



Universidad Tecnológica de Panamá

**Facultad de Ing. de Sistemas
Computacionales**

Departamento de Sistemas de Información,



Carrera: Licenciatura en Ingeniería en Sistemas de información

Cursado: Bases de Datos II

Profesor: Ing. Henry Lezcano

L7 | Implementación de Triggers en PL/SQL

Estudiantes:

Rolando Riley	[8-972-1033]
Johel Batista	[8-914-587]
Andrés Villareal	[8-970-1267]
Miguel Pinilla	[8-975-2460]

Grupo: 1IF-131

Fecha de Entrega: 07-11-2022

República de Panamá, II Semestre Académico 2022

IMPLEMENTACIÓN 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 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 abierto 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 e-campus por equipo.

Implementación de Triggers

Laboratorio N°4

--TABLA DE AUDITORIA

```
CREATE TABLE auditoria_Colab (  
    num_transac NUMBER,  
    modif_por VARCHAR2(25) NOT NULL,  
    modificacion VARCHAR2(25) NOT NULL,  
    fecha_modif DATE NOT NULL,  
    id_codcolaborador NUMBER NOT NULL,  
    status_ant CHAR,  
    status_act CHAR NOT NULL,  
    salario_ant NUMBER(15,2),  
    salario_act NUMBER(15,2) NOT NULL,  
    salario_dif NUMBER(15,2),  
    CONSTRAINT pk_auditoria_colab PRIMARY KEY (num_transac),  
);
```

```
CREATE SEQUENCE seq_aud_colab  
    MINVALUE 1  
    START WITH 1  
    INCREMENT BY 1  
    CACHE 20;
```

--TRIGGER

```
CREATE OR REPLACE TRIGGER Trig_Colaborador  
    BEFORE INSERT OR UPDATE OR DELETE ON Colaboradores  
    FOR EACH ROW
```

BEGIN

--SI INSERTA

IF INSERTING THEN

INSERT INTO auditoria_Colab (num_transac, modif_por, modificacion,
fecha_modif, id_codcolaborador, status_act, salario_act)

VALUES (seq_aud_colab.nextval, USER, 'Registro Nuevo', SYSDATE,
:new.id_codcolaborador, :new.status, :new.salario_mensual);

COMMIT;

END IF;

--SI SE ACTUALIZA UN REGISTRO

IF UPDATING THEN

INSERT INTO auditoria_Colab

VALUES (seq_aud_colab.nextval, USER, 'Registro Actualizado', SYSDATE,
:old.id_codcolaborador, :old.status, :new.status, :old.salario_mensual,
:new.salario_mensual, (:new.salario_mensual - :old.salario_mensual));

COMMIT;

END IF;

--SI SE ELIMINA UN REGISTRO

IF DELETING THEN

INSERT INTO auditoria_Colab (num_transac, modif_por, modificacion,
fecha_modif, id_codcolaborador, status_ant, status_act, salario_ant, salario_act,
salario_dif)

VALUES (seq_aud_colab.nextval, USER, 'Registro Eliminado', SYSDATE,
:old.id_codcolaborador, :old.status, :new.status, :old.salario_mensual,
:new.salario_mensual, (:new.salario_mensual - :old.salario_mensual));

END IF;

EXCEPTION

WHEN DUP_VAL_ON_INDEX THEN

DBMS_OUTPUT.PUT_LINE ('Registro Duplicado');

```

        WHEN NO_DATA_FOUND THEN

            DBMS_OUTPUT.PUT_LINE ('Registro No Encontrado');

        WHEN OTHERS THEN

            DBMS_OUTPUT.PUT_LINE ('Error');

    END Trig_Colaborador;

```

Laboratorio N°6

--TRIGGER AUDITORIA PRESTAMOS

```

CREATE OR REPLACE TRIGGER Trig_Prestamos

```

```

    BEFORE INSERT OR UPDATE OR DELETE ON Prestamos

```

```

    FOR EACH ROW

```

```

    BEGIN

```

```

        --SI INSERTA

```

```

        IF INSERTING THEN

```

```

            INSERT INTO aud_Prestamos (num_transac, modif_por, modificacion,
fecha_modif, id_cliente, monto_aprobado_act, monto_pagado_act, saldoactual_act)

```

```

            VALUES (seq_aud_prestamos.nextval, USER, 'Registro Nuevo',
TO_DATE(SYSDATE, 'DD/MM/YYYY HH:MI:SS'), :new.id_Cliente, :new.monto_aprobado, 0,
(:new.monto_aprobado));

```

```

        END IF;

```

```

        --SI SE ACTUALIZA UN REGISTRO

```

```

        IF UPDATING THEN

```

```

            INSERT INTO aud_Prestamos (num_transac, modif_por, modificacion,
fecha_modif, id_cliente, monto_aprobado_ant, monto_aprobado_act,
monto_pagado_ant, monto_pagado_act, saldoactual_ant, saldoactual_act)

```

```

            VALUES (seq_aud_prestamos.nextval, USER, 'Registro Actualizado',
TO_DATE(SYSDATE, 'DD/MM/YYYY HH:MI:SS'), :old.id_Cliente, :old.monto_aprobado,
:new.monto_aprobado, :old.monto_pagado, :new.monto_pagado, :old.saldoactual,
(:new.monto_aprobado - :new.monto_pagado));

```

```

        END IF;

```

--SI SE ELIMINA UN REGISTRO

IF DELETING THEN

INSERT INTO aud_Prestamos (num_transac, modif_por, modificacion,
fecha_modif, id_cliente, monto_aprobado_ant, monto_pagado_ant, saldoactual_ant)

VALUES (seq_aud_prestamos.nextval, USER, 'Registro Eliminado',
TO_DATE(SYSDATE, 'DD/MM/YYYY HH:MI:SS'), :old.id_Cliente, :old.monto_aprobado,
:old.monto_pagado, :old.saldoactual);

END IF;

EXCEPTION

WHEN DUP_VAL_ON_INDEX THEN

DBMS_OUTPUT.PUT_LINE ('Registro Duplicado');

WHEN NO_DATA_FOUND THEN

DBMS_OUTPUT.PUT_LINE ('Registro No Encontrado');

WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE ('Error');

END Trig_Prestamos;

--EJEMPLO

INSERT INTO SUCURSAL VALUES (1, 'San Antonio', 0)

INSERT INTO cliente values (seq_cliente.nextval, '8-975-2460', 'Miguel',
'Pinilla', 'M', TO_DATE('4/11/2001', 'DD/MM/YYYY'), 21, 1)

INSERT INTO TIPO_DE_PRESTAMO VALUES (1, 'PERSONAL')

INSERT INTO PRESTAMOS VALUES (1, 1, 1, SYSDATE, USER, 1500, 0.15,
60, 0, NULL, 1500, 0, 1)

UPDATE PRESTAMOS SET monto_aprobado = 2000 WHERE id_Cliente = 1

delete prestamos where id_cliente = 1

-- TABLA DE AUDITORIA SUCURSAL

```
CREATE TABLE aud_Sucursal (  
    num_transac NUMBER,  
    modif_por VARCHAR2(25) NOT NULL,  
    modificacion VARCHAR2(25) NOT NULL,  
    fecha_modif DATE NOT NULL,  
    montoprestamo_ant NUMBER,  
    montoprestamo_act NUMBER,  
    CONSTRAINT pk_aud_Sucursal PRIMARY KEY (num_transac),  
    CONSTRAINT ck_aud_Sucursal_prestamo_ant CHECK (montoprestamo_ant >= 0),  
    CONSTRAINT ck_aud_Sucursal_prestamo_act CHECK (montoprestamo_act >= 0)  
);
```

CREATE SEQUENCE seq_aud_Sucursal

```
    MINVALUE 1  
    START WITH 1  
    INCREMENT BY 1  
    CACHE 20;
```

-- TRIGGER AUDITORIA SUCURSAL

CREATE OR REPLACE TRIGGER Trig_Sucursal

```
    BEFORE INSERT OR UPDATE OR DELETE ON Sucursal  
    FOR EACH ROW  
  
    BEGIN  
        --AL ACTUALIZAR  
  
        IF UPDATING THEN  
            INSERT INTO aud_Sucursal VALUES (seq_aud_Sucursal.nextval, USER, 'Registro  
Actualizado', TO_DATE(SYSDATE, 'DD/MM/YYYY HH:MI:SS'), :old.montoprestamo,  
:new.montoprestamo);  
        END IF;
```

END Trig_Sucursal;

Evidencias de Implementación

Tabla Préstamos

ID_CLIENTE	ID_TIPO_PRESTAMO	NUM_PRESTAMOS	FECHA_APROBADA	USUARIO	MONTO_APROBADO
21	1	1	08-NOV-22	APEX_PUBLIC_USER	2700
1	1	1	08-NOV-22	APEX_PUBLIC_USER	4000

[Download CSV](#)

Tabla Auditoría de Sucursal

NUM_TRANSAC	MODIF_POR	MODIFICACION	FECHA_MODIF	MONTOPRESTAMO_ANT	MONTOPRESTAMO_ACT
9	APEX_PUBLIC_USER	Registro Actualizado	08-NOV-22	4000	6700
8	APEX_PUBLIC_USER	Registro Actualizado	08-NOV-22	0	4000

[Download CSV](#)

Tabla de Auditoría Préstamos

NUM_TRANSAC	MODIF_POR	MODIFICACION	FECHA_MODIF	ID_CLIENTE	MONTO_APROBADO_ANT	M
5	APEX_PUBLIC_USER	Registro Nuevo	08-NOV-22	21	-	1
6	APEX_PUBLIC_USER	Registro Actualizado	08-NOV-22	21	1000	2
4	APEX_PUBLIC_USER	Registro Eliminado	08-NOV-22	21	3700	

[Download CSV](#)