

Procedimiento que dado un código de un empleado mostrar en pantalla el código de empleado, nombre y apellidos de todos sus subordinados.

```
CREATE OR REPLACE PROCEDURE LISTADO_SUBORDINADOS (CODIGO  
EMPLOYEES.EMPLOYEE_ID%TYPE) IS
```

```
CURSOR SUBORDINADOS IS
```

```
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME  
FROM EMPLOYEES  
WHERE MANAGER_ID=CODIGO;
```

```
NOMBRE EMPLOYEES.FIRST_NAME%TYPE;  
APELLIDO EMPLOYEES.LAST_NAME%TYPE;  
CODIGOEMP EMPLOYEES.EMPLOYEE_ID%TYPE;
```

```
BEGIN
```

```
OPEN SUBORDINADOS;  
LOOP  
  FETCH SUBORDINADOS INTO CODIGOEMP, NOMBRE, APELLIDO;  
  EXIT WHEN SUBORDINADOS%NOTFOUND;  
  DBMS_OUTPUT.PUT_LINE(CODIGOEMP||' '||NOMBRE||' '||APELLIDO);  
END LOOP;
```

```
CLOSE SUBORDINADOS;
```

```
END;
```

```
EXECUTE LISTADO_SUBORDINADOS(101);
```

**Versión 2:**

```
CREATE OR REPLACE PROCEDURE LISTADO_SUBORDINADOS2 (CODIGO  
EMPLOYEES.EMPLOYEE_ID%TYPE) IS
```

```
CURSOR SUBORDINADOS IS
```

```
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME  
FROM EMPLOYEES  
WHERE MANAGER_ID=CODIGO;
```

```
REGISTRO SUBORDINADOS%ROWTYPE;
```

```
BEGIN
```

```
OPEN SUBORDINADOS;  
LOOP  
  FETCH SUBORDINADOS INTO REGISTRO;  
  EXIT WHEN SUBORDINADOS%NOTFOUND;
```

```

        DBMS_OUTPUT.PUT_LINE(REGISTRO.EMPLOYEE_ID || ' ' || REGISTRO.FIRST_NAME || '
' || REGISTRO.LAST_NAME);
    END LOOP;
    CLOSE SUBORDINADOS;

END;

```

### **Versión 3:**

CREATE OR REPLACE PROCEDURE LISTADO\_SUBORDINADOS3 (CODIGO  
EMPLOYEES.EMPLOYEE\_ID%TYPE) IS

```

CURSOR SUBORDINADOS IS
    SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME
    FROM EMPLOYEES
    WHERE MANAGER_ID=CODIGO;

```

BEGIN

FOR REGISTRO IN SUBORDINADOS LOOP

```

        DBMS_OUTPUT.PUT_LINE(REGISTRO.EMPLOYEE_ID || ' ' || REGISTRO.FIRST_NAME || '
' || REGISTRO.LAST_NAME);

```

END LOOP;

END;

- Procedimiento que reciba nombre de departamento y emita:

a)Nombre y apellidos de todos sus empleados.

b)Número de empleados total.

c)Suma salarial de sus empleados.

### **Versión corta:**

CREATE OR REPLACE PROCEDURE INFO\_DEPARTAMENTOS2 (NOMBRE  
DEPARTMENTS.DEPARTMENT\_NAME%TYPE) IS

```

CURSOR EMPLEADOS_DEPT IS
    SELECT FIRST_NAME, LAST_NAME, SALARY
    FROM EMPLOYEES JOIN DEPARTMENTS USING (DEPARTMENT_ID)
    WHERE DEPARTMENT_NAME=NOMBRE;

```

TOTAL\_EMPLEADOS NUMBER(4):=0;

SUM\_SALARIAL NUMBER(9,2):=0;

BEGIN

FOR REGISTRO IN EMPLEADOS\_DEPT LOOP

```

        DBMS_OUTPUT.PUT_LINE(REGISTRO.FIRST_NAME || ' ' || REGISTRO.LAST_NAME);

```

```

SUM_SALARIAL:=SUM_SALARIAL+REGISTRO.SALARY;
TOTAL_EMPLEADOS:=TOTAL_EMPLEADOS+1;

END LOOP;
DBMS_OUTPUT.PUT_LINE('Total empleados: ' || TOTAL_EMPLEADOS);
DBMS_OUTPUT.PUT_LINE('Suma salarial: ' || SUM_SALARIAL);
END;

```

### Versión larga:

```

CREATE OR REPLACE PROCEDURE INFO_DEPARTAMENTOS (NOMBRE
DEPARTMENTS.DEPARTMENT_NAME%TYPE) IS

```

```

CURSOR EMPLEADOS_DEPT IS
SELECT FIRST_NAME, LAST_NAME
FROM EMPLOYEES JOIN DEPARTMENTS USING (DEPARTMENT_ID)
WHERE DEPARTMENT_NAME=NOMBRE;

```

```

TOTAL_EMPLEADOS NUMBER(2);
SUM_SALARIAL NUMBER(9,2);

```

```

BEGIN

```

```

SELECT COUNT(*) INTO TOTAL_EMPLEADOS
FROM EMPLOYEES JOIN DEPARTMENTS USING (DEPARTMENT_ID)
WHERE DEPARTMENT_NAME=NOMBRE;

```

```

SELECT SUM(SALARY) INTO SUM_SALARIAL
FROM EMPLOYEES JOIN DEPARTMENTS USING (DEPARTMENT_ID)
WHERE DEPARTMENT_NAME=NOMBRE;

```

```

FOR REGISTRO IN EMPLEADOS_DEPT LOOP

```

```

    DBMS_OUTPUT.PUT_LINE(REGISTRO.FIRST_NAME || ' ' || REGISTRO.LAST_NAME);

```

```

END LOOP;
DBMS_OUTPUT.PUT_LINE('Total empleados: ' || TOTAL_EMPLEADOS);
DBMS_OUTPUT.PUT_LINE('Suma salarial: ' || SUM_SALARIAL);
END;

```

- Dado el nombre de una localidad, mostrar un listado de empleados indicando al final el nombre del empleado con mayor salario de todos.

### CREATE OR REPLACE PROCEDURE INFO\_LOCALIDAD (CIUDAD LOCATIONS.CITY%TYPE) IS

```
CURSOR EMPLEADOS IS  
SELECT FIRST_NAME, LAST_NAME, SALARY  
FROM EMPLOYEES JOIN DEPARTMENTS USING (DEPARTMENT_ID)  
JOIN LOCATIONS USING (LOCATION_ID)  
WHERE CITY=CIUDAD  
ORDER BY SALARY;
```

**REGISTRO2 EMPLEADOS%ROWTYPE;**

**BEGIN**  
**FOR REGISTRO IN EMPLEADOS LOOP**

```
DBMS_OUTPUT.PUT_LINE(REGISTRO.FIRST_NAME||' '||REGISTRO.LAST_NAME);
REGISTRO2:=REGISTRO;
END LOOP;
```

```
DBMS_OUTPUT.PUT_LINE('EMPLEADOS CON MAYOR SALARIO: ');
```

```
FOR REGISTRO IN EMPLEADOS LOOP
  IF REGISTRO.SALARY=REGISTRO2.SALARY THEN
    DBMS_OUTPUT.PUT_LINE(REGISTRO.FIRST_NAME||' '||REGISTRO.LAST_NAME);
  END IF;
END LOOP;
END;
```

```
EXECUTE INFO_LOCALIDAD('Seattle');
```

- Escribir procedimiento que dada una ciudad muestre todos aquellos empleados cuyo salario está por encima de la media salarial de la ciudad.

### Versión 1:

**CREATE OR REPLACE PROCEDURE INFO\_LOCALIDAD2 (CIUDAD LOCATIONS.CITY%TYPE) IS**

```
CURSOR EMPLEADOS IS  
SELECT FIRST_NAME, LAST_NAME, SALARY  
FROM EMPLOYEES JOIN DEPARTMENTS USING (DEPARTMENT_ID)  
JOIN LOCATIONS USING (LOCATION_ID)  
WHERE CITY=CIUDAD AND SALARY > (SELECT AVG(SALARY)  
FROM EMPLOYEES JOIN DEPARTMENTS USING (DEPARTMENT_ID)  
JOIN LOCATIONS USING (LOCATION_ID)  
WHERE CITY=CIUDAD)  
ORDER BY SALARY;
```

BEGIN

```
FOR REGISTRO IN EMPLEADOS LOOP
    DBMS_OUTPUT.PUT_LINE(REGISTRO.FIRST_NAME || ' ' || REGISTRO.LAST_NAME || '
' || REGISTRO.SALARY);
END LOOP;
END;
```

**Versión 2:**

CREATE OR REPLACE PROCEDURE INFO\_LOCALIDAD3 (CIUDAD LOCATIONS.CITY%TYPE) IS

CURSOR EMPLEADOS IS

```
SELECT FIRST_NAME, LAST_NAME, SALARY
FROM EMPLOYEES JOIN DEPARTMENTS USING (DEPARTMENT_ID)
      JOIN LOCATIONS USING (LOCATION_ID)
WHERE CITY=CIUDAD;
```

MEDIA\_SALARIAL EMPLOYEES.SALARY%TYPE;

BEGIN

```
SELECT AVG(SALARY) INTO MEDIA_SALARIAL
FROM EMPLOYEES JOIN DEPARTMENTS USING (DEPARTMENT_ID)
      JOIN LOCATIONS USING (LOCATION_ID)
WHERE CITY=CIUDAD;
```

```
FOR REGISTRO IN EMPLEADOS LOOP
    IF REGISTRO.SALARY > MEDIA_SALARIAL THEN
        DBMS_OUTPUT.PUT_LINE(REGISTRO.FIRST_NAME || ' ' || REGISTRO.LAST_NAME);
    END IF;
END LOOP;
END;
```