

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Licenciatura en Ingeniería de Sistemas de Información

Departamento de Sistemas de Información, Control y Evaluación de Recursos Informáticos

Sistemas de Bases de Datos II

Laboratorio No.6. undamentos del lenguaje-cursores-procedimientos y funciones

Facilitador:

Ing. Henry Lezcano

Integrantes:

David Cedeño 8-951-990

Simon Billeke E-8-154322

Gabriel Grimaldo 8-949-56

Daniel Rodriguez 3-740-2242

Grupo:

1IF131

II Semestre, 2020

1. **Agregar el atributo edad a la tabla de clientes proporcionada.**

alter table Clientes ADD Edad number not null;

1. **Agregar una tabla al modelo físico que almacenes las sucursales de la empresa la financiera con las restricciones correspondientes:**

Sucursal (cod\_surcursal, nombresucursal, tipoprestamo, montoprestamos)

Create table Sucursal (

cod\_sucursal varchar(5) not null,

Nombresucursal varchar(100) not null,

constraint pk\_Sucursal\_cod\_tip primary key (cod\_sucursal)

);

Create Table Sucursal\_Prestamo(

tipoprestamo varchar(5) not null,

cod\_sucursal varchar (10) not null,

montopres number not null,

constraint fk\_Suc\_Pres\_tipoP foreign key (tipoprestamo) References Tipo\_Prestamos (ID\_Tipo),

constraint fk\_Suc\_Pres\_codC foreign key (cod\_sucursal) References Sucursal (cod\_sucursal),

constraint pk\_Suc\_Pres primary key (tipoprestamo,cod\_sucursal));

1. **Agregar a la tabla clientes y préstamos el atributo de código de sucursal.**

alter table Clientes ADD cod\_sucursal varchar(10) not null;

alter table Clientes ADD constraint fk\_Clientes\_Sucursal foreign key(cod\_sucursal) References Sucursal (cod\_sucursal);

alter table Prestamos ADD cod\_sucursal varchar(10) not null;

alter table Prestamos ADD constraint fk\_Prestamos\_Sucursal foreign key(cod\_sucursal) References Sucursal (cod\_sucursal);

1. **Agregar a la tabla del prestamos los siguientes atributos**:

saldoactual, que cuando el préstamo es aprobado es equivalente al monto aprobado, interéspagado que será cargado a partir de los pagos realizados por cliente en proceso de aplicación, fechamodificacion (ddmmyyyy hhmmss), usuario)

alter table Prestamos ADD (

saldoactual number not null,

interespagado number not null,

fechamodificacion date not null,

usuario varchar(30) not null);

1. **Agregar una tabla transaccional para recibir los pagos de los clientes con las restricciones correspondiente que contendrá el siguiente atributo:**

Transacpagos (cod\_sucursal, id\_transaccion, id\_cliente, tipoprestamo, fechatransaccion, monto del pago, fechainsercion(dd/mm/yyyy, hh:mm:ss, usuario)

Create table Transacpagos(

cod\_sucursal varchar(10) not null,

id\_transa number not null,

IDCliente number not null,

tipopres varchar(5) not null,

f\_transa date not null,

monto\_pago number not null,

F\_inser date not null,

usuario varchar(30) not null,

constraint pk\_Transacpagos\_idtran primary key (id\_transa),

constraint fk\_Transacpagos\_codsur foreign key (cod\_sucursal) references Sucursal(cod\_sucursal),

constraint fk\_Transacpagos\_tipoP foreign key (tipopres) references Tipo\_Prestamos (ID\_Tipo),

constraint fk\_Transacpagos\_IDCli foreign key (IDCliente) references Clientes (IDCliente));

1. **Procedimiento almacenado para la carga o inserción de las tablas paramétricas**
   1. **Tipo de Teléfonos**

CREATE OR REPLACE PROCEDURE Crear\_Tipo\_Telefono(

p\_IDTipo Tipo\_Telefono.IDTipo%TYPE,

p\_Descripcion Tipo\_Telefono.Descripcion%TYPE,

p\_error out varchar2)AS

BEGIN

INSERT INTO Tipo\_Telefono (IDTipo,Descripcion)

Values(p\_IDTipo,p\_Descripcion);

p\_error := 'Registro Creado';

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

p\_error := 'Tipo de telefono existente';

WHEN OTHERS THEN

p\_error := 'No se creo el registro';

COMMIT;

END Crear\_tipo\_Telefono;

* + 1. **Prueba**

DECLARE

v\_error varchar2(100);

BEGIN

Crear\_Tipo\_Telefono(1,'Celular',v\_error);

Crear\_Tipo\_Telefono(2,'Telefono de Residencia',v\_error);

Crear\_Tipo\_Telefono(3,'Telfono Familiar',v\_error);

Crear\_Tipo\_Telefono(4,'Telefono de Conyugue',v\_error);

dbms\_output.put\_line(v\_error);

END;

* 1. **Tipo de Correos**

CREATE OR REPLACE PROCEDURE Crear\_Tipo\_Email(

p\_IDTipo Tipo\_Email.IDTipo%TYPE,

p\_Descripcion Tipo\_Email.Descripcion%TYPE,

p\_error out varchar2)AS

BEGIN

INSERT INTO Tipo\_Email (IDTipo,Descripcion)

Values(p\_IDTipo,p\_Descripcion);

p\_error := 'Registro Creado';

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

p\_error := 'Tipo de correo existente';

WHEN OTHERS THEN

p\_error := 'No se creo el registro';

COMMIT;

END Crear\_Tipo\_Email;

* + 1. **Prueba**

DECLARE

v\_error varchar2(100);

BEGIN

Crear\_Tipo\_Email(1,'Personal',v\_error);

Crear\_Tipo\_Email(2,'Laboral',v\_error);

Crear\_Tipo\_Email(3,'Academico',v\_error);

dbms\_output.put\_line(v\_error);

END;

* 1. **Profesiones,**

CREATE OR REPLACE PROCEDURE Crear\_Profesion(

p\_IDProfesion Profesion.IDProfesion%TYPE,

p\_Descripcion Profesion.Descripcion%TYPE,

p\_error out varchar2)AS

BEGIN

INSERT INTO Profesion (IDProfesion,Descripcion)

Values(p\_IDProfesion,p\_Descripcion);

p\_error := 'Registro Creado';

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

p\_error := 'Profesion existente';

WHEN OTHERS THEN

p\_error := 'No se creo el registro';

COMMIT;

END Crear\_Profesion;

* + 1. **Prueba**

DECLARE

v\_error varchar2(100);

BEGIN

Crear\_Profesion(101,'Abogado',v\_error);

Crear\_Profesion(102,'Farmaceutico',v\_error);

Crear\_Profesion(103,'Ejecutivo',v\_error);

Crear\_Profesion(104,'Arquitecto',v\_error);

Crear\_Profesion(105,'Doctor',v\_error);

dbms\_output.put\_line(v\_error);

END;

* 1. **Sucursales**

CREATE OR REPLACE PROCEDURE Crear\_Sucursal(

p\_cod\_suc Sucursal.cod\_sucursal%TYPE,

p\_Nombresucursal Sucursal.Nombresucursal%TYPE,

p\_error out varchar2)AS

BEGIN

INSERT INTO Sucursal (cod\_sucursal,Nombresucursal)

Values(p\_cod\_suc,p\_Nombresucursal);

p\_error := 'Registro Creado';

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

p\_error := 'Sucursal existente';

WHEN OTHERS THEN

p\_error := 'No se creo el registro';

COMMIT;

END Crear\_Sucursal;

* + 1. **Prueba**

DECLARE

v\_error varchar2(100);

BEGIN

Crear\_Sucursal('S-BDG','S.Brisas Del Golf',v\_error);

Crear\_Sucursal('S-VL','S.Villa Lucre',v\_error);

Crear\_Sucursal('S-CDE','S.Costa Del Este',v\_error);

Crear\_Sucursal('S-HAP','S.Hato Pintado',v\_error);

dbms\_output.put\_line(v\_error);

END;

* 1. **Tipos de prestamos**

CREATE OR REPLACE PROCEDURE Crear\_Tipos\_Prestamos(

p\_ID\_Tipo Tipo\_Prestamos.ID\_Tipo%TYPE,

p\_Nombre Tipo\_Prestamos.Nombre%TYPE,

p\_Interes Tipo\_Prestamos.Interes%TYPE,

p\_error out varchar2)AS

BEGIN

INSERT INTO Tipo\_Prestamos(ID\_Tipo,Nombre,Interes)

Values(p\_ID\_Tipo,p\_Nombre,p\_Interes);

p\_error := 'Registro Creado';

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

p\_error := 'Tipo de prestamo existente';

WHEN OTHERS THEN

p\_error := 'No se creo el registro';

COMMIT;

END Crear\_Tipos\_Prestamos;

* + 1. **Prueba**

DECLARE

v\_error varchar2(100);

BEGIN

Crear\_Tipos\_Prestamos('PERS','Personal',0.09,v\_error);

Crear\_Tipos\_Prestamos('AUT','Auto',0.075,v\_error);

Crear\_Tipos\_Prestamos('HIPO','Hipoteca',0.065,v\_error);

Crear\_Tipos\_Prestamos('GAAH','Garantizado/ahorros',0.065,v\_error);

dbms\_output.put\_line(v\_error);

END;

1. **Procedimiento almacenado para la carga o inserción de los clientes con toda la información correspondiente. Este procedimiento debe invocar una Función que calcule la edad de los clientes.**

alter table Clientes ADD F\_nacim date not null;

CREATE OR REPLACE FUNCTION Calcular\_edad(

p\_F\_nacim Clientes.F\_nacim%TYPE) RETURN NUMBER AS

BEGIN

RETURN floor(months\_between(trunc(sysdate),p\_F\_nacim)/12);

END Calcular\_edad;

CREATE OR REPLACE PROCEDURE Crear\_Clientes(

p\_IDCliente Clientes.IDCliente%TYPE,

p\_Profesion Clientes.Profesion%TYPE,

p\_cedula Clientes.cedula%TYPE,

p\_Nombre Clientes.Nombre%TYPE,

p\_Apellido Clientes.Apellido%TYPE,

p\_Sexo Clientes.Sexo%TYPE,

p\_cod\_suc Clientes.cod\_sucursal%TYPE,

p\_F\_nacim Clientes.F\_nacim%TYPE,

p\_error out varchar2) AS

v\_Edad Clientes.Edad%TYPE;

BEGIN

p\_error := 'Registro Creado';

v\_Edad :=Calcular\_edad(p\_F\_nacim);

INSERT INTO Clientes(IDCliente,Profesion,cedula,Nombre,Apellido,Sexo,Edad,cod\_sucursal,F\_nacim)

Values(p\_IDCliente,p\_Profesion,p\_cedula,p\_Nombre,p\_Apellido,p\_Sexo,v\_Edad,p\_cod\_suc,p\_F\_nacim);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

p\_error := 'Cliente existente';

WHEN OTHERS THEN

p\_error := 'No se creo el registro';

COMMIT;

END Crear\_Clientes;

* 1. **Pruebas**

DECLARE

v\_error varchar2(100);

BEGIN

Crear\_Clientes(1,105,'4-909-111','Marianela','Camacho','Femenino','S-BDG',to\_date('13/09/1969','DD/MM/YYYY'),v\_error);

Crear\_Clientes(2,104,'8-423-834','Jose','Barrios','Masculino','S-VL',to\_date('01/01/1969','DD/MM/YYYY'),v\_error);

Crear\_Clientes(3,103,'8-812-2312','Maria','Errigo','Femenino','S-CDE',to\_date('06/05/1971','DD/MM/YYYY'),v\_error);

Crear\_Clientes(4,102,'8-111-1111','Elizabeth','Ohlsen','Femenino','S-CDE',to\_date('23/08/1990','DD/MM/YYYY'),v\_error);

Crear\_Clientes(5,101,'7-598-731','Mario','Alcedo','Masculino','S-HAP',to\_date('01/12/1989','DD/MM/YYYY'),v\_error);

dbms\_output.put\_line(v\_error);

END;

* 1. **Clientes\_Telefono**

CREATE OR REPLACE PROCEDURE Crear\_Clien\_Telef(

p\_Id\_clien Clientes\_Telefono.IDCliente%TYPE,

p\_Id\_Tipo Clientes\_Telefono.IDTipo%TYPE,

p\_telef clientes\_telefono.telefono%TYPE,

p\_error out varchar2) AS

BEGIN

p\_error := 'Registro Creado';

INSERT INTO Clientes\_Telefono(IDCliente,IDTipo,Telefono)

VALUES(p\_Id\_clien,p\_Id\_Tipo,p\_telef);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

p\_error := 'Prestamo existente';

WHEN OTHERS THEN

p\_error := 'No se creo el registro';

COMMIT;

END Crear\_Clien\_Telef;

* + 1. **Prueba**

DECLARE

v\_error varchar2(100);

BEGIN

Crear\_Clien\_Telef(1,1,'6622-3388');

Crear\_Clien\_Telef(1,2,'474-7848');

Crear\_Clien\_Telef(1,3,'6284-1248');

Crear\_Clien\_Telef(2,1,'6600-0000');

Crear\_Clien\_Telef(2,2,'466-0666');

Crear\_Clien\_Telef(2,3,'6600-0217');

Crear\_Clien\_Telef(3,1,'6875-6572');

Crear\_Clien\_Telef(3,3,'6665-3751');

Crear\_Clien\_Telef(4,1,'6587-4408');

Crear\_Clien\_Telef(4,2,'474-1234');

Crear\_Clien\_Telef(4,3,'6323-1124');

Crear\_Clien\_Telef(4,4,'6662-8987');

dbms\_output.put\_line(v\_error);

END;

* 1. **Clientes\_Email**

CREATE OR REPLACE PROCEDURE Crear\_Client\_Email(

p\_Id\_clien Clientes\_Email.IDCliente%TYPE,

p\_Id\_Tipo Clientes\_Email.IDTipo%TYPE,

p\_correo Clientes\_Email.Correo%TYPE,

p\_error out varchar2) AS

BEGIN

p\_error := 'Registro Creado';

INSERT INTO Clientes\_Email(IDCliente,IDTipo,Correo)

VALUES(p\_Id\_clien,p\_Id\_Tipo,p\_correo);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

p\_error := 'Prestamo existente';

WHEN OTHERS THEN

p\_error := 'No se creo el registro';

COMMIT;

END Crear\_Client\_Email;

* + 1. **Prueba**

DECLARE

v\_error varchar2(100);

BEGIN

Crear\_Client\_Email(1,1,'MarianelaCa@gmail.com');

Crear\_Client\_Email(1,2,'Marianela1@empresa.com');

Crear\_Client\_Email(2,1,'JoseBA@gmail.com');

Crear\_Client\_Email(2,2,'JoseBA23@empresa.com');

Crear\_Client\_Email(3,1,'MariaErri@gmail.com');

Crear\_Client\_Email(3,2,'Maria\_Errigo@empresa.com');

Crear\_Client\_Email(4,1,'ElizabethOHl@gmail.com');

Crear\_Client\_Email(4,2,'ElizabethOHl@empresa.com');

Crear\_Client\_Email(5,1,'Mario\_Alce@gmail.com');

Crear\_Client\_Email(5,2,'Mario\_Alce21@empresa.com');

Crear\_Client\_Email(5,3,'Mario\_Alce@usma.com.pa');

dbms\_output.put\_line(v\_error);

END;

1. **Procedimiento almacenado para la carga o inserción de los préstamos aprobados con toda la información correspondiente e igualmente este proceso deberá actualizar la información de préstamo en la tabla de sucursales.**

CREATE OR REPLACE PROCEDURE Crear\_Prestamos(

p\_IDclien Prestamos.ID\_Cliente%TYPE,

p\_IDtip Prestamos.ID\_Tipo%TYPE,

p\_Nprest Prestamos.N\_prestamo%TYPE,

p\_codSuc Sucursal.cod\_sucursal%TYPE,

p\_Faprob Prestamos.F\_aprobado%TYPE,

p\_Maprob Prestamos.M\_aprobado%TYPE,

p\_Lmens Prestamos.L\_mensual%TYPE,

p\_Mpag Prestamos.M\_pagado%TYPE,

p\_Fpago Prestamos.F\_pago%TYPE,

p\_saldoac Prestamos.saldoactual%TYPE,

p\_interes Prestamos.interespagado%TYPE,

p\_usuario Prestamos.usuario%TYPE,

p\_error out varchar2) AS

BEGIN

p\_error := 'Registro Creado';

INSERT INTO Prestamos(ID\_Cliente,ID\_Tipo,N\_prestamo,cod\_sucursal,F\_aprobado,M\_aprobado

,L\_mensual,M\_pagado,F\_pago,saldoactual,interespagado,

fechamodificacion,usuario)

VALUES(p\_IDclien,p\_IDtip,p\_Nprest,p\_codSuc,p\_Faprob,p\_Maprob,p\_Lmens,p\_Mpag,p\_Fpago,p\_saldoac,p\_interes,sysdate,p\_usuario);

UPDATE Sucursal\_Prestamo

SET montopres = montopres+ p\_Maprob Where tipoprestamo=p\_IDtip AND cod\_sucursal=p\_codSuc;

IF SQL%ROWCOUNT=0 THEN

INSERT INTO Sucursal\_Prestamo(tipoprestamo,cod\_sucursal,montopres)

VALUES(p\_IDtip, p\_codSuc, p\_Maprob);

COMMIT;

END IF;

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

p\_error := 'Prestamo existente';

WHEN OTHERS THEN

p\_error := 'No se creo el registro';

COMMIT;

END Crear\_Prestamos;

* 1. **Prueba**

DECLARE

v\_error varchar2(100);

BEGIN

Crear\_Prestamos(1,'PERS',1,'S-BDG',to\_date('17/10/2020','DD/MM/YYYY'),10000,250,0,to\_date('17/10/2020','DD/MM/YYYY'),10000,900,user,v\_error);

Crear\_Prestamos(2,'AUT',2,'S-VL',to\_date('17/10/2020','DD/MM/YYYY'),30000,447.91,0,to\_date('17/10/2020','DD/MM/YYYY'),30000,2250,user,v\_error);

Crear\_Prestamos(3,'HIPO',3,'S-CDE',to\_date('17/10/2020','DD/MM/YYYY'),200000,591.66,0,to\_date('17/10/2020','DD/MM/YYYY'),200000,252.77,user,v\_error);

Crear\_Prestamos(4,'GAAH',4,'S-CDE',to\_date('17/10/2020','DD/MM/YYYY'),15000,332.81,0,to\_date('17/10/2020','DD/MM/YYYY'),15000,108.35,user,v\_error);

dbms\_output.put\_line(v\_error);

END;

1. **Procedimiento almacenado para la carga o inserción de los pagos recibidos de los clientes que se almacena en la tabla transacpagos. Por lo menos uno para cada tipo de préstamos.**

CREATE OR REPLACE PROCEDURE Crear\_pagos(

p\_codSuc Transacpagos.cod\_sucursal%TYPE,

p\_IDtrans Transacpagos.id\_transa%TYPE,

p\_IDClien Transacpagos.IDCliente%TYPE,

p\_tipopres Transacpagos.tipopres%TYPE,

p\_ftransa Transacpagos.f\_transa%TYPE,

p\_montpa Transacpagos.monto\_pago%TYPE,

p\_finser Transacpagos.F\_inser%TYPE,

p\_usuario Transacpagos.usuario%TYPE,

p\_error out varchar2) AS

BEGIN

p\_error := 'Proceso completado';

INSERT INTO Transacpagos (cod\_sucursal,id\_transa,IDCliente,tipopres,f\_transa,monto\_pago,F\_inser,usuario)

VALUES(p\_codSuc,p\_IDtrans,p\_IDClien,p\_tipopres,p\_ftransa,p\_montpa,p\_finser,p\_usuario);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

p\_error := 'Pago existente';

WHEN OTHERS THEN

p\_error := 'ERROR..No se creo el registro';

COMMIT;

END Crear\_pagos;

* 1. **Pruebas**

DECLARE

v\_error varchar2(100);

BEGIN

Crear\_pagos('S-BDG',1,1,'PERS',to\_date('17/10/2020','DD/MM/YYYY'),250,sysdate,user,v\_error);

Crear\_pagos('S-VL',2,2,'AUT',to\_date('17/10/2020','DD/MM/YYYY'),447.91,sysdate,user,v\_error);

Crear\_pagos('S-CDE',2,3,'HIPO',to\_date('17/10/2020','DD/MM/YYYY'),591.66,sysdate,user,v\_error);

Crear\_pagos('S-CDE',3,4,'GAAH',to\_date('17/10/2020','DD/MM/YYYY'),332.81,sysdate,user,v\_error);

dbms\_output.put\_line(v\_error);

END;

1. **Procedimiento almacenado que actualice el pago recibo a los préstamos correspondientes. Deberá implementar un cursor que busque los pagos insertados uno a uno y los vaya actualizando en la tabla de préstamos y en la tabla sucursales.**
   1. **Para aplicar el pago este debe rebajarlo del saldo prestamos ( 1000- 20)=980.00. Pero debe tomar en cuenta lo siguiente el préstamo paga interés y este se calcula sobre el saldo del préstamo (saldo del préstamo \* tasa de interés%) el pago interés es mensual (1000 \* 1%) = 50.00.**
   2. **El cálculo del interés lo realiza una Función que es invocada desde procedimiento.**
   3. **El interés se cobra primero y de quedar alguna porción del monto pagado por el cliente se aplica al saldo del préstamo.**
   4. **La tabla de sucursales solo almacena los montos prestados por la empresa**
   5. **financiera en función de esto, aplicar las actualizaciones.**

CREATE OR REPLACE FUNCTION Calc\_Interes(

p\_saldoac Prestamos.saldoactual%TYPE,

p\_Interes Tipo\_Prestamos.Interes%TYPE)RETURN NUMBER AS

BEGIN

Return p\_saldoac\*p\_Interes;

END Calc\_Interes;

/

CREATE OR REPLACE PROCEDURE Actua\_Pago (p\_mensaje OUT varchar2)AS

CURSOR C\_Tansacpagos IS

SELECT COD\_SUCURSAL,ID\_TRANSA,IDCLIENTE,TIPOPRES,MONTO\_PAGO

FROM Transacpagos;

v\_SalAct Prestamos.saldoactual%TYPE;

v\_pagointe Prestamos.interespagado%TYPE;

v\_montp Prestamos.M\_pagado%TYPE;

v\_intereses Prestamos.interespagado%TYPE;

BEGIN

p\_mensaje := 'Se actualizo la base de datos';

FOR v\_indice IN C\_Tansacpagos loop

SELECT Interes INTO v\_intereses FROM Tipo\_Prestamos WHERE ID\_Tipo=v\_indice.TIPOPRES;

SELECT saldoactual INTO v\_SalAct FROM Prestamos WHERE ID\_Tipo=v\_indice.TIPOPRES AND ID\_Cliente=v\_indice.IDCLIENTE;

v\_pagointe:= Calc\_Interes(v\_SalAct,v\_intereses);

v\_montp := v\_pagointe- v\_indice.monto\_pago;

IF(v\_montp>0) THEN

UPDATE Prestamos

SET

M\_pagado = v\_montp,

F\_pago = sysdate,

saldoactual=saldoactual-v\_montp,

interespagado = v\_pagointe

WHERE

ID\_Cliente=v\_indice.IDCLIENTE AND ID\_Tipo=v\_indice.TIPOPRES;

UPDATE Sucursal\_Prestamo

SET

montopres =montopres-v\_montp

WHERE

tipoprestamo = v\_indice.TIPOPRES AND

cod\_sucursal=v\_indice.cod\_sucursal;

ELSE

UPDATE Prestamos

SET

M\_pagado = v\_montp,

F\_pago = sysdate,

interespagado = v\_pagointe

WHERE

ID\_Cliente=v\_indice.IDCLIENTE AND ID\_Tipo=v\_indice.TIPOPRES;

END IF;

end loop;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

p\_mensaje:='No se encontro ningun dato';

WHEN others THEN

p\_mensaje:='No se actualizo la tabla';

COMMIT;

END Actua\_Pago;

* 1. **Prueba**

DECLARE

v\_error varchar2(100);

BEGIN

Actua\_Pago(v\_error);

dbms\_output.put\_line(v\_error);

END;

1. **Vistas**

* Muetsra los clientes y el nombre de la sucursal a la que pertenecen

Create view Clientes\_Suc as

Select c.Cedula,c.Nombre,c.Apellido,s.Nombresucursal

FROM Clientes c join Sucursal s on c.cod\_sucursal = s.cod\_sucursal;

* Muestra el Nombre del cliente, el monto aprobado y el saldo actual de su prestamo

Create view Clien\_Prest as

Select c.Cedula,c.Nombre ||' '|| c.Apellido as Nombre ,p.m\_aprobado,p.saldoactual

FROM Clientes c join Prestamos p on c.idcliente = p.id\_cliente;

* Muestra el total que tiene casa sucursal en prestmos

Create view Suc\_Pres as

Select s.nombresucursal,tp.Nombre,sp.montopres

FROM sucursal s join sucursal\_prestamo sp on s.cod\_sucursal=sp.cod\_sucursal

join tipo\_prestamos tp on sp.tipoprestamo=tp.id\_tipo;