

```

CREATE OR REPLACE FUNCTION F_PROMEDIO_JUGADOR (pJug
jugadores.codigo%type, pTemp estadisticas.temporada%type)
return number
is
    V_PUNTOS_TOTAL_EQ_JUGADOR NUMBER;
    V_PARTIDOS_JUGADOS NUMBER;
    V_JUG_EQ NUMBER;
    V_MEDIA_JUG NUMBER;

begin

    -- CALCULAMOS LOS PUNTOS TOTALES DEL EQUIPO DEL JUGADOR POR
    TEMPORADA Y
    -- CALCULAMOS LOS NUMEROS DE PARTIDOS JUGADOS
    SELECT SUM(PUNTOS),SUM(NUMPARTIDOS) INTO
    V_PUNTOS_TOTAL_EQ_JUGADOR,V_PARTIDOS_JUGADOS
    FROM (
        SELECT SUM(PA.PUNTOS_VISITANTE) AS PUNTOS,NVL(COUNT(PA.CODIGO),0)
    AS NUMPARTIDOS
        FROM PARTIDOS PA, EQUIPOS E, JUGADORES J
        WHERE E.NOMBRE = PA.EQUIPO_VISITANTE
        AND E.NOMBRE = J.NOMBRE_EQUIPO
        AND J.CODIGO = pJug
        AND PA.TEMPORADA = pTemp
        UNION ALL
        SELECT SUM(PA.PUNTOS_LOCAL),NVL(COUNT(PA.CODIGO),0)
        FROM PARTIDOS PA, EQUIPOS E, JUGADORES J
        WHERE E.NOMBRE = PA.EQUIPO_LOCAL
        AND E.NOMBRE = J.NOMBRE_EQUIPO
        AND J.CODIGO = pJug
        AND PA.TEMPORADA = pTemp
    );

    -- SI NO HAY PUNTOS, PUES DIRECTAMENTE PONGO UN CERO
    IF (V_PUNTOS_TOTAL_EQ_JUGADOR = 0) THEN
        return 0;
    ELSE
        -- NECESITO SABER CUANTOS JUGADORES TIENE EL EQUIPO
        SELECT COUNT(CODIGO) INTO V_JUG_EQ
        FROM JUGADORES
        WHERE NOMBRE_EQUIPO = (SELECT NOMBRE_EQUIPO FROM JUGADORES
WHERE CODIGO = pJug);

        -- CALCULAMOS LA MEDIA DE PUNTOS POR JUGADORES
        V_MEDIA_JUG :=
        ((V_PUNTOS_TOTAL_EQ_JUGADOR/V_PARTIDOS_JUGADOS)/V_JUG_EQ);

        return V_MEDIA_JUG;

```

END IF;

end;

CREATE OR REPLACE FUNCTION ObtenerMejorCliente(pNomPro
PRODUCTOS.NOMBRE%TYPE)
RETURN VARCHAR2

IS

V_NOMCLI CLIENTES.NOMBRECLIENTE%TYPE;

BEGIN

SELECT C.NOMBRECLIENTE INTO V_NOMCLI
FROM CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP, PRODUCTOS PR
WHERE C.CODIGOCLIENTE = P.CODIGOCLIENTE
AND P.CODIGOPEDIDO = DP.CODIGOPEDIDO
AND DP.CODIGOPRODUCTO = PR.CODIGOPRODUCTO
AND UPPER(PR.NOMBRE) = UPPER(pNomPro)
GROUP BY NOMBRECLIENTE
HAVING SUM(DP.CANTIDADDP.PRECIOUNIDAD)=(SELECT
MAX(SUM(DP.CANTIDADDP.PRECIOUNIDAD))
FROM CLIENTES C, PEDIDOS P, DETALLEPEDIDOS DP,
PRODUCTOS PR
WHERE C.CODIGOCLIENTE = P.CODIGOCLIENTE
AND P.CODIGOPEDIDO = DP.CODIGOPEDIDO
AND DP.CODIGOPRODUCTO = PR.CODIGOPRODUCTO
AND UPPER(PR.NOMBRE) = UPPER(pNomPro)
GROUP BY NOMBRECLIENTE);

RETURN V_NOMCLI;

EXCEPTION

WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('Ocurrió el error ' ||SQLCODE ||' ---mensaje: ' || SQLERRM);
RETURN SQLERRM;

END;

create or replace FUNCTION quitarAcento(p_palabra VARCHAR2)

return VARCHAR2

is

begin

```

        return (translate(UPPER(p_palabra),'ÁÉÍÓÚ','AEIOU'));

end;

CREATE OR REPLACE FUNCTION fNumDept
(p_numdept DEPT.DEPTNO%TYPE, P_fecha DATE)
RETURN INTEGER
IS
v_cantidad_empleados integer;

BEGIN
    SELECT COUNT(EMPNO) into v_cantidad_empleados
    FROM EMP
    WHERE TO_DATE(HIREDATE, 'DD/MM/YY')= p_fecha
    AND DEPTNO= P_numdept;

return v_cantidad_empleados;
EXCEPTION
WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Ocurrió el error ' ||SQLCODE ||' ---mensaje: ' || SQLERRM);
    RETURN -1;
END;
/
begin
    dbms_output.put_line(
FnUMDEPT(10, TO_DATE('17/11/81', 'DD/MM/YY')));
end;
/
SELECT COUNT(EMPNO)
FROM EMP
WHERE TO_DATE(HIREDATE, 'DD/MM/YY')= TO_DATE('17/11/81', 'DD/MM/YY')
AND DEPTNO=10;

/*
13. Realizar una función que me devuelva la suma de pagos que
ha realizado. Pasa el codigo por parametro. Controla en caso
de que no se encuentre, en ese caso devuelve un -1.
*/

CREATE OR REPLACE FUNCTION F_SUMA_PAGOS
(P_COD_CLI IN PAGOS.CODIGOCLIENTE%TYPE)
RETURN NUMBER
IS

V_SUMA_PAGOS PAGOS.CODIGOCLIENTE%TYPE;

```

```

BEGIN
select sum(p.cantidad) INTO V_SUMA_PAGOS
from clientes c, pagos p
where p.codigocliente= c.codigocliente
and c.codigocliente= P_COD_CLI;

if v_suma_pagos is null then
    raise no_data_found;
else
    RETURN V_SUMA_PAGOS;
end if;

EXCEPTION

WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('El codigo ' ||P_COD_CLI ||' no es código de ningún pagos ');
    RETURN -1;

WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Ocurrió el error ' ||SQLCODE ||' ---mensaje: ' || SQLERRM);
    RETURN -1;
END;

```

```

select F_SUMA_PAGOS(1000) from dual;

```

--FUNCION QUE PASEMOS UNA FECHA, NUMERO DE AÑOS Y SUMAR ESA CANTIDAD DE AÑOS A LA FECHA

```

CREATE OR REPLACE FUNCTION fSumaAnhos( p_fecha date, p_numero integer)
return date
is
    v_anho integer;
    v_anho_nuevo integer;
    v_fecha_nueva date;
begin

    return (add_months(p_fecha, 12*p_numero));
/*
    v_anho := to_number(to_char(p_fecha,'yyyy'));
    v_anho_nuevo := v_anho + p_numero;

    v_fecha_nueva :=
to_date(to_char(p_fecha,'dd/mm')||'/'||to_char(v_anho_nuevo),'dd/mm/yyyy');

    return v_fecha_nueva;
*/

```

```
end;
```

```
SELECT fSumaAnhos(to_date('01/06/2009','dd/mm/yyyy'),2)
FROM DUAL;
```

```
/*
```

```
CREA UNA FUNCION QUE AL COLOCAR UN NUMERO DE DEPARTAMENTO  
DEVUELVA EL TOTAL DE SALARIO DEL DEPARTAMENTO
```

```
*/
```

```
CREATE OR REPLACE FUNCTION F_SALARIO_DEPT
(P_CODDEPT IN DEPT.DEPTNO%TYPE)
RETURN NUMBER
IS
```

```
    V_SAL_DEPT NUMBER; --GUARDA EL RESULTADO DE LA SUMA DEL CODIGO
DEPT
    V_CODIGO_DEPT DEPT.DEPTNO%TYPE; --GUARDA LA FILA DE LA CONSULTA
BEGIN
```

```
    SELECT DEPTNO INTO V_CODIGO_DEPT
    FROM DEPT
    WHERE DEPTNO = P_CODDEPT;
```

```
    SELECT NVL(SUM(SAL+NVL(COMM,0)),0) INTO V_SAL_DEPT
    FROM EMP
    WHERE DEPTNO = P_CODDEPT;
```

```
    RETURN V_SAL_DEPT;
```

```
EXCEPTION
```

```
WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('El codigo ' || p_coddept || ' no es código de ningún
departamento ');
    RETURN -1;
```

```
WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Ocurrió el error ' || SQLCODE || ' ---mensaje: ' || SQLERRM);
    RETURN -1;
END;
```

```
SELECT F_SALARIO_DEPT(10) FROM DUAL;
```

```
-- Crear una funcion F_QUIEN_ES_MI_JEFE que reciba un nombre de empleado y nos
devuelva el nombre del jefe directo
```

```

-- de ese empleado. Controlar todas las excepciones posibles.
CREATE OR REPLACE FUNCTION F_QUIEN_ES_MI_JEFE(P_NOM_EMP IN
EMP.ENAME%TYPE)
RETURN VARCHAR2
IS
    V_NOM_JEFE EMP.ENAME%TYPE;

BEGIN

    SELECT J.ENAME INTO V_NOM_JEFE
    FROM EMP E, EMP J
    WHERE E.MGR = J.EMPNO
    AND UPPER(E.ENAME) = UPPER(P_NOM_EMP);

    RETURN V_NOM_JEFE;

EXCEPTION

WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Ocurrió el error ' ||SQLCODE ||' ---mensaje: ' || SQLERRM);
    RETURN SQLERRM;

END;

SELECT J.ENAME
FROM EMP E, EMP J
WHERE E.MGR = J.EMPNO
AND UPPER(E.ENAME) = UPPER('KING');

BEGIN
    DBMS_OUTPUT.PUT_LINE(F_QUIEN_ES_MI_JEFE('king'));
END;

-- crear función que dado un nombre de empleado, me devuelva el nombre de su jefe.
-- controlar todas las excepciones necesarias
create or replace function f_nombre_jefe(p_nom_emp emp.ename%type)
return varchar2
is
    v_nom_jefe emp.ename%type;
begin

    select NVL(j.ename,'NO TENGO JEFE') into v_nom_jefe
    from emp e, emp j
    where e.mgr = j.empno(+)

```

```

AND upper(e.ename) = upper(p_nom_emp);

return v_nom_jefe;

EXCEPTION
WHEN NO_DATA_FOUND THEN
    return 'NO SOY UN EMPLEADO';

WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Ocurrió el error ' || SQLCODE || ' ---mensaje: ' || SQLERRM);
    return 'ERROR';
END;

--
BEGIN
    DBMS_OUTPUT.PUT_LINE(f_nombre_jefe('CLARK'));
    DBMS_OUTPUT.PUT_LINE(f_nombre_jefe('KING'));
    DBMS_OUTPUT.PUT_LINE(f_nombre_jefe('NELSON'));
END;

--Crea una función que reciba como parámetros el número de un departamento y una fecha,
-- y que devuelva el número de empleados que tenía ese departamento en esa fecha.
-- si no tiene que devuelva cero.

set serveroutput on;
--Mostrar un listado de todos los clientes (nombre y ciudad) que no hayan hecho pagos
DECLARE

    CURSOR C_CLI_CIU IS
        SELECT NOMBRECLIENTE, CIUDAD
        FROM CLIENTES
        WHERE CODIGOCLIENTE NOT IN (SELECT DISTINCT CODIGOCLIENTE FROM
PAGOS);

    V_NOM CLIENTES.NOMBRECLIENTE%TYPE;
    V_CIU CLIENTES.CIUDAD%TYPE;

BEGIN

    OPEN C_CLI_CIU;
    LOOP
        FETCH C_CLI_CIU INTO V_NOM, V_CIU;
        EXIT WHEN C_CLI_CIU%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('EL CLIENTE '||V_NOM||' DE LA CIUDAD '||V_CIU||' NO
HA HECHO PAGOS');

    END LOOP;

```

```

CLOSE C_CLI_CIU;

EXCEPTION
WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Ocurrió el error ' || SQLCODE || ' ---mensaje: ' || SQLERRM);

END;

```

```

SET SERVEROUTPUT ON;

```

```

/*

```

14. Realizar una función que devuelva el total en euros de un pedido, pasale el codigo por parametro. Controla en caso de que no se encuentre, en ese caso devuelve un 0. Pasale otro parametro, si supera ese limite, lanzaremos una excepcion propia y devolveremos un 0.

```

*/

```

```

set SERVEROUTPUT ON;

```

```

create or replace function total_pedido_func

```

```

    (v_codigopedido pedidos.codigopedido%type, v_limite number)

```

```

return number

```

```

as

```

```

    v_total number(8) := 0;

```

```

    limite_superado exception;

```

```

begin

```

```

    select sum(dp.cantidad * dp.PRECIOUNIDAD) into v_total

```

```

    from pedidos p, detallepedidos dp

```

```

    where p.codigopedido = dp.codigopedido and p.codigopedido = v_codigopedido;

```

```

    if v_total is null then

```

```

        raise no_data_found;

```

```

    else

```

```

        if v_limite < v_total then

```

```

            raise limite_superado;

```

```

        else

```

```

            return v_total;

```

```

        end if;

```

```

    end if;

```

```

exception

```

```

    when no_data_found then

```

```

        return -1;

```

```

    when limite_superado then

```

```

        DBMS_OUTPUT.PUT_LINE('Limite superado');

```



```
        return 0;

end;
/

declare
    v_codigopedido pedidos.codigopedido%type := &codigo;
    v_total number(8);
    v_limite number(8) := &limite;
begin
    v_total := total_pedido_func(v_codigopedido , v_limite);

    if v_total = -1 then
        DBMS_OUTPUT.PUT_LINE('no existe el pedido');
    else
        DBMS_OUTPUT.PUT_LINE('El pedido total es ' || v_total);
    end if;

end;
```