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

Equipo #1

Laboratorio #6: “Implementación de Programación Almacenada de Base de Datos”

Facilitador:

Ing. Henry Lezcano

Integrantes:

Aguilar, Milagros 3-740-771

Atencio, Anel 8-950-868

Márquez, Paola 8-949-1108

Rojas, Reynaldo 8-950-792

Grupo:

1IF131

II Semestre, 2020

**-- Agregar el atributo** edad **a la tabla de clientes**

alter table CLIENTE

add edad number not null;

**--Tabla sucursal**

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

);

------------------------------------------------------------------------------------------------

**-- Agregar a la tabla clientes y préstamos**

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));

--------------------------------------------------------------------------------------------

**-- Agregar a la tabla del prestamos**

alter table PRESTAMO

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

**---------------------------------------------------------------------------------------------**

**-- Agregar una tabla transaccional para recibir los pagos de los clientes**

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

);

--------------------------------------------------------------------------------------------

**--Procedimientos de tablas paramétricas**

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;

/

**--Funcion para calcular edad**

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;

/

**--Procedimiento de tabla Cliente**

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;

/

---------------------------------------------------------------------------------------

**--Procedimiento tablas Prestamo y Sucursal\_prestamo**

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;

/

---------------------------------------------------------------------------

**--Procedimiento tabla transact\_pagos**

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;

/

---------------------------------------------------------------------------------------------------

**--Funcion calcular interes**

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;

/

-----------------------------------------------------------------------------------------------

**--Procedimiento actualizar 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**

**--Email**

BEGIN

Add\_email (1, 'Personal');

Add\_email (2, 'Académico');

Add\_email (3, 'Laboral');

END;

/

---------------------------------------------------------------------

**--Telefono**

BEGIN

Add\_telefono (1, 'Residencial');

Add\_telefono (2, 'Celular');

Add\_telefono (3, 'Conyuge');

Add\_telefono (4, 'Familiar');

END;

/

-------------------------------------------------------------

**--Tipo prestamo**

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;

/

---------------------------------------------------------------------------

**--Profesion**

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;

/

**--Sucursal**

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;

/

------------------------------------------------------------------

**--Transaccion**

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;

/

-----------------------------------------------------------------------------

**--Cliente**

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;

/

**--email\_cliente**

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;

/

---------------------------------------------------------------------------------

**--telefono\_cliente**

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;

/

**--prestamo**

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;

/

-----------------------------------------------------------------------

**--Vistas**

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;

**--Secuencias**

CREATE SEQUENCE email\_sequence

START WITH 1

INCREMENT BY 1;

CREATE SEQUENCE telefono\_sequence

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE profesion\_sequence

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE tipo\_prestamo\_sequence

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE cliente\_sequence

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE pagos\_sequence

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE sucursales\_sequence

START WITH 1

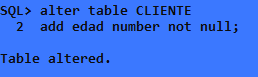
INCREMENT BY 1

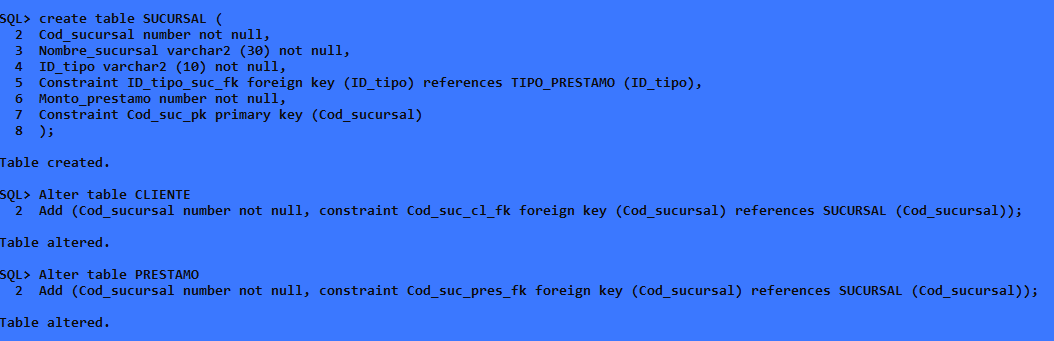
NOCACHE

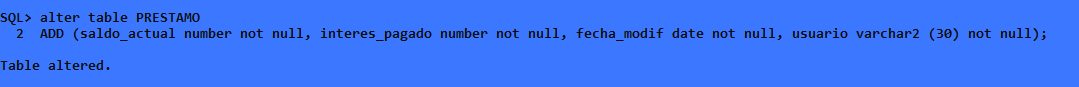
NOCYCLE;

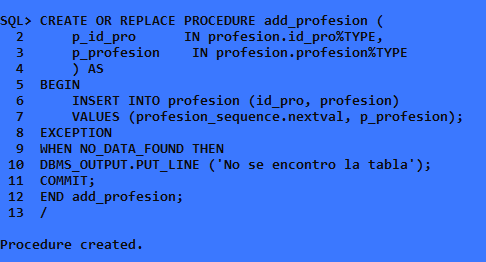
**Capturas de pantalla**

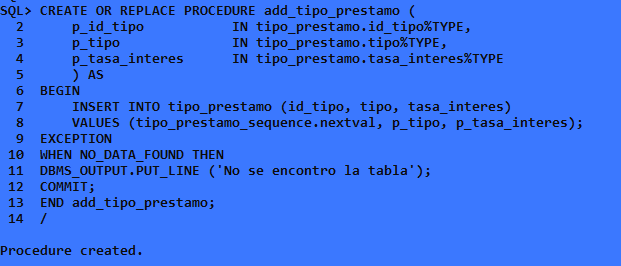
**Procedimientos, creaciones y funciones**

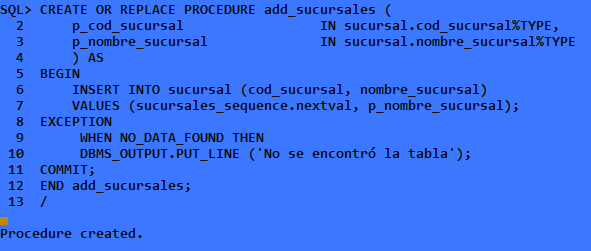


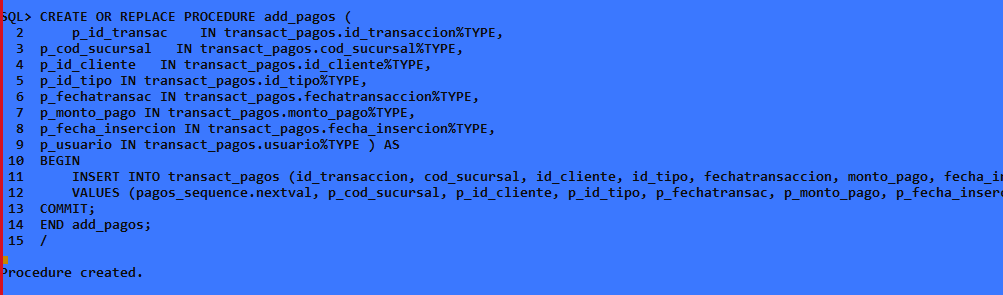


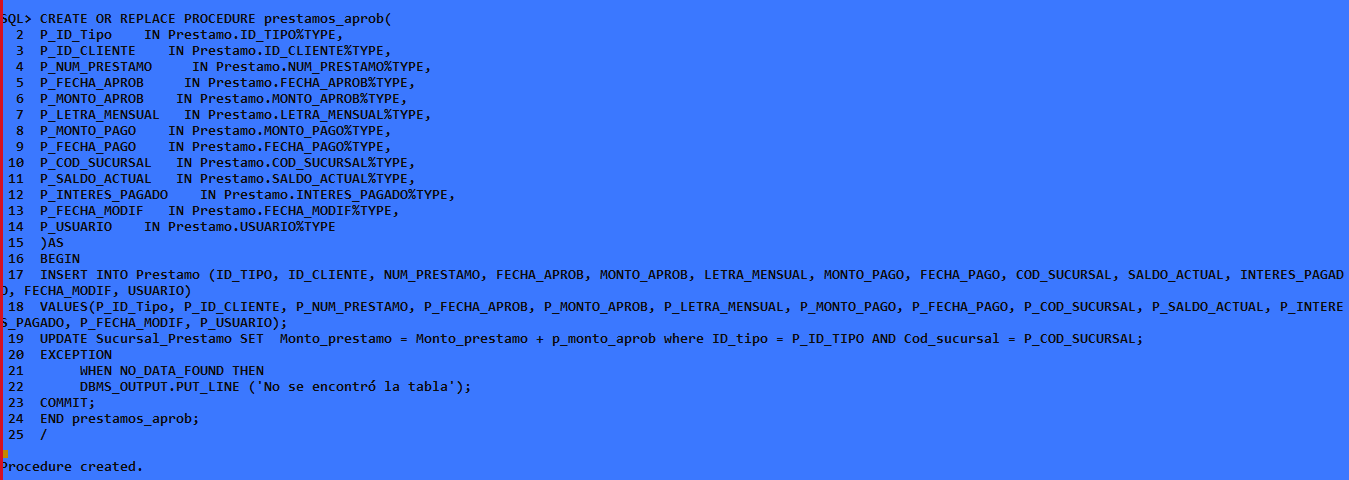


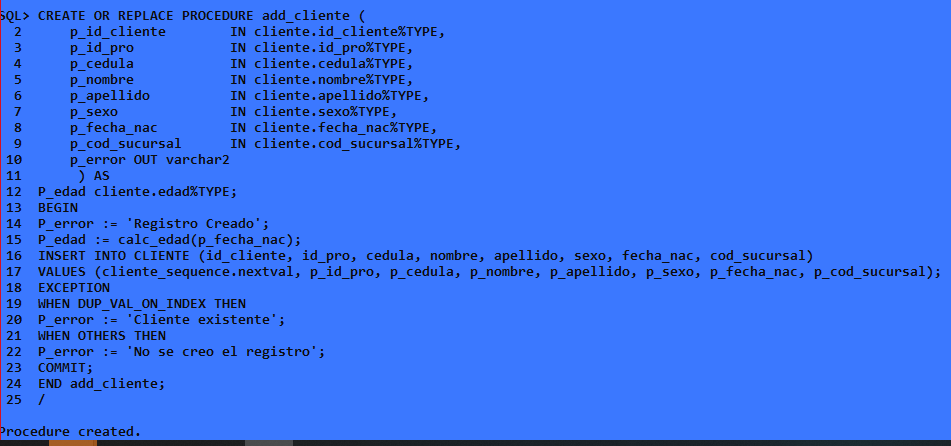


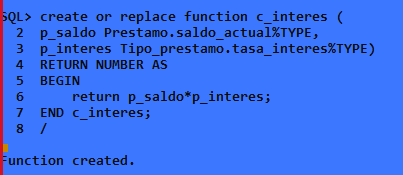




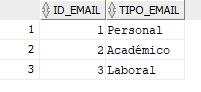


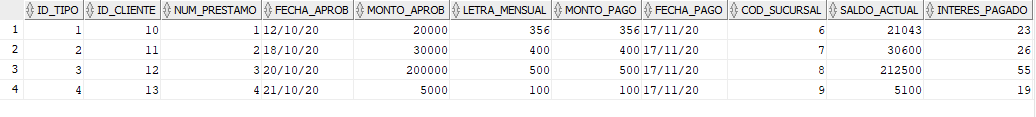




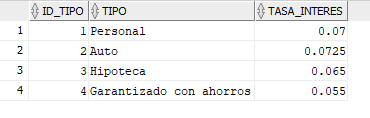


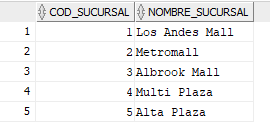
**Invocaciones**

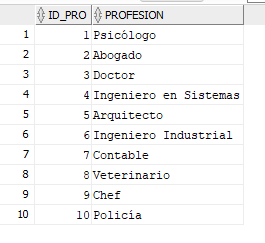












**Vistas**

