**UNIVERSIDAD CATÓLICA DEL NORTE**



**TALLER N° 4**

**Informe de trabajo**

Presentado en Cumplimiento de la asignatura de: **Bases de datos**

**Integrado por:**

José Quinteros

Ignacio Rivera

**Profesor(a) de asignatura:**

Luis Lobos

**Antofagasta, Domingo 07 de Diciembre del 201**

Ejercicio 1

-- EJ 1

create or replace function digitoVerificador(rut in varchar2)

return varchar2

is

resultado number:=0;

contador number(3):=2;

numero varchar2(255);

begin

for pos in reverse 1..length(rut) loop

numero:=substr(rut,pos,1);

if contador>7 then

contador:=2;

end if;

resultado:=resultado + (to\_number(numero)\*contador);

contador:=contador+1;

end loop;

resultado:=11-mod(resultado,11);

if resultado=11 then

numero:='k';

else if resultado =10 then

numero:='0';

else

numero:=to\_char(resultado);

end if;

end if;

return rut||'-'||numero;

end;

set serveroutput on;

declare

rut varchar2(255);

begin

select digitoVerificador(&rutt)into rut from table1;

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

dbms\_output.put\_line('Rut: '||rut);

end;

Ejercicio 2

Declare

cursor c is select \* from MOVIMIENTO;

cant NUMBER;

Procedure actualiza (u\_numero in CUENTA.NUMERO%TYPE, u\_nombre in CUENTA.NOMBRE%TYPE , u\_direccion in CUENTA.DIRECCION%TYPE ,

u\_ciudad in CUENTA.CIUDAD%TYPE, u\_fono in CUENTA.FONO%TYPE , u\_saldo in CUENTA.SALDO%TYPE)

IS

od\_numero CUENTA.NUMERO%TYPE;

od\_nombre CUENTA.NOMBRE%TYPE;

od\_direccion CUENTA.DIRECCION%TYPE;

od\_ciudad CUENTA.CIUDAD%TYPE;

od\_fono CUENTA.FONO%TYPE;

od\_saldo CUENTA.SALDO%TYPE;

begin

select NUMERO , NOMBRE , DIRECCION , CIUDAD , FONO , SALDO

into od\_numero , od\_nombre , od\_direccion , od\_ciudad , od\_fono , od\_saldo

from CUENTA

where NUMERO = u\_numero;

update CUENTA

set

NOMBRE = nvl(u\_nombre,od\_nombre),

DIRECCION = nvl(u\_direccion,od\_direccion),

CIUDAD = nvl(u\_ciudad,od\_ciudad),

FONO = nvl(u\_fono,od\_fono),

SALDO = nvl(u\_saldo,od\_saldo)

where NUMERO = u\_numero;

end actualiza;

Procedure inserta (i\_numero in CUENTA.NUMERO%TYPE, i\_nombre in CUENTA.NOMBRE%TYPE , i\_direccion in CUENTA.DIRECCION%TYPE ,

i\_ciudad in CUENTA.CIUDAD%TYPE, i\_fono in CUENTA.FONO%TYPE , i\_saldo in CUENTA.SALDO%TYPE)

IS

begin

insert into CUENTA

values (i\_numero , i\_nombre , i\_direccion , i\_ciudad , i\_fono , i\_saldo );

end inserta;

Procedure borra(d\_numero in CUENTA.NUMERO%TYPE)

IS

begin

delete from CUENTA

where NUMERO = d\_numero ;

end borra;

begin

for r in c loop

select count(\*)

into cant

from CUENTA

where NUMERO = r.NUMERO;

if cant > 0 and (r.OPERACION = 'I' or r.OPERACION = 'U') then

actualiza(r.NUMERO,r.NOMBRE,r.DIRECCION, r.CIUDAD , r.FONO , r.SALDO);

elsif r.OPERACION = 'I' or r.OPERACION = 'U' then

inserta(r.NUMERO,r.NOMBRE,r.DIRECCION, r.CIUDAD , r.FONO , r.SALDO);

elsif cant > 0 then

borra(r.NUMERO);

end if;

end loop;

end;

Ejercicio 3

*Nota: por motivos de conflicto en los indicadores se optó por cambiar el nombre de la tabla en lugar de eliminar la tabla del ejercicio dos al realizar la prueba del código.*

create or replace trigger valCompra

before insert on COMPRA

for each row

Declare

suma NUMBER;

begin

select sum(MONTOCOMPRA)

into suma

from COMPRA

where NUMCUENTA = :new.NUMCUENTA and RUTCLIENTE = :new.RUTCLIENTE and trunc(FECHACOMPRA) = trunc(:new.FECHACOMPRA);

if (suma + :new.MONTOCOMPRA > 200000) then

raise\_application\_error(-20000, 'monto diario exedido');

end if;

end;

**Distribución del Trabajo**

|  |  |  |
| --- | --- | --- |
|  | Ignacio rivera | José Quinteros |
| codificación | 4 horas | 3 horas |
| Corrección de errores | 0,3 horas | 1 hora |
| documentación | - | - |
| informe | 0,2 horas | 0,1 horas |