

EXÁMENES PRÁCTICOS BBDD (tipo 1)

1) Muestre el nombre del empleado, el número del empleado, el nombre y el número de su responsable (manager) para todos los empleados de la tabla *emp* que contengan una T en su nombre.

```
SELECT empleado.ename EMPLEADO,  
       empleado.empno NUM_EMP,  
       manager.ename MANAGER,  
       manager.empno NUM_MANAG
```

```
FROM emp empleado, emp manager
```

```
WHERE empleado.mgr = manager.empno
```

```
AND empleado.ename LIKE '%T%';
```

2) Mostrar el nombre, número de departamento y el salario de los empleados, los cuales su salario y comisión coinciden ambos con el salario y comisión de algún otro empleado que trabaja en un departamento localizado en New York.

(Mucho cuidado con los valores Nulos....usar NVL)

(Usamos una tabla auxiliar 'nuevayork' poniendo la subconsulta en el FROM!!!!)

```
SELECT empleado.ename NOMBRE,  
       empleado.deptno DEPARTAM,  
       NVL(empleado.sal,0) SALARIO
```

```
FROM emp empleado, (  
    SELECT NVL(e.sal,0) salario,  
           NVL(e.comm,0) comision,  
           empno numero  
    FROM emp e  
    WHERE e.deptno = (SELECT deptno  
                     FROM dept  
                     WHERE loc='NEW YORK')  
    ) nuevayork
```

```
WHERE NVL(empleado.sal,0) = nuevayork.salario
```

```
AND NVL(empleado.comm,0) = nuevayork.comision
```

```
AND empleado.empno <> nuevayork.numero;
```

3) Muestre el número, nombre, localización y número de empleados de todos los departamentos, incluso para aquellos departamentos que no tienen empleados.

```
SELECT deptno NUMERO,  
       dname NOMBRE,  
       loc LOCALIZACION,  
       tablanueva.empleados  
  
FROM dept,  
     (  
       SELECT d.deptno departamento,  
              COUNT(e.ename) empleados  
       FROM emp e,dept d  
       WHERE e.deptno(+) = d.deptno  
       GROUP BY d.deptno  
     ) tablanueva  
  
WHERE deptno = tablanueva.departamento;
```

4) Muestre el tipo de trabajo y el número de empleados que tiene cada tipo de trabajo, para aquellos empleados que trabajan en los departamentos RESEARCH y SALES.

```
SELECT  tablanueva.trabajo TIPO,  
        COUNT(tablanueva.trabajo) NUM_EMPLEADOS  
  
FROM  
     (  
       SELECT e.job trabajo  
       FROM emp e,dept d  
       WHERE e.deptno = d.deptno  
       AND d.dname IN ('RESEARCH','SALES')  
     ) tablanueva  
  
GROUP BY tablanueva.trabajo;
```

5) Cree una consulta que muestre el nombre del empleado, el salario, y la comisión, para cualquier empleado cuyo salario y comisión coincidan ambos con el salario y la comisión de cualquier empleado que trabaje en DALLAS. Excluya aquellos empleados que tengan un salario inferior a la media de salario de toda la empresa.

```
SELECT  e.ename NOMBRE,
        NVL(e.sal,0) SALARIO,
        NVL(e.comm,0) COMISION

FROM emp e,
        (
            SELECT  NVL(e.sal,0) salario,
                    NVL(e.comm,0) comision,
                    e.empno numero
            FROM emp e, dept d
            WHERE e.deptno = d.deptno
            AND d.loc = 'DALLAS'
        ) dallas,
        (
            SELECT AVG( NVL(sal,0) ) media
            FROM emp
        ) tablamedia

WHERE  NVL(e.sal,0)      =  dallas.salario

AND    NVL(e.comm,0)    =  dallas.comision

AND    e.empno          < >  dallas.numero

AND    NVL(e.sal,0)      > =  tablamedia.media;
```

6) Mostrar el número de departamento, nombre, la media del salario de todos los empleados que trabajan en ese departamento y el número de empleados que trabajan en él para el/los departamento/s con menor número de empleados.

Nota: No tiene sentido considerar a los departamentos que no tienen empleados!!!!

```
SELECT  d.deptno NUMERO_DEPT,  
        d.dname  NOMBRE_DEPT,  
        AVG(NVL(e.sal,0)) MEDIA_SAL,  
        COUNT(e.ename) EMPLEADOS
```

```
FROM emp e, dept d
```

```
WHERE e.deptno = d.deptno
```

```
GROUP BY d.deptno, d.dname
```

/ Ahora de todas las filas me quedo con la de menor n° de 'EMPLEADOS' */*

```
HAVING COUNT(e.ename) = (SELECT MIN( COUNT(e.ename) )  
                        FROM emp e  
                        GROUP BY e.deptno);
```

7) Muestre en columnas el número de personas que no son jefes en cada departamento. Ponga un alias a cada columna con el número de departamento.

DUDA: (EN ROJO, EN CADA DEPARTAMENTO = EN TODOS LOS DEPARTAMENTOS) ????

```
SELECT  COUNT( DECODE(deptno,10,1) ) DIEZ,  
        COUNT( DECODE(deptno,20,1) ) VEINTE,  
        COUNT( DECODE(deptno,30,1) ) TREINTA
```

```
FROM emp
```

```
WHERE mgr IS NOT NULL;
```

8) Crear una consulta que muestre, de manera tabular (matriz), para cada departamento el n° de empleados que entraron a trabajar en los años 80, 81, 82, 83; así como el **nivel (grado) salarial medio** del departamento. Del departamento se mostrará además el nombre y su localización entre paréntesis: NOMBRE (LOCALIZACIÓN).

```
SELECT d.dname || '(' || d.loc || ')'      "Departamento (Localización)",
       ROUND( AVG(s.grade) )              "Grado Salarial Medio",
       COUNT( DECODE( TO_CHAR(e.hiredate,'yy'),'80',1 ) ) "AÑO 80",
       COUNT( DECODE( TO_CHAR(e.hiredate,'yy'),'81',1 ) ) "AÑO 81",
       COUNT( DECODE( TO_CHAR(e.hiredate,'yy'),'82',1 ) ) "AÑO 82",
       COUNT( DECODE( TO_CHAR(e.hiredate,'yy'),'83',1 ) ) "AÑO 83"

FROM emp e, dept d, salgrade s

WHERE e.deptno = d.deptno

AND      e.sal BETWEEN s.losal AND s.hisal

GROUP BY d.dname,d.loc;
```

9) Obtener para cada empleado (**jefe**), el sueldo medio de sus subordinados.

```
SELECT jefe.empno EMPNO,
       jefe.ename NOMBRE,
       AVG( NVL(subordinado.sal,0) ) SUELDO_MED_SUBORD

FROM emp jefe, emp subordinado

WHERE jefe.empno = subordinado.mgr

GROUP BY jefe.empno, jefe.ename;
```

10) Mostrar los empleados que ganen más que la media salarial de su departamento.

```
SELECT      e.ename          NOMBRE,
            NVL(e.sal,0)      SALARIO,
            tablanueva.media  MEDIASAL,
            e.deptno         DEPARTAMENTO
FROM emp e, (
            SELECT deptno, AVG(NVL(sal,0)) media
            FROM emp
            GROUP BY deptno
            )tablanueva
WHERE e.deptno = tablanueva.deptno
AND      e.sal > tablanueva.media;
```

**NOTA IMPORTANTE: SIEMPRE QUE NOS PIDAN
FORMA TABULAR(MATRIZ), ES DECIR, MOSTRAR
EN COLUMNAS.....HABRÁ QUE USAR *DECODE*
!!!!!!!!!!!!!!**

EXÁMENES PRÁCTICOS BBDD (tipo 2)

1) Crear una consulta que muestre el n° de departamento, nombre del departamento y el n° de trabajadores que trabajan en el departamento que tiene un menor n° de empleados.

```
SELECT    d.deptno  NUMERO,  
          d.dname   NOMBRE,  
          COUNT(e.ename) TRABAJADORES
```

```
FROM emp e, dept d
```

```
WHERE e.deptno(+) = d.deptno
```

```
GROUP BY d.deptno, d.dname
```

*/*Ahora nos quedamos sólo con el departm con menor n° de trabajadores*/*

```
HAVING COUNT(e.ename) = ( SELECT MIN(COUNT(e.ename))  
                          FROM emp e,dept d  
                          WHERE e.deptno(+) = d.deptno  
                          GROUP BY d.deptno);
```

NOTA: Si NO queremos considerar el departamento con 0 empleados (departamento 40), entonces bastará con quitar los dos (+) !!!!!

2) Crear una consulta que muestre, *para todos los trabajadores que trabajan en un departamento localizado en Dallas*, el n° de empleado, nombre, salario, n° de departamento, y la media del salario del departamento donde trabaja.

La clave está en crear una subtabla (tablanueva) que contendrá el número de departamento y la media salarial de ese departamento (localizado en Dallas)

Tablanueva

<u>DEPARTAMENTO</u>	<u>MEDIA</u>
20	2175

```
SELECT  e.empno  NUM_EMPLEADO,
        e.ename  NOMBRE,
        NVL(e.sal,0) SALARIO,
        tablanueva.departamento NUM_DEPART,
        tablanueva.media  MEDIA_SALARIAL

FROM emp e,
(
    SELECT  e.deptno      departamento,
            AVG( NVL(e.sal,0) )  media
    FROM emp e, dept d
    WHERE  e.deptno = d.deptno
    AND    d.loc     = 'DALLAS'
    GROUP BY e.deptno
) tablanueva

/* Sólo nos interesan los del departamento 20 de Dallas */
WHERE e.deptno = tablanueva.departamento;
```


3) Muestre el nombre del departamento, media del salario de los empleados de dicho departamento, el nombre y el salario de los empleados que trabajan en él; para aquel departamento que tenga la media más alta. Además, no se deberán mostrar aquellos trabajadores cuyo salario es inferior a dicha media.

La clave está en crear una subtabla (tablanueva) que contendrá el número de departamento y la media salarial de ese departamento (DEL QUE TENGA LA MEDIA SALARIAL MÁS ALTA)

Tablanueva

<u>DEPARTAMENTO</u>	<u>MEDIAALTA</u>
10	2187,5

```

SELECT      d.dname NOMBRE_DEPART,
            tablanueva.mediaalta MEDIA_SALARIAL,
            e.ename NOMBRE_EMPLEADO,
            NVL(e.sal,0) SALARIO

FROM emp e, dept d,
    (
        SELECT      e.deptno      departamento,
                    AVG( NVL(e.sal,0) ) mediaalta
        FROM emp e
        GROUP BY e.deptno
        HAVING AVG( NVL(e.sal,0) ) = (
                                SELECT MAX( AVG( NVL(e.sal,0) ) )
                                FROM emp e
                                GROUP BY e.deptno)
    ) tablanueva

WHERE      e.deptno = d.deptno

/* Sólo nos interesan los del departamento que tenía la media más alta */
AND        e.deptno = tablanueva.departamento

/*Ahora, no se deben mostrar aquellos cuyo salario sea inferior a dicha MEDIAALTA*/
AND        NVL(e.sal,0) >= tablanueva.mediaalta;

```

4) Cree una consulta que muestre el número del departamento, el nombre del departamento, el n° de empleados y promedio de salario en cada uno de los departamentos. Muestre además el salario, el empleo (job) y el nombre de todos los empleados que trabajan en cada departamento.

La clave está en crear una subtabla (tablanueva) que contendrá el número de cada departamento (deptno), el nombre de cada departamento (dname), el número de empleados de cada departamento (empleados) y la media salarial de cada departamento (mediasal)

Tablanueva				
<u>DEPTNO</u>	<u>DNAME</u>	<u>EMPLEADOS</u>	<u>MEDIASAL</u>	
10	ACCOUNTING	4	2187,5	
20	RESEARCH	5	2175	
30	SALES	6	1566,66667	
40	OPERATIONS	0	0	

Nota: Las dos tablas (emp y tablanueva) las voy a relacionar mediante los DEPTNO!!
Nota: Ponemos los (+) para que nos imprima el departamento 40!!!!

```

SELECT  e.ename          ENAME,
        e.job            JOB,
        NVL(e.sal,0)      SAL,
        tablanueva.deptno DEPTNO,
        tablanueva.dname DNAME,
        tablanueva.empleados "Num empleados ",
        tablanueva.mediasal "Media salarial "

FROM    emp e,
        (
            SELECT  d.deptno DEPTNO,
                    d.dname DNAME,
                    COUNT( e.ename ) EMPLEADOS,
                    AVG( NVL(sal,0) ) MEDIASAL
            FROM    emp e, dept d
            WHERE   e.deptno(+) = d.deptno
            GROUP BY d.deptno, d.dname
        ) tablanueva

WHERE   e.deptno(+) = tablanueva.deptno

ORDER BY tablanueva.deptno;

```

5) Crear una tabla dep_caracteristicas que contenga para cada departamento

- El número del departamento
- El nombre del departamento
- El número de empleados que trabajan en ese departamento.

NOTA: Mostrar también los departamentos sin empleados y poner un mensaje 'NO HAY EMPLEADOS'.

NOTA: La idea es justamente la misma que la del ejercicio anterior. Creamos la subtabla y luego las enlazamos mediante los deptno. También usamos (+). El que diga crear una tabla simplemente tiene como novedad lo que aparece en color naranja.

CREATE TABLE dep_caracteristicas

AS

```
SELECT  NVL(e.ename,'No hay empleados') ENAME,  
        NVL(e.job,'No hay empleados')   JOB,  
        NVL(e.sal,0)                     SAL,  
        tablanueva.deptno                DEPTNO,  
        tablanueva.dname                 DNAME,  
        tablanueva.empleados             EMPLEADOS
```

FROM emp e,

```
(  
    SELECT      d.deptno DEPTNO,  
                d.dname DNAME,  
                COUNT(e.ename) EMPLEADOS  
    FROM emp e, dept d  
    WHERE e.deptno(+) = d.deptno  
    GROUP BY d.deptno, d.dname  
    ) tablanueva
```

WHERE e.deptno(+) = tablanueva.deptno

ORDER BY tablanueva.deptno;

6) Muestra el número de empleado, nombre de empleado, salario y número de departamento para aquellos empleados cuyo nivel salarial sea **mayor** que el nivel salarial medio de la ciudad en la que trabajan. Ordena el resultado por el número de departamento.

DUDA IMPORTANTE!!! : ¿¿Qué entendemos por Nivel o Grado Salarial Medio de la ciudad en la que trabajan???

Opción 1: HACER LA MEDIA DE LOS GRADOS!!!

Opción 2: Calcular el SALARIO MEDIO y ver qué grado le corresponde!!!

El ejercicio se ha resuelto considerando la PRIMERA de las opciones!!!

La subtabla tablanueva es esta:

DEPTNO	GRADOMEDIO
10	3,66666667
20	2,8
30	2,5

```

SELECT  e.empno          EMPNO,
        e.ename          ENAME,
        NVL(e.sal,0)      SALARY,
        e.deptno         DEPTNO,
        s.grade          GRADO_SALARIAL,
        tablanueva.gradomedio GRADO_SALARIAL_MEDIO

FROM emp e, salgrade s,
(
    SELECT  e.deptno DEPTNO,
            AVG(s.grade) GRADOMEDIO
    FROM emp e,salgrade s
    WHERE e.sal BETWEEN s.losal AND s.hisal
    GROUP BY e.deptno
) tablanueva

WHERE e.sal BETWEEN s.losal AND s.hisal

AND e.deptno = tablanueva.deptno
AND s.grade > tablanueva.gradomedio
/* Me quedo sólo con los de grado salarial MAYOR que el grado salarial medio de su
departamento */

ORDER BY e.deptno;
```

7) Cree una consulta que muestre la siguiente información:

-DEPARTAMENTO (Nombre del departamento)
 -NO_EMP (Total de empleados de cada departamento)
 -P_SALARIOS (Promedio de salario de cada departamento)
 -MEJOR_PAGADOS (Cantidad de trabajadores cuyo salario es superior a la media salarial de su departamento)

Notas: Se considera con valor 0 aquellos departamentos que no tengan asignados empleados. Y, se consideran con valor 0 aquellos que no tengan asignado un salario

DEPARTAMENTO	NO_EMP	P_SALARIOS	MEJOR_PAGADOS
.....
.....
.....

*El problema está en calcular los mejor pagados. Por ello, nos creamos nuestra tabla auxiliar (tablanueva) que contendrá el valor de las MEDIAS!!! La tabla auxiliar tendrá un campo **deptno** para así poder relacionarse con la tabla emp!!! (como siempre)*

SELECT

tablanueva.dname DEPARTAMENTO,
 tablanueva.empleados NO_EMP,
 tablanueva.mediasal P_SALARIOS,
 COUNT(e.ename) MEJORPAGADOS

FROM emp e,

(
 SELECT d.deptno DEPTNO,
 d.dname DNAME,
 COUNT(e.ename) EMPLEADOS,
 AVG(NVL(e.sal,0)) MEDIASAL
 FROM emp e, dept d
 WHERE e.deptno(+) = d.deptno
 GROUP BY d.deptno, d.dname
) tablanueva

WHERE e.deptno(+) = tablanueva.deptno /*Para relacionar las 2 tablas*/

/*Hay que poner los paréntesis del AND!*//*El OR se pone para que imprima el dept 40!*/
 AND (NVL(e.sal,0) > tablanueva.mediasal OR tablanueva.mediasal=0)
 GROUP BY tablanueva.dname, tablanueva.empleados, tablanueva.mediasal;

8) Mostrar los departamentos en los que todos los empleados ganan lo mismo o menos que su manager.

/ La idea es: **A todos los departamentos** le quitamos (MINUS) **los departamentos en los que todos los empleados ganan más que su manager** */*

**SELECT deptno
FROM emp**

MINUS

**SELECT DISTINCT empleado.deptno
FROM emp empleado, emp manager
WHERE empleado.mgr = manager.empno
AND empleado.sal > manager.sal;**

9) Obtener el nombre de los managers (empleados con subordinados) que tiene a su cargo únicamente empleados que trabajan en el mismo departamento que él y que además tienen menor nivel (grado) salarial que él.

(Nota: La comparación final forzosamente debe hacerse con el < >, lo que nos obliga a usar NOT IN. Sino no funciona)

SELECT DISTINCT manager.ename

**FROM emp manager,
 emp subordinado,
 salgrade managersalg,
 salgrade subordinadosalg**

WHERE manager.empno = subordinado.mgr

AND manager.sal BETWEEN managersalg.losal AND managersalg.hisal

AND subordinado.sal BETWEEN subordinadosalg.losal AND subordinadosalg.hisal

AND *subordinadosalg.grade < managersalg.grade*

AND manager.ename NOT IN (

***SELECT DISTINCT manager.ename
FROM emp manager, emp subordinado
WHERE manager.empno = subordinado.mgr
AND manager.deptno < > subordinado.deptno***

);

EXÁMENES PRÁCTICOS BBDD (tipo 3)

11 Crear un bloque PL/SQL, que introduzca en la tabla TOP_DOGS el nombre y el salario de aquellos empleados (tabla emp) que posean el mismo salario. Dicho bloque debe cumplir lo siguiente:

12 Si por ejemplo los empleados SMITH y JAMES tienen de salario 3000 deben ser introducidos. Además, si también se cumple que ADAMS, WARD y MARTIN tienen salario de 2500 deben ser introducidos.

13 Solamente no se deberán introducir aquellos empleados cuyo salario sea diferente al resto de los empleados de la tabla emp.

DECLARE

```
v_empleado emp.ename%TYPE;
```

```
v_salario emp.sal%TYPE;
```

```
CURSOR cursor1 IS
```

```
    SELECT empleado1.ename, empleado1.sal  
    FROM emp empleado1, emp empleado2
```

```
    WHERE empleado1.sal = empleado2.sal  
    AND      empleado1.empno <>
```

```
empleado2.empno;
```

BEGIN

```
DELETE FROM top_dogs;
```

```
IF NOT cursor1%ISOPEN THEN OPEN cursor1;
```

```
END IF;
```

```
LOOP
```

```
    FETCH cursor1 INTO v_empleado, v_salario; /*OJO: No hay  
paréntesis*/
```

```
    EXIT WHEN cursor1%NOTFOUND;
```

```
    INSERT INTO top_dogs(name,salary) VALUES  
(v_empleado,v_salario);
```

```
END LOOP;
```

```
CLOSE cursor1;
```

```
END;
```

```
/
```

```
/*Para ver el resultado: SELECT * FROM top_dogs; */
```

14 Crear un bloque PL/SQL, que almacene en la tabla MESSAGES, para cada departamento su n° de departamento, su nombre, el nombre y salario de los empleados cuyo salario está comprendido entre el salario máximo y mínimo para dicho departamento. Cada salario deberá ser actualizado según los distintos porcentajes:

15 Si el empleado tiene un salario menor de 2000 se le aplica el 10% de aumento.

16 Si el empleado tiene un salario comprendido entre 2000 y 3000 se le aplica el 12% de descuento.

17 Si el empleado tiene un salario mayor de 3000 se le aplica un 15% de descuento. *Nota: Imprimir la ocurrencia de un posible error*

VARIABLE mensaje VARCHAR2(60)

DECLARE

```
v_depart    emp.deptno%TYPE;
v_nombre    dept.dname%TYPE;
V_empleado  emp.ename%TYPE;
v_salario    emp.sal%TYPE;
v_resultado  messages.results%TYPE;
CURSOR cursor1 IS
```

```
    SELECT e.deptno, d.dname, e.ename, e.sal
    FROM emp e, dept d,
         (SELECT deptno DEPARTAMENTO, MIN(sal) minimo, MAX(sal)
maximo
         FROM emp
         GROUP BY deptno) tablanueva
    WHERE e.deptno = d.deptno
    AND e.sal > tablanueva.minimo
    AND e.sal < tablanueva.maximo
    AND e.deptno = tablanueva.departamento;
```

BEGIN

```
DELETE FROM messages;
IF NOT cursor1%ISOPEN THEN OPEN cursor1;
END IF;
LOOP
    FETCH cursor1 INTO v_depart, v_nombre, v_empleado,
v_salario;
    EXIT WHEN cursor1%NOTFOUND;
    IF v_salario < 2000 THEN v_salario :=
v_salario*1.10;
    ELSIF v_salario BETWEEN 2000 AND 3000 THEN v_salario :=
v_salario*0.88;
    ELSE v_salario := v_salario*0.85;
```



```

        END IF;
        v_resultado := v_depart||' '||v_nombre||' '||v_empleado||'
        ||v_salario;
        INSERT INTO messages(results) VALUES (v_resultado);
    END LOOP;
    CLOSE cursor1;
    :mensaje := 'BLOQUE REALIZADO CON ÉXITO';

EXCEPTION
    WHEN OTHERS THEN    :mensaje := 'HA OCURRIDO ALGUN ERROR';

END;
/
PRINT mensaje

```

18 Cree un procedimiento PL/SQL que al inicio solicite un valor, e inserte en la tabla MESSAGES el nombre del empleado, el salario, el nombre del departamento donde trabaja, para los empleados cuyo salario, sea mayor que el valor introducido al inicio, y además sea superior a la media del salario del departamento donde trabajan. El procedimiento debe controlar las siguientes excepciones:

19 Si el valor introducido es superior a 6000 debe aparecer el mensaje 'El salario debe ser inferior a 6000'.

20 Si el valor introducido es inferior a 100 debe aparecer el mensaje 'El salario debe ser superior a 100'.

21 Si ocurre cualquier error se debe mostrar el mensaje 'Ha ocurrido un error'.

Nota: La solución de este ejercicio se debe realizar utilizando un cursor.

```

ACCEPT valor          PROMPT 'Introduzca valor: '
VARIABLE mensaje VARCHAR2(60)

```

```

DECLARE

```

```

    excepcion1    EXCEPTION;
    excepcion2    EXCEPTION;
    v_valor        emp.sal%TYPE := &valor;
    v_nombre       emp.ename%TYPE;
    v_salario      emp.sal%TYPE;
    v_departam     dept.dname%TYPE;
    v_resultado    messages.results%TYPE;
    CURSOR cursor1 IS
        SELECT e.ename, NVL(e.sal,0), d.dname
        FROM   emp e, dept d, (
                SELECT  e.deptno  DEPART,  AVG(  NVL(e.sal,0)  )

```

```

MEDIASAL

```

```

        FROM emp e
        GROUP BY deptno
        )tablanueva
    WHERE  e.deptno = d.deptno
    AND    e.deptno = tablanueva.depart
    AND    e.sal    > tablanueva.mediasal
    AND    e.sal    > v_valor;

```

```

BEGIN
    DELETE FROM messages;
    IF      v_valor > 6000 THEN RAISE excepcion1;
    ELSIF   v_valor < 100  THEN RAISE excepcion2;
    END IF;
    IF NOT cursor1%ISOPEN THEN OPEN cursor1;
    END IF;
    LOOP
        FETCH      cursor1      INTO      v_nombre, v_salario,
v_departam;
        EXIT WHEN cursor1%NOTFOUND;
        v_resultado := v_nombre || ' ' || v_salario || ' ' ||
v_departam;
        INSERT      INTO      messages(results)      VALUES
(v_resultado);
    END LOOP;
    :mensaje := 'Salario introducido CORRECTO!!';
    CLOSE cursor1;
EXCEPTION
    WHEN excepcion1 THEN :mensaje := 'El salario debe ser
inferior a 6000';
    WHEN excepcion2 THEN :mensaje := 'El salario debe ser
superior a 100';
    WHEN OTHERS THEN :mensaje := 'HA OCURRIDO
ERROR';
END;
/
PRINT mensaje

```

22 Se desea aumentar el salario de los empleados, destinando un incremento de 2000\$ para todos los emp que tienen el mismo cargo, es decir, que se destinan 2000 a los vendedores, analistas, etc. Esta cantidad debe repartirse a partes iguales para todos los empleados que tienen el mismo cargo. Escriba un bloque PL/SQL para modificar el sueldo de los empleados según este criterio y comprobar el resultado.

DECLARE

```

    v_salarionuevo      emp.sal%TYPE;
    v_empleado          emp.empno%TYPE;
    CURSOR cursor1 IS
        SELECT          tablaauxiliar.salarionuevo,
        tablaauxiliar.codigo
        FROM
            (
                SELECT e.empno codigo,
                       e.deptno departament,
                       NVL(e.sal,0) salario,
                       NVL(e.sal,0)+(2000/tablanueva.numero)
        salarionuevo
                FROM emp e, (
                    SELECT  e.deptno departamento,
                           COUNT(e.ename) numero
                    FROM emp e
                    GROUP BY deptno
                )tablanueva
                WHERE      e.deptno =
        tablanueva.departamento
            )tablaauxiliar;

```

```

BEGIN
    IF NOT cursor1%ISOPEN THEN OPEN cursor1;
    END IF;
    LOOP
        FETCH    cursor1    INTO    v_salarionuevo,
v_empleado;
        EXIT WHEN cursor1%NOTFOUND;
        UPDATE emp
        SET sal = v_salarionuevo
        WHERE empno = v_empleado;
    END LOOP;
    CLOSE cursor1;

END;
/

```

23 Cree un bloque PL/SQL que, después de solicitar el número de un departamento de la empresa, almacene en la tabla MESSAGES la siguiente información:

24	Nombre del departamento.
25	El número de empleados del departamento.
26	Promedio de salario del departamento.
27	Mayor salario del departamento.

Además, se deben controlar las siguientes excepciones:

21 Si el departamento no tiene empleados, debe indicarse el siguiente mensaje: ¡El departamento no tiene empleados!

22 Si el promedio de salario del departamento es inferior al de toda la empresa, debe indicarse el mensaje: ¡Departamento con trabajadores en precario!

23 Si el departamento introducido no existe, debe indicarse el mensaje: ¡Departamento introducido NO existente!

```

ACCEPT valor    PROMPT    'Introduzca departamento: '
VARIABLE mensaje VARCHAR2(60)

```

```

DECLARE

```

```

excepcion1 EXCEPTION;
excepcion2 EXCEPTION;
excepcion3 EXCEPTION;
v_valor      dept.deptno%TYPE := &valor;
v_dname      dept.dname%TYPE;
v_numero     NUMBER;
v_mediasal   emp.sal%TYPE;
v_maximo     emp.sal%TYPE;
resultado    messages.results%TYPE;
mediaempresa emp.sal%TYPE;
CURSOR cursor1 IS
    SELECT d.dname NOMBREDEPT,
           COUNT(e.ename) NUMERO,
           AVG( NVL(e.sal,0) ) MEDIASAL,
           MAX( NVL(e.sal,0) ) MAXIMO
    FROM emp e, dept d
    WHERE e.deptno (+) = d.deptno
    GROUP BY d.deptno, d.dname
    HAVING d.deptno = v_valor;

BEGIN
    SELECT AVG( NVL(e.sal,0) )      /* Se pueden poner subconsultas en el
BEGIN!! */
    INTO mediaempresa
    FROM emp e;
    DELETE FROM messages;
    IF NOT cursor1%ISOPEN THEN OPEN cursor1;
    END IF;
    LOOP
        FETCH      cursor1      INTO      v_dname, v_numero, v_mediasal,
v_maximo;
        EXIT WHEN cursor1%NOTFOUND;
        resultado := v_dname || ' ' || v_numero || ' ' || v_mediasal || ' ' ||
v_maximo;
        INSERT INTO messages (results) VALUES
(resultado);
    END LOOP;
    IF      v_numero = 0                      THEN RAISE
excepcion1;
    ELSIF   v_mediasal < mediaempresa THEN RAISE
excepcion2;
    ELSIF   v_dname IS NULL                  THEN RAISE
excepcion3;
    END IF;
    :mensaje := 'TODO OK';
    CLOSE cursor1;

```

```

EXCEPTION
    WHEN excepcion1 THEN :mensaje:='El departamento no tiene
empleados';
    WHEN excepcion2 THEN :mensaje:='Departamento con trabajadores en
precario';
    WHEN excepcion3 THEN :mensaje:='Departamento introducido NO
existente';

END;
/
PRINT mensaje

```

28 Crear un bloque PL/SQL, que almacene en la tabla MESSAGES, para cada departamento su n° de departamento, su nombre, su localización, el nombre y salario de los empleados cuyo salario está comprendido entre el salario máximo y mínimo para dicho departamento. Cada salario deberá ser actualizado según los distintos porcentajes:

29 Si el empleado tiene un salario menor de 2000 se le aplica el 12% de aumento.

30 Si el empleado tiene un salario comprendido entre 2000 y 3000 se le aplica el 14% de descuento.

31 Si el empleado tiene un salario mayor de 3000 se le aplica un 16% de descuento.

Nota: Imprimir en pantalla la ocurrencia de un posible error.

VARIABLE mensaje VARCHAR2(60)

DECLARE

resultado messages.results%TYPE;

v_deptno emp.deptno%TYPE;

v_dname dept.dname%TYPE;

v_loc dept.loc%TYPE;

v_ename emp.ename%TYPE;

v_sal emp.sal%TYPE;

CURSOR cursor1 IS

SELECT e.deptno, d.dname, d.loc, e.ename, e.sal
FROM emp e, dept d,

(
SELECT **d.deptno**
departam,
MAX(NVL(e.sal,0))
maximo,
MIN(NVL(e.sal,0))
minimo
FROM emp e, dept d
WHERE e.deptno (+) = d.deptno
GROUP BY d.deptno
)tablanueva
WHERE e.deptno = d.deptno
AND e.deptno = tablanueva.departam
AND e.sal > tablanueva.minimo
AND e.sal < tablanueva.maximo;

BEGIN

DELETE FROM messages;

IF NOT cursor1%ISOPEN THEN OPEN cursor1;

END IF;

LOOP

FETCH cursor1 INTO v_deptno, v_dname, v_loc, v_ename,
v_sal;

EXIT WHEN cursor1%NOTFOUND;

IF v_sal < 2000 THEN v_sal := v_sal * 1.12;

ELSIF v_sal BETWEEN 2000 AND 3000 THEN
v_sal := v_sal * 0.86;

ELSE v_sal := v_sal * 0.84;

```

        END IF;
        resultado := v_deptno|| ' ' ||v_dname|| ' ' ||v_loc|| ' ' ||v_ename|| '
' ||v_sal;
        INSERT INTO messages(results) VALUES
(resultado);
    END LOOP;
    :mensaje := 'Realizado con éxito';
    CLOSE cursor1;

EXCEPTION
    WHEN OTHERS THEN :mensaje := 'Ha ocurrido
algún error';

END;
/
PRINT mensaje

```

32 Realizar una consulta PL/SQL para insertar en la tabla MESSAGES el número del empleado, nombre y salario para el manager con mayor salario de cada departamento. Incrementar el salario según estos casos:

- 33 Si gana menos de 2000 incrementar en un 10%**
- 34 Si gana entre 2000 y 3000 incrementar un 12%**
- 35 Si gana más de 3000 incrementar un 15%**

VARIABLE mensaje **VARCHAR2**(60)

DECLARE

```
    resultado    messages.results%TYPE;
    v_empno      emp.empno%TYPE;
    v_ename      emp.ename%TYPE;
    v_salario    emp.sal%TYPE;
    CURSOR cursor1 IS
        SELECT      DISTINCT  manager.empno,  manager.ename,
manager.sal
        FROM emp manager, emp subordinado, (
            SELECT      manager.deptno
departamento,
                                MAX(manager.sal)
                                maximo
        FROM emp manager, emp
subordinado
        WHERE      manager.empno =
subordinado.mgr
                GROUP BY  manager.deptno

)tablanueva
        WHERE  manager.empno = subordinado.mgr
        AND  manager.deptno = tablanueva.departamento
        AND  manager.sal = tablanueva.maximo;
```

BEGIN

```
    DELETE FROM messages;
    IF NOT cursor1%ISOPEN THEN OPEN
cursor1;
    END IF;
    LOOP
        FETCH cursor1 INTO v_empno, v_ename,
v_salario;
        EXIT WHEN cursor1%NOTFOUND;
        IF v_salario<2000 THEN
v_salario:=v_salario*1.10;
        ELSIF v_salario BETWEEN 2000 AND 3000 THEN
v_salario:=v_salario*1.12;
        ELSE v_salario:=v_salario*1.15;
        END IF;
        resultado := v_empno||' '||v_ename||' '||v_salario;
        INSERT INTO messages(results) VALUES
(resultado);
    END LOOP;
    :mensaje := 'Operación realizada con éxito';
    CLOSE cursor1;
```

```
END;  
/  
PRINT mensaje
```

36 Realice un bloque PL/SQL para insertar un empleado (tabla emp) en el departamento con mayor n° de trabajadores de la empresa. El salario y la comisión que se asignará a este empleado coincidirá ambos con el salario y la comisión del empleado 7839, la fecha de alta corresponderá con la fecha actual. El identificador (empno) será siempre un n° consecutivo (uno más) al mayor ya almacenado en la tabla y el nombre del empleado será solicitado al inicio.

El procedimiento debe controlar además las siguientes excepciones:

37 Si después de insertar el empleado se comprueba que existen más de tres empleados cuyo salario y comisión coinciden ambos con el salario y comisión del empleado 7839, se debe mostrar el siguiente mensaje: “Hay más de 3 empleados con el mismo salario y comisión que el 7839”. Además se debe garantizar que la transacción no tenga validez. (Usaremos DELETE)

38 Si después de insertar el empleado el promedio de salario de su departamento es mayor que el promedio del resto (de cada uno). Se mostrará el mensaje: “Se ha sobrepasado el límite salarial”.

39 Bajo la ocurrencia de cualquier otro error se mostrará el código de error y el mensaje asociado al mismo.

*/*Nota: Al insertar una sola fila, no usaremos cursor. Mediante las subconsultas apropiadas iremos almacenando los valores correspondientes en las variables, usando INTO */*

```
ACCEPT nombre PROMPT 'Introduzca el nombre: '
```

```
VARIABLE mensaje VARCHAR2(300)
```

```
DECLARE
```

excepcion1	EXCEPTION;
excepcion2	EXCEPTION;
v_inser	emp.sal%TYPE;
v_mayor	emp.sal%TYPE;
v_contador	NUMBER;
v_empno	emp.empno%TYPE;
v_ename	emp.ename%TYPE := '&nombre';
v_fecha	emp.hiredate%TYPE;
v_sal	emp.sal%TYPE;
v_comision	emp.comm%TYPE;
v_deptno	emp.deptno%TYPE;
v_codigo_error	NUMBER;

```
v_mensaje_error VARCHAR2(255);
```

```
BEGIN
```

```
    SELECT e.deptno departamento
           INTO v_deptno
    FROM emp e
    GROUP BY e.deptno
    HAVING COUNT(e.ename) = (SELECT
MAX(COUNT(e.ename))
                                FROM emp e
                                GROUP BY e.deptno);
```

```
    SELECT SYSDATE
           INTO v_fecha
    FROM dual;
```

```
    SELECT sal, comm
           INTO v_sal, v_comision
    FROM emp
    WHERE empno=7839;
```

```
    SELECT MAX(empno)+1
           INTO v_empno
    FROM emp;
```

```
    INSERT INTO emp(empno, ename, hiredate, sal, comm,
deptno)
    VALUES (v_empno, v_ename, v_fecha, v_sal, v_comision,
v_deptno);
```

```
/*Para las restricciones . . .*/
```

```
/* Guardamos la media salarial del departamento (después de haber
insertado) */
```

```
    SELECT AVG( NVL(sal,0) )
           INTO v_inser
    FROM emp
    GROUP BY deptno
    HAVING deptno = v_deptno;
```

```
/* Guardamos la mayor media salarial (sea del departamento que sea) */
```

```
    SELECT MAX( AVG( NVL(sal,0) ) )
           INTO v_mayor
    FROM emp
```

GROUP BY deptno;

/ Contamos el n° de trabajadores con salario y comm igual a la del 7839.
OJO: Hay que usar los NVL, sino no cuenta, porque comm al ser nulo no
lo compararía!! */*

```
SELECT COUNT(ename)
      INTO v_contador
FROM emp
WHERE NVL(sal,0) = (SELECT NVL(sal,0) FROM emp WHERE
empno=7839)
AND NVL(comm,0) = (SELECT NVL(comm,0) FROM emp WHERE
empno=7839);
```

*/*Ya podemos lanzar las excepciones */*

```
IF v_contador>3 THEN RAISE excepcion1;
END IF;
```

```
IF v_inser=v_mayor THEN RAISE excepcion2;
END IF;
```

```
:mensaje := 'Todo OK';
```

EXCEPTION

```
WHEN excepcion1 THEN
:mensaje := 'Hay más de 3 empleados con el mismo salario y comisión que
el 7839';
```

```
DELETE FROM emp WHERE empno = v_empno;
```

```
WHEN excepcion2 THEN
:mensaje := 'Se ha sobrepasado el límite salarial';
```

```
WHEN OTHERS THEN /* Usamos esas funciones */
v_codigo_error := SQLCODE;
v_mensaje_error := SQLERRM;
:mensaje := v_codigo_error || ' ' || v_mensaje_error;
```

END;

/

```
PRINT mensaje
```

40 Cree un bloque PL/SQL que inserte en la tabla MESSAGES el número de empleado, el nombre y el salario del manager que gana más en cada departamento. Dicho salario deberá ser actualizado según los siguientes porcentajes:

41 Si el manager tiene un salario < 2000 se le aplica el 12% de aumento.

42 Si el manager tiene un salario comprendido entre 2000 y 3000 se le aplica el 14% de aumento.

43 Si el manager tiene un salario mayor de 3000 se le aplica un 16% de aumento.

DECLARE

```

v_empno      emp.empno%TYPE;
v_ename      emp.ename%TYPE;
v_sal        emp.sal%TYPE;
v_deptno     emp.deptno%TYPE;
resultado    messages.results%TYPE;
CURSOR cursor1 IS
    SELECT DISTINCT  manager.empno,
                      manager.ename,
                      manager.sal,
                      manager.deptno
    FROM emp manager, emp subordinado,
        (
            SELECT      managers.deptno,
                        MAX( managers.sal) maximo
            FROM
                (
                    SELECT DISTINCT  manager.empno,
                                    manager.ename,
                                    manager.sal,
                                    manager.deptno
                    FROM emp manager, emp subordinado
                    WHERE              manager.empno      =
subordinado.mgr
                ) managers
            GROUP BY managers.deptno
        ) tablanueva

```

```

WHERE manager.empno = subordinado.mgr
AND manager.deptno = tablanueva.deptno
AND manager.sal      = tablanueva.maximo;

```

```

BEGIN
DELETE FROM messages;
IF NOT cursor1%ISOPEN THEN OPEN cursor1;
END IF;
LOOP
    FETCH cursor1 INTO v_empno, v_ename,
v_sal, v_deptno;
    EXIT WHEN cursor1%NOTFOUND;
    IF v_sal<2000 THEN v_sal:=v_sal*1.12;
    ELSIF v_sal BETWEEN 2000 AND 3000 THEN
v_sal:=v_sal*1.14;
    ELSE v_sal:=v_sal*1.16;
    END IF;
    resultado := v_empno || ' ' || v_ename || ' ' || v_sal || ' ' || v_deptno;
    INSERT INTO messages(results) VALUES
(resultado);
END LOOP;
CLOSE cursor1;
END;
/

```

```

/*****
*****/
/***** LA CONSULTA EN CORRELADA SERÍA ASÍ
*****/
/***** (como siempre pero añadiendo el AND final )
*****/

```

```

SELECT fuera.empno, fuera.ename, fuera.sal, fuera.deptno
FROM
(
    SELECT DISTINCT manager.empno,
manager.ename,
manager.sal,
manager.deptno
FROM emp manager, emp subordinado
WHERE manager.empno = subordinado.mgr
)fuera

```

```

WHERE 1 > ( SELECT COUNT(dentro.ename)
            FROM
            (
                SELECT DISTINCT  manager.empno,
                                manager.ename,
                                manager.sal,
                                manager.deptno
                FROM emp manager, emp subordinado
                WHERE            manager.empno =
subordinado.mgr
            )dentro
            WHERE dentro.sal > fuera.sal
            AND  dentro.deptno = fuera.deptno );
/*****
*****/
/*****
*****/

```

44 Cree un bloque PL/SQL que inserte en la tabla MESSAGES el tipo de trabajo, nombre y salario de los dos empleados que tienen un mayor sueldo para cada tipo de trabajo. Además, también se deberá insertar el nombre del departamento y la media del salario del departamento al cual pertenece cada trabajador.

DECLARE

```

v_job      emp.job%TYPE;
v_dname    dept.dname%TYPE;
v_ename    emp.ename%TYPE;
v_sal      emp.sal%TYPE;
v_media    emp.sal%TYPE;
resultado  messages.results%TYPE;
CURSOR     cursor1 IS
SELECT  fuera.job, d.dname, fuera.ename, NVL(fuera.sal,0), tablanueva.media
FROM emp fuera, dept d,
(
  SELECT deptno departamento, AVG(NVL(sal,0)) media
  FROM emp
  GROUP BY deptno
)tablanueva
WHERE 2 > ( SELECT COUNT(dentro.ename)
            FROM emp dentro
            WHERE NVL(dentro.sal,0) > NVL(fuera.sal,0)
            AND  dentro.job = fuera.job )
AND fuera.deptno = d.deptno
AND fuera.deptno = tablanueva.departamento;

```

BEGIN

DELETE FROM messages;

IF NOT cursor1%ISOPEN THEN OPEN cursor1;

END IF;

LOOP

FETCH cursor1 INTO v_job, v_dname, v_ename, v_sal, v_media;

EXIT WHEN cursor1%NOTFOUND;

resultado := v_job || ' ' || v_dname || ' ' || v_ename || ' ' || v_sal || ' ' ||

v_media;

INSERT INTO messages(results) VALUES(resultado);

END LOOP;

CLOSE cursor1;

END;

/