**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 ll.**

**LABORATORIO 7**

**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 TECNOLOGICA DE PANAMA FACULTAD DE INGENIERA DE SISTEMAS COMPUTACIONALES LICENCIATURA EN INGENIERIA DE SISTEMAS DE INFORMACION

*Docente: Ing. Henry Lezcano MAS*

*III* Semestre de 2022

**BASE DE DATOS ll**

# LABORATORIO 7

# IMPLEMENTACION DE TRIGGERS

**Para el Laboratorio No.4** que indica ‘La Sociedad de Ingenieros y Arquitectos de Panamá, requiere que los Ingenieros en Sistemas Y computación que implementen el proceso de cálculo de salario quincenal para sus colaboradores, actualmente ellos cuentas con modelo físico que será proporcionado para esta tarea’.

## Implementar una batería de trigger (insert, update or delete) para auditar los cambios sobre la tabla critica colaboradores. Diseñe su propia estructura para completar el objetivo

**Para el Laboratorio No.6** que indica ‘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’

## Implementar una batería de trigger (insert, update or delete) para auditar los cambios sobre la tabla critica prestamos que permita garantizar la trazabilidad transaccional. Diseñe su propia estructura para completar el objetivo.

Por otra parte, implementar triggers para las acumulaciones en la tabla sucursal y sucursal\_tipoprestamos una vez se hagan cambios en los saldos de la tabla préstamos.

Evidenciar la implementación solicitada y subirla a ecampus por equipo.

-- LAB 4

create table colaboradores(

id\_codcolaborador number,

nombre varchar2(25) not null,

apellido varchar2(25) not null,

cedula varchar2(12) not null unique,

sexo char not null check (sexo IN('M', 'F')),

fecha\_nacimiento date not null,

fecha\_ingreso date not null,

status char not null check (status IN('A', 'V', 'R')),

salario\_mensual number(15,2) default 0,

constraint pk\_colaboradores primary key (id\_codcolaborador)

);

create table salario\_quincenal(

id\_salario number primary key,

id\_codcolaborador1 number not null,

fecha\_pago date not null,

salario\_quincenal number(15,2) default 0 not null,

seguro\_social number(15,2) default 0 not null,

seguro\_educativo number(15,2) default 0 not null,

salario\_neto number(15,2) default 0 not null,

constraint fk\_colaborador\_ foreign key(id\_codcolaborador1) references

colaboradores(id\_codcolaborador)

);

create table auditoria (

id\_auditoria number not null,

id\_colaborador number not null,

estado\_registro varchar(50) not null,

nombre varchar2(20),

salario\_anterior number(15,2),

salario\_nuevo number(15,2),

constraint auditoria\_pk primary key (id\_auditoria)

);

create sequence sec\_colaborador

start with 1

increment by 1;

create sequence sec\_auditoria

start with 1

increment by 1;

create or replace procedure agregar\_col(

p\_nombre colaboradores.nombre%TYPE,

p\_apellido colaboradores.apellido%TYPE,

p\_cedula colaboradores.cedula%TYPE,

p\_sexo colaboradores.sexo%TYPE,

p\_fecha\_nac colaboradores.fecha\_nacimiento%TYPE,

p\_fecha\_ingreso colaboradores.fecha\_ingreso%TYPE,

p\_status colaboradores.status%TYPE,

p\_salario\_men colaboradores.salario\_mensual%TYPE)

as

begin

insert into colaboradores (id\_codcolaborador, nombre, apellido, cedula, sexo,

fecha\_nacimiento, fecha\_ingreso, status, salario\_mensual)

values (sec\_colaborador.nextval, p\_nombre, p\_apellido, p\_cedula, p\_sexo, p\_fecha\_nac,

p\_fecha\_ingreso, p\_status, p\_salario\_men);

end agregar\_col;

begin

agregar\_col('Carlos','Solis','8-953-756','M','21-JUN-99','14-MAY-21','A',800.0);

agregar\_col('Julia','Sanjur','8-979-953','F','21-OCT-99','14-FEB-21','A',920.55);

agregar\_col('Raul','Jimenez','9-964-1234','M','11-OCT-98','14-JUL-21','A',1310.00);

agregar\_col('Pablo','Escobar','9-784-1234','M','11-OCT-99','18-JUL-21','A',1510.00);

end;

-- Función salario quincenal

create or replace function quincenal(p\_salario\_men IN colaboradores.salario\_mensual%type)

return number

as

v\_salario\_quincenal number;

begin

v\_salario\_quincenal := p\_salario\_men/2;

return v\_salario\_quincenal;

end quincenal;

-- función de seguro\_social

create or replace function seguro\_social(p\_salario\_men IN colaboradores.salario\_mensual%type)

return number

as

v\_seguro\_social number;

begin

v\_seguro\_social := ((p\_salario\_men/2) \* 0.0975);

return v\_seguro\_social;

end seguro\_social;

-- función de seguro\_educativo

create or replace function seguro\_educativo(p\_salario\_men IN colaboradores.salario\_mensual%type)

return number

as

v\_seguro\_educativo number;

begin

v\_seguro\_educativo := (p\_salario\_men/2 \* 0.0125);

return v\_seguro\_educativo;

end seguro\_educativo;

-- función de neto

create or replace function neto(p\_salario\_men IN colaboradores.salario\_mensual%type)

return number

as

v\_neto number;

begin

v\_neto := (p\_salario\_men/2) - (((p\_salario\_men/2) \* 0.0975) + (p\_salario\_men/2 \* 0.0125));

return v\_neto;

end neto;

declare

-- Declaración

v\_colaboradorID colaboradores.id\_codcolaborador%type;

v\_salarioMensual colaboradores.salario\_mensual%type;

v\_salarioQuincenal salario\_quincenal.salario\_quincenal%type;

v\_seguroSocial salario\_quincenal.seguro\_social%type;

v\_seguroEducativo salario\_quincenal.seguro\_educativo%type;

v\_salarioNeto salario\_quincenal.salario\_neto%type;

--Declarar Cursor

cursor c\_Colaboradores is

select id\_codcolaborador, salario\_mensual

from colaboradores

where status = 'A';

--Empieza el siguiente bloque

begin

--Ver si es día de pago

if to\_char(CURRENT\_DATE, 'dd') = '15' or to\_char(CURRENT\_DATE, 'dd')= '30'

then

open c\_Colaboradores;

loop

fetch c\_Colaboradores into v\_colaboradorID, v\_salarioMensual;

exit when c\_Colaboradores%NOTFOUND;

v\_salarioQuincenal := quincenal(v\_SalarioMensual);

v\_seguroSocial := seguro\_social(v\_salarioMensual);

v\_seguroEducativo := seguro\_educativo(v\_salarioMensual);

v\_salarioNeto := neto(v\_salarioMensual);

--v\_seguroSocial := v\_salarioQuincenal \* 0.0975;

--seguro\_social();

--v\_seguroEducativo := v\_salarioQuincenal \* 0.0125;

--v\_salarioNeto := v\_salarioQuincenal - (v\_seguroSocial + v\_seguroEducativo);

insert into salario\_quincenal values(salario\_sequence.nextval, v\_colaboradorID, CURRENT\_DATE, v\_salarioQuincenal, v\_seguroSocial, v\_seguroEducativo, v\_salarioNeto);

end loop;

close c\_Colaboradores;

else

dbms\_output.put\_line('No es día de pago');

end if;

exception

when NO\_DATA\_FOUND then

DBMS\_OUTPUT.PUT\_LINE('Este ID (identificador) no existe');

end;

-- Para el Laboratorio No.4 que indica ‘La Sociedad de Ingenieros y Arquitectos de Panamá,

-- requiere que los Ingenieros en Sistemas Y computación que implementen el proceso de

-- cálculo de salario quincenal para sus colaboradores, actualmente ellos cuentas con modelo

-- físico que será proporcionado para esta tarea’.

-- Implementar una batería de trigger (insert, update or delete) para auditar los cambios

-- sobre la tabla critica colaboradores. Diseñe su propia estructura para completar el objetivo

**-- Creación de triggers lab 4**

-- Trigger para insertar y actualizar colaborador

create or replace trigger tr\_colaborador

before insert or update or delete on colaboradores

for each row

begin

if inserting then

insert into auditoria (id\_auditoria,id\_colaborador,estado\_registro,nombre,salario\_anterior,salario\_nuevo)

values(sec\_auditoria.nextval,:new.id\_codcolaborador,'Insercion -- Datos insertados',:new.nombre,0,:new.salario\_mensual);

end if;

if updating then

insert into auditoria (id\_auditoria,id\_colaborador,estado\_registro,nombre,salario\_anterior,salario\_nuevo)

values(sec\_auditoria.nextval,:new.id\_codcolaborador,'Actualizacion -- Datos anteriores',:new.nombre,:old.salario\_mensual,:new.salario\_mensual);

insert into auditoria (id\_auditoria,id\_colaborador,estado\_registro,nombre,salario\_anterior,salario\_nuevo)

values(sec\_auditoria.nextval,:new.id\_codcolaborador,'Actualizacion -- Datos actualizados nuevos',:new.nombre,:old.salario\_mensual,:new.salario\_mensual);

end if;

if deleting then

insert into auditoria (id\_auditoria,id\_colaborador,estado\_registro,nombre,salario\_anterior,salario\_nuevo)

values(sec\_auditoria.nextval,:old.id\_codcolaborador,'Borrado -- Datos eliminados',:old.nombre,:old.salario\_mensual,:old.salario\_mensual);

end if;

exception

when others then

dbms\_output.put\_line('Error inesperado, algo salió mal');

end tr\_colaborador;

begin

agregar\_col('David','Rosas','7-784-1274','M','12-OCT-99','19-JUL-21','A',2510.00);

end;

delete from colaboradores where ID\_CODCOLABORADOR = 5;

select \* from colaboradores;

select \* from auditoria order by id\_auditoria;

-- Inserción de datos para comprobar el trigger

-- select \* from colaboradores;

-- select \* from auditoria order by id\_auditoria asc;

-- Update para comprobar que el trigger funciona correctamente

update colaboradores

set salario\_mensual = salario\_mensual + 1000

where id\_codcolaborador = 5;

create view Ver as

select c.id\_codcolaborador as codigo, c.nombre,

c.apellido , c.salario\_mensual, sk.salario\_quincenal,

sk.seguro\_social , sk.seguro\_educativo, sk.salario\_neto as salario\_Neto

from colaboradores c

inner join salario\_quincenal sk on c.id\_codcolaborador = sk.id\_codcolaborador1

where status = 'A'

order by c.id\_codcolaborador asc;

select \* from ver;

**Evidencias de la implementación**

**Resultado para lab 4 de la tabla auditoría (inserción, actualización y eliminación).**



**Lab 4 tabla colaborador insertamos (el id\_codcolaborador = 5)**



--lab 6 aplicación de triggers

--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)

);

create table sucursal(

id\_sucursal number primary key not null,

descripcion varchar2(30) not null

);

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)

);

--===========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\_ttelefono,descripcion)

values (1, 'personal');

insert into tipos\_telefonos(id\_ttelefono,descripcion)

values (2, 'residencial');

insert into tipos\_telefonos(id\_ttelefono,descripcion)

values (3, 'familiar');

insert into tipos\_telefonos(id\_ttelefono,descripcion)

values (4, 'cónyuge');

--====================clientes telefono====================

insert into clientes\_telefono(id\_cliente, id\_ttelefono, telefono)

values (1, 1, 67489215);

insert into clientes\_telefono(id\_cliente, id\_ttelefono, telefono)

values (1, 2, 2745690);

insert into clientes\_telefono(id\_cliente, id\_ttelefono, telefono)

values (2, 1, 68729087);

insert into clientes\_telefono(id\_cliente, id\_ttelefono, telefono)

values (2, 4, 69258481);

insert into clientes\_telefono(id\_cliente, id\_ttelefono, telefono)

values (3, 1, 64689871);

insert into clientes\_telefono(id\_cliente, id\_ttelefono, telefono)

values (4, 1, 65679809);

insert into clientes\_telefono(id\_cliente, id\_ttelefono, telefono)

values (5, 2, 2634560);

insert into clientes\_telefono(id\_cliente, id\_ttelefono, 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 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\_ttelefono

start with 5

increment by 1

maxvalue 99999

minvalue 1

nocycle;

create sequence id\_temail

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\_ttelefono, descripcion)

values(id\_ttelefono.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 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 profesiones.descripcion%type;

begin

ingresarprofesiones('electricista');

ingresarprofesiones('fotógrafo');

ingresarprofesiones('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;

-- creación de tablas para las auditorias

create table auditoria\_prestamo (

aup\_id\_prestamo number not null primary key,

estado\_registro varchar(50) not null,

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

);

create sequence aup\_id\_prestamo

start with 1

increment by 1;

-- implementar una batería de trigger (insert, update or delete) para auditar los cambios

-- sobre la tabla critica prestamos que permita garantizar la trazabilidad transaccional. diseñe

-- su propia estructura para completar el objetivo.

-- trigger para insertar,actualizar y eliminar prestamos

create or replace trigger tr\_prestamos

before insert or update or delete on prestamos

for each row

begin

if inserting then

insert into auditoria\_prestamo (aup\_id\_prestamo,estado\_registro,cliente\_fk,tipoprestamo\_fk,id\_prestamo,fecha\_aprobado,monto\_aprobado,tasa\_interes,letra\_mensual,

monto\_pagado,monto\_intereses,fecha\_pago)

values(aup\_id\_prestamo.nextval,'inserción -- datos insertados nuevos',:new.cliente\_fk,:new.tipoprestamo\_fk,:new.id\_prestamo,:new.fecha\_aprobado,:new.monto\_aprobado,:new.tasa\_interes,:new.letra\_mensual,

:new.monto\_pagado,:new.monto\_intereses,:new.fecha\_pago);

end if;

if updating then

-- para que aparezca registro de datos antes de actualizar

insert into auditoria\_prestamo (aup\_id\_prestamo,estado\_registro,cliente\_fk,tipoprestamo\_fk,id\_prestamo,fecha\_aprobado,monto\_aprobado,tasa\_interes,letra\_mensual,

monto\_pagado,monto\_intereses,fecha\_pago)

values(aup\_id\_prestamo.nextval,'actualizacion -- datos anteriores',:old.cliente\_fk,:old.tipoprestamo\_fk,:old.id\_prestamo,:old.fecha\_aprobado,:old.monto\_aprobado,:old.tasa\_interes,:old.letra\_mensual,

:old.monto\_pagado,:old.monto\_intereses,:old.fecha\_pago);

-- la nueva actualizacion

insert into auditoria\_prestamo (aup\_id\_prestamo,estado\_registro,cliente\_fk,tipoprestamo\_fk,id\_prestamo,fecha\_aprobado,monto\_aprobado,tasa\_interes,letra\_mensual,

monto\_pagado,monto\_intereses,fecha\_pago)

values(aup\_id\_prestamo.nextval,'actualizacion -- datos actualizados nuevos',:new.cliente\_fk,:new.tipoprestamo\_fk,:new.id\_prestamo,:new.fecha\_aprobado,:new.monto\_aprobado,:new.tasa\_interes,:new.letra\_mensual,

:new.monto\_pagado,:new.monto\_intereses,:new.fecha\_pago);

end if;

-- que aparezcan los datos eliminados

if deleting then

insert into auditoria\_prestamo (aup\_id\_prestamo,estado\_registro,cliente\_fk,tipoprestamo\_fk,id\_prestamo,fecha\_aprobado,monto\_aprobado,tasa\_interes,letra\_mensual,

monto\_pagado,monto\_intereses,fecha\_pago)

values(aup\_id\_prestamo.nextval,'borrado -- datos eliminados',:old.cliente\_fk,:old.tipoprestamo\_fk,:old.id\_prestamo,:old.fecha\_aprobado,:old.monto\_aprobado,:old.tasa\_interes,:old.letra\_mensual,

:old.monto\_pagado,:old.monto\_intereses,:old.fecha\_pago);

end if;

exception

when others then

dbms\_output.put\_line('error inesperado, algo salió mal');

end tr\_prestamos;

/

insert into prestamos (id\_prestamo, fecha\_aprobado, monto\_aprobado, tasa\_interes,

letra\_mensual, monto\_pagado, fecha\_pago,tipoprestamo\_fk, cliente\_fk ) values (cod\_prestamo.nextval,

'25-feb-21', 6000, 7.25, 100, 300, '09-may-22', 5, 2 );

select \* from auditoria\_prestamo order by aup\_id\_prestamo;

delete from prestamos where tipoprestamo\_fk = 5;

update prestamos

set monto\_aprobado = 2000

where tipoprestamo\_fk = 5;

-- por otra parte, implementar triggers para las acumulaciones en la tabla sucursal y

-- sucursal\_tipoprestamos una vez se hagan cambios en los saldos de la tabla préstamos.

create table auditoria\_sucursal(

id\_sucursal number primary key not null,

descripcion varchar2(30) not null

);

create table auditoria\_sucursal\_tipoprestamos(

id\_sucursal number not null primary key,

cod\_prestamo number not null,

monto\_prestamos number

);

**Evidencia**