

a) Create Sequence aud_registro

Min Value 1

Start with 1

Increment By 1

Cache 20; Create Sequence seq - abs. min value 1.1 start with 1 increment by 1 cache 20;

b) Create or Replace Trigger act_alumnos

Before Insert or Update on Alumnos

For Each Row

Begin

If inserting then

Insert into Auditoria Values (seq_aud_registro.nextval, 'A', 'I', user, sysdate);

and If;

If updating then

Insert into Auditoria Values (seq_aud_registro.nextval, 'A', 'U', user, sysdate);

and If;

If deleting then

Insert into Auditoria Values (seq_aud_registro.nextval, 'A', 'D', user, sysdate);

and If;

and Trigger act_alumnos;

d) Create or Replace procedure dm-deposito (

p-monto Alumnos.cho_saldo %TYPE;

p-aventa Alumnos.cho_aventa %TYPE;

p-num cliente Alumnos.cho_num_cliente %TYPE;

p-tipo cheque Alumnos.cho_tipo_cheque %TYPE;

mensaje out Varchar2(200);

Begin

Update Alumnos set cho_saldo = cho_saldo + p-monto

Where cho_num_cliente = p-num cliente;

And cho_tipo_cheque = p-tipo cheque;

And cho_aventa = p-aventa;

mensaje := 'La inserción del depósito se ha realizado de manera exitosa';

Exception

When No_Data_Found Then

mensaje := 'No se encontró información';

When Others Then

mensaje := 'Error in error';

End dm-deposito;

Declare

v-monto Alumnos.cho_saldo %TYPE;

v-num cliente Alumnos.cho_num_cliente %TYPE;

v-mensaje Varchar2(200);

Begin

v-monto := 50000;

v-num cliente := 1000000000;

Insert_deposito(v-monto, v-num cliente, mensaje);

dbms_output.put_line(mensaje);

End;

5) Create or Replace Procedure Petros_Aho(
P-aho-creata Aho-aho-creata TYPE;
P-creata Petros Aho-aho-creata saldo TYPE; AS
V-resultado out Varchar2 := 'Exit to register';
Begin
UPDATE Aho set aho-saldo = (aho-saldo - p-montoretino);
aho-retornad = (aho-retornad + p-montoretid);
fecha = (TO Date(SysDate, 'DD/MM/YYYY' 'H:M:SS'));
aho-creacion = USe where aho-nucleato = p-nucleato;
And = aho-tipo-aho = p-tipo-aho;
And = aho-creata = p-creata;

Exceptions
When NO Data Found then
V-resultado := 'Registro no encontrado';
When others then
V-resultado := 'Error encontrado';
End Petros_Aho;
-- Llamado a procedimiento Petros_Aho

Declaro
V-aho-creata Aho-aho-creata TYPE;
V-creata Petros Aho-aho-creata saldo TYPE;
V-resultado Varchar2(200);
Begin
V-aho-creata := 'P-creata-aho';
V-montoretino := 'P-montoretino';
Petros_Aho(V-aho-creata, V-creata, V-resultado);
Dbms_output.put_line(V-resultado);
End;

6) Create or Replace Function Inter_Aho Con(
P-aho-creata Aho-aho-creata TYPE;
Return number AS
V-interes Aho Con number;
V-aho-saldo Aho-aho-saldo;
Select aho-saldo into V-aho-saldo From Aho where aho-creata = P-aho-creata;
V-interes Aho Con := (V-aho-saldo * (0.04 * 30) / 365);
Return V-interes Aho Con;

Exceptions
When NO Data Found then
Dbms_output.put_line('Registro no encontrado');
When others then
Dbms_output.put_line('Error encontrado');

- Actualizar Saldo y ahoro mensual
Create or Replace Procedure UpdtAho Con () AS
Declaro
C-aho Con IS
Select aho-creata, aho-creata, aho-mensual From Aho
V-aho-creata = aho-creata;
V-interes = Inter_Aho Con(V-aho-creata);
For V-Aho Con in C-aho Con Loop
V-interes = Inter_Aho Con(V-Aho Con.aho-creata);
UPDATE Aho set aho-saldo = (V-Aho Con.aho-saldo + V-interes);
End Loop;
Commit;

Exceptions
When NO Data Found then
Dbms_output.put_line('Registro no encontrado');
When others then
Dbms_output.put_line('Error encontrado');
End UpdtAho Con;