

CREATE OR REPLACE PACKAGE BODY pkPedidosCompras IS

PROCEDURE spGenerarPedidosCompras(pFECINI DATE, pFECFIN DATE, pExtra
integer)

IS

CURSOR C_PED IS

SELECT P.FECHAPEDIDO, ADD_MONTHS(P.FECHAPEDIDO,1),
DP.CANTIDAD, DP.CODIGOPRODUCTO
FROM PEDIDOS P, DETALLEPEDIDOS DP
WHERE P.CODIGOPEDIDO =DP.CODIGOPEDIDO
AND P.FECHAPEDIDO >= pFECINI
AND P.FECHAPEDIDO <= pFECFIN;

V_FEC PEDIDOS.FECHAPEDIDO%TYPE;

V_FECMES PEDIDOS.FECHAPEDIDO%TYPE;

V_CANT DETALLEPEDIDOS.CANTIDAD%TYPE;

V_CODPRO PRODUCTOS.CODIGOPRODUCTO%TYPE;

V_NUMPED pedidosmaterialcompras.NUMPEDIDO%TYPE;

V_MAX NUMBER;

V_CUENTA NUMBER;

v_existe number;

BEGIN

OPEN C_PED;

LOOP

FETCH C_PED INTO V_FEC, V_FECMES,V_CANT,V_CODPRO;

EXIT WHEN C_PED%NOTFOUND;

-- ¿QUE DEBE DE PASAR AQUI?----

-- 2 COSAS

-- CALCULAR EL NUMPED

SELECT NVL(MAX(NUMPEDIDO),0), NVL(COUNT(NUMPEDIDO),0) INTO V_MAX,
V_CUENTA

FROM pedidosmaterialcompras;

IF V_MAX = V_CUENTA THEN

V_NUMPED := V_MAX +1;

ELSE

FOR K in 1..v_max

loop

select nvl(count(*),0) into v_existe

from pedidosmaterialcompras

where numpedido = k;

if v_existe =0 then

v_numped := k;

exit;

```

        end if;
    end loop;
END IF;
-- INSERTAR.
INSERT INTO pedidosmaterialcompras
    (numpedido,fechainicio,fechafin,codigoproducto,cantidadesolicitada)
VALUES
    (V_NUMPED, V_FEC,V_FECMES,V_CODPRO,V_CANT
    );
END LOOP;
CLOSE C_PED;

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

/*7.

Crear un procedimiento spPagosPedido con parámetros de entrada pFECINI y pFECFIN, tal que por cada pedido situado entre esas dos fechas, se genere un pago con la cantidad de ese pedido que cuesta ese pedido, al cliente que la ha realizado y al mes siguiente de haber realizado el pedido. Controlar todos los errores posibles que se pueden obtener. Crear el procedimiento dentro de un paquete llamado pkPedidos. (3 puntos)*/

PASOS

CONSULTA SQL:

```

select p.codigocliente,p.fechapedido,
sum(dp.preciounidad * dp.cantidad) as pedido_total_gasto
from pedidos p,
detallepedidos dp
where p.codigopedido= dp.codigopedido
and fechapedido >= to_date('17/01/2006', 'dd/mm/yyyy')
and fechapedido <= to_date('20/06/2008', 'dd/mm/yyyy')
group by p.codigocliente,p.fechapedido
order by p.fechapedido;
```

GENERAR PROCEDIMIENTO

```

CREATE OR REPLACE PROCEDURE spPagosPedido (
    pFECINI pedidos.fechapedido%type,
    pFECFIN pedidos.fechapedido%type
) IS
    CURSOR c_PagosPedido is
```

```

select p.codigocliente,p.fechapedido,
sum(dp.preciounidad * dp.cantidad)
from pedidos p, detallepedidos dp
where p.codigopedido= dp.codigopedido
and fechapedido >= pFECINI
and fechapedido <= pFECFIN
group by p.codigocliente,p.fechapedido
order by p.fechapedido;

v_codigocliente pedidos.codigocliente%type;
v_fechapedido PEDIDOS.fechapedido%type;
v_pedido_total_gasto integer;
BEGIN
OPEN c_PagosPedido;
LOOP
FETCH c_PagosPedido INTO v_codigocliente,
v_fechapedido, v_pedido_total_gasto;
EXIT WHEN c_PagosPedido%NOTFOUND;

INSERT INTO pagos (
CODIGOCLIENTE,
formapago,
idtransaccion,
fechapago,
cantidad
) VALUES (
v_codigocliente,
'sin informacion',
'sin informacion',
ADD_MONTHS(v_fechapedido,1),
v_pedido_total_gasto
);

END LOOP;
CLOSE c_PagosPedido;

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

CREAR EL PAQUETE GUIA
CREATE OR REPLACE PACKAGE pkPedidos
IS
PROCEDURE spPagosPedido (pFECINI pedidos.fechapedido%type,pFECFIN
pedidos.fechapedido%type);

END;

```

CREAR Y COMPILAR EL CUERPO DEL PAQUETE

CREATE OR REPLACE PACKAGE BODY pkPedidos

IS

PROCEDURE spPagosPedido (

pFECINI pedidos.fechapedido%type,

pFECFIN pedidos.fechapedido%type

) IS

CURSOR c_PagosPedido is

select p.codigocliente,p.fechapedido,

sum(dp.preciounidad * dp.cantidad)

from pedidos p, detallepedidos dp

where p.codigopedido= dp.codigopedido

and fechapedido >= pFECINI

and fechapedido <= pFECFIN

group by p.codigocliente,p.fechapedido

order by p.fechapedido;

v_codigocliente pedidos.codigocliente%type;

v_fechapedido PEDIDOS.fechapedido%type;

v_pedido_total_gasto integer;

BEGIN

OPEN c_PagosPedido;

LOOP

FETCH c_PagosPedido INTO v_codigocliente,

v_fechapedido, v_pedido_total_gasto;

EXIT WHEN c_PagosPedido%NOTFOUND;

INSERT INTO pagos (

CODIGOCLIENTE,

formapago,

idtransaccion,

fechapago,

cantidad

) VALUES (

v_codigocliente,

'sin informacion',

'sin informacion',

ADD_MONTHS(v_fechapedido,1),

v_pedido_total_gasto

);

END LOOP;

CLOSE c_PagosPedido;

EXCEPTION

WHEN OTHERS THEN

```

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

```

```

/*-----
DENTRO DEL PAQUETE UTILIDADES CREAR UNA PROCEDIMIENTO QUE RECIBA
UN NUMERO pNum
Y MUESTRE POR PANTALLAUNA FIGURA ASI, pNum es el número de líneas hacia
abajo
-----

```

```

*
* *
* * *
* * * *
* * * * *
*/

```

```

CREATE OR REPLACE PACKAGE UTILIDADES
IS
    PROCEDURE SP_PINTAR_LINEAS(P_NUM INTEGER);
END;

```

```

CREATE OR REPLACE PACKAGE BODY UTILIDADES
IS
    PROCEDURE SP_PINTAR_LINEAS(P_NUM IN INTEGER)

    IS
        V_RELLENO VARCHAR2(20):= "";
    BEGIN
        FOR K IN 1..P_NUM LOOP

            V_RELLENO:= V_RELLENO || '*';
            DBMS_OUTPUT.PUT_LINE(V_RELLENO);
        END LOOP;
    END;
END;

```

```

BEGIN
    UTILIDADES.SP_PINTAR_LINEAS(5);
END;

/* PAQUETE CON VARIOS PROCEDIMIENTOS*/
CREATE OR REPLACE PACKAGE PK_EMPLEADOS
IS
    PROCEDURE spInsertNuevoEmpIVAN(p_apellido EMP.ENAME%TYPE);
    PROCEDURE spInsertNuevoEmp(p_apellido EMP.ENAME%TYPE);
    PROCEDURE spMedia;
END;

CREATE OR REPLACE PACKAGE BODY PK_EMPLEADOS
IS
    PROCEDURE spInsertNuevoEmpIVAN(p_apellido EMP.ENAME%TYPE)
    IS
        v_empno emp.empno%type:= 0;
        v_sal NUMBER:= 0;
        v_deptno emp.deptno%type :=0;
    BEGIN
        select max(empno)+1 into v_empno
        from emp;

        select deptno INTO v_deptno from dept where upper(loc) = 'SEVILLA';

        select avg(sal) INTO v_sal from emp;

        -- 2 formas distintas que espero que funcionen
        INSERT INTO EMP (EMPNO, ENAME ,JOB , hiredate ,sal, comm, DEPTNO)
        VALUES (v_empno, p_apellido, 'Novato', sysdate, v_sal, NULL, v_deptno);

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

    PROCEDURE spInsertNuevoEmp(p_apellido EMP.ENAME%TYPE)
    IS
    BEGIN
        INSERT INTO EMP (EMPNO, ENAME ,JOB , hiredate ,sal, comm, DEPTNO)
        VALUES ((select max(empno)+1 from emp), p_apellido, 'Novato', sysdate, (select
        avg(sal) from emp), NULL, (select deptno from dept where upper(loc) = 'SEVILLA'));

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

```

```

PROCEDURE spMedia
IS
    CURSOR C_EMP IS
        SELECT EMPNO, SAL, JOB
        FROM EMP;

    V_EMPNO EMP.EMPNO%TYPE;
    V_SAL EMP.SAL%TYPE;
    V_JOB EMP.JOB%TYPE;
    V_MEDIA_JOB NUMBER;
BEGIN
    OPEN C_EMP;
    LOOP
        FETCH C_EMP INTO V_EMPNO, V_SAL, V_JOB;
        EXIT WHEN C_EMP%NOTFOUND;

        SELECT AVG(SAL) INTO V_MEDIA_JOB
        FROM EMP
        WHERE UPPER(JOB) = UPPER(V_JOB);

        IF V_SAL <= V_MEDIA_JOB THEN
            UPDATE EMP
            SET SAL = SAL * 1.5
            WHERE EMPNO = V_EMPNO;
        END IF;
    END LOOP;
    CLOSE C_EMP;
END;
END;

EXEC PK_EMPLEADOS.spInsertNuevoEmp('ELVIS');
select *
from emp
order by ENAME desc;

```