



**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 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 TECNOLÓGICA DE PANAMA FACULTAD DE  
INGENIERIA DE SISTEMAS COMPUTACIONALES LICENCIATURA EN  
INGENIERIA DE SISTEMAS DE INFORMACION**

**BASE DE DATOS II**

**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 cuentan 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 crítica 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 crítica préstamos 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 e-campus 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).

ID_AUDITORIA	ID_COLABORADOR	ESTADO_REGISTRO	NOMBRE	SALARIO_ANTERIOR	SALARIO_NUEVO
1	5	Insercion -- Datos insertados	David	0	2510
2	5	Actualizacion -- Datos anteriores	David	2510	3510
3	5	Actualizacion -- Datos actualizados nuevos	David	2510	3510
4	5	Borrado -- Datos eliminados	David	3510	3510

[Download CSV](#)  
4 rows selected.

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

ID_CODCOLABORADOR	NOMBRE	APELLIDO	CEDULA	SEXO	FECHA_NACIMIENTO	FECHA_INGRESO	STATUS	SALARIO_MENSUAL
1	Carlos	Solis	8-953-756	M	21-JUN-99	14-MAY-21	A	800
2	Julia	Sanjur	8-979-953	F	21-OCT-99	14-FEB-21	A	920.55
3	Raul	Jimenez	9-964-1234	M	11-OCT-98	14-JUL-21	A	1310
4	Pablo	Escobar	9-784-1234	M	11-OCT-99	18-JUL-21	A	1510
5	David	Rosas	9-784-1274	M	11-OCT-99	18-JUL-21	A	3510

[Download CSV](#)

--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_telefono) references tipos_telefonos(id_telefono)  
);
```

```
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_sucursaltiprestamo 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_telefono
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_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 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_fecha_transaccion transacpagos.fecha_transaccion%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, fecha_transaccion,
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_fecha_transaccion, 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_fecha_transaccion
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
ingresartipocorreos('coorporativo');
ingresartipocorreos('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,monito\_aprobado,tasa\_interes,letra\_mensual,

monito\_pagado,monito\_intereses,fecha\_pago)

values(aup\_id\_prestamo.nextval,'inserción -- datos insertados

nuevos',:new.cliente\_fk,:new.tipoprestamo\_fk,:new.id\_prestamo,:new.fecha\_apr

obado,:new.monito\_aprobado,:new.tasa\_interes,:new.letra\_mensual,

:new.monito\_pagado,:new.monito\_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,monito\_aprobado,tasa\_interes,letra\_mensual,

monito\_pagado,monito\_intereses,fecha\_pago)

values(aup\_id\_prestamo.nextval,'actualizacion -- datos

anteriores',:old.cliente\_fk,:old.tipoprestamo\_fk,:old.id\_prestamo,:old.fecha\_aprob

ado,:old.monito\_aprobado,:old.tasa\_interes,:old.letra\_mensual,

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

-- la nueva actualizacion

insert into auditoria\_prestamo

(aup\_id\_prestamo,estado\_registro,cliente\_fk,tipoprestamo\_fk,id\_prestamo,fecha\_aprobado,monito\_aprobado,tasa\_interes,letra\_mensual,

monito\_pagado,monito\_intereses,fecha\_pago)

values(aup\_id\_prestamo.nextval,'actualizacion -- datos actualizados

nuevos',:new.cliente\_fk,:new.tipoprestamo\_fk,:new.id\_prestamo,:new.fecha\_apr

obado,:new.monito\_aprobado,:new.tasa\_interes,:new.letra\_mensual,

:new.monito\_pagado,:new.monito\_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_apro
bado,: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

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
1	Inserción -- Datos insertados nuevos	2	5	7	25-FEB-21	6000	7.25	100	300	0	09-MAY-22
2	Actualizacion -- Datos anteriores	2	5	7	25-FEB-21	6000	7.25	100	300	0	09-MAY-22
3	Actualizacion -- Datos actualizados nuevos	2	5	7	25-FEB-21	2000	7.25	100	300	0	09-MAY-22
4	Borrado -- Datos eliminados	2	5	7	25-FEB-21	2000	7.25	100	300	0	09-MAY-22

[Download CSV](#)  
4 rows selected.