**Universidad Tecnológica de Panamá**

**Facultad de Ingeniería de Sistemas Computacionales**

**Licenciatura en Ingeniería de Sistemas de Información**

**Departamento de Sistemas de Información**

Sistemas de Bases de Datos II

Equipo #2

Laboratorio #7

Facilitador:

Ing. Henry Lezcano

Integrantes grupo 2:

Karen Cabrera 20-14-5403

Medardo Logreira 8-879-1679

Sebastian Zamora 20-15-4392

Georlenys Martinez 8-926-2256

Grupo:

1IF131

II Semestre, 2020

**UNIVERSIDAD TECNOLOGICA DE PANAMA**

**FACULTAD DE INGENIERIA DE SISTEMAS COMPUTACIONES**

**LICENCIATURA EN INGENIERIA DE SISTEMAS DE INFORMACION**

**SISTEMAS DE BASE DE DATOS II**

**LABORATORIO NO.7**

**FUNDAMENTOS DEL LENGUAJE-CURSORES-PROCEDIMIENTOS Y FUNCIONES**

**y TRIGGERS ‘APLICAR PROGRAMACION ALMACENADA DE BASE DE DATOS**

**PL/SQL ORACLE’**

SQL> Create Table Sucursal (

Cod\_Sucursal Varchar (2) Primary key not Null,

NomSucursal Varchar (50)

);

Table created.

SQL> Create table TipoPrestamo (

Tip\_Cod Varchar (2) Primary Key not Null,

TipPrestamo varchar2 (50),

TasaInteres number (15,2)

);

Table created.

SQL> Create table SucursalPrestamo (

Cod\_Sucursal Varchar (2),

Tip\_Cod Varchar2 (2),

Monto\_Prestamo Number (15,2),

Primary Key (Cod\_Sucursal, Tip\_Cod)

);

SQL> Alter Table SucursalPrestamo Add Constraint fk\_CodSuc Foreign Key (Cod\_Sucursal) references Sucursal (cod\_Sucursal);

Table altered.

SQL> Create Table Tipoaho (

Cod\_Ahorro varchar (2),

MotoAhorrado Number (15,2),

Descripcion Varchar2 (50)

);

Table created.

SQL> Alter Table Tipoaho Modify (Cod\_Ahorro primary key);

Table alter

SQL> Create table TiposDeAhorros (

Cod\_Ahorro varchar (2),

Descripcion varchar2 (50),

Tasa\_Interes varchar (2),

Foreign key (Cod\_Ahorro) References Tipoaho (Cod\_Ahorro)

);

Table created.

Create Table Ahorros (

Cod\_Sucursal varchar (2),

Cod\_Ahorro varchar (2),

IDCliente varchar (4),

NumeroCuenta varchar (24),

Fecha\_Apertura date,

Tasa\_Interes Varchar (2),

LetraDepMensual varchar (3),

SaldoAhorro Number (15,2),

SaldoInteres Number (15,2),

FechaDeposito Date,

FechaRetiro Date,

Usuario Varchar2 (20),

FechaModificacion Date,

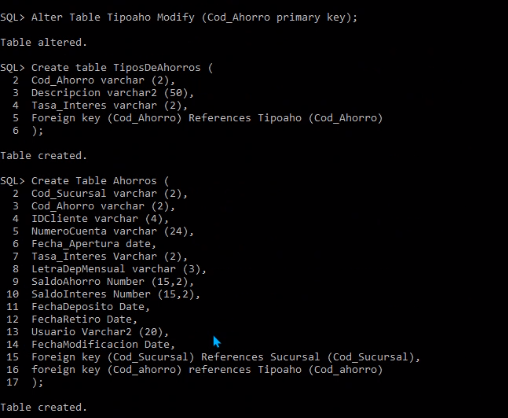
Foreign key (Cod\_Sucursal) References Sucursal (Cod\_Sucursal),

foreign key (Cod\_ahorro) references Tipoaho (Cod\_ahorro)

);

SQL> Alter Table TransadePoreti Add Constraint fk\_IDCliente Foreign key (IDCliente) References Ahorros (IDCliente);

Table altered.



Procedimientos

**Procedimiento** almacenado para la carga o inserción de la tabla paramétrica

* Tipo de ahorros

SQL> Create Or Replace Procedure InserTipAhorro (

p\_CodAhorro IN TiposDeAhorros.Cod\_Ahorro%Type,

p\_Descrip IN TiposDeAhorros.Descripcion%Type,

p\_TasaInte IN TiposDeAhorros.Tasa\_Interes%Type,

p\_Error OUT Varchar2) AS

Begin

Insert into TiposDeAhorros(Cod\_Ahorro, Descripcion, Tasa\_Interes)

Values (p\_CodAhorro, p\_Descrip, p\_TasaInte);

p\_Error := 'Ahorro Registrado';

Exception

When Others Then

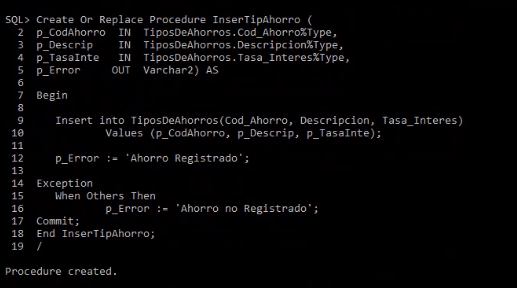
p\_Error := 'Ahorro no Registrado';

Commit;

End InserTipAhorro;

/

Procedure created.



**Procedimiento** almacenado para la apertura o inserción de los ahorros aprobados

con toda la información correspondiente.

SQL> Create or replace Procedure InserAhorros(

p\_CodSuc IN Ahorros.Cod\_Sucursal%Type,

p\_IDClien IN Ahorros.IDCliente%Type,

P\_NumC IN Ahorros.NumeroCuenta%Type,

p\_FechaAp IN Ahorros.Fecha\_Apertura%Type,

p\_Interes IN Ahorros.Tasa\_Interes%Type,

p\_LDepMen IN Ahorros.LetraDepMensual%Type,

p\_SalAho IN Ahorros.SaldoAhorro%Type,

P\_SalInt IN Ahorros.SaldoInteres%Type,

p\_FDep IN Ahorros.FechaDeposito%Type,

p\_FRet IN Ahorros.FechaRetiro%Type,

p\_User IN Ahorros.Usuario%Type,

p\_FMod IN Ahorros.FechaModificacion%Type,

p\_Error OUT Varchar2) AS

Begin

Insert into Ahorros (Cod\_Sucursal, IDCliente, NumeroCuenta, Fecha\_Apertura, Tasa\_Interes, LetraDepMensual, SaldoAhorro, SaldoInteres, FechaDeposito, FechaRetiro, Usuario, FechaModificacion)

Values (p\_CodSuc, p\_IDClien, P\_NumC, p\_FechaAp, p\_Interes, p\_LDepMen, p\_SalAho, P\_SalInt, p\_FDep, p\_FRet, p\_User, p\_FMod);

p\_Error := 'Apertura creada';

Exception

When Others Then

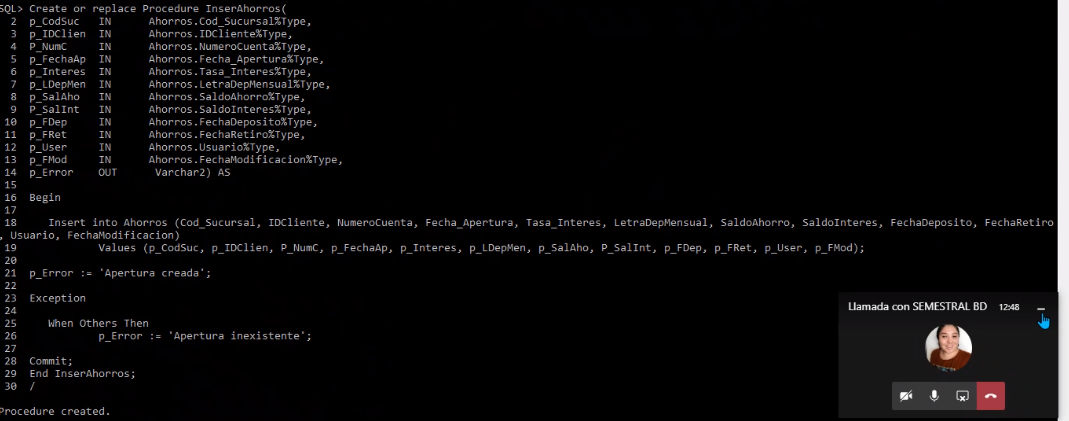
p\_Error := 'Apertura inexistente';

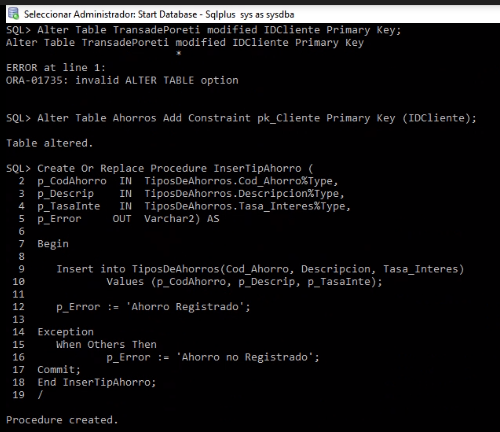
Commit;

End InserAhorros;

/

Procedure created.





**Procedimiento** almacenado para la carga o inserción de los depósitos o retiros

recibidos de los clientes para sus cuentas de ahorros que se almacena en la tabla

transadeporeti. Por lo menos uno para cada tipo de ahorro.

SQL> Create Or Replace Procedure InserPoreti (

p\_CoDSuc IN TransadePoreti.Cod\_Sucursal%Type,

p\_IDTran IN TransadePoreti.ID\_Transaccion%Type,

p\_IDClin IN TransadePoreti.IDCliente%Type,

p\_TipAho IN TransadePoreti.Tip\_Ahorro%Type,

p\_FeTran IN TransadePoreti.FechaTransaccion%Type,

p\_TiTran IN TransadePoreti.Tip\_Transac%Type,

p\_MontDR IN TransadePoreti.MontoDep\_Retiro%Type,

p\_FInser IN TransadePoreti.FechaInserccion%Type,

p\_User IN TransadePoreti.Usuario%Type,

p\_Error OUT Varchar2 ) AS

Begin

Insert Into TransadePoreti (Cod\_Sucursal, ID\_Transaccion, IDCliente, Tip\_Ahorro, FechaTransaccion, Tip\_Transac, MontoDep\_Retiro, FechaInserccion, Usuario)

Values (p\_CoDSuc, p\_IDTran, p\_IDClin, p\_TipAho, p\_FeTran, p\_TiTran, p\_MontDR, p\_FInser, p\_User);

p\_Error := 'Insercion Creada';

Exception

When Others Then

p\_Error := 'Insercion no Creada';

Commit;

End InserPoreti;

/

Procedure created.

 **Procedimiento** almacenado que actualice los depósitos o retiros de las cuentas de

ahorro correspondientes. Deberá implementar un **cursor** que busque el

depósitos/retiros insertados en la tabla uno a uno y los vaya actualizando en la

tabla de ahorros de cada cliente (proceso en lote o proceso en línea usted decide).

De la siguiente forma:

* Si el tipo de ahorro es navidad o escolar por cada deposito realizado debe calcular el interés que corresponde montodeposito \* tasade interes% que calculo que lo debe realizar una función diseñada previamente. El procedimiento debe actualizar el saldo de ahorro y el saldo interés de la cuenta de ahorro de los clientes
* Si el tipo de ahorro es corriente simplemente se realizar la aplicación del depósito o retiro a la cuenta de ahorro del cliente correspondiente. De las únicas cuentas que se puede realizar retiros es de la cuenta de ahorro corriente por lo tanto el procedimiento debe controlar esta situación.

SQL> Create Or Replace Function CalcInteres (

f\_TipAho TransadePoreti.Tip\_Ahorro%Type)

Return Varchar2 IS

V\_Monto Number;

Begin

Select Tip\_Ahorro, MontoDep\_Retiro INTO

f\_TipAho, V\_Monto

From TransadePoreti

Where Tip\_Ahorro = f\_TipoAho;

If f\_TipAho = Tip\_Ahorro then

V\_Monto \* 0.06;

Else

V\_Monto \* 0.04;

End IF;

End CalcInteres;

/

Warning: Function created with compilation errors.

 **Triggers.** Para las actualizaciones en la tabla de sucursales

 **Triggers.** Para inserciones en la tabla de auditoria.

 **Procedimiento.** Llegamos a fin de mes y hay pagar los interés a la cuenta de ahorro

corriente. Deberá diseñar un cursor que consulte todos los ahorros corrientes de

forma controlada y calcule los interés que habrá que pagar mensualmente a estos

ahorros (**saldo de ahorros por la tasade interes%** ) este cálculo lo realizar una

función. El procedimiento deberá actualizar el saldo del ahorro y saldo de interés

de las cuentas.’ El interés es sumado al saldo de ahorro’.

CREATE OR REPLACE TRIGGER sucur\_trig AFTER UPDATE OR INSERT OR DELETE

ON Ahorros FOR EACH ROW

BEGIN

IF INSERTING THEN

INSERT INTO Temp\_suc (upd\_sucursal, Cod\_sucursal, Cod\_ahorro, monto\_ahor\_v, monto\_ahor\_n, usuario, fecha)

Values (trig\_sucur.nextval, :new.cod\_Sucursal, :new.Cod\_ahorro, :old.SaldoAhorro, :new.SaldoAhorro,

user, sysdate);

ELSIF UPDATING THEN

UPDATE Temp\_suc

SET

Upd\_sucursal=trig\_suc.nextval,

Cod\_Sucursal= :NEW.cod\_Sucursal,

Cod\_Ahorro= :new.Cod\_Ahorro,

monto\_ahor\_n=:old.SaldoAhorro + (:new.SaldoAhorro-:old.SaldoAhorro),

monto\_ahor\_v=:OLD.SaldoAhorro,

usuario=user,

fecha=sysdate

where Cod\_Ahorro=:new.Cod\_Ahorro AND Cod\_Sucursal=:new.Cod\_Sucursal;

END IF;

END sucur\_trig;

/