# **PRÁCTICA PL/SQL – 456**

https://www.oracletutorial.com/plsql-tutorial/

1. Crear un bloque PL que acepte un número como entrada, elija a los N empleados con el sueldo más BAJO de la tabla TEMPLE y los inserte dentro de la tabla “OTRA” (insertar nombre y salario), además, si estos empleados llevan contratados en la empresa más de 25 años, se les debe INCREMENTAR su comisión en un 10% de su salario **(Cursores)**

**SI SALARIO ES 1000 Y COMISIÓN ES 230 --> COMISIÓN= 330**

**SI SALARIO ES 1000 Y COMISIÓN ES NULL --> COMISIÓN =100**

SET SERVEROUTPUT ON;

DECLARE

V\_NUM NUMBER(3):=&V\_NUM;

CURSOR C IS

SELECT NUMEM, NOMEM, SALAR, COMIS, (TRUNC(MONTHS\_BETWEEN(SYSDATE, FECIN)/12))AS ANTIGUEDAD FROM TEMPLE ORDER BY SALAR;

V\_NUMEM TEMPLE.NUMEM%TYPE;

V\_NOMEM TEMPLE.NOMEM%TYPE;

V\_SALAR TEMPLE.SALAR%TYPE;

V\_ANTI NUMBER(3);

V\_COMIS TEMPLE.COMIS%TYPE;

BEGIN

OPEN C;

WHILE C%ROWCOUNT<=V\_NUM AND C%FOUND LOOP

FETCH C INTO V\_NUMEM, V\_NOMEM, V\_SALAR, V\_ANTI, V\_COMIS;

IF SALAR=1000 AND COMIS=230 THEN

UPDATE TEMPLE

SET COMIS=330;

ELSIF SALAR=1000 AND COMIN IS NULL THEN

UPDATE TEMPLE

SET COMIS=100;

ENDIF;

IF V\_ANTI<25 THEN

UPDATE TEMPLE

SET COMIS=COMIS+SALAR\*0.10;

ENDIF;

INSERT INTO OTRA VALUES (C.NOMEM, C.SALAR);

END LOOP;

CLOSE C;

END;

/

1. Crear un programa que muestre los nombres de los empleados de la tabla TEMPLE que trabajan un departamento introducido por teclado. **(Cursores)**
2. A partir de los siguientes tres ejemplos de cursores, completa el programa para que se muestre la información recuperada con o sin registros, según se indica, usando en cada ejemplo estructuras repetitivas distintas (podemos usar LOOP, WHILE LOOP Y FOR LOOP en este orden). En el AulaVirtual está la tabla provincias con algunos registros: **(Cursores)**

CURSOR 1 - SIN USAR REGISTRO

------------------------------

DECLARE

CURSOR MI\_CURSOR IS

SELECT PROVINCIA, DESCRIPCION, PREFIJO

FROM PROVINCIAS;

BEGIN

END;

/

EJEMPLO 2 - CON REGISTRO

---------------

DECLARE

CURSOR MI\_CURSOR IS

SELECT \*

FROM PROVINCIAS;

BEGIN

END;

/

EJEMPLO 3 - CON REGISTRO

-------------

DECLARE

CURSOR MI\_CURSOR IS

SELECT PROVINCIA, DESCRIPCION

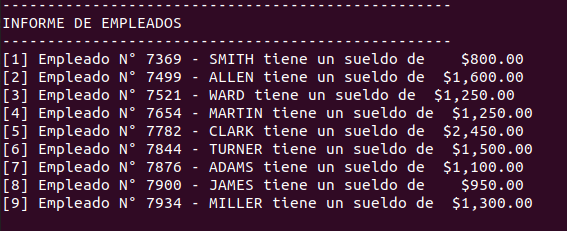
FROM PROVINCIAS;

BEGIN

END;

/

1. Genera un informe como el siguiente para todos los empleados con salario inferior a 2000 de la tabla EMP. **(Cursores)**



1. Crear un bloque PL que permita borrar a los empleados que no tienen comisión y los que la tienen dividirla entre dos. Se utilizará al crear el cursor, la cláusula FOR UPDATE y WHERE CURRENT OF. **(Cursores)**
2. Realizar un programa que muestre los códigos de los empleados y cuente aquellos cuya comisión es nula. Se utilizará la tabla TEMPLE. **(Cursores)**
3. Diseñar un bloque PL que muestre el nombre del departamento y el número de trabajadores que tiene cada uno. **(Cursores)**

**DECLARE**

**CURSOR C IS**

**SELECT NOMDE, COUNT(NUMEM)**

**FROM TEMPLE E, TDEPTO D**

**WHERE D.NUMDE = E.NUMDE(+)**

**GROUP BY NOMDE;**

**V\_DNOMBRE VARCHAR(20);**

**V\_NUM\_EMPLE NUMBER;**

**BEGIN**

**OPEN C;**

**FETCH C INTO V\_DNOMBRE, V\_NUM\_EMPLE;**

**WHILE C%FOUND LOOP**

**DBMS\_OUTPUT.PUT\_LINE(' EN '||V\_DNOMBRE||' HAY '||V\_NUM\_EMPLE||' EMPLEADOS ');**

**FETCH C INTO V\_DNOMBRE,V\_NUM\_EMPLE;**

**END LOOP;**

**CLOSE C;**

**END;**

**/**

1. Codificar un programa que visualice los dos empleados que ganan menos de cada oficio. **(Cursores)**

**DECLARE**

**CURSOR C IS**

**SELECT ENAME, JOB, SAL FROM EMP**

**ORDER BY JOB, SAL;**

**VR\_EMP C%ROWTYPE;**

**OFICIO\_ANT EMP.JOB%TYPE;**

**I NUMBER;**

**BEGIN**

**OPEN C;**

**OFICIO\_ANT:='\*';**

**FETCH C INTO VR\_EMP;**

**WHILE C%FOUND LOOP**

**IF OFICIO\_ANT <> VR\_EMP.JOB THEN**

**OFICIO\_ANT := VR\_EMP.JOB;**

**I :=0;**

**END IF;**

**IF I <2 THEN**

**DBMS\_OUTPUT.PUT\_LINE(VR\_EMP.JOB||' - ' ||VR\_EMP.ENAME||' - ' ||VR\_EMP.SAL);**

**I :=I+ 1;**

**END IF;**

**FETCH C INTO VR\_EMP;**

**END LOOP;**

**CLOSE C;**

**END;**

**/**

1. Escribir un bloque PL que reciba una cadena y visualice el apellido y el número de empleado de todos los empleados cuyo apellido contenga la cadena especificada. Al finalizar se visualizará el número de empleados mostrados. **(Cursores)**

**SET SERVEROUTPUT ON;**

**DECLARE**

**CADENA TEMPLE.NOMEM%TYPE:=&CADENA;**

**CURSOR C IS**

**SELECT NOMEM FROM TEMPLE WHERE NOMEM LIKE CADENA||'%';**

**VR\_TEMPLE C%ROWTYPE;**

**BEGIN**

**DBMS\_OUTPUT.PUT\_LINE('LA CADENA QUE SE BUSCA: '|| CADENA);**

**OPEN C;**

**FETCH C INTO VR\_TEMPLE;**

**WHILE (C%FOUND) LOOP**

**DBMS\_OUTPUT.PUT\_LINE(VR\_TEMPLE.NOMEM);**

**FETCH C INTO VR\_TEMPLE;**

**END LOOP;**

**DBMS\_OUTPUT.PUT\_LINE('NUMERO DE EMPLEADOS: '|| C%ROWCOUNT);**

**CLOSE C;**

**END;**

**/**

1. Escribir un bloque PL que permita **incrementar el salario un 5%** a un empleado introducido por teclado. Teniendo en cuenta que **sí la consulta no devuelve ninguna fila se visualizará un mensaje ‘Error. No encontrado’ y sí el salario es cero otro mensaje que diga ‘Error. Salario nulo’. (Excepciones)**

**INSERT INTO EMP (EMPNO, SAL, DEPTNO) VALUES (9998,NULL,10);**

**INSERT INTO EMP (EMPNO, SAL, DEPTNO) VALUES (9997,0,10);**

**SET SERVEROUTPUT ON;**

**DECLARE**

**V\_COD\_EMP EMP.EMPNO%TYPE:=&V\_COD\_EMP;**

**V\_SALAR EMP.SAL%TYPE;**

**ERROR\_SALARIO EXCEPTION;**

**BEGIN**

**SELECT SAL INTO V\_SALAR FROM EMP WHERE EMPNO=V\_COD\_EMP;**

**IF V\_SALAR<=0 THEN**

**RAISE ERROR\_SALARIO;**

**ELSE**

**DBMS\_OUTPUT.PUT\_LINE('CODIGO EMPLEADO: '||V\_COD\_EMP||' SALARIO: ' || V\_SALAR);**

**DBMS\_OUTPUT.PUT\_LINE('INCREMENTO DEL 5%');**

**UPDATE EMP**

**SET SAL=SAL\*1.05**

**WHERE EMPNO=V\_COD\_EMP;**

**DBMS\_OUTPUT.PUT\_LINE('SALARIO FINAL: ' || V\_SALAR\*1.05);**

**END IF;**

**EXCEPTION**

**WHEN NO\_DATA\_FOUND THEN**

**DBMS\_OUTPUT.PUT\_LINE('ERROR: NO ENCONTRADO');**

**WHEN ERROR\_SALARIO THEN**

**DBMS\_OUTPUT.PUT\_LINE('ERROR: SALARIO NULO');**

**WHEN OTHERS THEN**

**DBMS\_OUTPUT.PUT\_LINE('ERROR!');**

**END;**

**/**

1. Diseñar un bloque PL que permita calcular el salario de los empleados de los departamentos 111 y 112 de la tabla TEMPLE mediante un **cursor y un registro** de todos los empleados. Si el número de **hijos es igual a cero se tratará esta excepción mediante un bloque interno que evite la salida general del programa.** **(Excepciones)**

**SET SERVEROUTPUT ON;**

**DECLARE**

**ERROR\_HIJOS EXCEPTION;**

**CURSOR C IS SELECT NUMDE, SALAR, NUMHI FROM TEMPLE WHERE NUMDE=111 OR NUMDE=112 ORDER BY NUMHI DESC;**

**REGISTRO C%ROWTYPE;**

**V\_SAL111 NUMBER :=0;**

**V\_SAL112 NUMBER :=0;**

**BEGIN**

**WHILE C%FOUND LOOP**

**FETCH C INTO REGISTRO;**

**BEGIN**

**IF NUM=0 THEN**

**RAISE ERROR\_HIJOS;**

**EXCEPTION**

**WHEN ERROR\_HIJOS THEN**

**DBMS\_OUTPUT.PUT\_LINE(EXCEPCION: 0 HIJOS');**

**END;**

**IF REGISTRO.NUMDE=111 THEN**

**V\_SAL111:=V\_SAL111+REGISTRO.SALAR;**

**ELSE**

**V\_SAL112:=V\_SAL112+REGISTRO.SALAR;**

**END IF;**

**END LOOP;**

**CLOSE C;**

**END;**

**/**

1. Diseñar un bloque PL que incremente 50€ la comisión a un trabajador introducido por teclado, teniendo en cuenta que si esta fuese nula se lanzará una Excepción que inserte en una tabla temporal un registro con el nº de empleado y la descripción “COMISIÓN NULA” y si el empleado no existe, el registro se insertará con el nº de empleado y la descripción “EMPLEADO INEXISTENTE” **(Excepciones)**

DECLARE

V\_EMPNO NUMBER(4) := '&CADENA';

V\_COMM EMP.COMM%TYPE;

BEGIN

SELECT COMM INTO V\_COMM FROM EMP WHERE EMPNO = V\_EMPNO;

IF V\_COMM IS NOT NULL THEN

UPDATE EMP SET COMM=COMM+50 WHERE EMPNO=V\_EMPNO;

ELSE

RAISE COMM\_NULA;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

INSERT INTO TEMPO VALUES(V\_EMPNO, 'EMPLEADO NO EXISTE');

WHEN COMM\_NULA THEN

INSERT INTO TEMPO VALUES(V\_EMPNO, 'COMISIÓN NULA');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('ERROR NO IDENTIFICADO!');

DBMS\_OUTPUT.PUT\_LINE(SQLCODE);

DBMS\_OUTPUT.PUT\_LINE(SQLERRM);

END;

/