**CREAR UNA EXCEPCION PROPIA**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*set serveroutput on;

Declare

totalEstudiantes int;

begin

select count(\*) into totalEstudiantes from estudiantes;

if totalEstudiantes = 0 then

raise\_application\_error(-20000,'No hay estudiantes');--lanza y captura

else

dbms\_output.put\_line('Hay : ' || totalEstudiantes);

end if;

end;

/

**CREAR EXCEPCION DECLARANDOLA**

create or replace function nombreF(param tipo) return tipo

is

totalSal int;

noDatos exception;

begin

select sum(sal)into totalSal from emp where to\_char(hiredate, 'MM') = mes;

if totalSal > 0 then

return totalSal;

else

raise noDatos;

end if;

exception

when noDatos then

return -1;

end; /

**CURSOR MUCHOS**

declare

vNombre TABLA.nombre%type := '&nombre';

datosEstudiante estudiantes%rowtype;

cursor cEstudiante is

select \* from estudiantes where upper(nombre) = upper(vNombre);

total int := 0;

begin

for datosEstudiante in cEstudiante loop dbms\_output.put\_line(upper(datosEstudiante.apellidos) || ', ' ||upper(datosEstudiante.nombre));

end loop;

if total = 0 then

dbms\_output.put\_line('No hay datos');

end if;

end; /

**CURSOR IMPLICITO**

declare

vDni estudiantes.dni%type;

vApellidos estudiantes.apellidos%type;

vNombre estudiantes.nombre%type :='&dime\_nombre';

begin

select dni into vDni from estudiantes where nombre = vNombre;

select apellidos into vApellidos from estudiantes where nombre = vNombre;

dbms\_output.put\_line('DNI : ' ||vDni|| ' Apellidos :' ||vApellidos);

exception

when no\_data\_found then

dbms\_output.put\_line('No existe');

when too\_many\_rows then

dbms\_output.put\_line('Nombre repetido');

end;

/

**TABLA COUNT FIRST LAST DELETE**

declare

type tableN is table of int index by binary\_integer;

valTablaNum tableN;

filaCompleta empt%rowtype;

begin

for i in 1..10 loop

valTablaNum (i) :=i;

dbms\_output.put\_line(valTablaNum (i));

end loop;

dbms\_output.put\_line('Numero de elementos :' || valTablaNum.count);

for j in valTablaNum.first..valTablaNum.last loop

dbms\_output.put\_line(valTablaNum(j));

end loop;

valTablaNum.delete(valTablaNum.last);

dbms\_output.put\_line('Numero de elementos :' || valTablaNum.count);

if not (valTablaNum.exists(10)) then

dbms\_output.put\_line('No existe');

else

dbms\_output.put\_line('Si existe');

end if;

end; **/**

**REGISTRO**

declare

type Tpersona is record (

codigo number(10),

nombre varchar2(100),

edad integer

);

type tMascota is record(

nombre varchar(100),

propietario tPersona

);

persona1 tPersona;

mascota1 tMascota;

begin

persona1.codigo := 1;

persona1.nombre := 'Pedro';

persona1.edad := 25;

mascota1.propietario := persona1;

mascota1.nombre := 'Pepa';

dbms\_output.put\_line(persona1.codigo || '/' || persona1.nombre || '/' || persona1.edad );

dbms\_output.put\_line(mascota1.nombre || '/' || mascota1.propietario.codigo || '/' || mascota1.propietario.nombre );

end;

/

**REVERSE**

declare

palabra varchar(10) := '&dime\_palabra';

palabraInvertida varchar(10);

begin

for i in reverse 1..length(palabra) loop

palabraInvertida := palabraInvertida || substr(palabra,i,1);

dbms\_output.put\_line(palabraInvertida);

end loop;

end;

/

DOBLE BUCLE

declare

begin

for indice in 1..10 loop

for j in 1..10 loop

dbms\_output.put\_line(indice || 'x' || j || '=' || indice\*j);

**end loop;**

end loop;

end; /

**CASE**

declare

nota int := &dime\_la\_nota;

begin

case

when nota > 10 or nota < 0 then

dbms\_output.put\_line('La nota no es correcta');

when nota > 8 then

dbms\_output.put\_line('sobresaliente');

when nota between 8 and 6 then

dbms\_output.put\_line('notable');

when nota = 5 then

dbms\_output.put\_line('aprobado');

else

dbms\_output.put\_line('suspenso');

end case;

end;

**/**

**FUNCION**

create or replace function nombFunc (var tabla.campo%type) return varchar

Is

Var tabla.campo%type;

Begin

select campo into var from tabla where campo= campo;

if campo = campo then

update emp set emp.ename = nombre where empno = codigo;

return 'Actualizado';

end if;

exception

when no\_data\_found then

return 'No se actualizó';

end if;

end; /

**PROBAR FUNCION**

declare

num1 int :=7;

num2 int :=3;

begin dbms\_output.put\_line(restaNumerosEnteros(num1,num2)); --llamar a la funcion

end;

/

**PROCEDIMIENTO**

create or replace procedure consultarEmpleado(vVAR tabla.campo%type, vVar out tabla.campo%type,

vJob out emp.job%type)

is

begin

select campo1,campo2 into vVar,vVar from tabla where campo= vVar;

exception

when no\_data\_found then

dbms\_output.put\_line('No se encontraron datos');

end;

/

**PROBAR**

declare

vEmpno emp.empno%type;

vEname emp.ename%type;

vJob emp.job%type;

begin

vEmpno := 7839;

consultarEmpleado(vEmpno, vEname, vJob);

dbms\_output.put\_line(vEname || '|' || vJob);

end;

/

**CREAR PAQUETE**

create or replace package miPaquete  
 is  
 procedure is\_today;  
 procedure consultarEmpleado(v\_empno emp.empno%type,v\_ename out emp.ename%type,v\_job out emp.job%type);  
 end;  
 /  
 create or replace package body miPaquete  
 is  
 procedure is\_today  
 is  
 begin  
 dbms\_output.put\_line(sysdate);  
 end;  
 procedure consultarEmpleado(  
 v\_empno emp.empno%type,  
 v\_ename out emp.ename%type,  
 v\_job out emp.job%type)  
 is  
 noHayDatos exception; --opción3  
 numEmp emp.empno%type;  
 contador int := 0;  
 begin  
 select empno into numEmp from emp where empno = v\_empno; --opción 2  
 select ename,job into v\_ename,v\_job from emp where empno = v\_empno; --opción 2  
 select count(\*) into contador from emp where empno = v\_empno; --opción 3  
 if contador = 0 then --opción 3  
 raise noHayDatos;  
 else  
 select ename,job into v\_ename,v\_job from emp where empno = v\_empno;  
 end if;  
 select ename,job into v\_ename,v\_job from emp where empno = v\_empno; --opción 1  
 exception  
 when no\_data\_found then --opción 1 / opción 2  
 dbms\_output.put\_line('No se encontraron datos');  
 when noHayDatos then --opción 3  
 dbms\_output.put\_line('No se encontraron datos');  
 end;  
 end;  
 /

**USAR PAQUETE**

declare  
 nombre emp.ename%type;  
 puesto emp.job%type;  
 begin

miPaquete.is\_today; miPaquete.consultarEmpleado(7839,nombre,puesto);  
 dbms\_output.put\_line(nombre||'|'||puesto);  
 end;  
 /