



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA EN SISTEMAS COMPUTACIONALES
LIC. ING. EN SISTEMAS DE INFORMACIÓN

LABORATORIO 7

CURSO: SISTEMA BASE DE DATOS 2

PROFESOR:

HENRY LEZCANO

ESTUDIANTES:

MILAGROS CAMPOS 8-948-227

GUADALUPE CASTILLO 8-929-2252

ELIONAYS ROSAS 9-756-2182

ALEJANDRO URRIOLA 9-755-1141

GRUPO:

11F131

II SEMESTRE, 2020

Laboratorio 7

--CREACIÓN DE LA TABLA TIPO_AHORRO

```
Create table tipo_ahorro(  
cod_ahorro varchar2(5) primary key not null,  
descripcion varchar2(50),  
tasa_interes number  
);
```

```
SQL> Create table tipo_ahorro(  
2  cod_ahorro varchar2(5) primary key not null,  
3  descripcion varchar2(50),  
4  tasa_interes number  
5  );
```

Table created.

Create table Sucursal (

Cod_Sucursal varchar2(5) primary key not null,

Nom_Sucursal varchar2(25)

);

--MODIFICACIONES A LA TABLA SUCURSAL

ALTER TABLE Sucursal ADD (

cod_ahorro varchar2(5),

Constraint Fk_cod_ahorro foreign key (cod_ahorro) references tipo_ahorro (cod_ahorro),

monto_ahorrado number);

```
SQL> ALTER TABLE Sucursal ADD (  
2  cod_ahorro varchar2(5),  
3  Constraint Fk_cod_ahorro foreign key (cod_ahorro) references  
tipo_ahorro (cod_ahorro),  
4  monto_ahorrado number);
```

Table altered.

--CREACION DE LA TABLA AHORROS

```
Create table Ahorros(  
cod_ahorro varchar2(5) ,  
num_cta number, --secuencia---  
fecha_open date,  
interes number,  
letra number,  
sal_ahorro number,  
sal_interes number,  
fecha_dep date,  
fecha_ret date,  
usuario number,  
fecha_modif date,  
ID_Cliente number,  
Constraint Fk_otro_id_cliente foreign key (ID_Cliente) references Banco_Clientes (  
ID_Cliente),  
Cod_Sucursal varchar2(5),  
Constraint Fk_otro_cod_suc foreign key (Cod_Sucursal) references Sucursal  
(Cod_Sucursal),  
Constraint Fk_COD_ahorro2 foreign key (cod_ahorro ) references TIPO_AHORRO  
(cod_ahorro),  
Constraint pk_ahorro primary key(num_cta,cod_ahorro,id_cliente)  
);
```

```

SQL> Create table Ahorros(
  2  cod_ahorro varchar2(5) ,
  3  num_cta number, --secuencia---
  4  fecha_open date,
  5  interes number,
  6  letra number,
  7  sal_ahorro number,
  8  sal_interes number,
  9  fecha_dep date,
 10  fecha_ret date,
 11  usuario number,
 12  fecha_modif date,
 13  ID_Cliente number,
 14  Constraint Fk_otro_id_cliente foreign key (ID_Cliente) refere
nces Banco_Clientes (ID_Cliente),
 15  Cod_Sucursal varchar2(5),
 16  Constraint Fk_otro_cod_suc foreign key (Cod_Sucursal) referen
ces Sucursal (Cod_Sucursal),
 17  Constraint Fk_COD_ahorro2 foreign key (cod_ahorro ) reference
s TIPO_AHORRO (cod_ahorro),
 18  Constraint pk_ahorro primary key(num_cta,cod_ahorro,id_client
e)
 19  );

```

Table created.

```
--CREACION DE SECUENCIA PARA NUMERO DE CUENTA  
CREATE SEQUENCE seq_num_cta START WITH 1 INCREMENT BY 1;
```

```
SQL> CREATE SEQUENCE seq_num_cta START WITH 1 INCREMENT BY 1;  
  
Sequence created.
```

```
--Creación de la tabla Transac_dr  
Create table Transac_dr(  
ID_Transaccion number primary key,  
tipo_transac varchar2(10),  
fecha_transac date,  
monto_dr number,  
fecha_inser date,  
usuario number,  
num_cta number,  
Cod_Sucursal varchar2(5),  
ID_Cliente number,  
cod_ahorro varchar2(5),  
CONSTRAINTS FKvarios foreign key (num_cta,cod_ahorro,id_cliente)  
REFERENCES ahorros ( num_cta,cod_ahorro,id_cliente)  
);
```

```
SQL> Create table Transac_dr(  
2 ID_Transaccion number primary key,  
3 tipo_transac varchar2(10),  
4 fecha_transac date,  
5 monto_dr number,  
6 fecha_inser date,  
7 usuario number,  
8 num_cta number,  
9 Cod_Sucursal varchar2(5),  
10 ID_Cliente number,  
11 cod_ahorro varchar2(5),  
12 CONSTRAINTS FKvarios foreign key (num_cta,cod_ahorro,id_cliente)  
13 REFERENCES ahorros ( num_cta,cod_ahorro,id_cliente)  
);
```

```
--CREACION DE SECUENCIA PARA ID_Transaccion  
CREATE SEQUENCE seq_id_trans START WITH 1 INCREMENT BY 1;
```

```
SQL> CREATE SEQUENCE seq_id_trans START WITH 1 INCREMENT BY 1;  
  
Sequence created.
```

--CREACIÓN DE LA TABLA AUDITORÍA

```
Create table Auditoria(  
ID_Transaccion number PRIMARY KEY, --secuencia  
tipo_transac varchar2(10),  
saldo_cta_anterior number,  
monto_transac number,  
saldo_final number,  
num_cta number,  
usuario number,  
fecha date,  
reg_auditoria number,  
ID_Cliente number,  
cod_ahorro varchar2(5),  
CONSTRAINTS FKvarios1 foreign key (num_cta,cod_ahorro,id_cliente)  
REFERENCES ahorros ( num_cta,cod_ahorro,id_cliente)  
);
```

```
SQL> Create table Auditoria(  
2 ID_Transaccion number PRIMARY KEY, --secuencia  
3 tipo_transac varchar2(10),  
4 saldo_cta_anterior number,  
5 monto_transac number,  
6 saldo_final number,  
7 num_cta number,  
8 usuario number,  
9 fecha date,  
10 reg_auditoria number,  
11 ID_Cliente number,  
12 cod_ahorro varchar2(5),  
13 CONSTRAINTS FKvarios1 foreign key (num_cta,cod_ahorro,id_cliente)  
14 REFERENCES ahorros ( num_cta,cod_ahorro,id_cliente)  
15 );
```

Table created.

--creación de la secuencia para reg_auditoria

Create sequence seq_reg_auditoria start with 1 increment by 1;

```
SQL> Create sequence seq_reg_auditoria start with 1 increment by 1  
;
```

Sequence created.

```
ALTER TABLE AUDITORIA MODIFY FECHA TIMESTAMP;
```

```
SQL> ALTER TABLE AUDITORIA MODIFY FECHA TIMESTAMP;  
  
Table altered.
```

---Vista para TRIGGER Auditoria

```
CREATE VIEW Transac_Ahorro As
```

```
Select a.NUM_CTA,a.SAL_AHORRO, t.ID_TRANSACCION, t.TIPO_TRANSAC,  
t.MONTO_DR,t.USUARIO, t.ID_CLIENTE, t.COD_AHORRO
```

```
From transac_dr t join ahorros a on t.cod_ahorro = a.cod_ahorro;
```

```
SQL> CREATE VIEW Transac_Ahorro As  
2  Select a.NUM_CTA,a.SAL_AHORRO, t.ID_TRANSACCION, t.TIPO_TRANSAC, t.MONTO_DR,t.USUARIO, t.ID_CLIENTE, t.COD_AHORRO  
3  From transac_dr t join ahorros a on t.cod_ahorro = a.cod_ahorro;  
  
View created.
```

--PROCEDIMIENTO PARA ALMACENAMIENTO DE LA TABLA TIPO AHORRO

```
CREATE OR REPLACE PROCEDURE p_Tipo_Ahorro (  
p_cod_ahorro tipo_ahorro.cod_ahorro%TYPE,  
p_descripcion tipo_ahorro.descripcion%TYPE,  
p_tasa_interes tipo_ahorro.tasa_interes%TYPE  
)  
AS  
BEGIN  
INSERT INTO tipo_ahorro (cod_ahorro, descripcion, tasa_interes)  
Values (p_cod_ahorro, p_descripcion, p_tasa_interes);  
EXCEPTION  
WHEN OTHERS THEN  
DBMS_OUTPUT.PUT_LINE ('REGISTRO NO CREADO');  
COMMIT;  
END p_Tipo_Ahorro;  
/
```



```

SQL> CREATE OR REPLACE PROCEDURE p_Tipo_Ahorro (
  2  p_cod_ahorro tipo_ahorro.cod_ahorro%TYPE,
  3  p_descripcion tipo_ahorro.descripcion%TYPE,
  4  p_tasa_interes tipo_ahorro.tasa_interes%TYPE
  5  )
  6  AS
  7  BEGIN
  8  INSERT INTO tipo_ahorro (cod_ahorro, descripcion, tasa_interes)
  9      Values (p_cod_ahorro, p_descripcion, p_tasa_interes);
 10  EXCEPTION
 11  WHEN OTHERS THEN
 12  DBMS_OUTPUT.PUT_LINE ('REGISTRO NO CREADO');
 13  COMMIT;
 14  END p_Tipo_Ahorro;
 15  /

```

Procedure created.

--PROCEDIMIENTO PARA ALMACENAMIENTO DE LA TABLA AHORRO APROBADO

```

CREATE OR REPLACE PROCEDURE p_Ahorro_Aprobado (
p_cod_ahorro Ahorros.cod_ahorro%TYPE,
p_num_cta Ahorros.num_cta%TYPE,
p_fecha_open Ahorros.fecha_open%TYPE,
p_interes Ahorros.interes%TYPE,
p_letra Ahorros.letra%TYPE,
p_sal_ahorro Ahorros.sal_ahorro%TYPE,
p_sal_interes Ahorros.sal_interes%TYPE,
p_fecha_dep Ahorros.fecha_dep%TYPE,
p_fecha_ret Ahorros.fecha_ret%TYPE,
p_usuario Ahorros.usuario%TYPE,
p_fecha_modif Ahorros.fecha_modif%TYPE,
p_ID_Cliente Ahorros.ID_Cliente%TYPE,
p_Cod_Sucursal Ahorros.Cod_Sucursal%TYPE
)
AS
BEGIN
INSERT INTO Ahorros (cod_ahorro, num_cta, fecha_open , interes,
letra, sal_ahorro, sal_interes , fecha_dep, fecha_ret,
usuario, fecha_modif, ID_Cliente, Cod_Sucursal)
Values (p_cod_ahorro, p_num_cta, p_fecha_open , p_interes , p_letra, p_sal_ahor
ro, p_sal_interes , p_fecha_dep, p_fecha_ret, p_usuario, p_fecha_modif, p_ID_Cli
ente, p_Cod_Sucursal);
/*EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE ('REGISTRO NO CREADO');*/

```

```
COMMIT;  
END p_Ahorro_Aprobado;  
/
```

```
SQL> CREATE OR REPLACE PROCEDURE p_Ahorro_Aprobado (  
  2  p_cod_ahorro Ahorros.cod_ahorro%TYPE,  
  3  p_num_cta    Ahorros.num_cta%TYPE,  
  4  p_fecha_open Ahorros.fecha_open%TYPE,  
  5  p_interes    Ahorros.interes%TYPE,  
  6  p_letra      Ahorros.letra%TYPE,  
  7  p_sal_ahorro Ahorros.sal_ahorro%TYPE,  
  8  p_sal_interes Ahorros.sal_interes%TYPE,  
  9  p_fecha_dep  Ahorros.fecha_dep%TYPE,  
 10  p_fecha_ret  Ahorros.fecha_ret%TYPE,  
 11  p_usuario    Ahorros.usuario%TYPE,  
 12  p_fecha_modif Ahorros.fecha_modif%TYPE,  
 13  p_ID_Cliente Ahorros.ID_Cliente%TYPE,  
 14  p_Cod_Sucursal Ahorros.Cod_Sucursal%TYPE  
 15  )  
 16  AS  
 17  BEGIN  
 18  INSERT INTO Ahorros (cod_ahorro, num_cta, fecha_open , inte  
 19  letra, sal_ahorro, sal_interes , fecha_dep, fecha_ret, usua  
 20  Values (p_cod_ahorro, p_num_cta, p_fecha_open , p_interes ,  
 21  ario, p_fecha_modif, p_ID_Cliente, p_Cod_Sucursal);  
 22  /*EXCEPTION  
 23  WHEN OTHERS THEN  
 24  DBMS_OUTPUT.PUT_LINE ('REGISTRO NO CREADO');*/  
 25  COMMIT;  
 26  END p_Ahorro_Aprobado;  
 27  /
```

Procedure created.

```

--PROCEDIMIENTO PARA ALMACENAMIENTO DE LA TABLA TRANSAC_dr
CREATE OR REPLACE PROCEDURE p_Transacciones (
p_ID_Transaccion Transac_dr.ID_Transaccion%TYPE,
p_tipo_transac Transac_dr.tipo_transac%TYPE,
p_fecha_transac Transac_dr.fecha_transac%TYPE,
p_monto_dr Transac_dr.monto_dr%TYPE,
p_fecha_inser Transac_dr.fecha_inser%TYPE,
p_usuario Transac_dr.usuario%TYPE,
p_Cod_Sucursal Transac_dr.Cod_Sucursal%TYPE,
p_ID_Cliente Transac_dr.ID_Cliente%TYPE,
p_cod_ahorro Transac_dr.cod_ahorro%TYPE,
p_num_cta Transac_dr.num_cta%TYPE
)
AS
BEGIN
INSERT
INTO Transac_dr (ID_Transaccion, tipo_transac , fecha_transac , monto_dr , fecha
_inser,usuario, num_cta , Cod_Sucursal , ID_Cliente , cod_ahorro)
Values (p_ID_Transaccion, p_tipo_transac , p_fecha_transac , p_monto_dr , p_fecha_inser, p_usuario, p_num_cta, p_Cod_Sucursal , p_ID_Cliente ,p_cod_ahorro);

EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE ('REGISTRO NO CREADO');
COMMIT;
END p_Transacciones;
/

```

```

SQL> CREATE OR REPLACE PROCEDURE p_Transacciones (
  2  p_ID_Transaccion  Transac_dr.ID_Transaccion%TYPE,
  3  p_tipo_transac    Transac_dr.tipo_transac%TYPE,
  4  p_fecha_transac   Transac_dr.fecha_transac%TYPE,
  5  p_monto_dr        Transac_dr.monto_dr%TYPE,
  6  p_fecha_inser     Transac_dr.fecha_inser%TYPE,
  7  p_usuario         Transac_dr.usuario%TYPE,
  8  p_Cod_Sucursal    Transac_dr.Cod_Sucursal%TYPE,
  9  p_ID_Cliente      Transac_dr.ID_Cliente%TYPE,
10  p_cod_ahorro       Transac_dr.cod_ahorro%TYPE,
11  p_num_cta         Transac_dr.num_cta%TYPE
12 )
13 AS
14 BEGIN
15  INSERT INTO Transac_dr (ID_Transaccion, tipo_trans
_Cliente , cod_ahorro)
16  Values (p_ID_Transaccion, p_tipo_transac , p_fecha
ID_Cliente ,p_cod_ahorro);
17  /*EXCEPTION
18  WHEN OTHERS THEN
19  DBMS_OUTPUT.PUT_LINE ('REGISTRO NO CREADO');*/
20  COMMIT;
21  END p_Transacciones;
22  /

```

Procedure created.

```

--Procedimiento que actualiza las transacciones en la cuenta
CREATE OR REPLACE PROCEDURE update_transac
AS
CURSOR Recorrido IS
Select monto_dr, ID_Cliente, Cod_Sucursal, tipo_transac
, cod_ahorro, num_cta from Transac_dr;
v_sal_ahorro Ahorros.sal_ahorro%TYPE;
v_cal_int Ahorros.sal_interes%TYPE;
v_monto_dr Transac_dr.monto_dr%TYPE;
v_ID_Cliente Transac_dr.ID_Cliente%TYPE;
v_Cod_Sucursal Transac_dr.Cod_Sucursal%TYPE;
v_tipo_transac Transac_dr.tipo_transac%TYPE;
v_id_trans Transac_dr.id_transaccion%TYPE;
v_cod_ahorro Transac_dr.cod_ahorro%TYPE;
v_tasa_interes tipo_ahorro.tasa_interes%TYPE;
v_num_cta Transac_dr.num_cta%TYPE;
BEGIN
OPEN Recorrido;
LOOP
Fetch Recorrido INTO v_monto_dr, v_ID_Cliente, v_Cod_Sucursal, v_tipo_transac
, v_cod_ahorro, v_num_cta;
Select tasa_interes INTO v_tasa_interes
from tipo_ahorro where cod_ahorro= v_cod_ahorro;
Select sal_ahorro INTO v_sal_ahorro
FROM Ahorros
where num_cta=v_num_cta AND cod_ahorro= v_cod_ahorro
and ID_Cliente = v_ID_Cliente and Cod_Sucursal= v_Cod_Sucursal;
IF(( v_cod_ahorro= '01' OR v_cod_ahorro= '03') AND v_tipo_transac=
'DEPÓSITO') THEN
v_cal_int:= f_calcular_interes(v_monto_dr, v_tasa_interes);
UPDATE Ahorros
SET
sal_ahorro = sal_ahorro + v_monto_dr + v_cal_int,
sal_interes = sal_interes + v_cal_int,
fecha_dep= sysdate,
fecha_modif= sysdate
WHERE num_cta=v_num_cta AND cod_ahorro= v_cod_ahorro
and ID_Cliente = v_ID_Cliente ;
ELSIF (v_cod_ahorro= '02' AND v_tipo_transac= 'DEPOSITO') THEN
UPDATE Ahorros
SET
sal_ahorro= sal_ahorro + v_monto_dr,
fecha_dep= sysdate,
fecha_modif= sysdate
WHERE num_cta=v_num_cta AND cod_ahorro= v_cod_ahorro
and ID_Cliente = v_ID_Cliente ;

```

```

ELSIF ((v_cod_ahorro= '01' AND v_cod_ahorro= '03') AND v_tipo_transac=
'RETIRO') THEN
DBMS_OUTPUT.PUT_LINE ('NO PUEDES RETIRAR DE ESTA CUENTA');
ELSIF (v_cod_ahorro= '02' AND v_tipo_transac= 'RETIRO') THEN
  IF (v_sal_ahorro >= v_monto_dr) THEN
    UPDATE Ahorros
      SET
        sal_ahorro= sal_ahorro -v_monto_dr,
        fecha_ret= sysdate,
        fecha_modif= sysdate
    WHERE num_cta=v_num_cta AND cod_ahorro= v_cod_ahorro
      and ID_Cliente = v_ID_Cliente ;
  ELSE
    DBMS_OUTPUT.PUT_LINE ('NO SE PUDO REALIZAR LA TRANSACCION');
  END IF;
END IF;
EXIT WHEN Recorrido%NOTFOUND;
END LOOP;
CLOSE Recorrido;
END update_transac;
/

```

--procedimiento de consulta de ahorros corrientes

```
CREATE OR REPLACE PROCEDURE ahorro_corriente (  
v_cod_ahorro Ahorros.cod_ahorro%TYPE,  
v_sal_interes Ahorros.sal_interes%TYPE,  
v_sal_ahorro Ahorros.sal_ahorro%TYPE,  
v_interes Ahorros.interes%TYPE )  
AS  
CURSOR act_ahorro IS  
SELECT cod_ahorro, sal_interes, sal_ahorro, interes  
FROM Ahorros;  
BEGIN  
FOR i IN act_ahorro LOOP  
--SELECT interes into v_interes from Ahorros  
--where cod_ahorro = i.cod_ahorro;  
IF (v_cod_ahorro ='02') THEN  
UPDATE Ahorros  
SET  
sal_ahorro= v_sal_ahorro + f_interes(v_sal_ahorro, v_interes),  
sal_interes= v_sal_interes  
where cod_ahorro = i.cod_ahorro;  
END IF;  
END LOOP;  
EXCEPTION  
WHEN NO_DATA_FOUND THEN  
DBMS_OUTPUT.PUT_LINE ('No se encontro datos');  
COMMIT;  
END ahorro_corriente;  
/
```

Creacion de Funciones

```
--función que calcula el interés
CREATE OR REPLACE FUNCTION f_calcular_interes(
    f_interes tipo_ahorro.tasa_interes%TYPE,
    f_saldo Transac_dr.monto_dr%TYPE
)return number
AS
    interes_ahorro number(5,3);
BEGIN
    interes_ahorro := (f_saldo * (f_interes/100));
RETURN interes_ahorro;
END f_calcular_interes;
/
```

```
CREATE OR REPLACE FUNCTION f_interes(
    f_intereses Ahorros.interes%TYPE,
    f_sal_ahorro Ahorros.sal_ahorro%TYPE
)return number
AS
    interes_mensual number(5,3);
BEGIN
    interes_mensual := (f_sal_ahorro * (f_intereses/100));
RETURN interes_mensual;
END f_interes;
/
```


--trigger para las actualizaciones en la tabla sucursales

CREATE OR REPLACE TRIGGER act_sucursal

BEFORE INSERT OR DELETE OR UPDATE ON Ahorros FOR EACH ROW

BEGIN

IF INSERTING THEN

INSERT INTO temp_sucursal (Cod_sucursal,
Cod_ahorro, monto_ahorrado, fecha_modify, Usuario_modify)

VALUES (:new.cod_sucursal, :new.cod_ahorro, :new.sal_ahorro, SYSDATE,
:new.Usuario);

ELSIF UPDATING THEN

UPDATE temp_sucursal

SET monto_ahorrado= :new.sal_ahorro,

Usuario_modify = :new.Usuario

WHERE Cod_ahorro=:new.cod_ahorro AND Cod_sucursal = :new.cod_sucursal;

END IF;

END act_sucursal ;

/

--trigger para inserciones en la tabla auditoria

CREATE OR REPLACE TRIGGER t_aud

INSTEAD OF INSERT ON Transac_Ahorro FOR EACH ROW

BEGIN

INSERT INTO temp_auditoria (ID_TRANSACCION, TIPO_TRANSAC,
SALDO_CTA_ANTERIOR, MONTO_TRANSAC,
SALDO_FINAL, USUARIO, FECHA, REG_AUDITORIA, ID_CLIENTE)

VALUES (:new.ID_TRANSACCION, :new.TIPO_TRANSAC, :old.Sal_ahorro,
:new.Monto_DR,
:new.sal_ahorro, :new.usuario, sysdate, seq_reg_auditoria.NEXTVAL, :new.ID_client
e);

```
--END IF;  
END t_aud;  
/  

```

```
--INSERTION tabla tipo ahorro  
BEGIN  
p_Tipo_Ahorro('01', 'AHORRO DE NAVIDAD', 6);  
p_Tipo_Ahorro('02', 'AHORRO CORRIENTE', 4);  
p_Tipo_Ahorro('03', 'AHORRO ESCOLAR', 6);  
END;  
/  

```

```
SQL>  
SQL> BEGIN  
  2  p_Tipo_Ahorro('01', 'AHORRO DE NAVIDAD', 6);  
  3  p_Tipo_Ahorro('02', 'AHORRO CORRIENTE', 4);  
  4  p_Tipo_Ahorro('03', 'AHORRO ESCOLAR', 6);  
  5  END;  
  6  /  
  
PL/SQL procedure successfully completed.
```

--INSERTION tabla ahorros

BEGIN

p_Ahorro_Aprobado ('01', seq_num_cta.nextval, sysdate, 6, 10, 0,
0, sysdate, sysdate, 1231, sysdate, 1, 'VL');

p_Ahorro_Aprobado ('02', seq_num_cta.nextval, sysdate, 4, 25, 0,
0, sysdate, sysdate, 1131, sysdate, 2, 'CL');

p_Ahorro_Aprobado ('03', seq_num_cta.nextval, sysdate, 6, 5, 0,
0, sysdate, sysdate, 2231, sysdate, 3, 'LP');

END;

/

```
SQL> BEGIN
```

```
  2  p_Ahorro_Aprobado ('01', seq_num_cta.nextval, sysdate, 6, 10,  
    0, 0, sysdate, sysdate, 1231, sysdate, 1, 'VL');
```

```
  3  p_Ahorro_Aprobado ('02', seq_num_cta.nextval, sysdate, 4, 25,  
    0, 0, sysdate, sysdate, 1131, sysdate, 2, 'CL');
```

```
  4  p_Ahorro_Aprobado ('03', seq_num_cta.nextval, sysdate, 6, 5,  
    0, 0, sysdate, sysdate, 2231, sysdate, 3, 'LP');
```

```
  5  END;
```

```
  6  /
```

```
PL/SQL procedure successfully completed.
```

```
--INSERTION tabla transacciones  
BEGIN
```

```
p_Transacciones (seq_id_trans.nextval, 'DEPOSITO', sysdate, 25,sysdate,  
1243,4,'VL',1, '01' );
```

```
p_Transacciones (seq_id_trans.nextval, 'DEPOSITO', sysdate, 10,sysdate,  
6578,5,'CL',2,'02');
```

```
p_Transacciones (seq_id_trans.nextval, 'DEPOSITO', sysdate, 5,sysdate,  
9807,6,'LP', 3,'03');
```

```
END;  
/
```

```
BEGIN
```

```
Update_transac;
```

```
END;
```

```
/
```

```
BEGIN
```

```
Ahorro_corriente;
```

```
END;
```

```
/
```