700
MARTIN          15000      16800
BLAKE           34200
CLARK           29400
SCOTT           36000
ADAMS           13200
JAMES           11400
FORD            36000
MILLER          15600
KING            60000
TURNER          18000          0

14 rows selected.
```

10. Select the different salaries existing in the department 30.

SELECT DISTINCT SAL FROM EMP WHERE DEPTNO=30;

```
SQL> SELECT DISTINCT SAL FROM EMP WHERE DEPTNO=30;

       SAL
----------
      1250
      2850
      1600
       950
      1500
```

11. Obtain the department number of those department names that have an A anywhere, or an E or an I, as long as they are Dallas or Chicago.

SELECT DEPTNO FROM DEPT WHERE LOC LIKE '%A%' OR LOC LIKE '%E%'OR LOC LIKE '%I%';

```
SQL> SELECT DEPTNO FROM DEPT WHERE LOC LIKE '%A%' OR LOC LIKE '%E%'OR LOC LIKE '%I%';

    DEPTNO
----------
        10
        20
        30
```

12. Obtain all data of employees who actually have a commission.

SELECT ENAME FROM EMP WHERE COMM<>0;

```
SQL> SELECT * FROM EMP WHERE COMM<>0;

     EMPNO ENAME      JOB              MGR HIREDATE        SAL       COMM     DEPTNO
---------- ---------- --------- ---------- -------- ---------- ---------- ----------
      7499 ALLEN      SALESMAN         7698 20/02/81       1600        300         30
      7521 WARD       SALESMAN         7698 22/02/81       1250        500         30
      7654 MARTIN     SALESMAN         7698 28/09/81       1250       1400         30
```

13. Select the name and salary of employees with the aliases NAME and ANNUAL SALARY of those who are SALESMAN or MANAGER, joined the company before 1-1-83. Sort by oldest date first.

SELECT ENAME NOMBRE, SAL SALARIO FROM EMP WHERE JOB IN ('SALESMAN','MANAGER') AND HIREDATE< '01/01/83' ORDER BY HIREDATE DESC;

```
SQL> SELECT ENAME NOMBRE, SAL SALARIO FROM EMP WHERE JOB IN ('SALESMAN','MANAGER') AND HIREDATE< '01/01/83'
  2  ORDER BY HIREDATE DESC;

NOMBRE        SALARIO
---------- ----------
MARTIN           1250
TURNER           1500
CLARK            2450
BLAKE            2850
JONES            2975
WARD             1250
ALLEN            1600

7 rows selected.
```

14. Select the monthly salary increased by 5% cpi with the alias CPI SALARY of those whose salary is not between 3000 and 5000, or whose employee code has a 9 in its second character, or whose department is not 10 or 20, as long as they meet CLERK, ANALYST or SALESMAN. Sort by the lowest salary.

SELECT SAL*1.05 SALARIO_IPC FROM EMP WHERE (SAL NOT BETWEEN 3000 AND 5000 OR EMPNO LIKE '_9%' OR DEPTNO NOT IN (10,20)) AND JOB IN ('CLERK','ANALYST','SALESMAN') ORDER BY SAL ASC;

```
SQL> SELECT SAL*1.05 SALARIO_IPC FROM EMP WHERE (SAL NOT BETWEEN 3000 AND 5000 OR EMPNO LIKE '_9%' OR
  2  DEPTNO NOT IN (10,20)) AND JOB IN ('CLERK','ANALYST','SALESMAN') ORDER BY SAL ASC;

SALARIO_IPC
-----------
        840
     997,5
      1155
    1312,5
    1312,5
      1365
      1575
      1680
      3150

9 rows selected.
```

15. From the employees with zero commission, obtain the name of those whose boss is code 7839. Another condition to be fulfilled is that they joined the company between 1-1-80 and 31-12-83. Sort by name from A to Z.

```
SELECT ENAME NOMBRE FROM EMP WHERE (COMM=0 OR COMM IS
NULL) AND MGR=7839
AND HIREDATE BETWEEN '01/1/80' AND '31/12/83' ORDER BY
NOMBRE ASC;
```

```
SQL> SELECT ENAME NOMBRE FROM EMP WHERE (COMM=0 OR COMM IS NULL) AND MGR=7839
  2  AND HIREDATE BETWEEN '01/1/80' AND '31/12/83' ORDER BY NOMBRE ASC;

NOMBRE
----------
BLAKE
CLARK
JONES
```

16. Select the department number, name and monthly salary plus monthly commission (the given is annual), putting an alias called 'MONTHLY SALARY WITH COMMISSION' of the employees who are in the sales department, resident in Chicago, and whose position is salesman or clerk and, in any case, have a non-zero and non-zero commission. Order them by their commission from highest to lowest.

```
SELECT DEPTNO "Nº DEPARTAMENTO", ENAME NOMBRE,
(SAL+COMM)/12 "SALARIO MENSUAL CON COMISION"
FROM EMP
WHERE DEPTNO = (SELECT DEPTNO FROM DEPT WHERE
DNAME='SALES' AND LOC='CHICAGO')
AND JOB IN ('SALESMAN','CLERK') AND COMM IS NOT NULL AND
COMM<>0
ORDER BY COMM DESC;
```

```
SQL> SELECT DEPTNO "Nº DEPARTAMENTO", ENAME NOMBRE, (SAL+COMM)/12 "SALARIO MENSUAL CON COMISION"
  2  FROM EMP
  3  WHERE DEPTNO = (SELECT DEPTNO FROM DEPT WHERE DNAME='SALES' AND LOC='CHICAGO')
  4  AND JOB IN ('SALESMAN','CLERK') AND COMM IS NOT NULL AND COMM<>0
  5  ORDER BY COMM DESC;

Nº DEPARTAMENTO NOMBRE     SALARIO MENSUAL CON COMISION
--------------- ---------- ---------------------------
             30 MARTIN                       220,833333
             30 WARD                         145,833333
             30 ALLEN                        158,333333

SQL> SELECT DEPTNO "Nº DEPARTAMENTO", ENAME NOMBRE, (SAL+COMM)/12 "SALARIO MENSUAL CON COMISION"
  2  FROM EMP NATURAL JOIN DEPT
  3  WHERE DNAME='SALES' AND LOC='CHICAGO'
  4  AND JOB IN ('SALESMAN','CLERK') AND COMM IS NOT NULL AND COMM<>0
  5  ORDER BY COMM DESC;

Nº DEPARTAMENTO NOMBRE     SALARIO MENSUAL CON COMISION
--------------- ---------- ---------------------------
             30 MARTIN                       220,833333
             30 WARD                         145,833333
             30 ALLEN                        158,333333
```

```
SELECT DEPTNO "Nº DEPARTAMENTO", ENAME NOMBRE, (SAL+COMM)/12 "SALARIO MENSUAL CON COMISION"
FROM EMP NATURAL JOIN DEPT
WHERE DNAME='SALES' AND LOC='CHICAGO'
AND JOB IN ('SALESMAN','CLERK') AND COMM IS NOT NULL AND COMM<>0
ORDER BY COMM DESC;
```

17. Select the name of all employees with their corresponding department number only in the New York locality, sorted by department number.

```
SELECT DEPTNO "Nº DEPARTAMENTO", ENAME NOMBRE
FROM EMP NATURAL JOIN DEPT
WHERE LOC='NEW YORK' ORDER BY DEPTNO;
```

```
SQL> SELECT DEPTNO "Nº DEPARTAMENTO", ENAME NOMBRE
  2  FROM EMP NATURAL JOIN DEPT
  3  WHERE LOC='NEW YORK' ORDER BY DEPTNO;

Nº DEPARTAMENTO NOMBRE
--------------- ----------
             10 CLARK
             10 KING
             10 MILLER

SQL> SELECT DEPTNO "Nº DEPARTAMENTO", ENAME NOMBRE
  2  FROM EMP
  3  WHERE DEPTNO=(SELECT DEPTNO FROM DEPT WHERE LOC='NEW YORK')
  4  ORDER BY DEPTNO;

Nº DEPARTAMENTO NOMBRE
--------------- ----------
             10 CLARK
             10 KING
             10 MILLER
```

```
SELECT DEPTNO "Nº DEPARTAMENTO", ENAME NOMBRE
FROM EMP
WHERE DEPTNO=(SELECT DEPTNO FROM DEPT WHERE LOC='NEW YORK')
ORDER BY DEPTNO;
```

18. Select the name of all employees who have a monthly salary of 250 or more, provided they do not actually have a commission. Take the salary in the table as annual.

```
SELECT ENAME FROM EMP WHERE SAL/12>=250 AND (COMM=0 OR COMM IS NULL);
```

```
SQL> SELECT ENAME FROM EMP WHERE SAL/12>=250 AND (COMM=0 OR COMM IS NULL);

ENAME
----------
SCOTT
FORD
KING
```

19. Select the job title and name of the employees whose name has an A in the second character or another A in the third character or an S in the fifth character and whose name has six letters. They also have to fulfil that their boss has a code higher than 7600 or that their monthly salary is less than 200 and that their department is 20 or 30 or that their employee number is higher than 7700. Sort them by their manager's code. Take the salary in the table as annual.

```
SELECT ENAME, JOB FROM EMP
WHERE (ENAME LIKE '_A%____' OR ENAME LIKE '__A%___' OR ENAME
LIKE '____S%_')
AND (MGR>7600 OR SAL/12<200)
AND (DEPTNO IN (20,30) OR EMPNO>7700) ORDER BY MGR;
```

```
SQL> SELECT ENAME, JOB FROM EMP
  2  WHERE (ENAME LIKE '_A%____' OR ENAME LIKE '__A%___' OR ENAME LIKE '____S%_')
  3  AND (MGR>7600 OR SAL/12<200)
  4  AND (DEPTNO IN (20,30) OR EMPNO>7700) ORDER BY MGR;

ENAME      JOB
---------- ---------
MARTIN     SALESMAN
```

20. Select the employees with their department who have in their name an A in the third character and in the job an A in its second character and, in any case, their job is 7 characters long. At the same time tell me only those who do not belong to departments 10 and 20.

```
SELECT ENAME, DEPTNO FROM EMP
WHERE (ENAME LIKE '_A%' OR ENAME LIKE '__A%'
OR ENAME LIKE '____S%') AND ENAME LIKE '______'
AND (MGR>7600 OR SAL/12<200) AND (DEPTNO IN (20,30) OR
EMPNO>7700)
ORDER BY MGR;
```

```
SQL> SELECT ENAME, DEPTNO FROM EMP
  2  WHERE JOB LIKE '_A%' AND ENAME LIKE '__A%' AND JOB LIKE '_______'
  3  AND DEPTNO NOT IN(10,20);

ENAME          DEPTNO
---------- ----------
BLAKE              30
```

21. Select the name and entry date of the employees who belong to department 10, 20 or 30 and their name does not exceed 4 characters. Sort by the longest entry date.

```
SELECT ENAME NOMBRE, HIREDATE "FECHA INGRESO" FROM EMP
WHERE DEPTNO IN (10,20,30) AND LENGTH(ENAME)<5
ORDER BY HIREDATE DESC;
```

```
SQL> SELECT ENAME NOMBRE, HIREDATE "FECHA INGRESO" FROM EMP WHERE DEPTNO IN (10,20,30) AND LENGTH(ENAME)<5
  2  ORDER BY HIREDATE DESC;

NOMBRE     FECHA IN
---------- --------
FORD       03/12/81
KING       17/11/81
WARD       22/02/81
```

22. Select the monthly salary with the alias 'Monthly Salary' of the employees who have a Chief or their code is not less than 7900 and who do not belong to department 10 or 40.

```
SELECT SAL "SALARIO MENSUAL" FROM EMP
WHERE MGR IS NOT NULL OR MGR>=7900 AND DEPTNO NOT IN
(10,40)
ORDER BY DEPTNO ASC;
```

```
SQL> SELECT SAL "SALARIO MENSUAL" FROM EMP
  2  WHERE MGR IS NOT NULL OR MGR>=7900 AND DEPTNO NOT IN (10,40)
  3  ORDER BY DEPTNO ASC;

SALARIO MENSUAL
---------------
           2450
           1300
           3000
           3000
           1100
            800
           2975
            950
           1500
           2850
           1250
           1600
           1250

13 rows selected.
```

23. Select the name, job and employee code of those whose salary is between 600 and 1500 and not between 1000 and 3000. Sort by ascending employee code.

```
SELECT ENAME, JOB, EMPNO FROM EMP WHERE SAL BETWEEN 600
AND 1500 AND SAL NOT BETWEEN 1000 AND 3000 ORDER BY
DEPTNO ASC;
```

```
SQL> SELECT ENAME, JOB, EMPNO FROM EMP WHERE SAL BETWEEN 600 AND 1500 AND
  2  SAL NOT BETWEEN 1000 AND 3000 ORDER BY DEPTNO ASC;

ENAME      JOB              EMPNO
---------- --------- ----------
SMITH      CLERK             7369
JAMES      CLERK             7900
```

24. Select the monthly salary, calling it so, and the job of those employees who are SALESMAN and have a different salary between them. Order from highest salary to lowest.

```
SELECT DISTINCT JOB,SAL "SALARIO MENSUAL"
FROM EMP WHERE JOB='SALESMAN' ORDER BY 'SALARIO MENSUAL'
DESC;
```

```
SQL> SELECT DISTINCT JOB,SAL "SALARIO MENSUAL"
  2  FROM EMP WHERE JOB='SALESMAN' ORDER BY 'SALARIO MENSUAL' DESC;

JOB       SALARIO MENSUAL
--------- ---------------
SALESMAN             1600
SALESMAN             1250
SALESMAN             1500
```

25. Select the monthly salary plus monthly commission, converting the nulls to zero, with an alias, plus the name, but we want to increase the monthly salary with a CPI of 5% for all but CLERKs and employees who joined after 1/1/85.

```
SELECT ROUND(SAL+NVL(COMM,0),0)*1.05 "SALARIO CON
COMISION",ENAME FROM EMP
WHERE  JOB<> 'CLERK' AND HIREDATE<='01/01/85'
ORDER BY "SALARIO CON COMISION" DESC;
```

```
SQL> SELECT ROUND(SAL*1.05+NVL(COMM,0),0) "SALARIO CON COMISION",ENAME FROM EMP
  2  WHERE  JOB<> 'CLERK' AND HIREDATE<='01/01/85' ORDER BY "SALARIO CON COMISION" DESC;

SALARIO CON COMISION ENAME
-------------------- ----------
                5250 KING
                3150 FORD
                3124 JONES
                2993 BLAKE
                2713 MARTIN
                2573 CLARK
                1980 ALLEN
                1813 WARD
                1575 TURNER

9 rows selected.
```

26. Select the monthly salary plus monthly commission by converting the nulls, but we want to raise the salary only for SALESMAN or those with real commission, by 30%, leaving the CPI the same at 5%, and call the column "New Salary".

```
SELECT ROUND(SAL*1.3*1.05+NVL(COMM,0),0) "NUEVO
SUELDO",ENAME FROM EMP
WHERE JOB='SALESMAN' OR COMM IS NOT NULL AND COMM<>0;
```

```
SQL> SELECT ROUND(SAL*1.3*1.05+NVL(COMM,0),0) "NUEVO SUELDO",ENAME FROM EMP
  2  WHERE JOB='SALESMAN' OR COMM IS NOT NULL AND COMM<>0;

NUEVO SUELDO ENAME
------------ ----------
        2484 ALLEN
        2206 WARD
        3106 MARTIN
        2048 TURNER
```

27. All the same as in the previous Select but now we want to remove the increase to lower the salary by 10% and changing the CPI to 3%. The Where will now change and will affect those who are not MANAGER or PRESIDENT or do not have a real commission.

```
SELECT ROUND(SAL*1.03*0.90+NVL(COMM,0),0) "NUEVO
SUELDO",ENAME FROM EMP
WHERE JOB NOT IN ('MANAGER','PRESIDENT') OR COMM IS NULL OR
COMM=0;
```

```
SQL> SELECT ROUND(SAL*1.03*0.90+NVL(COMM,0),0) "NUEVO SUELDO",ENAME FROM EMP
  2  WHERE JOB NOT IN ('MANAGER','PRESIDENT') OR COMM IS NULL OR COMM=0;

NUEVO SUELDO ENAME
------------ ----------
         742 SMITH
        1783 ALLEN
        1659 WARD
        2758 JONES
        2559 MARTIN
        2642 BLAKE
        2271 CLARK
        2781 SCOTT
        1020 ADAMS
         881 JAMES
        2781 FORD
        1205 MILLER
        4635 KING
        1391 TURNER

14 rows selected.
```

28. Show the alias name of employees who meet either, earn more than 1400 or have actual commission, or be from department 10 and 20, or have joined before January '87 or their grade is between 2 and 4.
In any case your manager must have a code higher than 7800. Sort by the highest category.

```
SELECT ENAME
FROM EMP NATURAL JOIN DEPT NATURAL JOIN OFICIO
WHERE (COMM IS NOT NULL OR SAL>1400 OR COMM<>0 OR DEPTNO
IN (10,20) OR HIREDATE<'01/01/87' OR CATEGORIA IN (2,3,4))
AND MGR>7800 ORDER BY CATEGORIA DESC;
```

```
SQL> SELECT ENAME FROM EMP NATURAL JOIN DEPT NATURAL JOIN OFICIO
  2  WHERE (COMM IS NOT NULL OR SAL>1400 OR COMM<>0 OR DEPTNO IN (10,20)
  3  OR HIREDATE<'01/01/87' OR CATEGORIA IN (2,3,4)) AND MGR>7800
  4  ORDER BY CATEGORIA DESC;

ENAME
----------
CLARK
JONES
BLAKE
SMITH
```

29. Show for each job category, the number of employees together with their average salary with alias all of them, asking for locality and category, whenever this is greater than 1.
asking for the locality and the category, as long as this is greater than 1.

```
SELECT CATEGORIA,COUNT(EMPLEADOS),AVG(SAL) FROM EMP
NATURAL JOIN DEPT NATURAL JOIN OFICIO
WHERE LOC='&LOCALIDAD' GROUP BY CATEGORIA HAVING
CATEGORIA=&CATEGORIA AND CATEGORIA>1;
```

```
SQL> SELECT CATEGORIA,COUNT(EMPLEADOS),AVG(SAL) FROM EMP NATURAL JOIN DEPT NATURAL JOIN OFICIO
  2  WHERE LOC='&LOCALIDAD' GROUP BY CATEGORIA HAVING CATEGORIA=&CATEGORIA AND CATEGORIA>1;
Enter value for localidad: NEW YORK
Enter value for categoria: 4
old   2: WHERE LOC='&LOCALIDAD' GROUP BY CATEGORIA HAVING CATEGORIA=&CATEGORIA AND CATEGORIA>1
new   2: WHERE LOC='NEW YORK' GROUP BY CATEGORIA HAVING CATEGORIA=4 AND CATEGORIA>1

 CATEGORIA COUNT(EMPLEADOS)   AVG(SAL)
---------- --------------- ----------
         4               1       2450
```

30. Show the name of the department together with its number of employees, average, maximum, minimum and total salaries with the aliases of choice of those who either have an A or an E in their locality,
minimum and total with aliases to choose from for those who either have in their locality an A or an E anywhere, or have employees with category not anywhere, or have employees with category not between 1 and 3 and zero commission. We only wish to apply it to the Sales and Research departments and show first the department with the highest spend on salaries first. Present the numerical values rounded to two decimal places.

```
SELECT
DNAME PUESTO,
COUNT(EMPLEADOS)"Nº DE EMPLEADOS",
ROUND(AVG(SAL),0) "SALARIO MEDIO",
ROUND(MAX(SAL),0) "SALARIO MAXIMO",
ROUND(MIN(SAL),0) "SALARIO MINIMO",
ROUND(SUM(SAL),0) "SALARIO TOTAL"
FROM OFICIO NATURAL JOIN DEPT NATURAL JOIN EMP
```

```
GROUP BY DNAME HAVING DNAME IN ('SALES','RESEARCH')
ORDER BY "SALARIO TOTAL" DESC;
```

```
SQL> SELECT DNAME PUESTO,COUNT(EMPLEADOS)"Nº DE EMPLEADOS",ROUND(AVG(SAL),0) "SALARIO MEDIO",
  2  ROUND(MAX(SAL),0) "SALARIO MAXIMO",ROUND(MIN(SAL),0) "SALARIO MINIMO",ROUND(SUM(SAL),0) "SALARIO TOTAL"
  3  FROM OFICIO NATURAL JOIN DEPT NATURAL JOIN EMP
  4  GROUP BY DNAME HAVING DNAME IN ('SALES','RESEARCH') ORDER BY "SALARIO TOTAL" DESC;

PUESTO         Nº DE EMPLEADOS SALARIO MEDIO SALARIO MAXIMO SALARIO MINIMO SALARIO TOTAL
-------------- --------------- ------------- -------------- -------------- -------------
RESEARCH                     5          2175           3000            800         10875
SALES                        6          1567           2850            950          9400
```

31. Based on the previous select make the following changes,
- Now we also want to show the job title of the employees in each department.
- Sort by department name from Z to A.
- Create cross tables with Cube and Rollup indistinctly (check that they act in this case in the same way).
- If any field has null values, change them to a text of your choice.

```
SELECT DNAME DEPARTAMENTO,
NVL(JOB,'SUMATORIO') PUESTO,
ROUND(COUNT(EMPLEADOS),0) "Nº DE EMPLEADOS",
ROUND(AVG(SAL),0) "SALARIO MEDIO",
ROUND(MAX(SAL),0) "SALARIO MAXIMO",
ROUND(MIN(SAL),0) "SALARIO MINIMO",
ROUND(SUM(SAL),0) "SALARIO TOTAL"
FROM OFICIO NATURAL JOIN DEPT NATURAL JOIN EMP
GROUP BY ROLLUP(DNAME,JOB)
HAVING DNAME IN ('SALES','RESEARCH')
ORDER BY DEPARTAMENTO DESC.
```

```
SQL> SELECT DNAME DEPARTAMENTO,NVL(JOB,'SUMATORIO') PUESTO, ROUND(COUNT(EMPLEADOS),0) "Nº DE EMPLEADOS",ROUND(AVG(SAL),0) "SALARIO MEDIO",
  2  ROUND(MAX(SAL),0) "SALARIO MAXIMO",ROUND(MIN(SAL),0) "SALARIO MINIMO",ROUND(SUM(SAL),0) "SALARIO TOTAL"
  3  FROM OFICIO NATURAL JOIN DEPT NATURAL JOIN EMP
  4  GROUP BY CUBE(DNAME,JOB) HAVING DNAME IN ('SALES','RESEARCH') ORDER BY DEPARTAMENTO DESC;

DEPARTAMENTO   PUESTO    Nº DE EMPLEADOS SALARIO MEDIO SALARIO MAXIMO SALARIO MINIMO SALARIO TOTAL
-------------- --------- --------------- ------------- -------------- -------------- -------------
SALES          CLERK                   1           950            950            950           950
SALES          MANAGER                 1          2850           2850           2850          2850
SALES          SALESMAN                4          1400           1600           1250          5600
SALES          SUMATORIO               6          1567           2850            950          9400
RESEARCH       CLERK                   2           950           1100            800          1900
RESEARCH       ANALYST                 2          3000           3000           3000          6000
RESEARCH       MANAGER                 1          2975           2975           2975          2975
RESEARCH       SUMATORIO               5          2175           3000            800         10875

8 rows selected.

SQL> SELECT DNAME DEPARTAMENTO,NVL(JOB,'SUMATORIO') PUESTO, ROUND(COUNT(EMPLEADOS),0) "Nº DE EMPLEADOS",ROUND(AVG(SAL),0) "SALARIO MEDIO",
  2  ROUND(MAX(SAL),0) "SALARIO MAXIMO",ROUND(MIN(SAL),0) "SALARIO MINIMO",ROUND(SUM(SAL),0) "SALARIO TOTAL"
  3  FROM OFICIO NATURAL JOIN DEPT NATURAL JOIN EMP
  4  GROUP BY ROLLUP(DNAME,JOB) HAVING DNAME IN ('SALES','RESEARCH') ORDER BY DEPARTAMENTO DESC;

DEPARTAMENTO   PUESTO    Nº DE EMPLEADOS SALARIO MEDIO SALARIO MAXIMO SALARIO MINIMO SALARIO TOTAL
-------------- --------- --------------- ------------- -------------- -------------- -------------
SALES          CLERK                   1           950            950            950           950
SALES          MANAGER                 1          2850           2850           2850          2850
SALES          SALESMAN                4          1400           1600           1250          5600
SALES          SUMATORIO               6          1567           2850            950          9400
RESEARCH       CLERK                   2           950           1100            800          1900
RESEARCH       ANALYST                 2          3000           3000           3000          6000
RESEARCH       MANAGER                 1          2975           2975           2975          2975
RESEARCH       SUMATORIO               5          2175           3000            800         10875

8 rows selected.
```

32. We wish to show, for each job, which will be ordered by keyboard, and for each department number, the average salary with two decimal places, its job and its department number, sorted by major department.

```
SELECT ROUND(AVG(SAL),2) MEDIA_SALARIAL, JOB PUESTO, DEPTNO
DEPARTAMENTO
FROM EMP
GROUP BY DEPTNO,JOB HAVING JOB='&PUESTO'
ORDER BY DEPARTAMENTO DESC;
```

```
SQL> SELECT ROUND(AVG(SAL),2) MEDIA_SALARIAL, JOB PUESTO, DEPTNO DEPARTAMENTO
  2  FROM EMP
  3  GROUP BY DEPTNO,JOB HAVING DEPTNO='&DEPARTAMENTO'
  4  ORDER BY DEPARTAMENTO DESC;
Enter value for departamento: 20
old   3: GROUP BY DEPTNO,JOB HAVING DEPTNO='&DEPARTAMENTO'
new   3: GROUP BY DEPTNO,JOB HAVING DEPTNO='20'

MEDIA_SALARIAL PUESTO    DEPARTAMENTO
-------------- --------- ------------
          3000 ANALYST              20
           950 CLERK                20
          2975 MANAGER              20
```

Note to avoid mistakes - Del departamento 20 solo hay uno de cada uno****

```
SELECT ROUND(AVG(SAL),2) MEDIA_SALARIAL, JOB PUESTO, DEPTNO
DEPARTAMENTO
FROM EMP
GROUP BY DEPTNO HAVING DEPTNO='&DEPARTAMENTO'AND
JOB='&PUESTO'
ORDER BY DEPARTAMENTO DESC;
```

```
SQL> SELECT ROUND(AVG(SAL),2) MEDIA_SALARIAL, JOB PUESTO, DEPTNO DEPARTAMENTO
  2  FROM EMP
  3  GROUP BY DEPTNO HAVING DEPTNO='&DEPARTAMENTO'AND JOB='&PUESTO'
  4  ORDER BY DEPARTAMENTO DESC;
Enter value for departamento: 20
Enter value for puesto: CLERK
old   3: GROUP BY DEPTNO HAVING DEPTNO='&DEPARTAMENTO'AND JOB='&PUESTO'
new   3: GROUP BY DEPTNO HAVING DEPTNO='20'AND JOB='CLERK'
GROUP BY DEPTNO HAVING DEPTNO='20'AND JOB='CLERK'
                                  *
ERROR at line 3:
ORA-00979: not a GROUP BY expression
```

```
SELECT ROUND(AVG(SAL),2) MEDIA_SALARIAL, JOB PUESTO, DEPTNO
DEPARTAMENTO
FROM EMP
WHERE JOB='&JOB' GROUP BY DEPTNO HAVING
DEPTNO='&DEPARTAMENTO'
```

ORDER BY DEPARTAMENTO DESC;

```
SQL> SELECT ROUND(AVG(SAL),2) MEDIA_SALARIAL, JOB PUESTO, DEPTNO DEPARTAMENTO
  2  FROM EMP
  3  WHERE JOB='&JOB' GROUP BY DEPTNO HAVING DEPTNO='&DEPARTAMENTO'
  4  ORDER BY DEPARTAMENTO DESC;
Enter value for job: CLERK
Enter value for departamento: 20
old   3: WHERE JOB='&JOB' GROUP BY DEPTNO HAVING DEPTNO='&DEPARTAMENTO'
new   3: WHERE JOB='CLERK' GROUP BY DEPTNO HAVING DEPTNO='20'
SELECT ROUND(AVG(SAL),2) MEDIA_SALARIAL, JOB PUESTO, DEPTNO DEPARTAMENTO
                                         *
ERROR at line 1:
ORA-00979: not a GROUP BY expression
```

33. Display the following screen output: No aliases for constants.
The total number of actual commissioned employees is X, their average salary is Y, their department name is Z and their job category is W.

```
SELECT ' El total de empleados con comisión real es de  ' " ",
COUNT(EMPLEADOS) "TOTAL EMPLEADOS",
'su media salarial es de ' " ", ROUND(AVG(SAL),2) "MEDIA SALARIAL ",'
su nombre de departamento es el " ",
DNAME DEPARTAMENTO, 'categoria ' " ", CATEGORIA
FROM EMP NATURAL JOIN OFICIO NATURAL JOIN DEPT
WHERE COMM<>0 AND COMM IS NOT NULL
GROUP BY DNAME,CATEGORIA;
```

```
  1  SELECT 'El  ' " ", count(EMPLEADOS) "TOTAL EMPLEADOS",
  2  'su medi ' " ", ROUND(AVG(SAL),2) "MEDIA SALARIAL ",'su depart' " ",
  3  DNAME DEPARTAMENTO, 'categoria ' " ", CATEGORIA
  4  FROM EMP NATURAL JOIN OFICIO NATURAL JOIN DEPT
  5* WHERE COMM<>0 AND COMM IS NOT NULL GROUP BY DNAME,CATEGORIA
  6  ;

    TOTAL EMPLEADOS          MEDIA SALARIAL            DEPARTAMENTO              CATEGORIA
---- --------------- -------- --------------- --------- -------------- ---------- ----------
El                3 su medi          1366,67 su depart SALES          categoria           2
```

34. Show the unit of thousands of the salary and commission of the employees in the following format, for example 1451 should be shown as 1MIL, and the units of hundreds in the same way, for example 800 will be shown as 8CEN, also for those who receive more than 999 salary. In the case of no commission, we will set the value to zero.

```
SELECT SUBSTR(SAL,-4,1)||'mil' MIL_SAL,
SUBSTR(SAL,-3,1)||'cen' CEN_SAL,
DECODE(COMM,NULL,0,SUBSTR(COMM,-4,1))||'mil' MIL_COM,
DECODE(COMM,NULL,0,SUBSTR(COMM,-3,1))||'cen' CEN_COM,
SAL, COMM
FROM EMP WHERE SAL>999;
```

```
SQL> SELECT SUBSTR(SAL,-4,1)||'mil' MIL_SAL, SUBSTR(SAL,-3,1)||'cen' CEN_SAL,
  2  DECODE(COMM,NULL,0,SUBSTR(COMM,-4,1))||'mil' MIL_COM ,DECODE(COMM,NULL,0,SUBSTR(COMM,-3,1))||'cen' CEN_COM,
  3  SAL, COMM FROM EMP WHERE SAL>999;

MIL_SAL CEN_SAL MIL_COM                                  CEN_COM                                       SAL
------- ------- ---------------------------------------- ---------------------------------------- ---------- ----
1mil    6cen    mil                                      3cen                                          1600
1mil    2cen    mil                                      5cen                                          1250
2mil    9cen    0mil                                     0cen                                          2975
1mil    2cen    1mil                                     4cen                                          1250
2mil    8cen    0mil                                     0cen                                          2850
2mil    4cen    0mil                                     0cen                                          2450
3mil    0cen    0mil                                     0cen                                          3000
1mil    1cen    0mil                                     0cen                                          1100
3mil    0cen    0mil                                     0cen                                          3000
1mil    3cen    0mil                                     0cen                                          1300
5mil    0cen    0mil                                     0cen                                          5000
1mil    5cen    mil                                      cen                                           1500

12 rows selected.
```

35. Count the number of employees in each department by asking the user for the first number of the department.

```
SELECT COUNT(EMPLEADOS)
FROM DEPT NATURAL JOIN OFICIO  NATURAL JOIN EMP
WHERE DEPTNO LIKE '&VALOR%';
```

```
 SELECT COUNT(EMPLEADOS) FROM DEPT NATURAL JOIN OFICIO  NATURAL JOIN EMP
 WHERE DEPTNO LIKE '&VALOR%';
r value for valor: 1
  2: WHERE DEPTNO LIKE '&VALOR%'
  2: WHERE DEPTNO LIKE '1%'

T(EMPLEADOS)
------------
           3
```

36. Count the number of employees in each department by asking the user for the first number of the department.

```
SELECT COUNT(ENAME) FROM EMP WHERE ENAME LIKE 'M%';
```

```
SQL> SELECT COUNT(ENAME) FROM EMP WHERE ENAME LIKE 'M%';

COUNT(ENAME)
------------
           2
```

37. Count the different jobs against the total number of jobs.

```
SELECT COUNT(DISTINCT(JOB)) TOTAL_TRABAJOS FROM EMP;
```

```
SQL> SELECT COUNT(DISTINCT(JOB)) TOTAL_TRABAJOS FROM EMP;

TOTAL_TRABAJOS
--------------
             5
```

38. Select the names of employees who have been with the company for more than 35 years.

```
SELECT ENAME FROM EMP WHERE
MONTHS_BETWEEN(SYSDATE,HIREDATE)/12>35 ORDER BY ENAME;
```

```
SQL> SELECT ENAME FROM EMP WHERE MONTHS_BETWEEN(SYSDATE,HIREDATE)/12>35 ORDER BY ENAME;

ENAME
----------
ADAMS
ALLEN
BLAKE
CLARK
FORD
JAMES
JONES
KING
MARTIN
MILLER
SCOTT
SMITH
TURNER
WARD

14 rows selected.
```

39. Obtain by comparison the highest value between the sum of all salaries and the sum of all commissions of Sales' employees with real commission. Add aliases to the three columns.

```
SELECT SUM(SAL) SALARIO, SUM(COMM) COMISION,
GREATEST(SUM(SAL),SUM(COMM)) MAYOR
FROM EMP NATURAL JOIN DEPT
WHERE DNAME='SALES' AND COMM IS NOT NULL AND COMM<>0;
```

```
SQL> SELECT SUM(SAL) SALARIO, SUM(COMM) COMISION, GREATEST(SUM(SAL),SUM(COMM)) MAYOR
  2  FROM EMP NATURAL JOIN DEPT WHERE DNAME='SALES' AND COMM IS NOT NULL AND COMM<>0;

   SALARIO   COMISION      MAYOR
---------- ---------- ----------
      4100       2200       4100
```

40. Obtain the number of employees and the job position by grouping by job and changing the English positions by the Spanish values, taking where there are more than two employees with the same position. Use only the emp table.

```
SELECT COUNT(*),DECODE(JOB,'CLERK','DEPENDIENTE','SALESMAN',
```

```
'VENDEDOR','ANALYST','ANALISTA'
,'MANAGER','RESPONSABLE','PRESIDENT','PRESIDENT',JOB)
FROM EMP
GROUP BY JOB HAVING COUNT(*)>2;
```

```
SQL> SELECT COUNT(*),DECODE(JOB,'CLERK','DEPENDIENTE','SALESMAN','VENDEDOR','ANALYST','ANALISTA'
  2  ,'MANAGER','RESPONSABLE','PRESIDENT','PRESIDENT',JOB) FROM EMP
  3  GROUP BY JOB HAVING COUNT(*)>2;

  COUNT(*) DECODE(JOB,
---------- ----------
         4 DEPENDIENTE
         4 VENDEDOR
         3 RESPONSABLE
```

41. Show departments with more than two employees as long as they do not have category 4 employees.

```
SELECT DEPTNO,COUNT(DEPTNO) FROM EMP NATURAL JOIN OFICIO
WHERE CATEGORIA <>4 GROUP BY DEPTNO HAVING
COUNT(DEPTNO)>2;
```

```
SQL> SELECT DEPTNO,COUNT(DEPTNO) FROM EMP NATURAL JOIN OFICIO
  2  WHERE CATEGORIA <>4 GROUP BY DEPTNO HAVING COUNT(DEPTNO)>2;

    DEPTNO COUNT(DEPTNO)
---------- -------------
        30             5
        20             4
```

42. Show the name of the department that has employees with a salary of more than 500.

```
SELECT DISTINCT DNAME FROM EMP NATURAL JOIN DEPT WHERE
SAL>500;
```

```
SQL> SELECT DISTINCT DNAME FROM EMP NATURAL JOIN DEPT WHERE SAL>500;

DNAME
--------------
ACCOUNTING
RESEARCH
SALES
```

43. Select the department number and the name of the employee whose manager is an employee of department 10 or 20.

```
SELECT DEPTNO,ENAME FROM EMP WHERE MGR IN (SELECT EMPNO
FROM EMP WHERE DEPTNO IN (10,20));
```

```
SQL> SELECT DEPTNO,ENAME FROM EMP WHERE MGR IN (SELECT EMPNO FROM EMP WHERE DEPTNO IN (10,20));

    DEPTNO ENAME
---------- ----------
        20 SMITH
        20 JONES
        30 BLAKE
        10 CLARK
        20 SCOTT
        20 ADAMS
        20 FORD
        10 MILLER

8 rows selected.
```

44. Select the name, salary and commission of non-commissioned employees whose manager is an employee from the New York locality.

SELECT ENAME, SAL, COMM,MGR FROM EMP WHERE COMM IS NULL AND MGR IN (SELECT EMPNO FROM EMP WHERE DEPTNO=(SELECT DEPTNO FROM DEPT WHERE LOC='NEW YORK'));

```
SQL> SELECT ENAME, SAL, COMM,MGR FROM EMP WHERE COMM IS NULL
  2  AND MGR IN (SELECT EMPNO FROM EMP WHERE DEPTNO=(SELECT DEPTNO FROM DEPT WHERE LOC='NEW YORK'));

ENAME            SAL       COMM        MGR
---------- ---------- ---------- ----------
JONES            2975                  7839
BLAKE            2850                  7839
CLARK            2450                  7839
MILLER           1300                  7782
```

45. Select the name, code and date of entry of employees whose manager is an employee who does not have a manager.

SELECT ENAME, EMPNO, HIREDATE FROM EMP WHERE MGR IN (SELECT EMPNO FROM EMP WHERE MGR IS NULL);

```
SQL> SELECT ENAME, EMPNO, HIREDATE FROM EMP WHERE
  2  MGR IN (SELECT EMPNO FROM EMP WHERE MGR IS NULL);

ENAME           EMPNO HIREDATE
---------- ---------- --------
JONES            7566 02/04/81
BLAKE            7698 01/05/81
CLARK            7782 09/06/81
```

46. Select the monthly salary with an alias and the name of the employees whose boss is an employee with code less than 7900 and greater than 7800.

SELECT ROUND(SAL,0) "SALARIO MENSUAL", ENAME FROM EMP WHERE MGR IN(SELECT EMPNO FROM EMP WHERE EMPNO>7800 AND EMPNO<7900);

```
SQL> SELECT ROUND(SAL,0) "SALARIO MENSUAL", ENAME FROM EMP
  2  WHERE MGR IN( SELECT EMPNO FROM EMP WHERE EMPNO>7800 AND EMPNO<7900);

SALARIO MENSUAL ENAME
--------------- ----------
           2975 JONES
           2850 BLAKE
           2450 CLARK
```

47. Select the names of Sales employees whose boss is a non-Boston employee.

```
SELECT ENAME FROM EMP
WHERE DEPTNO IN
(SELECT DEPTNO FROM DEPT WHERE DNAME='SALES')
AND MGR IN (SELECT EMPNO FROM EMP WHERE DEPTNO
IN (SELECT DEPTNO FROM DEPT WHERE LOC<>'BOSTON'));
```

```
SQL> SELECT ENAME FROM EMP
  2  WHERE DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE DNAME='SALES') AND
  3  MGR IN (SELECT EMPNO FROM EMP WHERE DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE LOC<>'BOSTON'));

ENAME
----------
ALLEN
WARD
MARTIN
BLAKE
JAMES
TURNER

6 rows selected.
```

48. Select the name of the department that has employees with salaries higher than any employee in the New York locality. Always use a minimised select or virtual view instead of tables in all from clauses.

```
SELECT DNAME
FROM (SELECT DNAME, DEPTNO FROM DEPT) --VV DEPT
WHERE DEPTNO IN (SELECT DEPTNO FROM
(SELECT DEPTNO,SAL FROM EMP ) --VV EMP
WHERE SAL >SOME (SELECT SAL FROM
(SELECT SAL, DEPTNO FROM EMP) --VV EMP
WHERE DEPTNO IN (SELECT DEPTNO FROM
(SELECT DEPTNO,LOC FROM DEPT) --VV DEPT
WHERE LOC ='NEW YORK')));
```

```
DNAME
--------------
ACCOUNTING
RESEARCH
SALES
```

49. Select the code and name of the employee who is a Manager, not a Sales employee, and whose job is one of the existing jobs in the Chicago location.

```
SELECT EMPNO, ENAME FROM EMP  WHERE JOB='MANAGER'
AND DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE DNAME NOT
IN('SALES')
AND JOB IN (SELECT JOB FROM DEPT WHERE LOC='CHICAGO'));
```

```
SQL> SELECT EMPNO, ENAME FROM EMP WHERE JOB='MANAGER' AND
  2  DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE DNAME NOT IN('SALES')
  3  AND JOB IN (SELECT JOB FROM DEPT WHERE LOC='CHICAGO'));

     EMPNO ENAME
---------- ----------
      7782 CLARK
      7566 JONES
```

50. Select the name of the department whose department numeric value (1st nesting) is higher than the one held by the employee with empno 7698 and whose department number (2nd nesting) is one of the ones held by employees with Category lower than 2.

```
SELECT DNAME FROM DEPT WHERE DEPTNO>(SELECT DEPTNO
FROM EMP WHERE EMPNO=7698 AND
DEPTNO IN (SELECT DEPTNO FROM EMP NATURAL JOIN OFICIO
WHERE CATEGORIA<2));
```

```
SQL> SELECT DNAME FROM DEPT WHERE DEPTNO>(SELECT DEPTNO FROM EMP WHERE EMPNO=7698 AND
  2  DEPTNO IN (SELECT DEPTNO FROM EMP NATURAL JOIN OFICIO WHERE CATEGORIA<2));

DNAME
--------------
OPERATIONS
```

51. Select the name of the department whose numerical value (1st nesting) is greater than that held by employees with a salary between 4000 and 5000. Another condition to fulfil is that the numerical value of the department (2nd nesting) is greater than or equal to any of those held by commissioned employees. Sort by ascending department name.

```
SELECT DNAME FROM DEPT
WHERE DEPTNO > (SELECT DEPTNO FROM EMP WHERE SAL
BETWEEN 4000 AND 5000)
AND DEPTNO >= ANY (SELECT DEPTNO FROM EMP WHERE COMM IS
NOT NULL)
ORDER BY DNAME;
```

```
SQL> SELECT DNAME FROM DEPT
  2  WHERE DEPTNO > (SELECT DEPTNO FROM EMP WHERE SAL BETWEEN 4000 AND 5000)
  3  AND DEPTNO >= ANY (SELECT DEPTNO FROM EMP WHERE COMM IS NOT NULL)
  4  ORDER BY DNAME;

DNAME
--------------
OPERATIONS
SALES
```

52. Select the name, code, their functions and the locality of work of the employees whose salary (nested) is greater than or equal to any commission multiplied by 10. Use a view as a table containing all the fields of the tables we need.

```
SELECT ENAME, EMPNO, FUNCIONES, LOC
FROM
(SELECT ENAME, EMPNO, FUNCIONES,LOC,SAL FROM EMP, DEPT,
OFICIO
WHERE EMP.DEPTNO=DEPT.DEPTNO AND EMP.JOB=OFICIO.JOB)
WHERE SAL IN
(SELECT COMM*10 FROM
(SELECT COMM FROM EMP WHERE COMM IS NOT NULL))
ORDER BY EMPNO DESC;
```

```
SQL> SELECT ENAME, EMPNO, FUNCIONES, LOC
  2   FROM
  3   (SELECT ENAME, EMPNO, FUNCIONES,LOC,SAL FROM EMP, DEPT, OFICIO
  4   WHERE EMP.DEPTNO=DEPT.DEPTNO AND EMP.JOB=OFICIO.JOB)
  5   WHERE SAL IN
  6   (SELECT COMM*10 FROM (SELECT COMM FROM EMP WHERE COMM IS NOT NULL))
  7   ORDER BY EMPNO DESC;

ENAME           EMPNO FUNCIONES           LOC
---------- ---------- -------------------- ------------
FORD             7902 ECONOMISTA           DALLAS
KING             7839 FUNDADOR             NEW YORK
SCOTT            7788 ECONOMISTA           DALLAS
```

53. For employees whose category is not the one in the employee column of the trade table with a range not between 1 and 3, and work not corresponding to the New York locality, another condition is that the department name can have an A or an E anywhere; show their name and category. Remove the combination of tables where possible.

```
SELECT ENAME, CATEGORIA
FROM EMP, OFICIO
WHERE EMP.JOB=OFICIO.JOB AND CATEGORIA NOT IN ( SELECT
CATEGORIA FROM OFICIO WHERE EMPLEADOS NOT BETWEEN 1
AND 3 AND JOB IN ( SELECT JOB
FROM EMP WHERE DEPTNO IN ( SELECT DEPTNO FROM DEPT
WHERE LOC <> 'NEW YORK')))
```

```
AND DEPTNO IN ( SELECT DEPTNO FROM DEPT WHERE DNAME LIKE
'%A%' OR DNAME LIKE '%E%');
```

```
SQL> SELECT ENAME, CATEGORIA
  2  FROM EMP, OFICIO
  3  WHERE EMP.JOB=OFICIO.JOB AND CATEGORIA NOT IN ( SELECT CATEGORIA
  4  FROM OFICIO WHERE EMPLEADOS NOT BETWEEN 1 AND 3 AND JOB IN ( SELECT JOB
  5  FROM EMP WHERE DEPTNO IN ( SELECT DEPTNO FROM DEPT WHERE LOC <> 'NEW YORK')))
  6  AND DEPTNO IN ( SELECT DEPTNO FROM DEPT WHERE DNAME LIKE '%A%' OR
  7  DNAME LIKE '%E%');

ENAME       CATEGORIA
---------- ----------
JONES               4
BLAKE               4
CLARK               4
SCOTT               3
FORD                3
KING                5

6 rows selected.
```

54. Displays the name of employees whose salary is not greater than or equal to any of Dallas but is greater than any of the average salaries by department no. taking the average greater than or equal to 1000 and the number of craft employees is greater than 2.

```
SELECT ENAME FROM EMP
WHERE SAL < ANY ( SELECT SAL FROM EMP WHERE DEPTNO IN (
SELECT DEPTNO FROM DEPT WHERE LOC ='DALLAS')) AND SAL > ANY
( SELECT AVG(SAL) FROM EMP
WHERE JOB IN ( SELECT JOB FROM OFICIO WHERE EMPLEADOS >2)
GROUP BY DEPTNO HAVING AVG(SAL) >= 1000)
ORDER BY SAL DESC;
```

```
SQL> SELECT ENAME
  2  FROM EMP
  3  WHERE SAL < ANY ( SELECT SAL FROM EMP WHERE DEPTNO IN ( SELECT DEPTNO
  4  FROM DEPT WHERE LOC ='DALLAS')) AND SAL > ANY ( SELECT AVG(SAL)
  5  FROM EMP
  6  WHERE JOB IN ( SELECT JOB FROM OFICIO WHERE EMPLEADOS >2)
  7  GROUP BY DEPTNO
  8  HAVING AVG(SAL) >= 1000)
  9  ORDER BY SAL DESC;

ENAME
----------
JONES
BLAKE
CLARK
ALLEN
```

55. Select the name of the department whose department no. is greater than one that has employees with seniority prior to 1983 and whose work is not that of category 1 employees as long as the category is equal to half of the department no. Sort by department name from A to Z. Use a view in the tables as necessary to minimise the use of columns.

```
SELECT DNAME
FROM (SELECT DNAME, DEPTNO FROM DEPT)
WHERE DEPTNO > SOME ( SELECT DEPTNO
FROM (SELECT DEPTNO, HIREDATE, JOB FROM EMP)
WHERE HIREDATE < '01-01-1983' AND JOB NOT IN ( SELECT JOB
FROM (SELECT JOB, CATEGORIA FROM OFICIO)
WHERE CATEGORIA =1 AND CATEGORIA IN ( SELECT DEPTNO/2
FROM (SELECT DEPTNO FROM EMP))))
ORDER BY DNAME;
```

```
SQL> SELECT DNAME
  2  FROM (SELECT DNAME, DEPTNO FROM DEPT)
  3  WHERE DEPTNO > SOME ( SELECT DEPTNO
  4  FROM (SELECT DEPTNO, HIREDATE, JOB FROM EMP)
  5  WHERE HIREDATE < '01-01-1983' AND JOB NOT IN ( SELECT JOB
  6  FROM (SELECT JOB, CATEGORIA FROM OFICIO)
  7  WHERE CATEGORIA =1 AND CATEGORIA IN ( SELECT DEPTNO/2
  8  FROM (SELECT DEPTNO FROM EMP))))
  9  ORDER BY DNAME;

DNAME
--------------
OPERATIONS
RESEARCH
SALES
```

56. Show the name, salary, department name, functions and no. of employees (trade) of employees who, using a view as a table including the minimum possible fields, do not have a manager whose commission is higher than any of the monthly salaries of employees with jobs that include any of the following functions: clerk and salesperson. No combination of tables can be used except in the view.

```
SELECT ENAME, SAL, DNAME, FUNCIONES, EMPLEADOS
FROM (SELECT ENAME, SAL , DNAME, FUNCIONES, EMPLEADOS,
MGR, COMM, EMP.JOB,
OFICIO.JOB FROM EMP, DEPT, OFICIO
WHERE EMP.DEPTNO=DEPT.DEPTNO AND EMP.JOB=OFICIO.JOB)
WHERE MGR NOT IN
( SELECT EMPNO FROM EMP WHERE COMM > SOME (SELECT SAL/12
FROM EMP WHERE JOB IN ( SELECT JOB FROM OFICIO WHERE
FUNCIONES IN('DEPENDIENTE','VENDEDOR'))));
```

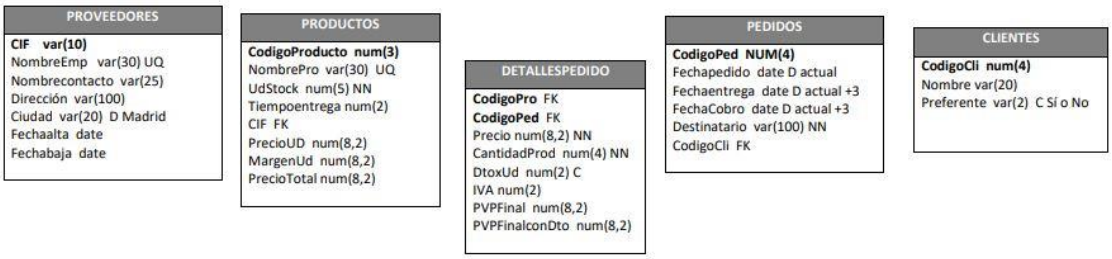
```
SQL> SELECT ENAME, SAL, DNAME, FUNCIONES, EMPLEADOS
  2  FROM (SELECT ENAME, SAL , DNAME, FUNCIONES, EMPLEADOS, MGR, COMM, EMP.JOB,
  3  OFICIO.JOB FROM EMP, DEPT, OFICIO
  4  WHERE EMP.DEPTNO=DEPT.DEPTNO AND EMP.JOB=OFICIO.JOB)
  5  WHERE MGR NOT IN (
  6  SELECT EMPNO FROM EMP WHERE COMM > SOME (SELECT SAL/12
  7  FROM EMP WHERE JOB IN (
  8  SELECT JOB FROM OFICIO WHERE FUNCIONES IN('DEPENDIENTE','VENDEDOR'))))
  9  ;

ENAME             SAL DNAME          FUNCIONES             EMPLEADOS
---------- ---------- -------------- -------------------- ----------
CLARK            2450 ACCOUNTING     RESPONSABLE                   3
BLAKE            2850 SALES          RESPONSABLE                   3
JONES            2975 RESEARCH       RESPONSABLE                   3
MILLER           1300 ACCOUNTING     DEPENDIENTE                   4
ADAMS            1100 RESEARCH       DEPENDIENTE                   4
TURNER           1500 SALES          VENDEDOR                      4
JAMES             950 SALES          DEPENDIENTE                   4
MARTIN           1250 SALES          VENDEDOR                      4
WARD             1250 SALES          VENDEDOR                      4
ALLEN            1600 SALES          VENDEDOR                      4
FORD             3000 RESEARCH       ECONOMISTA                    2
SCOTT            3000 RESEARCH       ECONOMISTA                    2
SMITH             800 RESEARCH       DEPENDIENTE                   4

13 rows selected.
```

DDL AND DML EXERCICES

57. Create the following table structure with the primary keys indicated in bold, foreign keys with FK, unique keys with UQ, non-null keys with NN, check keys with C and the default value with D,



In addition, the following restrictions must be met, all of them with restriction names,

- SUPPLIERS TABLE: The entry date must be equal to or prior to the withdrawal date.
- PRODUCTS TABLE: The TotalPrice will be the sum of the PUD Price plus the PUD Margin.
- TABLE ORDER DETAILS:
  - The VAT may contain the values of 4,10 and 21.
  - The PVPFinal will be the Price plus VAT always with two decimals. DtoxUd will only allow the values Null, 2, 5, 8 and 10.
  - The PVPFinalwithDto will be the Price plus VAT minus DtoxUd, always with two decimals.Se debe cumplir bien que para entre 0

y 50 el DtoxUd sea del 2, entre 51 y 100 del 5, entre 101 y 200 del 8, y mayor de 200 del 10.

Make the following insertions,

- One supplier with values of your choice.
- Two products with values of your choice except for PriceUd (5.25 and 4.25) and MarginUd (0.5 and 0.75), respectively.
- One customer with values of your choice.
- An order with values of your choice.
- Two partial records in OrderDetails, one for the Price of 5.75 of the table Products with 25 units and 21% VAT, and another for the Price of 5 of the table Products with 300 units, a DtoxUd of 10% and 21% VAT.

The result will be correct if we are presented with the following OrderDetails data,

CODIGOPRO	CODIGOPED	PRECIO	CANTIDADPROD	DTOXUD	IVA	PVPFINAL	PVPFINALCONDTO
1	1000	5,75	25		21	6,96	
2	1000	5	300	10	21	6,05	5,55

```
CREATE TABLE PROVEEDORES
(CIF VARCHAR2(10) CONSTRAINT PK_CIF_PRO PRIMARY KEY,
NOMBREEMP VARCHAR2(30) CONSTRAINT UQ_NE_PRO UNIQUE,
NOMBRECONTACTO VARCHAR2(25),
DIRECCION VARCHAR2(100),
CIUDAD VARCHAR2(20) DEFAULT 'MADRID',
FECHAALTA DATE,
FECHABAJA DATE,
CONSTRAINT F_PROV CHECK (FECHAALTA<=FECHABAJA));
```

```
SQL> DESC PROVEEDORES;
 Name                                                                    Null?    Type
 ----------------------------------------------------------------------- -------- ----------------
 CIF                                                                     NOT NULL VARCHAR2(10)
 NOMBREEMP                                                                        VARCHAR2(30)
 NOMBRECONTACTO                                                                   VARCHAR2(25)
 CIUDAD                                                                           VARCHAR2(20)
 FECHAALTA                                                                        DATE
 FECHABAJA                                                                        DATE
 DIRECCION                                                                        VARCHAR2(100)
```

```
CREATE TABLE PRODUCTOS(
CODIGOPRODUCTO NUMBER(3) CONSTRAINT
PK_CP_PROD PRIMARY KEY,
NOMBREPRO VARCHAR2(30) CONSTRAINT UQ_NP_PROD UNIQUE,
UDSTOCK NUMBER(5) CONSTRAINT NN_US_PROD NOT NULL,
TIEMPOENTREGA NUMBER(2),
CIF VARCHAR2(10) CONSTRAINT FK_CIF_PROD REFERENCES
PROVEEDORES ON DELETE CASCADE,
```

PRECIOUD NUMBER(8,2),
MARGENUD NUMBER(8,2),
PRECIOTOTAL NUMBER(8,2));

```
SQL> DESC PRODUCTOS;
 Name                                                                    Null?    Type
 ----------------------------------------------------------------------- -------- ----------------
 CODIGOPRODUCTO                                                          NOT NULL NUMBER(3)
 NOMBREPRO                                                                        VARCHAR2(30)
 UDSTOCK                                                                 NOT NULL NUMBER(5)
 TIEMPOENTREGA                                                                    NUMBER(2)
 CIF                                                                              VARCHAR2(10)
 PRECIOUD                                                                         NUMBER(8,2)
 MARGENUD                                                                         NUMBER(8,2)
 PRECIOTOTAL                                                                      NUMBER(8,2)
```

CREATE TABLE CLIENTES(
CODIGOCLI NUMBER(4) CONSTRAINT PK_CC_CLI PRIMARY KEY,
NOMBRE VARCHAR2(20),
PREFERENTE VARCHAR2(2) CONSTRAINT C_PREF_CLI CHECK (PREFERENTE IN ('SI','NO')));

```
SQL> DESC CLIENTES;
 Name                                                                    Null?    Type
 ----------------------------------------------------------------------- -------- ----------------
 CODIGOCLI                                                               NOT NULL NUMBER(4)
 NOMBRE                                                                           VARCHAR2(20)
 PREFERENTE                                                                       VARCHAR2(2)
```

CREATE TABLE PEDIDOS(
CODIGOPED NUMBER(4) CONSTRAINT PK_CODPED_PED PRIMARY KEY,
FECHAPEDIDO DATE DEFAULT SYSDATE,
FECHAENTREGA DATE DEFAULT SYSDATE+3,
FECHACOBRO DATE DEFAULT SYSDATE+3,
DESTINATARIO VARCHAR2(100) NOT NULL,
CODIGOCLI NUMBER(4) CONSTRAINT FK_CC_CLI REFERENCES CLIENTES ON DELETE CASCADE);

```
SQL> DESC PEDIDOS;
 Name                                                                    Null?    Type
 ----------------------------------------------------------------------- -------- ----------------
 CODIGOPED                                                               NOT NULL NUMBER(4)
 FECHAPEDIDO                                                                      DATE
 FECHAENTREGA                                                                     DATE
 FECHACOBRO                                                                       DATE
 DESTINATARIO                                                            NOT NULL VARCHAR2(100)
 CODIGOCLI                                                                        NUMBER(4)
```

CREATE TABLE DETALLESPEDIDO(
CODIGOPRO NUMBER(3) CONSTRAINT FK_CODPRO_DETPED REFERENCES PRODUCTOS ON DELETE CASCADE,
CODIGOPED NUMBER(4) CONSTRAINT FK_CODPED_DETPED REFERENCES PEDIDOS ON DELETE CASCADE,
PRECIO NUMBER(8,2) CONSTRAINT NN_PR_DETPED NOT NULL,

```
CANTIDADPROD NUMBER(4) CONSTRAINT NN_CANTPROD_DETPED
NOT NULL,
DTOXUD NUMBER(2) CHECK (DTOXUD IN (NULL,0,2,5,8,10)),
IVA NUMBER(2) CHECK( IVA IN (4,10,21)),
PVPFINAL NUMBER(8,2) AS
(ROUND(PRECIO*((PRECIO/PRECIO)+(IVA/100)),2)),
PVPFINALCONDTO NUMBER(8,2) AS
(ROUND((PRECIO*((PRECIO/PRECIO)+(IVA/100))-
(PRECIO*(DTOXUD/100)),2)),
CONSTRAINT PK_CODPROCODPED_DETPED PRIMARY KEY
(CODIGOPRO,CODIGOPED),
CONSTRAINT C_DP CHECK ((CANTIDADPROD BETWEEN 0 AND 50
AND DTOXUD=2) OR
(CANTIDADPROD BETWEEN 51 AND 100 AND DTOXUD=5) OR
(CANTIDADPROD BETWEEN 101 AND 200 AND DTOXUD=8) OR
(CANTIDADPROD>200 AND DTOXUD=10)));
```

```
SQL> DESC DETALLESPEDIDO;
 Name                                                                    Null?    Type
 ----------------------------------------------------------------------- -------- ------------
 CODIGOPRO                                                               NOT NULL NUMBER(3)
 CODIGOPED                                                               NOT NULL NUMBER(4)
 PRECIO                                                                  NOT NULL NUMBER(8,2)
 CANTIDADPROD                                                            NOT NULL NUMBER(4)
 DTOXUD                                                                           NUMBER(2)
 IVA                                                                              NUMBER(2)
 PVPFINAL                                                                         NUMBER(8,2)
 PVPFINALCONDTO                                                                   NUMBER(8,2)
```

```
INSERT INTO PROVEEDORES VALUES
('G87567823','MUEBLES BOOM','PEPE','CALLE RUEDA, 23',
'23-5-15',NULL,'VALENCIA');
```

```
CIF       NOMBREEMP                   NOMBRECONTACTO           CIUDAD              FECHAALT FECHABAJ
--------- --------------------------- ------------------------ ------------------- -------- --------
DIRECCION
-----------------------------------------------------------------------------------------------------
G87567823 MUEBLES BOOM                PEPE                     CALLE RUEDA, 23     23/05/15
VALENCIA
```

```
INSERT INTO PRODUCTOS  VALUES (
1,'TUERCAS',100,2,'G87567823',5.25,0.5);

INSERT INTO PRODUCTOS
VALUES (2,'ARANDELAS',100,2,'G87567823',4.25,0.75);
```

```
SQL> SELECT * FROM PRODUCTOS;

CODIGOPRODUCTO NOMBREPRO                     UDSTOCK TIEMPOENTREGA CIF         PRECIOUD   MARGENUD PRECIOTOTAL
-------------- ----------------------------- ------- ------------ ---------- ---------- ---------- -----------
             1 TUERCAS                           100            2 G87567823       5,25         ,5        5,75
             2 ARANDELAS                         100            2 G87567823       4,25        ,75           5
```

- We will add the new DATE column which will be DATE and unique to the PRODUCTS table.

```
ALTER TABLE PRODUCTOS ADD (FECHAALTA DATE CONSTRAINT UQ_FA_PROD UNIQUE);
```

```
SQL> DESC PRODUCTOS;
 Name                                                      Null?    Type
 ----------------------------------------------------------------- -------- --------------------
 CODIGOPRODUCTO                                            NOT NULL NUMBER(3)
 NOMBREPRO                                                          VARCHAR2(30)
 UDSTOCK                                                   NOT NULL NUMBER(5)
 TIEMPOENTREGA                                                      NUMBER(2)
 CIF                                                                VARCHAR2(10)
 PRECIOUD                                                           NUMBER(8,2)
 MARGENUD                                                           NUMBER(8,2)
 PRECIOTOTAL                                                        NUMBER(8,2)
 FECHAALTA                                                          DATE
```

- We will modify the NAME column of the CUSTOMERS table to increase the dimension to 50.

```
ALTER TABLE CLIENTES MODIFY (NOMBRE VARCHAR(50));
```

```
SQL> DESC CLIENTES;
 Name                                                      Null?    Type
 ----------------------------------------------------------------- -------- ------------------
 CODIGOCLI                                                 NOT NULL NUMBER(4)
 NOMBRE                                                             VARCHAR2(50)
 PREFERENTE                                                         VARCHAR2(2)
```

- We will change the name of the column TARGET to CUSTOMER of the table ORDERS.

```
ALTER TABLE PEDIDOS RENAME COLUMN DESTINATARIO TO CLIENTE;
```

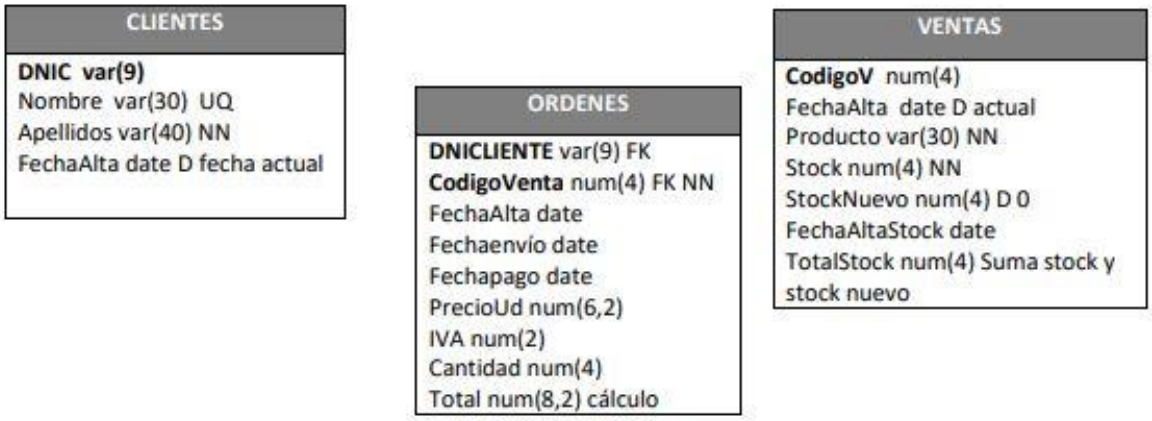
```
SQL> DESC PEDIDOS;
 Name                                                      Null?    Type
 ----------------------------------------------------------------- -------- ----------------
 CODIGOPED                                                 NOT NULL NUMBER(4)
 FECHAPEDIDO                                                        DATE
 FECHAENTREGA                                                       DATE
 FECHACOBRO                                                         DATE
 CLIENTE                                                   NOT NULL VARCHAR2(100)
 CODIGOCLI                                                          NUMBER(4)
```

- We will add the default constraint value of 0 in the DTOXUD column of the table DETAILSOrder.

```
ALTER TABLE DETALLESPEDIDO MODIFY (DTOXUD NUMBER(2) DEFAULT (0));
```

```
SQL> DESC DETALLESPEDIDO;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ------------
 CODIGOPRO                                             NOT NULL NUMBER(3)
 CODIGOPED                                             NOT NULL NUMBER(4)
 PRECIO                                                NOT NULL NUMBER(8,2)
 CANTIDADPROD                                          NOT NULL NUMBER(4)
 DTOXUD                                                         NUMBER(2)
 IVA                                                            NUMBER(2)
 PVPFINAL                                                       NUMBER(8,2)
 PVPFINALCONDTO                                                 NUMBER(8,2)
```

58. Create the following table structure by setting the primary keys in bold, foreign keys with FK, unique keys with UQ, non-null keys with NN, check keys with C and the default value with D,



In addition, the following constraints, all with constraint names, must be met,
ORDER TABLE:

- The Entry Date must be prior to or equal to the Send Date.
- The Ship Date must be prior to or equal to the Payment Date.
- The Total will be the calculation of the PriceUDP plus VAT for the quantity of products purchased.
- Perform the following insertions,
    - One customer and one sale with values of your choice.
    - One order with values to choose from except for PriceUDP (10), VAT (10) and Quantity (50).

```
CREATE TABLE CLIENTES ( DNIC VARCHAR2(9) CONSTRAINT
PK_DNIC_CLI PRIMARY KEY,
NOMBRE VARCHAR2(30) CONSTRAINT UQ_N_CLI UNIQUE,
APELLIDOS VARCHAR2(40) NOT NULL,
FECHAALTA DATE DEFAULT SYSDATE);
```

```
SQL> DESC CLIENTES;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ----------------
 DNIC                                                  NOT NULL VARCHAR2(9)
 NOMBRE                                                         VARCHAR2(30)
 APELLIDOS                                             NOT NULL VARCHAR2(40)
 FECHAALTA                                                      DATE
```

```
CREATE TABLE VENTAS (
CODIGOV NUMBER(4) CONSTRAINT PK_CODV_V PRIMARY KEY,
FECHAALTA DATE DEFAULT SYSDATE,
PRODUCTO VARCHAR2(30) NOT NULL,
STOCK NUMBER(4) NOT NULL,
STOCKNUEVO NUMBER(4) DEFAULT 0,
FECHAALTASTOCK DATE,
TOTALSTOCK NUMBER(4) AS (STOCK+STOCKNUEVO));
```

```
SQL> DESC VENTAS;
 Name                                                      Null?    Type
 --------------------------------------------------------- -------- ------------
 CODIGOV                                                   NOT NULL NUMBER(4)
 FECHAALTA                                                          DATE
 PRODUCTO                                                  NOT NULL VARCHAR2(30)
 STOCK                                                     NOT NULL NUMBER(4)
 STOCKNUEVO                                                         NUMBER(4)
 FECHAALTASTOCK                                                     DATE
 TOTALSTOCK                                                         NUMBER(4)
```

```
CREATE TABLE ORDENES (
DNICLIENTE VARCHAR2(9) CONSTRAINT FK_DC_ORD REFERENCES
CLIENTES ON DELETE CASCADE,
CODIGOVENTA NUMBER(4) NOT NULL CONSTRAINT FK_CV_ORD
REFERENCES VENTAS ON DELETE CASCADE,
FECHAALTA DATE,
FECHAENVIO DATE,
FECHAPAGO DATE,
PRECIOUD NUMBER(6,2),
IVA NUMBER(2),
CANTIDAD NUMBER(4),
TOTAL NUMBER(8,2) AS (PRECIOUD*(1+(IVA/100))*CANTIDAD),
CONSTRAINT PK_DNICCV_ORD PRIMARY KEY
(DNICLIENTE,CODIGOVENTA),
CONSTRAINT C_FAFE_ORD CHECK (FECHAALTA<=FECHAENVIO),
CONSTRAINT C_FEFP_ORD CHECK (FECHAENVIO<=FECHAPAGO));
```

```
SQL> DESC ORDENES;
 Name                                                      Null?    Type
 --------------------------------------------------------- -------- ------------
 DNICLIENTE                                                NOT NULL VARCHAR2(9)
 CODIGOVENTA                                               NOT NULL NUMBER(4)
 FECHAALTA                                                          DATE
 FECHAENVIO                                                         DATE
 FECHAPAGO                                                          DATE
 PRECIOUD                                                           NUMBER(6,2)
 IVA                                                                NUMBER(2)
 CANTIDAD                                                           NUMBER(4)
 TOTAL                                                              NUMBER(8,2)
```

Make the following insertions,

- One customer and one sale with values of your choice.

```
INSERT INTO CLIENTES VALUES
(05311954L,'EDUARDO','MATOS','16/09/96');
```

```
SQL> SELECT * FROM CLIENTES;

DNIC      NOMBRE                         APELLIDOS                                FECHAALT
--------- ------------------------------ ---------------------------------------- --------
53119549  EDUARDO                       MATOS                                    16/09/96
```

- One order with values to choose from except for PriceUd (10), VAT (10) and Quantity (50)

```
INSERT INTO VENTAS (CODIGOV, FECHAALTA, PRODUCTO, STOCK,
STOCKNUEVO, FECHAALTASTOCK) VALUES
(1234,SYSDATE,'CALEFACTOR',50,100,SYSDATE);
```

```
  CODIGOV FECHAALT PRODUCTO                       STOCK STOCKNUEVO FECHAALT TOTALSTOCK
--------- -------- ------------------------------ ----- ---------- -------- ----------
     1234 14/10/22 CALEFACTOR                        50        100 14/10/22        150
```

- Create the following tables according to the expressed restrictions:

FABRICANTES

COD_FABRICANTE	NUMBER(3)
NOMBRE	VARCHAR2(15)
PAIS	VARCHAR2(15)

Restrictions on the manufacturers table:

- The primary key is MANUFACTURER_CODE which is also NOT NULL.
- The default values for NAME and COUNTRY shall be UNdefined and Spain.

```
CREATE TABLE FABRICANTES(
COD_FABRICANTE NUMBER(3) CONSTRAINT NN_CF_FAB NOT NULL
CONSTRAINT PK_CF_FAB PRIMARY KEY,
NOMBRE VARCHAR2(15) DEFAULT 'NO DEFINIDO',
PAIS VARCHAR2(15) DEFAULT 'ESPANA'));
```

```
SQL> DESC FABRICANTES;
 Name                                                      Null?    Type
 --------------------------------------------------------- -------- ------------
 COD_FABRICANTE                                            NOT NULL NUMBER(3)
 NOMBRE                                                             VARCHAR2(15)
 PAIS                                                               VARCHAR2(15)
```

ARTICULOS

ARTICULO	VARCHAR2(20)
COD_FABRICANTE	NUMBER(3)
PESO	NUMBER(3)
CATEGORIA	VARCHAR2(10)
PRECIO_VENTA	NUMBER(4)
PRECIO_COSTO	NUMBER(4)
EXISTENCIAS	NUMBER(5)

Restrictions to the items table:

- The primary key will be called PK and is formed by ARTICLE,MANUFACTURER_CODE, WEIGHT and CATEGORY that have to be NOT NULL at the same time.
- MANUFACTURER_CODE is a foreign key that refers to the MANUFACTURERS table and we want it to be deleted in cascade.
- SALE_PRICE, COST_PRICE and WEIGHT must be greater than zero.
- CATEGORY must be First, Second or Third and the constraint will be called CATEG.

```
CREATE TABLE ARTICULOS(
ARTICULO VARCHAR2(20) NOT NULL,
COD_FABRICANTE NUMBER(3) NOT NULL CONSTRAINT FK_CODF_ART
REFERENCES FABRICANTES ON DELETE CASCADE,
PESO NUMBER(3) CONSTRAINT NN_P_ART NOT NULL CONSTRAINT
C_P_ART CHECK (PESO>0),
CATEGORIA VARCHAR2(10) CONSTRAINT NN_C_ART NOT NULL,
PRECIO_VENTA NUMBER(4) CHECK (PRECIO_VENTA>0),
PRECIO_COSTO NUMBER(4) CHECK (PRECIO_COSTO>0),
EXISTENCIAS NUMBER(5),
CONSTRAINT PK_ARTCODPCAT PRIMARY KEY
(ARTICULO,COD_FABRICANTE,PESO,CATEGORIA))
```

```
SQL> DESC ARTICULOS;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------
 ARTICULO                                  NOT NULL VARCHAR2(20)
 COD_FABRICANTE                            NOT NULL NUMBER(3)
 PESO                                      NOT NULL NUMBER(3)
 CATEGORIA                                 NOT NULL VARCHAR2(10)
 PRECIO_VENTA                                       NUMBER(4)
 PRECIO_COSTO                                       NUMBER(4)
 EXISTENCIAS                                        NUMBER(5)
```

Then insert two records in each of them, one complete and one partial record of your choice.

```
INSERT INTO FABRICANTES  (COD_FABRICANTE) VALUES (123);
```

```
SQL> SELECT * FROM FABRICANTES;

COD_FABRICANTE NOMBRE          PAIS
-------------- --------------- ---------------
           123 NO DEFINIDO     ESPANA
```

INSERT INTO ARTICULOS VALUES ('ZAPATILLAS',123,4,'NIKE',300,200,12345);

```
SQL> SELECT * FROM ARTICULOS;

ARTICULO           COD_FABRICANTE       PESO CATEGORIA  PRECIO_VENTA PRECIO_COSTO EXISTENCIAS
------------------ -------------- ---------- ---------- ------------ ------------ -----------
ZAPATILLAS                    123          4 NIKE                300          200       12345
```

Modify two fields of any record of your choice.

UPDATE ARTICULOS SET PRECIO_VENTA=800 WHERE ARTICULO='ZAPATILLAS';

```
SQL> SELECT * FROM ARTICULOS;

ARTICULO   COD_FABRICANTE       PESO CATEGORIA  PRECIO_VENTA PRECIO_COSTO EXISTENCIAS
---------- -------------- ---------- ---------- ------------ ------------ -----------
ZAPATILLAS            123          4 NIKE                800          200       12345
```

Delete an entire record.

DELETE FROM ARTICULOS WHERE ARTICULO='ZAPATILLAS';

```
SQL> SELECT * FROM ARTICULOS;

no rows selected
```

59. Create the following tables according to the expressed restrictions and then delete them:

TIENDAS

NIF	VARCHAR2(10)
NOMBRE	VARCHAR2(15)
DIRECCIÓN	VARCHAR2(20)
POBLACIÓN	VARCHAR2(10)
PROVINCIA	VARCHAR2(10)
CODPOSTAL	NUMBER(5)
FEC_APERTURA	DATE

Restrictions on the shops table:

- The primary key is the VAT number.
- The default values for NAME AND ADDRESS will be Undefined and Unknown Street.

- PROVINCE is NOT NULL and will be a foreign key referencing the SALES table.
- CODPOSTAL is a UNIQUE column and the OPEN_DATE will default to the system date of the day.

```
CREATE TABLE TIENDAS(
NIF VARCHAR2(10) CONSTRAINT PK_TIENDAS PRIMARY KEY,
NOMBRE VARCHAR2(15) DEFAULT 'NO DEFINIDO',
DIRECCION VARCHAR2(20) DEFAULT 'CALLE DESCONOCIDA',
POBLACION VARCHAR2(20),
PROVINCIA VARCHAR2(10) CONSTRAINT NN_T NOT NULL,
CODPOSTAL NUMBER(5) CONSTRAINT U1 UNIQUE,
FEC_APERTURA DATE DEFAULT SYSDATE);
```

```
SQL> desc TIENDAS;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ------------
 NIF                                                   NOT NULL VARCHAR2(10)
 NOMBRE                                                         VARCHAR2(15)
 DIRECCION                                                      VARCHAR2(20)
 POBLACION                                                      VARCHAR2(20)
 PROVINCIA                                             NOT NULL VARCHAR2(10)
 CODPOSTAL                                                      NUMBER(5)
 FEC_APERTURA                                                   DATE
```

VENTAS

NIF	VARCHAR2(10)
PROVINCIA	VARCHAR2(10)
UD_VENTAS	NUMBER(4)
ARTICULO	VARCHAR2(10)
FEC_VENTA	DATE

Restrictions on the sales table:

- The primary key is PROVINCE
- The NIF is a foreign key referring to the table TIENDAS and NOT NULL.
- UD_SALES has a minimum value of 100 units.

```
CREATE TABLE VENTAS(
NIF VARCHAR2(10) CONSTRAINT NN_V NOT NULL,
PROVINCIA VARCHAR2(10) CONSTRAINT PK_VENTAS PRIMARY KEY,
UD_VENTAS NUMBER(4) CONSTRAINT CV1 CHECK(UD_VENTAS
>=100),
ARTICULO VARCHAR2(10),
FEC_VENTA DATE);
```

```
SQL> desc VENTAS;
 Name                                                      Null?    Type
 --------------------------------------------------------- -------- ----------------
 NIF                                                       NOT NULL VARCHAR2(10)
 PROVINCIA                                                 NOT NULL VARCHAR2(10)
 UD_VENTAS                                                          NUMBER(4)
 ARTICULO                                                           VARCHAR2(10)
 FEC_VENTA                                                          DATE
```

Then insert two records in each of them, one complete and one partial record of your choice. Modify two fields of a record of your choice and delete a whole record.

```
ALTER TABLE TIENDAS
ADD CONSTRAINT FK_TIENDAS FOREIGN KEY(PROVINCIA)
REFERENCES VENTAS(PROVINCIA)
ON DELETE CASCADE;
```

```
SQL> ALTER TABLE TIENDAS
  2  ADD CONSTRAINT FK_TIENDAS FOREIGN KEY(PROVINCIA) REFERENCES VENTAS(PROVINCIA)
  3  ON DELETE CASCADE;

Table altered.
```

```
ALTER TABLE TIENDAS
DISABLE CONSTRAINT FK_TIENDAS;
```

```
SQL> ALTER TABLE TIENDAS
  2   DISABLE CONSTRAINT FK_TIENDAS;
```

```
INSERT INTO TIENDAS VALUES(
'123333-X','MUEBS','CALLE
MADRID','LEGANES','MADRID',28914,SYSDATE);
```

```
SQL> SELECT * FROM TIENDAS;

NIF        NOMBRE          DIRECCION           POBLACION           PROVINCIA   CODPOSTAL FEC_APER
---------- --------------- ------------------- ------------------- ---------- ---------- --------
123333-X   MUEBS           CALLE MADRID        LEGANES             MADRID          28914 19/10/22
```

```
INSERT INTO VENTAS
(NIF, PROVINCIA) VALUES('1212121-F','TOLEDO');
```

```
SQL> INSERT INTO VENTAS(NIF, PROVINCIA) VALUES('1212121-F','TOLEDO');

1 row created.
```

DCL EXERCISES

Default TABLESPACE creation

```
CREATE TABLESPACE CLASE
ONLINE
DATAFILE 'D:\CLASE.DBF'
 SIZE 300M REUSE
DEFAULT STORAGE
(INITIAL 10M
NEXT 10M
PCTINCREASE 10);
```

Temporary TABLESPACE creation

```
CREATE TEMPORARY TABLESPACE TEMPORAL
TEMPFILE 'D:\APP\TEMPORAL.DBF'
SIZE 100M AUTOEXTEND ON NEXT 10M MAXSIZE 1G;
```

60. Grant Scott the DBA role to be able to work without having to switch to the DBA. Create a role, called R_RRHH, that includes three users, called RRHH1/RRHH1, RRHH2/RRHH2, RRHH3/RRHH3, with quota of 10M and all of them with access to Select on Emp and Dept, to be able to connect and to be able to create synonyms. We want to store them in the tablespace RRHH, inside the file RRHH.ora, with a size of 100M, initial block of 10M, next block of 10M and increment of 10%, always online.
We will set its profile to the name P_RRHH, with a maximum connection time of 400 minutes, inactivity of 30 minutes, a single session, and a maximum of 2 blocks read per query. Delete the role, tablespace, profile and the three users. Ensure that Scott is removed from the DBA role.

Revoke dba from Scott;

```
CONN SYSTEM/*****
GRANT DBA TO SCOTT;
```

CONN SCOTT/****

Role creation

```
CREATE ROLE R_RRHH;
```

Tablespace creation

```
 CREATE TABLESPACE RRHH
 DATAFILE 'RRHH.ORA' SIZE 100M
 DEFAULT STORAGE(
 INITIAL 10M
```

```
 NEXT 10M
 PCTINCREASE 10);
```

Profile creation

```
CREATE PROFILE P_RRHH
 LIMIT
 CONNECT_TIME 400
 IDLE_TIME 30
 SESSIONS_PER_USER 1
 LOGICAL_READS_PER_CALL 2;
```

Users RRHH1,RRHH2 and RRHH3 creation

```
CREATE USER RRHH1
IDENTIFIED BY RRHH1
DEFAULT TABLESPACE RRHH
QUOTA 10M ON RRHH
PROFILE P_RRHH;
CREATE USER RRHH2
IDENTIFIED BY RRHH2
DEFAULT TABLESPACE RRHH
QUOTA 10M ON RRHH
PROFILE P_RRHH;
CREATE USER RRHH3
IDENTIFIED BY RRHH3
DEFAULT TABLESPACE RRHH
QUOTA 10M ON RRHH
PROFILE P_RRHH;
```

Assignament object' privileges

```
GRANT SELECT ON EMP TO R_RRHH;
GRANT SELECT ON DEPT TO R_RRHH;
```

Assignament sistem' privileges

```
GRANT CREATE SESSION, CREATE SYNONYM TO R_RRHH;
GRANT R_RRHH TO RRHH1,RRHH2,RRHH3;

DROP TABLESPACE RRHH INCLUDING CONTENTS AND DATAFILES
CASCADE CONSTRAINTS;
DROP PROFILE P_RRHH CASCADE;
DROP USER RRHH1 CASCADE;
DROP USER RRHH2 CASCADE;
```

```
DROP USER RRHH3 CASCADE;
DROP ROLE R_RRHH CASCADE;
CONN system/manager
REVOKE DBA FROM SCOTT;
```

61. Create an auto numeric field called Cycle that has a maximum and minimum limit of 1345 and 56, respectively. It starts at 1345, but decrements in threes and forms a repeating loop. Subsequently, create the table Numbers with the field Code number(4) and primary key, stored in Class. Insert in it the first five values given by the autonumber field created. Check the result with a Select.

```
CONN SYSTEM/*****
```

Sequence creation

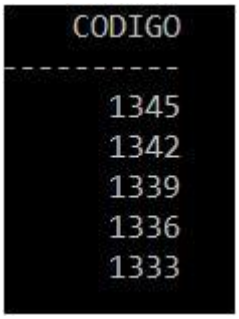
```
CREATE SEQUENCE CICLO
 MAXVALUE 1345
 MINVALUE 56
 START WITH 1345
 INCREMENT BY -3 CYCLE;
```

Table creation

```
CREATE TABLE numeros
(CODIGO NUMBER(4) CONSTRAINT PK_NUMEROS PRIMARY KEY)
TABLESPACE CLASE;
```

SECUENCE INCREMENTATION BY 3

```
INSERT INTO numeros VALUES (CICLO.NEXTVAL);
INSERT INTO numeros VALUES (CICLO.NEXTVAL);
INSERT INTO numeros VALUES (CICLO.NEXTVAL);
INSERT INTO numeros VALUES (CICLO.NEXTVAL);
INSERT INTO numeros VALUES (CICLO.NEXTVAL);
SELECT * FROM NUMEROS;
```



```
DROP TABLE NUMEROS;
DROP SEQUENCE CICLO;
```

62. Create a tablespace called Sales, containing the file Sales.ora with a size of 50M and set it Offline. Assign to it a user called Currito/Currito with 10M usage. Also assign the user Boss/Boss to the Sales tablespace with unlimited usage. Then activate the tablespace and grant the DBA role to the Boss and the CONNECT role to Currito. Also assign all object privileges to the Boss and only the Select to Currito on the EMP table. Delete BOSS, CURRITO and SALES.

```
CONN SYSTEM/*****
```

Tablespace creation

```
CREATE TABLESPACE VENTAS
OFFLINE DATAFILE 'VENTAS.ORA' SIZE 50M;
```

User CURRITO and JEFE creation

```
CREATE USER CURRITO
IDENTIFIED BY CURRITO
DEFAULT TABLESPACE VENTAS
QUOTA 10M ON VENTAS;

CREATE USER JEFE
IDENTIFIED BY JEFE
DEFAULT TABLESPACE CLASE
QUOTA UNLIMITED ON CLASE;
```

Bringing online VENTAS TABLESPACE to be available for Users

```
ALTER TABLESPACE VENTAS ONLINE;
```

Assignament sistem' privileges

```
GRANT DBA TO JEFE;
GRANT CONNECT TO CURRITO;
```

Assignament object' privileges

```
GRANT ALL ON SCOTT.EMP TO JEFE;
GRANT SELECT ON SCOTT.EMP TO CURRITO;

DROP TABLESPACE VENTAS INCLUDING CONTENTS AND DATAFILES;
```

```
DROP USER JEFE;
DROP USER CURRITO;
```

Create a user called JIMENO with password EL_CID, stored in the CLASS tablespace and with a quota of 4M. In turn, grant him the necessary privileges so that he can create another user called CHAMPION identified by EL, stored in CLASS and with a quota of 5M. In turn, grant him the necessary privileges to create another user named FERNANDO identified by REY, stored in CLASS and quota of 2M. Delete all the objects created.

```
CONN SYSTEM/*****
```

User' JIMENO creation

```
CREATE USER JIMENO
IDENTIFIED BY EL_CID
DEFAULT TABLESPACE CLASE
QUOTA 4M ON CLASE;
```

Assignament sistem' privileges

```
GRANT CREATE SESSION, CREATE USER TO JIMENO WITH ADMIN
OPTION;
```

Connecting as JIMENO

```
CONN JIMENO/***
CREATE USER CAMPEADOR
IDENTIFIED BY EL
DEFAULT TABLESPACE CLASE
QUOTA 5M ON CLASE;
```

Assignament sistem' privileges

```
GRANT CREATE SESSION, CREATE USER TO CAMPEADOR WITH
ADMIN OPTION;
```

Connecting as CAMPEADOR

```
CONN CAMPEADOR/***
```

User' Fernando creation

```
CREATE USER FERNANDO
```

```
IDENTIFIED BY REY
DEFAULT TABLESPACE CLASE
QUOTA 2M ON CLASE;

CONN system/manager
DROP USER FERNANDO;
DROP USER CAMPEADOR;
DROP USER JIMENO;
```

63. Create a Tablespace called COMPANY that is stored in the file property.ora, has a size of 100 Mb, its first extension is 10 Mb, the next one 20 Mb and no increment. Then create the user Director/Director, assign it to this tablespace by default with unlimited space. Assign to this user the role MANAGER that will be composed of the roles DBA and RESOURCE, and the profile OWNER with unlimited sessions, connection time, blocks, and all possible parameters unlimited.

We also want DIRECTOR/DIRECTOR to create three users called ONE/ONE, TWO/TWO and THREE/THREE with a quota of 1Mb each and the EMPLOYEE profile consisting of a single session. Assign to all three the WORK role consisting only of the privilege to create sessions. Then add the privilege of creating tables to ONE only.

ONE has to create a table called JOB with the columns CODE NUMBER and COMPANY VARCHAR(15). Insert in it two rows with the name of the companies ABENGOA and FYCSA and for the code I want to use an auto numeric field starting with the number 500, increasing by 500, with a maximum limit of 10,000 and being cyclic. Make a query that returns only the last value of the sequence used without putting WHERE.

The MANAGER has to ensure that employee TWO can make queries on the WORK table of employee ONE and that employee THREE can modify, insert and delete in the WORK table of employee ONE.

```
CONN system/manager
```

Tablespace creation

```
CREATE TABLESPACE EMPRESA
DATAFILE 'PROPIEDAD.ORA'
SIZE 100M
DEFAULT STORAGE
(INITIAL 10M
NEXT 20M
PCTINCREASE 0);
```

User DUENO and DIRECTOR creation

```
CREATE PROFILE DUENO LIMIT
SESSIONS_PER_USER UNLIMITED
CONNECT_TIME UNLIMITED
IDLE_TIME UNLIMITED
CPU_PER_CALL UNLIMITED
LOGICAL_READS_PER_CALL UNLIMITED
LOGICAL_READS_PER_SESSION UNLIMITED;

CREATE USER DIRECTOR
IDENTIFIED BY DIRECTOR
DEFAULT TABLESPACE EMPRESA
QUOTA UNLIMITED ON EMPRESA
PROFILE DUENO;
```

Role creation

```
CREATE ROLE GERENTE;
```

Assignament sistem' privileges

```
GRANT DBA, RESOURCE TO GERENTE;
GRANT GERENTE TO DIRECTOR;

CONN DIRECTOR/****

CREATE PROFILE EMPLEADO LIMIT
SESSIONS_PER_USER 1;
```

Role creation

```
CREATE ROLE TRABA;

GRANT CREATE SESSION TO TRABA;
CREATE USER UNO IDENTIFIED BY UNO
DEFAULT TABLESPACE EMPRESA
QUOTA 1M ON EMPRESA
PROFILE EMPLEADO;
```

User DOS and TRES creation

```
CREATE USER DOS IDENTIFIED BY DOS
DEFAULT TABLESPACE EMPRESA
```

```
QUOTA 1M ON EMPRESA
PROFILE EMPLEADO;

CREATE USER TRES IDENTIFIED BY TRES
DEFAULT TABLESPACE EMPRESA
QUOTA 1M ON EMPRESA
PROFILE EMPLEADO;
```

Assignament sistem' privileges

```
GRANT TRABA TO UNO,DOS,TRES;
GRANT CREATE TABLE TO UNO;
```

Table creation

```
CONN UNO/UNO
CREATE TABLE TRABAJO (CODIGO NUMBER(5), EMPRESA
VARCHAR2(15));
```

Sequence creation

```
CONN DIRECTOR/****

CREATE SEQUENCE SEC1
START WITH 500
INCREMENT BY 500
MAXVALUE 10000
CYCLE;
```

Assignament system' privileges

```
GRANT SELECT ANY SEQUENCE TO UNO;

CONN UNO/****

INSERT INTO TRABAJO
VALUES(DIRECTOR.SEC1.NEXTVAL,'ABENGOA');
INSERT INTO TRABAJO VALUES(DIRECTOR.SEC1.NEXTVAL,'FYCSA');
SELECT DIRECTOR.SEC1.CURRVAL FROM DUAL;
```

Assignament object' privileges

```
GRANT SELECT ON UNO.TRABAJO TO DOS;
GRANT UPDATE, INSERT, DELETE ON UNO.TRABAJO TO TRES;
```

```
CONN system/manager
DROP TABLESPACE EMPRESA INCLUDING CONTENTS AND DATAFILES;
DROP USER UNO CASCADE;
DROP USER DOS CASCADE;
DROP USER TRES CASCADE;
DROP SEQUENCE DIRECTOR.SEC1;
DROP USER DIRECTOR CASCADE;
DROP ROLE TRABA;
DROP PROFILE DUENO CASCADE;
DROP PROFILE EMPLEADO;
DROP ROLE GERENTE;
```