**TABLAS**

create table PROFESION (

ID\_Pro number not null,

profesion varchar2 (30) not null,

constraint ID\_Pro\_pk primary key (ID\_Pro)

);

create table TIPO\_PRESTAMO (

ID\_tipo number not null,

tipo varchar2 (30) not null,

tasa\_interes number not null,

constraint ID\_tipo\_pk primary key (ID\_tipo)

);

create table EMAIL (

ID\_email number not null,

Tipo\_email varchar2 (20) not null,

constraint ID\_email\_pk primary key (ID\_email)

);

create table TELEFONO (

ID\_telefono number not null,

Tipo\_telefono varchar2 (20) not null,

constraint ID\_telefono\_pk primary key (ID\_telefono)

);

create table CLIENTE (

ID\_Cliente number not null,

ID\_Pro number not null,

cedula varchar2 (20) not null,

nombre varchar2 (30) not null,

apellido varchar2 (30) not null,

sexo char not null,

fecha\_nac date not null,

constraint ID\_Cliente\_pk primary key (ID\_Cliente),

constraint ID\_prof\_fk foreign key (ID\_Pro) references PROFESION (ID\_Pro)

);

create table EMAIL\_CLIENTE (

ID\_email number not null,

ID\_Cliente number not null,

email varchar2 (40) not null,

constraint ID\_email\_fk foreign key (ID\_email) references EMAIL (ID\_email),

constraint ID\_Cl\_email\_fk foreign key (ID\_Cliente) references CLIENTE (ID\_Cliente),

constraint ID\_email\_cl\_pk primary key (ID\_email, ID\_Cliente)

);

create table TELEFONO\_CLIENTE (

ID\_telefono number not null,

ID\_Cliente number not null,

telefono varchar2 (20) not null,

constraint ID\_tel\_fk foreign key (ID\_telefono) references TELEFONO (ID\_telefono),

constraint ID\_cl\_tel\_fk foreign key (ID\_Cliente) references CLIENTE (ID\_Cliente),

constraint ID\_telef\_cl\_pk primary key (ID\_telefono, ID\_Cliente)

);

create table PRESTAMO (

ID\_tipo number not null,

ID\_Cliente number not null,

Num\_prestamo number not null,

fecha\_aprob date not null,

monto\_aprob number not null,

letra\_mensual number not null,

monto\_pago number not null,

fecha\_pago date not null,

constraint ID\_tipo\_fk foreign key (ID\_tipo) references TIPO\_PRESTAMO (ID\_tipo),

constraint ID\_cliente\_prest\_fk foreign key (ID\_Cliente) references CLIENTE (ID\_Cliente),

constraint ID\_cl\_tipo\_prest\_pk primary key (ID\_tipo, ID\_Cliente)

);

create unique index numero\_prestamo on PRESTAMO (Num\_prestamo);

alter table CLIENTE

add edad number not null;

create table SUCURSAL (

Cod\_sucursal number not null,

Nombre\_sucursal varchar2 (30) not null,

Constraint Cod\_suc\_pk primary key (Cod\_sucursal)

);

create table SUCURSAL\_PRESTAMO(

Cod\_sucursal number not null,

ID\_tipo number not null,

Constraint ID\_tipo\_suc\_fk foreign key (ID\_tipo) references TIPO\_PRESTAMO (ID\_tipo),

Constraint Cod\_sucursal\_p\_fk foreign key (Cod\_sucursal) references SUCURSAL (Cod\_sucursal),

Constraint Cod\_id\_suc\_pk primary key (Cod\_sucursal, ID\_tipo),

Monto\_prestamo number not null

);

Alter table CLIENTE

Add (Cod\_sucursal number not null, constraint Cod\_suc\_cl\_fk foreign key (Cod\_sucursal) references SUCURSAL (Cod\_sucursal));

Alter table PRESTAMO

Add (Cod\_sucursal number not null, constraint Cod\_suc\_pres\_fk foreign key (Cod\_sucursal) references SUCURSAL (Cod\_sucursal));

alter table PRESTAMO

ADD (saldo\_actual number not null, interes\_pagado number not null, fecha\_modif date not null, usuario varchar2 (30) not null);

create table transact\_pagos (

ID\_transaccion number not null,

constraint ID\_transaccion\_pk primary key (ID\_transaccion),

Cod\_sucursal number not null,

ID\_Cliente number not null,

ID\_tipo number not null,

Fechatransaccion date not null,

Monto\_pago number not null,

Fecha\_insercion date not null,

Usuario varchar2 (30) not null

);

------------------------------------------------------------------------------------- LAB7

create table Tipo\_ahorro (

ID\_ahorro number not null,

ahorro varchar2(20) not null,

interes\_ahorro number not null,

constraint ID\_ahorro\_pk primary key (ID\_ahorro)

);

create table Sucursal\_ahorro (

Cod\_sucursal number not null,

ID\_ahorro number not null,

montoahorrado number not null,

constraint Sucursal\_ahorro\_fk foreign key (Cod\_sucursal) references Sucursal (Cod\_sucursal),

constraint Id\_ah\_suc\_fk foreign key (ID\_ahorro) references Tipo\_ahorro (ID\_ahorro),

constraint suc\_ahorro\_pk primary key (Cod\_sucursal, ID\_ahorro)

);

create table Ahorros (

Cod\_sucursal number not null,

ID\_Cliente number not null,

ID\_ahorro number not null,

Num\_ahorro number not null,

fecha\_apertura date not null,

letra\_dep number not null,

saldo\_ahorro number not null,

saldo\_interes number not null,

fecha\_deposito date,

fecha\_retiro date,

usuario varchar2(20) not null,

fecha\_modif date not null,

constraint Suc\_ahorro\_fk foreign key (Cod\_sucursal) references Sucursal (Cod\_sucursal),

constraint ID\_aho\_ahor\_fk foreign key (Id\_ahorro) references Tipo\_ahorro (ID\_ahorro),

constraint ID\_cliente\_aho\_fk foreign key (ID\_cliente) references Cliente (ID\_cliente),

constraint Cliente\_ahorro\_pk primary key (Num\_ahorro)

);

create table Tipo\_transaccion (

ID\_tipo\_transc number not null,

transaccion varchar2(20) not null,

constraint ID\_transac\_pk primary key (ID\_tipo\_transc)

);

create table Auditoria (

Id\_transaccion number not null,

Tabla varchar2(20) not null,

Id\_cliente number not null,

Id\_ahorro number not null,

Id\_tipo\_transc number not null,

saldo\_anterior number not null,

monto\_dr number not null,

saldo\_final number not null,

usuario varchar2(20) not null,

fecha date not null,

constraint aud\_id\_cl\_fk foreign key (ID\_cliente) references Cliente (ID\_Cliente),

constraint aud\_id\_ahorro\_fk foreign key (ID\_ahorro) references Tipo\_ahorro (ID\_ahorro),

constraint aud\_id\_transc\_fk foreign key (ID\_tipo\_transc) references Tipo\_transaccion (ID\_tipo\_transc),

constraint aud\_id\_pk primary key (Id\_transaccion)

);

create table Transadeporeti (

Cod\_sucursal number not null,

id\_transaccion number not null,

id\_cliente number not null,

num\_ahorro number not null,

id\_ahorro number not null,

fechatransaccion date not null,

id\_tipo\_transc number not null,

monto\_dr number not null,

fechainsercion date not null,

usuario varchar2(20) not null,

constraint cod\_suc\_tran\_fk foreign key (Cod\_sucursal) references Sucursal (Cod\_sucursal),

constraint id\_cl\_tran\_fk foreign key (ID\_Cliente) references Cliente (ID\_Cliente),

constraint id\_ahor\_tran\_fk foreign key (id\_ahorro) references Tipo\_ahorro (id\_ahorro),

constraint id\_tipo\_tran\_fk foreign key (id\_tipo\_transc) references Tipo\_transaccion (id\_tipo\_transc),

constraint num\_ahorro\_fk foreign key (num\_ahorro) references Ahorros (num\_ahorro),

constraint id\_tran\_pk primary key (id\_transaccion)

);

create table Auditoria\_saldo (

Id\_transaccion number,

Tabla varchar2(20),

accion char,

Id\_cliente number,

Id\_ahorro number,

saldo\_anterior number,

saldo\_final number,

usuario varchar2(20),

fecha date

);

create table Auditoria\_transac (

Id\_transaccion number,

Tabla varchar2(20),

accion char,

Id\_cliente number,

Id\_ahorro number,

id\_tipo\_transc number,

monto\_dr number,

usuario varchar2(20),

fecha date

);

create table temp\_suc(

upd\_suc number,

cod\_sucursal number,

id\_ahorro number,

monto\_aho\_v number,

monto\_aho\_n number,

usuario varchar2(20),

fecha date

);

**SECUENCIAS**

CREATE SEQUENCE id\_transaccion

start with 1

increment by 1;

CREATE SEQUENCE num\_cuenta

START WITH 1

INCREMENT BY 1;

create sequence trig\_suc

start with 1

increment by 1;

create sequence trig\_transac

start with 1

increment by 1;

create sequence trig\_aud

start with 1

increment by 1;

**FUNCIONES**

CREATE OR REPLACE FUNCTION calc\_edad (fecha\_nac CLIENTE.fecha\_nac%TYPE)

RETURN NUMBER AS

BEGIN

RETURN FLOOR (months\_between (sysdate, fecha\_nac)/12);

END calc\_edad;

/

create or replace function c\_interes (

p\_saldo Prestamo.saldo\_actual%TYPE,

p\_interes Tipo\_prestamo.tasa\_interes%TYPE)

RETURN NUMBER AS

BEGIN

return p\_saldo\*p\_interes;

END c\_interes;

/

CREATE OR REPLACE FUNCTION interes\_deposito (

p\_monto\_dep transadeporeti.monto\_dr%TYPE,

p\_interes tipo\_ahorro.interes\_ahorro%TYPE )

RETURN NUMBER AS

BEGIN

RETURN (p\_monto\_dep \* p\_interes)/100;

END interes\_deposito;

/

create or replace FUNCTION interes\_corriente (

p\_montod ahorros.saldo\_ahorro%TYPE,

p\_interes tipo\_ahorro.interes\_ahorro%TYPE)

RETURN NUMBER AS

BEGIN

RETURN (p\_montod \* p\_interes)/100;

END interes\_corriente;

/

**TRIGGERS**

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

ON Ahorros FOR EACH ROW

BEGIN

IF INSERTING THEN

INSERT INTO Temp\_suc (upd\_suc, cod\_sucursal, id\_ahorro, monto\_aho\_v, monto\_aho\_n, usuario, fecha)

VALUES (trig\_suc.nextval, :new.cod\_sucursal, :new.id\_ahorro, :old.saldo\_ahorro, :new.saldo\_ahorro, user, sysdate);

ELSIF UPDATING THEN

UPDATE Temp\_suc

SET

Upd\_suc= trig\_suc.nextval,

Cod\_sucursal = :NEW.cod\_sucursal,

Id\_ahorro = :new.id\_ahorro,

monto\_aho\_n = :old.saldo\_ahorro +(:new.saldo\_ahorro- :old.saldo\_ahorro),

monto\_aho\_v = :OLD.saldo\_ahorro,

usuario = user,

fecha = sysdate

where id\_ahorro = :new.id\_ahorro AND cod\_sucursal = :new.cod\_sucursal;

END IF;

END suc\_trig;

/

CREATE OR REPLACE TRIGGER Aud\_trig\_saldo

BEFORE UPDATE OR DELETE OR INSERT ON Ahorros

FOR EACH ROW

DECLARE

v\_TipoCambio CHAR(1);

BEGIN

IF UPDATING THEN

v\_TipoCambio := 'U';

INSERT INTO AUDITORIA\_saldo

VALUES (trig\_aud.nextval, 'Ahorros', v\_tipoCambio, :NEW.ID\_cliente, :NEW.id\_ahorro,:OLD.saldo\_ahorro,:NEW.saldo\_ahorro, user, sysdate);

ELSIF INSERTING THEN

v\_TipoCambio := 'I';

INSERT INTO AUDITORIA\_saldo

VALUES (trig\_aud.nextval, 'Ahorros', v\_tipoCambio, :NEW.ID\_cliente, :NEW.id\_ahorro,:OLD.saldo\_ahorro,:NEW.saldo\_ahorro, user, sysdate);

ELSIF DELETING THEN

v\_TipoCambio := 'D';

INSERT INTO AUDITORIA\_saldo

VALUES (trig\_aud.nextval, 'Ahorros', v\_tipoCambio, :NEW.ID\_cliente, :NEW.id\_ahorro,:OLD.saldo\_ahorro,:NEW.saldo\_ahorro, user, sysdate);

END IF;

END Aud\_trig\_saldo;

/

CREATE OR REPLACE TRIGGER Aud\_trig\_transac

BEFORE UPDATE OR DELETE OR INSERT ON transadeporeti

FOR EACH ROW

DECLARE

v\_TipoCambio CHAR(1);

BEGIN

IF UPDATING THEN

v\_TipoCambio := 'U';

INSERT INTO AUDITORIA\_transac

VALUES (trig\_transac.nextval, 'Ahorros', v\_tipoCambio, :NEW.ID\_cliente, :NEW.id\_ahorro,:new.id\_tipo\_transc, :new.monto\_dr, user, sysdate);

ELSIF INSERTING THEN

v\_TipoCambio := 'I';

INSERT INTO AUDITORIA\_transac

VALUES (trig\_transac.nextval, 'Ahorros', v\_tipoCambio, :NEW.ID\_cliente, :NEW.id\_ahorro,:new.id\_tipo\_transc, :new.monto\_dr, user, sysdate);

ELSIF DELETING THEN

v\_TipoCambio := 'D';

INSERT INTO AUDITORIA\_transac

VALUES (trig\_transac.nextval, 'Ahorros', v\_tipoCambio, :NEW.ID\_cliente, :NEW.id\_ahorro,:new.id\_tipo\_transc, :new.monto\_dr, user, sysdate);

END IF;

END Aud\_trig\_transac;

/

**PROCEDIMIENTOS**

CREATE OR REPLACE PROCEDURE add\_email (

p\_id\_email IN email.id\_email%TYPE,

p\_tipo\_email IN email.tipo\_email%TYPE

) AS

BEGIN

INSERT INTO email (id\_email, tipo\_email)

VALUES (p\_id\_email, p\_tipo\_email);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

COMMIT;

END add\_email;

/

CREATE OR REPLACE PROCEDURE add\_telefono (

p\_id\_telefono IN telefono.id\_telefono%TYPE,

p\_tipo\_telefono IN telefono.tipo\_telefono%TYPE

) AS

BEGIN

INSERT INTO telefono (id\_telefono, tipo\_telefono)

VALUES (p\_id\_telefono, p\_tipo\_telefono);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

COMMIT;

END add\_telefono;

/

CREATE OR REPLACE PROCEDURE add\_profesion (

p\_id\_pro IN profesion.id\_pro%TYPE,

p\_profesion IN profesion.profesion%TYPE

) AS

BEGIN

INSERT INTO profesion (id\_pro, profesion)

VALUES (p\_id\_pro, p\_profesion);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

COMMIT;

END add\_profesion;

/

CREATE OR REPLACE PROCEDURE add\_tipo\_prestamo (

p\_id\_tipo IN tipo\_prestamo.id\_tipo%TYPE,

p\_tipo IN tipo\_prestamo.tipo%TYPE,

p\_tasa\_interes IN tipo\_prestamo.tasa\_interes%TYPE

) AS

BEGIN

INSERT INTO tipo\_prestamo (id\_tipo, tipo, tasa\_interes)

VALUES (p\_id\_tipo, p\_tipo, p\_tasa\_interes);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

COMMIT;

END add\_tipo\_prestamo;

/

CREATE OR REPLACE PROCEDURE add\_sucursales (

p\_cod\_sucursal IN sucursal.cod\_sucursal%TYPE,

p\_nombre\_sucursal IN sucursal.nombre\_sucursal%TYPE

) AS

BEGIN

INSERT INTO sucursal (cod\_sucursal, nombre\_sucursal)

VALUES (p\_cod\_sucursal, p\_nombre\_sucursal);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontró la tabla');

COMMIT;

END add\_sucursales;

/

CREATE OR REPLACE PROCEDURE add\_cliente (

p\_id\_cliente IN cliente.id\_cliente%TYPE,

p\_id\_pro IN cliente.id\_pro%TYPE,

p\_cedula IN cliente.cedula%TYPE,

p\_nombre IN cliente.nombre%TYPE,

p\_apellido IN cliente.apellido%TYPE,

p\_sexo IN cliente.sexo%TYPE,

p\_fecha\_nac IN cliente.fecha\_nac%TYPE,

p\_cod\_sucursal IN cliente.cod\_sucursal%TYPE,

p\_error OUT varchar2

) AS

P\_edad cliente.edad%TYPE;

BEGIN

P\_error := 'Registro Creado';

P\_edad := calc\_edad(p\_fecha\_nac);

INSERT INTO cliente (id\_cliente, id\_pro, cedula, nombre, apellido, sexo, fecha\_nac, edad, cod\_sucursal)

VALUES (p\_id\_cliente,p\_id\_pro, p\_cedula, p\_nombre, p\_apellido, p\_sexo, p\_fecha\_nac, p\_edad, p\_cod\_sucursal);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

P\_error := 'Cliente existente';

WHEN OTHERS THEN

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

COMMIT;

END add\_cliente;

/

CREATE OR REPLACE PROCEDURE add\_email\_cliente (

p\_id\_email IN email\_cliente.id\_email%TYPE,

p\_id\_cliente IN email\_cliente.id\_cliente%TYPE,

p\_email IN email\_cliente.email%TYPE

) AS

BEGIN

INSERT INTO email\_cliente(id\_email, id\_cliente, email)

VALUES (p\_id\_email, p\_id\_cliente, p\_email);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontró la tabla');

COMMIT;

END add\_email\_cliente;

/

CREATE OR REPLACE PROCEDURE add\_telefono\_cliente (

p\_id\_telefono IN telefono\_cliente.id\_telefono%TYPE,

p\_id\_cliente IN telefono\_cliente.id\_cliente%TYPE,

p\_telefono IN telefono\_cliente.telefono%TYPE

) AS

BEGIN

INSERT INTO telefono\_cliente(id\_telefono, id\_cliente, telefono)

VALUES (p\_id\_telefono, p\_id\_cliente, p\_telefono);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontró la tabla');

COMMIT;

END add\_telefono\_cliente;

/

CREATE OR REPLACE PROCEDURE prestamos\_aprob(

P\_ID\_Tipo IN Prestamo.ID\_TIPO%TYPE,

P\_ID\_CLIENTE IN Prestamo.ID\_CLIENTE%TYPE,

P\_NUM\_PRESTAMO IN Prestamo.NUM\_PRESTAMO%TYPE,

P\_FECHA\_APROB IN Prestamo.FECHA\_APROB%TYPE,

P\_MONTO\_APROB IN Prestamo.MONTO\_APROB%TYPE,

P\_LETRA\_MENSUAL IN Prestamo.LETRA\_MENSUAL%TYPE,

P\_MONTO\_PAGO IN Prestamo.MONTO\_PAGO%TYPE,

P\_FECHA\_PAGO IN Prestamo.FECHA\_PAGO%TYPE,

P\_COD\_SUCURSAL IN Prestamo.COD\_SUCURSAL%TYPE,

P\_SALDO\_ACTUAL IN Prestamo.SALDO\_ACTUAL%TYPE,

P\_INTERES\_PAGADO IN Prestamo.INTERES\_PAGADO%TYPE,

P\_FECHA\_MODIF IN Prestamo.FECHA\_MODIF%TYPE,

P\_USUARIO IN Prestamo.USUARIO%TYPE

)AS

BEGIN

INSERT INTO Prestamo (ID\_TIPO, ID\_CLIENTE, NUM\_PRESTAMO, FECHA\_APROB, MONTO\_APROB, LETRA\_MENSUAL, MONTO\_PAGO, FECHA\_PAGO, COD\_SUCURSAL, SALDO\_ACTUAL, INTERES\_PAGADO, FECHA\_MODIF, USUARIO)

VALUES(P\_ID\_Tipo, P\_ID\_CLIENTE, P\_NUM\_PRESTAMO, P\_FECHA\_APROB, P\_MONTO\_APROB, P\_LETRA\_MENSUAL, P\_MONTO\_PAGO, P\_FECHA\_PAGO, P\_COD\_SUCURSAL, P\_SALDO\_ACTUAL, P\_INTERES\_PAGADO, P\_FECHA\_MODIF, P\_USUARIO);

UPDATE Sucursal\_Prestamo

SET

Monto\_prestamo = Monto\_prestamo + p\_monto\_aprob

where ID\_tipo = P\_ID\_TIPO AND Cod\_sucursal = P\_COD\_SUCURSAL;

IF SQL%ROWCOUNT=0 THEN

INSERT INTO Sucursal\_prestamo (id\_tipo, cod\_sucursal, monto\_prestamo)

Values (p\_id\_tipo, p\_cod\_sucursal,p\_monto\_aprob );

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontró la tabla');

COMMIT;

END prestamos\_aprob;

/

CREATE OR REPLACE PROCEDURE add\_pagos (

p\_id\_transac IN transact\_pagos.id\_transaccion%TYPE,

p\_cod\_sucursal IN transact\_pagos.cod\_sucursal%TYPE,

p\_id\_cliente IN transact\_pagos.id\_cliente%TYPE,

p\_id\_tipo IN transact\_pagos.id\_tipo%TYPE,

p\_fechatransac IN transact\_pagos.fechatransaccion%TYPE,

p\_monto\_pago IN transact\_pagos.monto\_pago%TYPE,

p\_fecha\_insercion IN transact\_pagos.fecha\_insercion%TYPE,

p\_usuario IN transact\_pagos.usuario%TYPE ) AS

BEGIN

INSERT INTO transact\_pagos (id\_transaccion, cod\_sucursal, id\_cliente, id\_tipo, fechatransaccion, monto\_pago, fecha\_insercion, usuario)

VALUES (p\_id\_transac, p\_cod\_sucursal, p\_id\_cliente, p\_id\_tipo, p\_fechatransac, p\_monto\_pago, p\_fecha\_insercion, p\_usuario);

COMMIT;

END add\_pagos;

/

create or replace procedure update\_pago

AS

CURSOR act\_pago IS

SELECT monto\_pago, cod\_sucursal, id\_tipo, id\_cliente, id\_transaccion

FROM Transact\_pagos;

v\_pago prestamo.interes\_pagado%TYPE;

v\_tasa tipo\_prestamo.tasa\_interes%TYPE;

v\_saldo transact\_pagos.monto\_pago%TYPE;

v\_monto prestamo.monto\_aprob%TYPE;

v\_saldo\_act prestamo.saldo\_actual%TYPE;

v\_apago prestamo.interes\_pagado%TYPE;

BEGIN

FOR i IN act\_pago LOOP

SELECT tasa\_interes into v\_tasa from tipo\_prestamo

where id\_tipo = i.id\_tipo;

SELECT saldo\_actual into v\_saldo\_act from prestamo

where id\_tipo=i.id\_tipo and id\_cliente=i.id\_cliente;

v\_apago :=c\_interes (v\_saldo\_act, v\_tasa);

v\_saldo:= v\_apago-i.monto\_pago;

IF (v\_saldo > 0) THEN

UPDATE PRESTAMO

SET

Monto\_pago = v\_saldo,

Interes\_pagado = v\_apago,

Saldo\_actual = saldo\_actual – v\_saldo,

Fecha\_pago = sysdate,

Usuario= user

WHERE id\_cliente=i.id\_cliente and id\_tipo = i.id\_tipo;

UPDATE SUCURSAL\_PRESTAMO

SET

monto\_prestamo = monto\_prestamo - v\_saldo

where id\_tipo = i.id\_tipo and cod\_sucursal = i.cod\_sucursal;

ELSE

UPDATE PRESTAMO

SET

Monto\_pago = v\_saldo,

Interes\_pagado = v\_apago,

Fecha\_pago = sysdate,

Usuario= user

WHERE

id\_cliente = i.id\_cliente and id\_cliente=i.id\_cliente;

END IF;

END LOOP;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro datos');

COMMIT;

END update\_pago;

/

------------------------------------------------------------------------------------------ Invocaciones

BEGIN

Add\_email (1, 'Personal');

Add\_email (2, 'Académico');

Add\_email (3, 'Laboral');

END;

/

BEGIN

Add\_telefono (1, 'Residencial');

Add\_telefono (2, 'Celular');

Add\_telefono (3, 'Conyuge');

Add\_telefono (4, 'Familiar');

END;

/

BEGIN

add\_tipo\_prestamo (1, 'Personal', 0.07);

add\_tipo\_prestamo (2, 'Auto', 0.0725);

add\_tipo\_prestamo (3, 'Hipoteca', 0.0650);

add\_tipo\_prestamo (4, 'Garantizado con ahorros', 0.055);

END;

/

BEGIN

add\_profesion (1, 'Psicólogo');

add\_profesion (2, 'Abogado');

add\_profesion (3, 'Doctor');

add\_profesion (4, 'Ingeniero en Sistemas');

add\_profesion (5, 'Arquitecto');

add\_profesion (6, 'Ingeniero Industrial');

add\_profesion (7, 'Contable');

add\_profesion (8, 'Veterinario');

add\_profesion (9, 'Chef');

add\_profesion (10,'Policía');

END;

/

BEGIN

add\_sucursales (1, 'Los Andes Mall');

add\_sucursales (2, 'Metromall');

add\_sucursales (3, 'Albrook Mall');

add\_sucursales (4, 'Multi Plaza');

add\_sucursales (5, 'Alta Plaza');

END;

/

BEGIN

add\_pagos(1, 1, 1, 1, to\_date('17/11/2020','dd/mm/yyyy'), 356, sysdate, user);

add\_pagos(2, 2, 2, 2, to\_date('17/11/2020','dd/mm/yyyy'), 400, sysdate, user);

add\_pagos(3, 3, 3, 3, to\_date('17/11/2020','dd/mm/yyyy'), 500, sysdate, user);

add\_pagos(4, 4, 4, 4, to\_date('17/11/2020','dd/mm/yyyy'), 100, sysdate, user);

END;

/

set serveroutput on;

DECLARE

v\_error varchar2 (100);

BEGIN

add\_cliente (1, 1, '8-789-4532', 'Carlos', 'González', 'M', TO\_DATE('13/10/1995', 'DD/MM/YYYY'), 1, v\_error);

add\_cliente (2, 2, '8-925-546', 'Alicia', 'Navarro', 'F', TO\_DATE('10/01/1993','DD/MM/YYYY'), 3, v\_error);

add\_cliente (3, 3, '8-948-754', 'Jorge', 'Perez', 'M', TO\_DATE('27/05/1997','DD/MM/YYYY'), 2, v\_error);

add\_cliente (4, 4, '8-950-479', 'Sonia', 'Ortiz', 'F', TO\_DATE('24/03/1998','DD/MM/YYYY'), 9, v\_error);

add\_cliente (5, 5, '8-960-521', 'Juan', 'Castillo', 'M', TO\_DATE('29/12/1997','DD/MM/YYYY'), 4, v\_error);

DBMS\_OUTPUT.PUT\_LINE (v\_error);

END;

/

BEGIN

add\_email\_cliente (1, 1, 'carlosg06@hotmail.com');

add\_email\_cliente (2, 1, 'carlos.gonzalez@utp.ac.pa');

add\_email\_cliente (1, 2, 'alicianav87@gmail.com');

add\_email\_cliente (1, 3, 'jorge8778@gmail.com');

add\_email\_cliente (1, 5, 'ortizson2554@gmail.com');

add\_email\_cliente (2, 5, 'sonia.ortiz@utp.ac.pa');

END;

/

BEGIN

Add\_telefono\_cliente (1, 1, '268-8564');

Add\_telefono\_cliente (2, 1, '6654-8574');

Add\_telefono\_cliente (1, 2, '352-7584');

Add\_telefono\_cliente (2, 3, '6587-8214');

Add\_telefono\_cliente (1, 3, '268-8795');

Add\_telefono\_cliente (3, 2, '6574-8941');

Add\_telefono\_cliente (4, 3, '6978-3210');

Add\_telefono\_cliente (1, 5, '236-8741');

END;

/

BEGIN

prestamos\_aprob (1, 1, 1, to\_date('12-OCT-20'), 20000, 356,0,TO\_DATE ('17-NOV-20'), 1, 20000, 0, to\_date('18-NOV-20'), user);

prestamos\_aprob (2, 2, 2, to\_date('18-OCT-20'), 30000, 400, 0,to\_date('17-NOV-20'), 2, 30000, 0, to\_date('18-NOV-20'), user);

prestamos\_aprob (3, 3, 3, to\_date('20-OCT-20'), 200000, 500, 0,to\_date('17-NOV-20'), 3, 200000, 0, to\_date('18-NOV-20'), user);

prestamos\_aprob (4, 5, 4, to\_date('21-OCT-20'), 5000, 100, 0,to\_date('17-NOV-20'), 4, 5000, 0, to\_date('18-NOV-20'), user);

END;

/

----------------------------------------------------------------------------------- Procedimientos Lab7

CREATE OR REPLACE PROCEDURE add\_t\_ahorro (

p\_id\_ahorro IN tipo\_ahorro.id\_ahorro%TYPE,

p\_ahorro IN tipo\_ahorro.ahorro%TYPE,

p\_interes IN tipo\_ahorro.interes\_ahorro%TYPE

) AS

BEGIN

INSERT INTO tipo\_ahorro (id\_ahorro, ahorro, interes\_ahorro)

VALUES (p\_id\_ahorro, p\_ahorro, p\_interes);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

COMMIT;

END add\_t\_ahorro;

/

CREATE OR REPLACE PROCEDURE ahorros\_aprob(

p\_cod\_sucursal IN ahorros.cod\_sucursal%TYPE,

P\_ID\_CLIENTE IN ahorros.ID\_CLIENTE%TYPE,

p\_id\_ahorro IN ahorros.id\_ahorro%TYPE,

P\_fecha\_apert IN ahorros.fecha\_apertura%TYPE,

P\_LETRA\_MENSUAL IN ahorros.LETRA\_dep%TYPE,

P\_saldo\_ahorro IN ahorros.saldo\_ahorro%TYPE,

P\_saldo\_interes IN ahorros.saldo\_interes%TYPE,

P\_fecha\_deposito IN ahorros.fecha\_deposito%TYPE,

P\_fecha\_retiro IN ahorros.fecha\_retiro%TYPE

)AS

BEGIN

INSERT INTO Ahorros (cod\_sucursal, ID\_CLIENTE, id\_ahorro, num\_ahorro, fecha\_apertura, LETRA\_dep, saldo\_ahorro, saldo\_interes, fecha\_deposito, fecha\_retiro, usuario, FECHA\_MODIF)

VALUES(P\_cod\_sucursal, P\_ID\_CLIENTE, P\_id\_ahorro, num\_cuenta.nextval, P\_fecha\_apert, P\_LETRA\_MENSUAL, P\_saldo\_ahorro, P\_saldo\_interes, P\_fecha\_deposito, P\_fecha\_retiro, user, sysdate);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontró la tabla');

COMMIT;

END ahorros\_aprob;

/

create or replace PROCEDURE add\_transadeporeti (

p\_cod\_sucursal IN transadeporeti.cod\_sucursal%TYPE,

p\_id\_cliente IN transadeporeti.id\_cliente%TYPE,

p\_num\_ahorro IN transadeporeti.num\_ahorro%TYPE,

p\_id\_ahorro IN transadeporeti.id\_ahorro%TYPE,

p\_id\_tipo\_transc IN transadeporeti.id\_tipo\_transc%TYPE,

p\_monto\_dr IN transadeporeti.monto\_dr%TYPE

)AS

BEGIN

INSERT INTO transadeporeti (cod\_sucursal, id\_transaccion, id\_cliente, num\_ahorro, id\_ahorro, fechatransaccion, id\_tipo\_transc, monto\_dr, fechainsercion, usuario)

VALUES (p\_cod\_sucursal, id\_transaccion.nextval, p\_id\_cliente, p\_num\_ahorro, p\_id\_ahorro, sysdate, p\_id\_tipo\_transc, p\_monto\_dr, sysdate, user);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

COMMIT;

END add\_transadeporeti;

/

CREATE OR REPLACE PROCEDURE ahorros\_aprob(

p\_cod\_sucursal IN ahorros.cod\_sucursal%TYPE,

P\_ID\_CLIENTE IN ahorros.ID\_CLIENTE%TYPE,

p\_id\_ahorro IN ahorros.id\_ahorro%TYPE,

P\_fecha\_apert IN ahorros.fecha\_apertura%TYPE,

P\_LETRA\_MENSUAL IN ahorros.LETRA\_dep%TYPE,

P\_saldo\_ahorro IN ahorros.saldo\_ahorro%TYPE,

P\_saldo\_interes IN ahorros.saldo\_interes%TYPE,

P\_fecha\_deposito IN ahorros.fecha\_deposito%TYPE,

P\_fecha\_retiro IN ahorros.fecha\_retiro%TYPE

)AS

BEGIN

INSERT INTO Ahorros (cod\_sucursal, ID\_CLIENTE, id\_ahorro, num\_ahorro, fecha\_apertura, LETRA\_dep, saldo\_ahorro, saldo\_interes, fecha\_deposito, fecha\_retiro, usuario, FECHA\_MODIF)

VALUES(P\_cod\_sucursal, P\_ID\_CLIENTE, P\_id\_ahorro, num\_cuenta.nextval, P\_fecha\_apert, P\_LETRA\_MENSUAL, P\_saldo\_ahorro, P\_saldo\_interes, P\_fecha\_deposito, P\_fecha\_retiro, user, sysdate);

UPDATE Sucursal\_ahorro

SET

montoahorrado=montoahorrado+p\_saldo\_ahorro

WHERE

id\_ahorro=p\_id\_ahorro and cod\_sucursal=p\_cod\_sucursal;

IF SQL%ROWCOUNT=0 THEN

INSERT INTO Sucursal\_ahorro (id\_ahorro, cod\_sucursal, montoahorrado)

Values (p\_id\_ahorro, p\_cod\_sucursal,p\_saldo\_ahorro);

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontró la tabla');

COMMIT;

END ahorros\_aprob;

/

create or replace PROCEDURE update\_dr

AS

CURSOR c\_depret IS

SELECT cod\_sucursal, id\_cliente, id\_ahorro, monto\_dr, id\_tipo\_transc, num\_ahorro

from Transadeporeti;

p\_sucursal transadeporeti.cod\_sucursal%TYPE;

p\_cliente transadeporeti.id\_cliente%TYPE;

p\_transac transadeporeti.id\_transaccion%TYPE;

p\_tipotrans transadeporeti.id\_tipo\_transc%TYPE;

p\_montod transadeporeti.monto\_dr%TYPE;

p\_funcion ahorros.saldo\_interes%TYPE;

p\_tipoa transadeporeti.id\_ahorro%TYPE;

p\_interes tipo\_ahorro.interes\_ahorro%TYPE;

p\_saldo\_actual ahorros.saldo\_ahorro%TYPE;

p\_num\_ahorro transadeporeti.num\_ahorro%TYPE;

BEGIN

OPEN c\_depret;

LOOP

FETCH c\_depret INTO p\_sucursal, p\_cliente, p\_tipoa, p\_montod, p\_tipotrans, p\_num\_ahorro;

SELECT interes\_ahorro INTO p\_interes

FROM tipo\_ahorro WHERE id\_ahorro = p\_tipoa;

SELECT saldo\_ahorro INTO p\_saldo\_actual FROM Ahorros WHERE id\_ahorro = p\_tipoa and id\_cliente =p\_cliente and cod\_sucursal=p\_sucursal;

IF ((p\_tipoa = 1 OR p\_tipoa = 3) AND p\_tipotrans = 1) THEN

p\_funcion := interes\_deposito (p\_montod, p\_interes);

UPDATE Ahorros

SET

saldo\_ahorro =saldo\_ahorro+p\_montod+p\_funcion,

saldo\_interes = saldo\_interes+p\_funcion,

fecha\_deposito = sysdate,

usuario = user,

fecha\_modif = sysdate

WHERE id\_cliente = p\_cliente and num\_ahorro=p\_num\_ahorro;

ELSIF (p\_tipoa = 2 AND p\_tipotrans = 1) THEN

UPDATE Ahorros

SET

saldo\_ahorro =saldo\_ahorro+p\_montod,

fecha\_deposito = sysdate,

usuario = user,

fecha\_modif = sysdate

WHERE id\_cliente = p\_cliente and num\_ahorro=p\_num\_ahorro;

ELSIF ((p\_tipoa=1 OR p\_tipoa=3) AND p\_tipotrans=2) THEN

DBMS\_OUTPUT.PUT\_LINE ('NO SE PUEDE RETIRAR DE ESTA CUENTA');

ELSIF (p\_tipoa = 2 AND p\_tipotrans = 2 )THEN

IF (p\_saldo\_actual >= p\_montod) THEN

UPDATE Ahorros

SET

saldo\_ahorro =saldo\_ahorro-p\_montod,

fecha\_retiro = sysdate,

usuario = user,

fecha\_modif = sysdate

WHERE id\_cliente = p\_cliente and num\_ahorro=p\_num\_ahorro;

ELSE

DBMS\_OUTPUT.PUT\_LINE ('No hay suficientes fondos');

END IF;

END IF;

EXIT WHEN c\_depret%NOTFOUND;

END LOOP;

CLOSE c\_depret;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontró los datos');

COMMIT;

END update\_dr;

/

CREATE OR REPLACE PROCEDURE update\_corriente

AS

CURSOR c\_corriente IS

SELECT id\_cliente, id\_ahorro, saldo\_ahorro, num\_ahorro

FROM Ahorros where id\_ahorro=2;

v\_cliente Ahorros.id\_cliente%TYPE;

v\_ahorro Ahorros.id\_ahorro%TYPE;

v\_saldoa Ahorros.saldo\_ahorro%TYPE;

v\_funcion Ahorros.saldo\_interes%TYPE;

v\_interes Tipo\_ahorro.interes\_ahorro%TYPE;

v\_num Ahorros.num\_ahorro%TYPE;

BEGIN

OPEN c\_corriente;

LOOP

FETCH c\_corriente INTO v\_cliente, v\_ahorro, v\_saldoa, v\_num;

SELECT interes\_ahorro INTO v\_interes

FROM tipo\_ahorro WHERE id\_ahorro = v\_ahorro;

v\_funcion := interes\_corriente (v\_saldoa, v\_interes);

UPDATE Ahorros

SET

saldo\_ahorro = saldo\_ahorro + v\_funcion,

saldo\_interes = saldo\_interes + v\_funcion,

usuario = user,

fecha\_modif = sysdate

WHERE id\_cliente = v\_cliente AND num\_ahorro = v\_num;

EXIT WHEN c\_corriente%NOTFOUND;

END LOOP;

CLOSE c\_corriente;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontró los datos');

COMMIT;

END update\_corriente;

/

CREATE OR REPLACE PROCEDURE add\_t\_transac (

p\_id\_tipo IN tipo\_transaccion.id\_tipo\_transc%TYPE,

p\_transac IN tipo\_transaccion.transaccion%TYPE

) AS

BEGIN

INSERT INTO tipo\_transaccion (id\_tipo\_transc, transaccion)

VALUES (p\_id\_tipo, p\_transac);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

COMMIT;

END add\_t\_transac;

/

-------------------------------------------------------------------------- Invocaciones

(aquí copiar 1 por 1)

BEGIN

ADD\_T\_AHORRO (01,'Navidad', 6);

ADD\_T\_AHORRO (02,'Corriente',4);

ADD\_T\_AHORRO (03,'Escolar', 6);

END;

/

BEGIN

ADD\_T\_TRANSAC (1, 'Depósito');

ADD\_T\_TRANSAC (2, 'Retiro');

END;

/

BEGIN

AHORROS\_APROB (1,1,1,TO\_DATE('27-OCT-20'),100,0,0,NULL,NULL);

AHORROS\_APROB (2,2,2,TO\_DATE('31-OCT-20'),50,0,0,NULL,NULL);

AHORROS\_APROB (3,3,3,TO\_DATE('01-NOV-20'),200,0,0,NULL,NULL);

AHORROS\_APROB (4,5,1,TO\_DATE('01-NOV-20'),100,0,0,NULL,NULL);

AHORROS\_APROB (3,2,2,TO\_DATE('05-NOV-20'),200,0,0,NULL,NULL);

END;

/

select \* from ahorros;

BEGIN

ADD\_TRANSADEPORETI (1,1,1,1,1,100);

ADD\_TRANSADEPORETI (2,2,2,2,1,100);

ADD\_TRANSADEPORETI (3,3,3,3,1,200);

ADD\_TRANSADEPORETI (4,5,4,1,1,100);

ADD\_TRANSADEPORETI (3,2,5,2,1,200);

END;

/

BEGIN

update\_dr;

END;

/

select \* from ahorros;

BEGIN

update\_corriente;

END;

/

select \* from ahorros;

select \* from temp\_suc;

select \* from auditoria\_saldo;

select \* from auditoria\_transac;

**VISTAS Y SELECCIONES**

create view CLIENTE\_TIPO as select c.nombre, c.apellido, c.cedula, t.tipo from cliente c join prestamo p on p.id\_cliente = c.id\_cliente join tipo\_prestamo t on t.id\_tipo = p.id\_tipo;

create view PROFESION\_CLIENTE as select p.profesion, c.nombre, c.apellido from cliente c join profesion p on c.id\_pro = p.id\_pro;

create view Cliente\_monto as select c.nombre, c.apellido, p.monto\_aprob, p.letra\_mensual from cliente c join prestamo p on c.id\_cliente = p.id\_cliente;

create view CLIENTE\_AHORRO as select c.nombre, c.apellido, c.cedula, t.ahorro from cliente c join ahorros a on a.id\_cliente = c.id\_cliente join tipo\_ahorro t on t.id\_ahorro = a.id\_ahorro;

create view Cliente\_montoah as select c.nombre, c.apellido, a.saldo\_ahorro from cliente c join ahorros a on c.id\_cliente = a.id\_cliente;

select \* from cliente\_tipo;

select \* from profesion\_cliente;

select \* from cliente\_monto;

select \* from ahorros;

select \* from cliente;

select \* from email;

select \* from email\_cliente;

select \* from prestamo;

select \* from profesion;

select \* from sucursal;

select \* from sucursal\_ahorro;

select \* from sucursal\_prestamo;

select \* from telefono;

select \* from telefono\_cliente;

select \* from tipo\_ahorro;

select \* from tipo\_prestamo;

select \* from tipo\_transaccion;

select \* from transact\_pagos;

select \* from transadeporeti;