



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS
COMPUTACIONALES



DEPARTAMENTO DE SISTEMAS DE
INFORMACIÓN,
CONTROL Y EVALUACIÓN DE RECURSOS INFORMÁTICOS

LICENCIATURA EN INGENIERÍA DE SISTEMAS DE INFORMACIÓN
SISTEMAS DE BASES DE DATOS II.

LABORATORIO 6

Profesor
Henry Lezcano

Estudiantes

Samudio, Nedith 8-968-1471

Sánchez, Ana 8-967-832

Solis, Michael 8-958-1219

Tejada, Ángel 8-969-974

Urriola, Vicente 8-892-2296

Grupo
1IF131

UNIVERSIDAD TECNOLÓGICA DE PANAMA
FACULTAD DE INGENIERIA DE SISTEMAS COMPUTACIONALES
LICENCIATURA EN INGENIERIA DE SISTEMAS DE INFORMACIONN

SISTEMAS DE BASE DE DATOS II

LABORATORIO 6

FUNDAMENTOS DEL LENGUAJE-CURSORES-PROCEDIMIENTOS Y FUNCIONES

APLICAR PROGRAMACION ALMACENADA DE BASE DE DATOS-PL/SQLORACLE

Tomando en cuenta lo proporcionado por la compañía Financiera Márquez-Cedeño, S.A del Laboratorio No.5

'Una Compañía financiera necesita levantar un modelo físico de datos a partir de un modelo lógico relacional para una sección de su proceso de negocio (otorgamiento de préstamos), en lo corresponde a la entrada de la información de los cliente y los préstamos e información relacionada a esta sección del proceso a dicha base de datos'

- I. **Realizar las siguientes tareas que afectan el Modelo Físico implementado previamente que permitirá optimizar la sección del proceso de negocio implementado 'Proceso de Prestamos' ya que se han aperturado varias sucursales, se han aprobado préstamos y se necesita iniciar con el proceso retorno de la cartera prestada:**

1. Agregar el atributo **edad** a la tabla de clientes proporcionada.
2. Agregar una tabla al modelo físico que almacenes las sucursales de la empresa la financiera con las restricciones correspondientes:
Sucursal (cod_sucursal, nombresucursal, montoprestamo)
importante conocer que cada sucursales tiene la autoridad de otorgar muchos tipos de préstamos, y cada tipo préstamo puede ser otorgado por muchas sucursales
3. Agregar a la tabla clientes y préstamos el atributo de **código de sucursal**.
4. 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**)

5. Agregar una tabla transaccional para recibir los pagos de los clientes con las restricciones correspondiente que contendrá la siguiente atributos:

Transacpagos (cod_sucursal, id_transaccion, id_cliente, tipoprestamo, fechatransaccion, monto del pago, estado char, fechainsercion(dd/mm/yyyy, hh:mm:ss, usuario)

Todo los campos id que son infinitos en las tablas de la base de datos deben ser controlados por una secuencia (id_cliente, numero de prestamos, id_transaccion)

Importante en el Modelo Físico validar la Integridad referencial para los nuevos atributos de ser necesarios y para las nuevas estructuras que formaran parte del Modelo Físico Optimizado. Esto cambios impactan el modelo Lógico Relacional de la Base de Datos.

Procesos que deben ser implementado con la programación almacenada y que serán probados en la sustentación con las invocaciones correspondiente:

- **Procedimiento** almacenado controlado para la carga o inserción de las tablas paramétricas
 - Tipo de Teléfonos
 - Tipo de Correos
 - Profesiones,
 - Sucursales,
 - Tipos de prestamos
- **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.
- **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.
- **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.
- **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.
 - 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 ($1000 * 1\% = 50.00$).

- El cálculo del interés lo realiza una **Función** que es invocada desde procedimiento.
- El interés se cobra primero y de quedar alguna porción del monto pagado por el cliente se aplica al saldo del préstamo.
- La tabla de sucursales solo almacena los monto prestados por la empresa financiera en función de esto, aplicar las actualizaciones.

Este laboratorio desarrollo en equipo plazo una semana-

Una vez los equipos hayan presentado el laboratorio será subirlo a la plataforma Moodle.

Ejemplo de Objetos

Sucursal (suc_codigo varchar2(2) primary key not nul, suc_descripcion varchar(50));

Tipoprestamos (tip_codigo varchar(2) primary key not null, tip_descripcion varchar2(50), tasaintereres number(15,2));

--Creación de las tablas

```
CREATE TABLE tipos_correos(  
id_temail number PRIMARY KEY NOT NULL,  
descripcion varchar2(50) NOT NULL  
);
```

```
create table tipos_prestamos(  
cod_prestamo number not null,  
descripcion varchar2(100) not null,  
tasa_interes number,  
constraint tipos_prestamos_pk primary key (cod_prestamo)  
);
```

```
create table profesiones(  
id_profesion number not null,  
descripcion varchar2(100) not null,  
constraint profesion_pk primary key (id_profesion)  
);
```

```
create table clientes(  

```

```

id_cliente number not null,
cedula varchar2(10) not null unique,
nombre varchar2(100) not null,
apellido varchar2(100) not null,
sexo varchar2(2) not null check (sexo in ('M', 'F')),
fecha_nacimiento date not null,
cod_profesion number not null,
constraint clientes_pk primary key (id_cliente),
constraint profesion_fk foreign key (cod_profesion) references
Profesiones(id_profesion)
);

```

```

create table Clientes_email(
id_cliente number NOT NULL,
id_temail number NOT NULL,
email varchar2(100) NOT NULL,
PRIMARY KEY(id_cliente, id_temail),
CONSTRAINT cliente_temail_fk
FOREIGN KEY(id_cliente) REFERENCES Clientes(id_cliente),
CONSTRAINT email_fk
FOREIGN KEY(id_temail) REFERENCES tipos_correos(id_temail)
);

```

```

create table tipos_telefonos(
id_ttelefono number NOT NULL,
descripcion varchar2(50) NOT NULL,
constraint tipos_telefonos1 primary key (id_ttelefono),
constraint tipos_telefonos_u unique (descripcion)
);

```

```

create table Clientes_telefono(
id_cliente number NOT NULL,
id_ttelefono number NOT NULL,
telefono number NOT NULL CHECK (telefono not like '%[^0-9]%' ),
PRIMARY KEY(id_cliente, id_ttelefono),
CONSTRAINT cliente_ttelefono_fk
FOREIGN KEY(id_cliente) REFERENCES Clientes(id_cliente),
CONSTRAINT telefono_fk
FOREIGN KEY (id_ttelefono) REFERENCES tipos_telefonos(id_ttelefono)
);

```

```

create table Prestamos(
cliente_fk number NOT NULL,
tipoprestamo_fk number NOT NULL,
id_prestamo number NOT NULL,
fecha_aprobado date NOT NULL,

```

```

monto_aprobado number NOT NULL,
tasa_interes number NOT NULL,
letra_mensual number NOT NULL,
monto_pagado number DEFAULT 0 NOT NULL,
monto_intereses number DEFAULT 0 NOT NULL,
fecha_pago date,
PRIMARY KEY(cliente_fk, tipoprestamo_fk),
CONSTRAINT tipoprestamo_fk
FOREIGN KEY(tipoprestamo_fk) REFERENCES
tipos_prestamos(cod_prestamo),
CONSTRAINT cliente_fk
FOREIGN KEY(cliente_fk) REFERENCES Clientes(id_cliente)
);

```

--=====PROFESIONES =====

```

insert into Profesiones (id_profesion, descripcion) values (1, 'Contador');
insert into Profesiones (id_profesion, descripcion) values (2, 'Ingeniero');
insert into Profesiones (id_profesion, descripcion) values (3, 'Profesor');

```

--=====CLIENTES=====

```

INSERT INTO CLIENTES(id_cliente, cedula, nombre, apellido, sexo,
fecha_nacimiento,
cod_profesion)
VALUES (1, '8-981-765', 'Carlos', 'Navarrete', 'M', '25-OCT-98', 3);
INSERT INTO CLIENTES(id_cliente, cedula, nombre, apellido, sexo,
fecha_nacimiento,
cod_profesion)
VALUES (2, '4-794-245', 'Sofía', 'De Gracia', 'F', '13-MAY-78', 1);
INSERT INTO CLIENTES(id_cliente, cedula, nombre, apellido, sexo,
fecha_nacimiento,
cod_profesion)
VALUES (3, '8-964-1765', 'Nicole', 'Russell', 'F', '02-FEB-00', 3);
INSERT INTO CLIENTES(id_cliente, cedula, nombre, apellido, sexo,
fecha_nacimiento,
cod_profesion)
VALUES (4, '8-889-112', 'Valentín', 'Pérez', 'M', '18-MAY-01', 2);
INSERT INTO CLIENTES(id_cliente, cedula, nombre, apellido, sexo,
fecha_nacimiento,
cod_profesion)
VALUES (5, '2-994-1772', 'Jorge', 'Gomez', 'M', '27-JUN-67', 3);

```

--=====TIPO EMAIL=====

```

INSERT INTO tipos_correos(id_temail,descripcion)
values (1, 'Personal');
INSERT INTO tipos_correos(id_temail,descripcion)
values (2, 'Laboral');
INSERT INTO tipos_correos(id_temail,descripcion)
values (3, 'Académico');

```

--=====CLIENTES EMAIL=====

```
insert into Clientes_email (id_cliente, id_temail, email) values (1,
1,'Carlosmontiel29@gmail.com' );
insert into Clientes_email (id_cliente, id_temail, email) values (2,
1,'Mariacastillo22@gmail.com' );
insert into Clientes_email (id_cliente, id_temail, email) values (2,
2,'Mariacastillo22@gmail.com' );
insert into Clientes_email (id_cliente, id_temail, email) values (3,
1,'Marioarriaza@gmail.com' );
insert into Clientes_email (id_cliente, id_temail, email) values (3,
3,'Marioarriaza@utp.ac.pa' );
```

--=====TIPOS TELEFONOS=====

```
INSERT INTO tipos_telefonos(id_telefono,descripcion)
values (1, 'Personal');
INSERT INTO tipos_telefonos(id_telefono,descripcion)
values (2, 'Residencial');
INSERT INTO tipos_telefonos(id_telefono,descripcion)
values (3, 'Familiar');
INSERT INTO tipos_telefonos(id_telefono,descripcion)
values (4, 'Cónyuge');
```

--=====CLIENTES TELEFONO=====

```
INSERT INTO Clientes_telefono(id_cliente, id_telefono, telefono)
values (1, 1, 67489215);
INSERT INTO Clientes_telefono(id_cliente, id_telefono, telefono)
values (1, 2, 2745690);
INSERT INTO Clientes_telefono(id_cliente, id_telefono, telefono)
values (2, 1, 68729087);
INSERT INTO Clientes_telefono(id_cliente, id_telefono, telefono)
values (2, 4, 69258481);
INSERT INTO Clientes_telefono(id_cliente, id_telefono, telefono)
values (3, 1, 64689871);
INSERT INTO Clientes_telefono(id_cliente, id_telefono, telefono)
values (4, 1, 65679809);
INSERT INTO Clientes_telefono(id_cliente, id_telefono, telefono)
values (5, 2, 2634560);
INSERT INTO Clientes_telefono(id_cliente, id_telefono, telefono)
values (5, 1, 62678901);
```

--=====TIPO PRESTAMO =====

```
INSERT INTO tipos_prestamos(cod_prestamo,descripcion)
values (1, 'Personal');
INSERT INTO tipos_prestamos(cod_prestamo,descripcion)
values (2, 'Automóvil');
INSERT INTO tipos_prestamos(cod_prestamo,descripcion)
values (3, 'Hipoteca');
INSERT INTO tipos_prestamos(cod_prestamo,descripcion)
values (4, 'Garantizado con ahorros');
```

```

--=====PRESTAMOS=====
insert into Prestamos (id_prestamo, fecha_aprobado, monto_aprobado,
tasa_interes,
letra_mensual, monto_pagado, fecha_pago,tipoprestamo_fk, cliente_fk ) values
(1,
'23-FEB-21', 5000, 6.25, 400, 200, '07-MAY-21', 1, 1 );
insert into Prestamos (id_prestamo, fecha_aprobado, monto_aprobado,
tasa_interes,
letra_mensual, monto_pagado, fecha_pago,tipoprestamo_fk, cliente_fk ) values
(2,
'26-MAR-21', 4000, 5.24, 200, 150, '07-MAY-21', 2, 2);
insert into Prestamos (id_prestamo, fecha_aprobado, monto_aprobado,
tasa_interes,
letra_mensual, monto_pagado, fecha_pago,tipoprestamo_fk, cliente_fk ) values
(3,
'02-MAY-21', 3500, 8.24, 500, 150, '17-JUL-21', 1, 3);
insert into Prestamos (id_prestamo, fecha_aprobado, monto_aprobado,
tasa_interes,
letra_mensual, monto_pagado, fecha_pago,tipoprestamo_fk, cliente_fk ) values
(4,
'25-MAY-20', 10500, 5, 250, 1050, '01-JUN-20', 2, 4);
insert into Prestamos (id_prestamo, fecha_aprobado, monto_aprobado,
tasa_interes,
letra_mensual, monto_pagado, fecha_pago,tipoprestamo_fk, cliente_fk ) values
(5,
'12-JUL-19', 1200, 4.5, 120, 483, '25-SEP-21', 4, 5);
insert into Prestamos (id_prestamo, fecha_aprobado, monto_aprobado,
tasa_interes,
letra_mensual, monto_pagado, fecha_pago,tipoprestamo_fk, cliente_fk ) values
(6,
'14-MAR-21', 25500, 3, 300, 12000, '09-MAY-21', 3, 1);
insert into Prestamos (id_prestamo, fecha_aprobado, monto_aprobado,
tasa_interes,
letra_mensual, monto_pagado, fecha_pago,tipoprestamo_fk, cliente_fk ) values
(7,
'15-FEB-21', 34500, 8.24, 200, 150, '30-MAY-21', 3, 5);
insert into Prestamos (id_prestamo, fecha_aprobado, monto_aprobado,
tasa_interes,
letra_mensual, monto_pagado, fecha_pago,tipoprestamo_fk, cliente_fk ) values
(8,
'25-SEP-21', 20000, 10.24, 2500, 1500, '5-NOV-21', 2, 5);
-- Alter punto 4 de la Parte I
ALTER TABLE Clientes
ADD Edad number;
ALTER TABLE Prestamos
ADD saldoactual number DEFAULT 0 NOT NULL;

```



```
ALTER TABLE Prestamos
ADD interespagado number DEFAULT 0 NOT NULL;
ALTER TABLE Prestamos
ADD fechamodificacion date;
ALTER TABLE Prestamos
ADD usuario varchar2(20) DEFAULT USER NOT NULL;
ALTER TABLE Prestamos
ADD id_sucursal number DEFAULT 1 NOT NULL;
ALTER TABLE Clientes
ADD id_sucursal number DEFAULT 1 NOT NULL;
```

```
CREATE TABLE Sucursal(
id_sucursal number PRIMARY KEY NOT NULL,
descripcion varchar2(30) NOT NULL
);
```

```
CREATE TABLE transacpagos(
id_transaccion number PRIMARY KEY NOT NULL,
id_sucursal number NOT NULL,
id_cliente number NOT NULL,
tipoprestamo number NOT NULL,
fechatransaccion date NOT NULL,
monto_pago number NOT NULL,
status_pago varchar2(20) DEFAULT 'Pendiente' NOT NULL,
fechainsercion date NOT NULL,
usuario varchar2(20) DEFAULT USER NOT NULL,
CONSTRAINT fk_id_sucursal FOREIGN KEY (id_sucursal) REFERENCES
Sucursal(id_sucursal),
CONSTRAINT fk_tipoprestamo FOREIGN KEY(tipoprestamo) REFERENCES
tipos_prestamos(cod_prestamo),
CONSTRAINT fk_id_cliente FOREIGN KEY(id_cliente) REFERENCES
Clientes(id_cliente)
);
create sequence id_cliente
start with 1
increment by 1
maxvalue 99999
minvalue 1
nocycle;
```

```
create sequence numero_prestamo
start with 1
increment by 1
maxvalue 99999
minvalue 1
nocycle;
```

```
create sequence id_transaccion  
start with 1  
increment by 1  
maxvalue 99999  
minvalue 1  
nocycle;
```

```
create sequence id_telefono  
start with 5  
increment by 1  
maxvalue 99999  
minvalue 1  
nocycle;
```

```
create sequence id_email  
start with 4  
increment by 1  
maxvalue 99999  
minvalue 1  
nocycle;
```

```
create sequence id_profesion  
start with 4  
increment by 1  
maxvalue 99999  
minvalue 1  
nocycle;
```

```
create sequence id_sucursal  
start with 1  
increment by 1  
maxvalue 99999  
minvalue 1  
nocycle;
```

```
create sequence cod_prestamo  
start with 5  
increment by 1  
maxvalue 99999  
minvalue 1  
nocycle;
```

```
create sequence s_cliente  
start with 6  
increment by 1
```

```
maxvalue 99999
minvalue 1
nocycle;
```

```
CREATE OR REPLACE PROCEDURE IngresarTipoTelefono(
p_descripcion IN tipos_telefonos.descripcion%TYPE)
IS
BEGIN
INSERT INTO tipos_telefonos(id_telefono, descripcion)
VALUES(id_telefono.nextval, p_descripcion);
COMMIT;
EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
dbms_output.put_line('La descripcion ya existe');
WHEN OTHERS THEN
dbms_output.put_line('Ocurrió un error en la inserción de los datos');
END;
/
```

```
-----
-- Procedimientos --
```

```
CREATE OR REPLACE PROCEDURE IngresarTipoCorreo(
p_descripcion IN tipos_correos.descripcion%TYPE)
IS
BEGIN
INSERT INTO tipos_correos(id_temail, descripcion)
VALUES(id_temail.nextval, p_descripcion);
COMMIT;
EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
dbms_output.put_line('La descripcion ya existe');
WHEN OTHERS THEN
dbms_output.put_line('Ocurrió un error en la inserción de los datos');
END;
/
```

```
-----
CREATE OR REPLACE PROCEDURE IngresarProfesiones(
p_descripcion IN Profesiones.descripcion%TYPE)
IS
BEGIN
INSERT INTO Profesiones(id_profesion, descripcion)
VALUES(id_profesion.nextval, p_descripcion);
COMMIT;
EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
dbms_output.put_line('La descripcion ya existe');
WHEN OTHERS THEN
```

```
dbms_output.put_line('Ocurrió un error en la inserción de los datos');  
END;  
/
```

```
-----  
CREATE OR REPLACE PROCEDURE IngresarSucursales(  
p_descripcion IN Sucursal.descripcion%TYPE)  
IS  
BEGIN  
INSERT INTO Sucursal(id_sucursal, descripcion)  
VALUES(id_sucursal.nextval, p_descripcion);  
COMMIT;  
EXCEPTION  
WHEN DUP_VAL_ON_INDEX THEN  
dbms_output.put_line('La descripcion ya existe');  
WHEN OTHERS THEN  
dbms_output.put_line('Ocurrió un error en la inserción de los datos');  
END;  
/
```

```
-----  
CREATE OR REPLACE PROCEDURE IngresarTipoPrestamo(  
p_descripcion IN tipos_prestamos.descripcion%TYPE)  
IS  
BEGIN  
INSERT INTO tipos_prestamos(cod_prestamo, descripcion)  
VALUES(cod_prestamo.nextval, p_descripcion);  
COMMIT;  
EXCEPTION  
WHEN DUP_VAL_ON_INDEX THEN  
dbms_output.put_line('La descripcion ya existe');  
WHEN OTHERS THEN  
dbms_output.put_line('Ocurrió un error en la inserción de los datos');  
END;  
/
```

```
-----  
CREATE TABLE SucursalTipoPrestamo(  
id_sucursal number not null,  
cod_prestamo number not null,  
monto_prestamos number,  
constraint pk_sucursaltprestamo PRIMARY KEY (id_sucursal, cod_prestamo),  
CONSTRAINT fk_idsucursal FOREIGN KEY (id_sucursal) REFERENCES  
Sucursal(id_sucursal),  
CONSTRAINT fk_idtprestamo FOREIGN KEY (cod_prestamo) REFERENCES  
tipos_prestamos(cod_prestamo)  
);
```

```

-----
CREATE OR REPLACE PROCEDURE IngresarSucursalTipoPrestamo(
p_id_sucursal IN Sucursal.id_sucursal%TYPE,
p_cod_prestamo IN tipos_prestamos.cod_prestamo%TYPE)
IS
BEGIN
INSERT INTO SucursalTipoPrestamo(id_sucursal, cod_prestamo)
VALUES(p_id_sucursal, p_cod_prestamo);
COMMIT;
EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
dbms_output.put_line('Datos ya existentes');
WHEN OTHERS THEN
dbms_output.put_line('Ocurrió un error en la inserción de los datos');
END;
/

```

```

-----
--Función de calcular edad
CREATE OR REPLACE FUNCTION fn_CalcularEdad
(p_fechanacimiento IN Clientes.fecha_nacimiento%TYPE)
RETURN number
IS
v_edadcalculada number; --almacena la edad que se va a calcular
BEGIN
v_edadcalculada := ROUND(TRUNC(MONTHS_BETWEEN(SYSDATE,
p_fechanacimiento))/12, 0);
RETURN v_edadcalculada;
EXCEPTION
WHEN OTHERS THEN
dbms_output.put_line('Ocurrió un error desconocido en el cálculo de la edad');
END fn_CalcularEdad;
/

```

```

-- Procedimiento de inserción de clientes
CREATE OR REPLACE PROCEDURE IngresarCliente(
p_cedula IN Clientes.cedula%TYPE,
p_nombre IN Clientes.nombre%TYPE,
p_apellido IN Clientes.apellido%TYPE,
p_sexo IN Clientes.sexo%TYPE,
p_fechanacimiento IN Clientes.fecha_nacimiento%TYPE,
p_cod_profesion IN Clientes.cod_profesion%TYPE,
p_id_sucursal IN Clientes.id_sucursal%TYPE
)
IS
v_edad number(3,0) := 0;
BEGIN

```

```

v_edad := fn_CalcularEdad(p_fechanacimiento);
INSERT INTO Clientes (id_cliente, cedula, nombre, apellido, sexo,
fecha_nacimiento,
cod_profesion, edad, id_sucursal)
VALUES (s_cliente.nextval, p_cedula, p_nombre, p_apellido, p_sexo,
p_fechanacimiento, p_cod_profesion, v_edad, p_id_sucursal);
EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
dbms_output.put_line('Datos repetidos');
WHEN VALUE_ERROR THEN
dbms_output.put_line('Error causado por el tamaño de los datos ingresados');
WHEN OTHERS THEN
dbms_output.put_line('Ocurrió un error en la inserción de los datos');
END;
/

```

```

-----
CREATE OR REPLACE PROCEDURE IngresarPrestamoAprobado
(
p_id_cliente IN Clientes.id_cliente%TYPE,
p_idprestamo IN Prestamos.id_prestamo%TYPE,
p_tipoprestamo IN Prestamos.tipoprestamo_fk%TYPE,
p_fecha_aprobado IN Prestamos.fecha_aprobado%TYPE,
p_monto_aprobado IN Prestamos.monto_aprobado%TYPE,
p_tasa_interes IN Prestamos.tasa_interes%TYPE,
p_letra_mensual IN Prestamos.letra_mensual%TYPE,
p_id_sucursal IN Prestamos.id_sucursal%TYPE
)
IS
BEGIN
INSERT INTO Prestamos(cliente_fk, tipoprestamo_fk, id_prestamo,
fecha_aprobado,
monto_aprobado, tasa_interes, letra_mensual, id_sucursal)
VALUES (p_id_cliente, p_tipoprestamo, numero_prestamo.nextval,
p_fecha_aprobado, p_monto_aprobado, p_tasa_interes, p_letra_mensual,
p_id_sucursal);
UPDATE SucursalTipoPrestamo
SET monto_prestamos = monto_prestamos + p_monto_aprobado
WHERE id_sucursal = p_id_sucursal AND cod_prestamo = p_tipoprestamo;
EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
dbms_output.put_line('Datos repetidos');
WHEN VALUE_ERROR THEN
dbms_output.put_line('Error causado por el tamaño de los datos ingresados');
WHEN OTHERS THEN
dbms_output.put_line('Ocurrió un error en la inserción de los datos');

```

END;

/

CREATE OR REPLACE PROCEDURE IngresarPagosRecibidos

(

p_idsucursal IN transacpagos.id_sucursal%TYPE,

p_idcliente IN transacpagos.id_cliente%TYPE,

p_tipoprestamo IN transacpagos.tipoprestamo%TYPE,

p_fechatransaccion IN transacpagos.fechatransaccion%TYPE,

p_montopagado IN transacpagos.monto_pago%TYPE

)

IS

BEGIN

INSERT INTO transacpagos(id_transaccion,id_sucursal, id_cliente,

tipoprestamo,

fechatransaccion, monto_pago, fechainsercion)

VALUES(id_transaccion.nextval, p_idsucursal, p_idcliente, p_tipoprestamo,

p_fechatransaccion, p_montopagado, SYSDATE);

EXCEPTION

WHEN DUP_VAL_ON_INDEX THEN

dbms_output.put_line('Un pago con el mismo ID ya fue registrado.');

WHEN OTHERS THEN

dbms_output.put_line('Ocurrió un error en la inserción de los datos.');

END;

/

CREATE OR REPLACE FUNCTION fn_CalculoPrestamo

(p_prestamoPagado IN Prestamos.monto_pagado%TYPE,

p_tasa_interes IN Prestamos.tasa_interes%TYPE)

RETURN number

as

v_MontoInteres number; --almacena la edad que se va a calcular

BEGIN

v_MontoInteres := p_prestamoPagado * (p_tasa_interes/100);

RETURN v_MontoInteres;

EXCEPTION

WHEN OTHERS THEN

dbms_output.put_line('Ocurrió un error desconocido en el cálculo de la tasa');

END fn_CalculoPrestamo;

/

CREATE OR REPLACE PROCEDURE ActualizarPagos

IS

v_id_cliente transacpagos.id_cliente%TYPE;

v_tipoprestamo transacpagos.tipoprestamo%TYPE;

```

v_id_sucursal transacpagos.id_sucursal%TYPE;
v_monto_pago transacpagos.monto_pago%TYPE;
v_fechatransaccion transacpagos.fechatransaccion%TYPE;
v_tasa_interes Prestamos.tasa_interes%TYPE;
v_monto_aprobado Prestamos.monto_aprobado%TYPE;
v_monto_pagado Prestamos.monto_pagado%TYPE;
v_saldoprestamo Prestamos.monto_aprobado%TYPE;
v_monto_interes Prestamos.monto_intereses%TYPE;
v_monto_al_saldo Prestamos.monto_aprobado%TYPE;
v_id_transaccion transacpagos.id_transaccion%TYPE;
CURSOR c_transaccion IS
SELECT id_cliente, tipoprestamo, id_sucursal, monto_pago, fechatransaccion,
id_transaccion
FROM transacpagos
WHERE status_pago = 'Pendiente';
BEGIN
open c_transaccion;
LOOP
FETCH c_transaccion INTO v_id_cliente, v_tipoprestamo, v_id_sucursal,
v_monto_pago, v_fechatransaccion, v_id_transaccion;
EXIT WHEN c_transaccion%NOTFOUND;
SELECT tasa_interes, monto_aprobado, monto_pagado
INTO v_tasa_interes, v_monto_aprobado, v_monto_pagado
FROM Prestamos
WHERE cliente_fk = v_id_cliente AND tipoprestamo_fk = v_tipoprestamo;
v_saldoprestamo := v_monto_aprobado - v_monto_pagado;
v_monto_interes := fn_CalculoPrestamo(v_saldoprestamo, v_tasa_interes);
v_monto_al_saldo := v_monto_pago - v_monto_interes;
UPDATE Prestamos
SET monto_pagado = monto_pagado + v_monto_al_saldo, monto_intereses =
monto_intereses + v_monto_interes, fecha_pago = v_fechatransaccion
WHERE cliente_fk = v_id_cliente AND tipoprestamo_fk = v_tipoprestamo;
UPDATE SucursalTipoPrestamo
SET monto_prestamos = monto_prestamos - v_monto_al_saldo
WHERE id_sucursal = v_id_sucursal AND cod_prestamo = v_tipoprestamo;
UPDATE transacpagos
SET status_pago = 'Procesado'
WHERE status_pago = 'Pendiente' AND id_transaccion = v_id_transaccion;
END LOOP;
CLOSE c_transaccion;
EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
dbms_output.put_line('el pago ya existe');
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('Este ID no existe');
-- WHEN OTHERS THEN

```



```
-- dbms_output.put_line('Ocurrió un error en la inserción de los datos');
END;
/
```

```
---Actualizar edad de los clientes ingresados previos a la creación del atributo
edad
```

```
DECLARE
v_codcliente Clientes.id_cliente%TYPE;
v_fecha_nac Clientes.fecha_nacimiento%TYPE;
v_edadcalc Clientes.edad%TYPE;
---Declaración del cursor
CURSOR c_edad IS
SELECT id_cliente, fecha_nacimiento
FROM Clientes
WHERE edad = 0;
BEGIN
OPEN c_edad; --Apertura del cursor
LOOP
FETCH c_edad INTO v_codcliente, v_fecha_nac;
EXIT WHEN c_edad%NOTFOUND;
v_edadcalc := fn_CalcularEdad(v_fecha_nac);
UPDATE Clientes
SET edad = v_edadcalc
WHERE id_cliente = v_codcliente;
END LOOP;
DBMS_OUTPUT.PUT_LINE('Edad Actualizada');
END;
/
```

```
-----BLOQUE ANONIMO TIPO DE
TELEFONO-----
```

```
DECLARE
v_idtipotelefono tipos_telefonos.id_ttelefono%TYPE;
v_descripcion tipos_telefonos.descripcion%TYPE;
BEGIN
IngresarTipoTelefono('Suegra');
IngresarTipoTelefono('Papá');
EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
dbms_output.put_line('La descripcion del telefono ya existe');
WHEN OTHERS THEN
dbms_output.put_line('Ocurrió un error en la inserción de los datos');
END;
/
```

```
-----BLOQUE ANONIMO TIPO
CORREO-----
```

```
DECLARE
```

```

v_descripcion tipos_correos.descripcion%TYPE;
BEGIN
IngresarTipoCorreo('Coorporativo');
IngresarTipoCorreo('Promocional');
EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
dbms_output.put_line('La descripcion del correo ya existe');
WHEN OTHERS THEN
dbms_output.put_line('Ocurrió un error en la inserción de los datos');
END;
/

=====BLOQUE ANONIMO TIPO
PROFESIONES=====
DECLARE
v_descripcion Profesionες.descripcion%TYPE;
BEGIN
IngresarProfesionες('Electricista');
IngresarProfesionες('Fotógrafo');
IngresarProfesionες('Piloto');
EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
dbms_output.put_line('La descripcion de la profesión ya existe');
WHEN OTHERS THEN
dbms_output.put_line('Ocurrió un error en la inserción de los datos');
END;
/

=====BLOQUE ANONIMO TIPO
SUCURSALES=====
DECLARE
v_descripcion Sucursal.descripcion%TYPE;
BEGIN
IngresarSucursales('La Chorrera');
IngresarSucursales('Bethania');
EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
dbms_output.put_line('El lugar de la sucursal ya existe');
WHEN OTHERS THEN
dbms_output.put_line('Ocurrió un error en la inserción de los datos');
END;
/

=====
DECLARE
p_id_cliente Clientes.id_cliente%TYPE;
p_id_prestamo Prestamos.id_prestamo%TYPE;
p_tipoprestamo Prestamos.tipoprestamo_fk%TYPE;
p_fecha_aprobado Prestamos.fecha_aprobado%TYPE;

```

```

p_monto_aprobado Prestamos.monto_aprobado%TYPE;
p_tasa_interes Prestamos.tasa_interes%TYPE;
p_letra_mensual Prestamos.letra_mensual%TYPE;
p_id_sucursal Prestamos.id_sucursal%TYPE;
BEGIN
IngresarPrestamoAprobado(1, 2, 3,'02-MAY-21', 500, 3, 25, 4);
--IngresarPrestamoAprobado(1, 2, 3,'02-MAY-21', 500, 3, 25, 4);
EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
dbms_output.put_line('Un préstamo con el mismo ID ya fue registrado. ');
WHEN OTHERS THEN
dbms_output.put_line('Ocurrió un error en la inserción de los datos');
END;
/

=====BLOQUE ANONIMO TIPO DE
PRESTAMOS=====
DECLARE
v_descripcion tipos_prestamos.descripcion%TYPE;
v_tasa_interes tipos_prestamos.tasa_interes%TYPE;
BEGIN
IngresarTipoPrestamo('Estudiantil' );
IngresarTipoPrestamo('Empresarial');
EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
dbms_output.put_line('La descripcion del Prestamo ya existe');
WHEN OTHERS THEN
dbms_output.put_line('Ocurrió un error en la inserción de los datos');
END;
/

=====BLOQUE ANONIMO CLIENTES=====
DECLARE
v_cedula Clientes.cedula%TYPE;
v_nombre Clientes.nombre%TYPE;
v_apellido Clientes.apellido%TYPE;
v_sexo Clientes.sexo%TYPE;
v_fecha_nacimiento Clientes.fecha_nacimiento%TYPE;
v_profesion Clientes.cod_profesion%TYPE;
v_id_sucursal Clientes.id_sucursal%TYPE;
BEGIN
IngresarCliente('8-934-312', 'Ana', 'Gutierrez', 'F', '02-MAY-97', 2, 2);
IngresarCliente('4-239-113', 'Mario', 'Arriaza', 'M', '23-OCT-74', 3, 1);
IngresarCliente('9-223-2008', 'Felipe', 'Contreras', 'M', '19-JUL-01', 1, 3);
EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
dbms_output.put_line('Ya se encuentra registrado un cliente con estos datos');
WHEN OTHERS THEN

```

```

dbms_output.put_line('Ocurrió un error en la inserción de los datos');
END;
/

=====BLOQUE ANONIMO PRESTAMOS=====
DECLARE
v_id_cliente Clientes.id_cliente%TYPE;
BEGIN
--id_cliente - tipoprestamo (id) - fecha_aprobado - monto_aprobado -
tasa_interes -
--letra_mensual - id_sucursal
IngresarPrestamoAprobado(1, 4, numero_prestamo.nextval, '02-OCT-20', 7000,
5, 120, 2);
IngresarPrestamoAprobado(1, 2, numero_prestamo.nextval, '27-JUL-19', 30000,
7, 200, 1);
IngresarPrestamoAprobado(4, 1, numero_prestamo.nextval, '18-FEB-21', 9000,
4, 90, 1);
EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
dbms_output.put_line('Ya se encuentra registrado un cliente con estos datos');
WHEN OTHERS THEN
dbms_output.put_line('Ocurrió un error en la inserción de los datos');
END;
/

=====BLOQUE ANONIMO PAGOS RECIBIDOS=====
DECLARE
v_id_sucursal transacpagos.id_sucursal%TYPE := 1;
v_id_cliente transacpagos.id_cliente%TYPE := 1;
v_tipoprestamo transacpagos.tipoprestamo%TYPE := 4;
v_fechatransaccion transacpagos.fechatransaccion%TYPE := '02-NOV-21';
v_monto_pago transacpagos.monto_pago%TYPE := 150;
BEGIN
IF v_tipoprestamo IS NOT NULL THEN
IngresarPagosRecibidos(v_id_sucursal, v_id_cliente, v_tipoprestamo,
v_fechatransaccion, v_monto_pago);
DBMS_OUTPUT.PUT_LINE('Pago quincenal registrado exitosamente');
END IF;
END;
/

DECLARE
v_id_sucursal transacpagos.id_sucursal%TYPE := 2;
v_id_cliente transacpagos.id_cliente%TYPE := 4;
v_tipoprestamo transacpagos.tipoprestamo%TYPE := 1;
v_fechatransaccion transacpagos.fechatransaccion%TYPE := '18-MAR-21';
v_monto_pago transacpagos.monto_pago%TYPE := 100;

```

```

BEGIN
IF v_tipoprestamo IS NOT NULL THEN
IngresarPagosRecibidos(v_id_sucursal, v_id_cliente, v_tipoprestamo,
v_fechatransaccion, v_monto_pago);
DBMS_OUTPUT.PUT_LINE('Pago quincenal registrado exitosamente');
END IF;
END;
/

```

```

DECLARE
BEGIN
ActualizarPagos();
END;
/

```

```

create view ver_todo as select c.id_cliente,concat(concat(c.nombre, ' '
),c.apellido) as cliente, p.descripcion,tr.tipoprestamo, tr.monto_pago,
tr.status_pago, ct.telefono from profesiones p
inner join clientes c on c.id_cliente = p.id_profesion
inner join clientes_telefono ct on c.id_cliente = ct.id_cliente
inner join transacpagos tr on c.id_cliente = tr.id_cliente;

```

```
select * from ver_todo;
```

Vista para el usuario

View created.

ID_CLIENTE	CLIENTE	DESCRIPCION	TIPOPRESTAMO	MONTO_PAGO	STATUS_PAGO	TELEFONO
1	Carlos Navarrete	Contador	4	150	Procesado	67489215
1	Carlos Navarrete	Contador	4	150	Procesado	2745690
4	Valentín Pérez	Electricista	1	100	Pendiente	65679809

[Download CSV](#)

3 rows selected.