

## **2. DESARROLLO:**

- 1) Permita que el usuario ingrese un empleo y que el bloque anónimo: A) Muestre el promedio de sueldo semanal si el empleo es analista B) Muestre la cantidad de salarios semanales distintos en el caso de que el empleo sea supervisor o prog, C) Que muestre la diferencia entre el mayor salario y menor salario si el empleo es vendedor y D) Que muestre el mensaje correspondiente si el empleo no existe. (18 puntos)

### **RESPUESTA:**

```
SET SERVEROUTPUT ON;
set verify off;
DECLARE
V_cant number;
V_max number;
v_min number;
V_empleo empleado.empleo%type;
```

```
Begin
```

```
V_empleo := '&empleo';
```

```
If v_empleo = 'analista' then
```

```
Select avg(salario_sem) into v_cant from empleado where empleo = v_empleo ;
Dbms_output.put_line('promedio de sueldo - analista: ' || v_cant);
```

```
Elsif v_empleo = 'supervisor' or v_empleo = 'prog' then
```

```
Select count(distinct salario_sem) into v_cant from empleado where empleo =
v_empleo;
Dbms_output.put_line('Cantidad de salarios distintos para ' || v_empleo || ': ' || v_cant);
```

```
Elsif v_empleo = 'vendedor' then
```

```
Select max(salario_sem), min(salario_sem) into v_max, v_min from empleado
where empleo = v_empleo;
v_cant := v_max - v_min;
Dbms_output.put_line('Diferencia de salarios para ' || v_empleo || ': ' || v_cant);
```

```
Else
```

```
Dbms_output.put_line('No existe ese empleo');
```

```
End if;
```

```
End;
```

2) Escriba un bloque anónimo que reciba un nombre de empleo cumpliendo los siguientes puntos:

- ✓ Si el nombre pasado no corresponde al empleo de ningún empleado, tratar esta situación mostrando el mensaje “Ningún empleado tiene el empleo de <nombre dado>”.
- ✓ Si el nombre dado corresponde al empleo de más de un empleado, tratar la situación mostrando el mensaje “Más de un empleado posee el empleo de <nombre dado>”.
- ✓ Si el nombre pasado corresponde al empleo de un único empleado, mostrar el nombre de dicho empleado. (20 puntos)

### **RESPUESTA:**

```
declare
v_nombre empleado.nom_emp%type;
v_empleo empleado.empleo%type:=&empleo;
begin

select nom_emp into v_nombre
from empleado
where empleo = v_empleo;
dbms_output.put_line('Nombre del empleado:'|| ' '||v_nombre);

exception
when no_data_found then
dbms_output.put_line('Ningún empleado tiene el empleo de'|| ' '||v_empleo);
when too_many_rows then
dbms_output.put_line('Más de un empleado posee el empleo de'|| ' '||v_empleo);

end;
```

### **RESULTADO:**

```
- 'Analista':
Más de un empleado posee el empleo de analista

-'Gerente':
Nombre del empleado: Karla

- 'prof':
Ningún empleado tiene el empleo de prof
```

- 3) Implemente un cursor que muestre la cantidad de empleados, el nombre del departamento al cual pertenecen, de los empleados que ingresaron el año 1981 donde la cantidad de empleados sea mayor a 3. (28 puntos)

**RESPUESTA:**

```
Set serveroutput on;
begin

for C1 in (select count(fecha_ingreso) cant, nom_depto
from empleado, depto
where empleado.cod_depto = depto.cod_depto
and fecha_ingreso like '%81'
group by nom_depto
having count(fecha_ingreso) > 3)
loop
dbms_output.put_line(c1.cant || ' ' || c1.nom_depto);
end loop;

end;
```

**RESULTADO:**

```
bloque anónimo terminado
6 Ventas
4 Operación
```

- 4) Algunos empleados tienen una comisión asignada en cero. Por lo anterior, se pide actualizar este tipo de comisiones en la tabla empleado. Dependiendo de en qué rango se encuentre el salario semanal según la tabla sueldo, la comisión se debe actualizar para que sea un 20% del sueldo mínimo según el rango al que corresponda el salario semanal del empleado. (34 puntos)

**RESPUESTA:**

```
SET SERVEROUTPUT ON;
DECLARE
    CURSOR c_comision IS
        select cod_emp, salario_sem
        from empleado where comision=0 FOR UPDATE;
    v_emp empleado.cod_emp%TYPE;
    v_salario empleado.salario_sem%type;
    v_sueldo sueldo.sueldo_minimo%type;
BEGIN
    OPEN c_comision;
    FETCH c_comision INTO v_emp, v_salario;

    WHILE c_comision%found
    LOOP
        select sueldo_minimo into v_sueldo from sueldo
        where v_salario >= sueldo_minimo and v_salario <= sueldo_maximo;

        UPDATE empleado
        SET comision = v_sueldo* 0.2
        WHERE CURRENT OF c_comision;
        DBMS_OUTPUT.PUT_LINE( v_emp||' '|| round(v_sueldo*0.2));

        FETCH c_comision INTO v_emp, v_salario;
    END LOOP;
    CLOSE c_comision;
    COMMIT;
END;
```

**RESULTADO: Actualizaría**

<b>Cod_emp</b>	<b>comision</b>
7839	600
7566	400
7902	400
7369	140
7698	400
7782	400

7788	400
7844	280
7876	140
7900	140
7934	240
7697	600
7901	400
7368	140
7565	400
7838	600

16 filas afectadas