

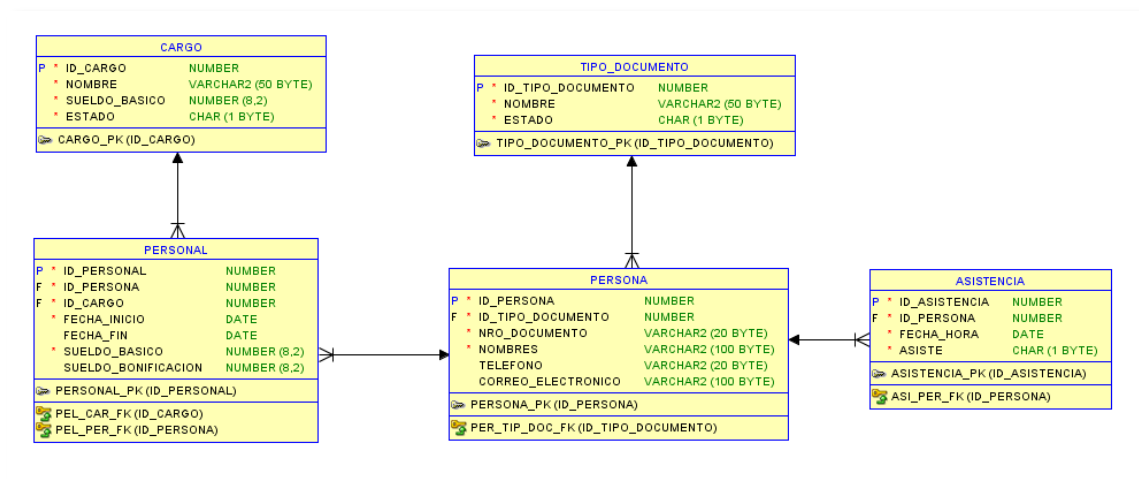
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

Bases de Datos
5ta. Práctica Dirigida
(Primer Semestre 2021)

CURSORES – TRIGGERS

Tema propuesto:

La empresa Rent SAC es una empresa emergente (startup) recién formada que ha conseguido financiamiento y colaboración de personas de distintas partes del mundo. Desea mantener el registro de su personal en un Sistema de Recursos Humanos elaborado por su propio personal. Para este fin, se elaboró una base de datos cuyo modelo se muestra a continuación:



Ejecute el contenido del archivo **INF246_2021-1_LAB5_00_drops.sql** para limpiar las tablas de la sesión dirigida 4 y recrear el modelo nuevamente, si alguna sentencia del drop falla, no hay problema, eso se debe a que no existe la tabla a borrar.

A continuación ejecute el contenido del archivo **INF246_2021-1_LAB5_01_ddl.sql** para crear las tablas de la figura anterior. Luego ejecutar el contenido del archivo **INF246_2021-1_LAB5_02_dml.sql** para popular las tablas.

Ejercicios cursores:

1.- Se pide realizar una función que calcule la bonificación sobre el sueldo básico de un empleado, basado en la siguiente tabla (para la función puede usar directamente los id_cargo):

Cargo	% aumento
Gerente General	20%
Gerente de Operaciones	
Administrador	
Jefe de Imagen	
Analista Comercial	15%
Jefe de Soporte	
Jefe de Proyectos	
Analista Programador	

Programador Programador Junior Analista de Marketing	10%
--	-----

```
CREATE OR REPLACE FUNCTION FN_CALCULAR_BONIF(P_SUELDO NUMBER, P_ID_CARGO
NUMBER) RETURN NUMBER
IS
BEGIN
    IF P_ID_CARGO IN (1, 2, 3, 4) THEN
        RETURN P_SUELDO * 1.2;
    END IF;
    IF P_ID_CARGO IN (5, 6, 7, 8) THEN
        RETURN P_SUELDO * 1.15;
    END IF;
    IF P_ID_CARGO IN (9, 10, 11) THEN
        RETURN P_SUELDO * 1.1;
    END IF;
    RETURN 0; -- SI INGRESA UN CARGO QUE NO EXISTE
END;
```

Pruebas:

```
SELECT FN_CALCULAR_BONIF(25000,1) AS BONIFICACION FROM DUAL;
SELECT FN_CALCULAR_BONIF(7000,7) AS BONIFICACION FROM DUAL;
SELECT FN_CALCULAR_BONIF(2500,11) AS BONIFICACION FROM DUAL;
```

```

BONIFICACION
-----
30000

BONIFICACION
-----
8050

BONIFICACION
-----
2750

```

2.- Se pide crear un procedimiento que permita recorrer a partir de la tabla de **CARGOS** todos los registros de la tabla **PERSONAL** para poder calcular y actualizar el valor del campo **SUELDO_BONIFICACION** usando la función anterior.

```
CREATE OR REPLACE PROCEDURE PR_CALCULAR_BONIF IS
    CURSOR C_CARGOS IS
        SELECT C.ID_CARGO FROM CARGO C WHERE ESTADO = 'A';

    V_ID_CARGO NUMBER;
BEGIN
    OPEN C_CARGOS;
    LOOP
        FETCH C_CARGOS INTO V_ID_CARGO;
        EXIT WHEN C_CARGOS%NOTFOUND;

        UPDATE PERSONAL SET SUELDO_BONIFICACION = FN_CALCULAR_BONIF(SUELDO_BASICO,
V_ID_CARGO)
        WHERE ID_CARGO = V_ID_CARGO;

    END LOOP;
```

```

CLOSE C_CARGOS;
END;

```

Prueba:

```

EXEC PR_CALCULAR_BONIF;

SELECT * FROM PERSONAL;

```

ID_PERSONAL	ID_PERSONA	ID_CARGO	FECHA_INICIO	FECHA_FIN	SUELDO_BASICO	SUELDO_BONIFICACION
1	1	1	02-FEB-16	02-FEB-21	25000	30000
2	2	2	02-FEB-16	02-FEB-21	15000	18000
3	3	3	02-FEB-16	02-FEB-21	16000	19200
4	4	4	02-FEB-16	(null)	10000	11500
5	5	5	02-FEB-16	(null)	5500	6325
6	6	6	01-APR-16	(null)	7000	8050
7	7	7	01-APR-16	(null)	2000	2200
8	8	8	01-APR-16	(null)	4000	4400
9	9	9	05-MAY-16	(null)	5500	6325
10	10	10	16-APR-17	08-AUG-18	2500	2750
11	11	11	01-JAN-20	(null)	12000	14400
12	12	12	01-JUL-20	01-AUG-20	2500	2750

3.- Se pide crear un procedimiento que permita recorrer a partir de la tabla **TIPO_DOCUMENTO** y agregar al número de documento la letra **P** en caso se trate de pasaporte y **C** en caso sea Carnet de Extranjería, además se debe recorrer la tabla **PERSONA** y si se trata de personas con nacionalidad peruana se debe imprimir sus nombres, teléfono y la fecha en la que inicio a laborar. Se debe agregar el código **+51** en los números de teléfonos para mostrarlos.

```

CREATE OR REPLACE PROCEDURE PR_TIPO_DOCUMENTO IS
    CURSOR C_TIP_DOC IS
        SELECT TD.ID_TIPO_DOCUMENTO FROM TIPO_DOCUMENTO TD
        WHERE TD.ESTADO = 'A';

    V_ID_TIP_DOC NUMBER;

    CURSOR C_PERUANO(N_ID_TIP_DOC NUMBER) IS
        SELECT P.NOMBRES, P.TELEFONO, PL.FECHA_INICIO
        FROM PERSONA P, PERSONAL PL
        WHERE P.ID_PERSONA = PL.ID_PERSONA AND P.ID_TIPO_DOCUMENTO =
N_ID_TIP_DOC;
BEGIN
    OPEN C_TIP_DOC;
    LOOP
        FETCH C_TIP_DOC INTO V_ID_TIP_DOC;
        EXIT WHEN C_TIP_DOC%NOTFOUND;

        IF V_ID_TIP_DOC = 2 THEN
            UPDATE PERSONA SET NRO_DOCUMENTO = 'P-' || NRO_DOCUMENTO
            WHERE ID_TIPO_DOCUMENTO = V_ID_TIP_DOC;
        END IF;

        IF V_ID_TIP_DOC = 3 THEN
            UPDATE PERSONA SET NRO_DOCUMENTO = 'C-' || NRO_DOCUMENTO
            WHERE ID_TIPO_DOCUMENTO = V_ID_TIP_DOC;
        END IF;
    END LOOP;
END;

```

```

        IF V_ID_TIP_DOC = 1 THEN
            FOR REG_PERUANO in C_PERUANO(V_ID_TIP_DOC) LOOP
                DBMS_OUTPUT.PUT_LINE('Nombre:' || REG_PERUANO.NOMBRES || '+51-'
                || REG_PERUANO.TELEFONO || REG_PERUANO.FECHA_INICIO);
            END LOOP;
        END IF;
    END LOOP;
    CLOSE C_TIP_DOC;
END;

```

Prueba:

```

SET SERVEROUTPUT ON;
EXEC PR_TIPO_DOCUMENTO;

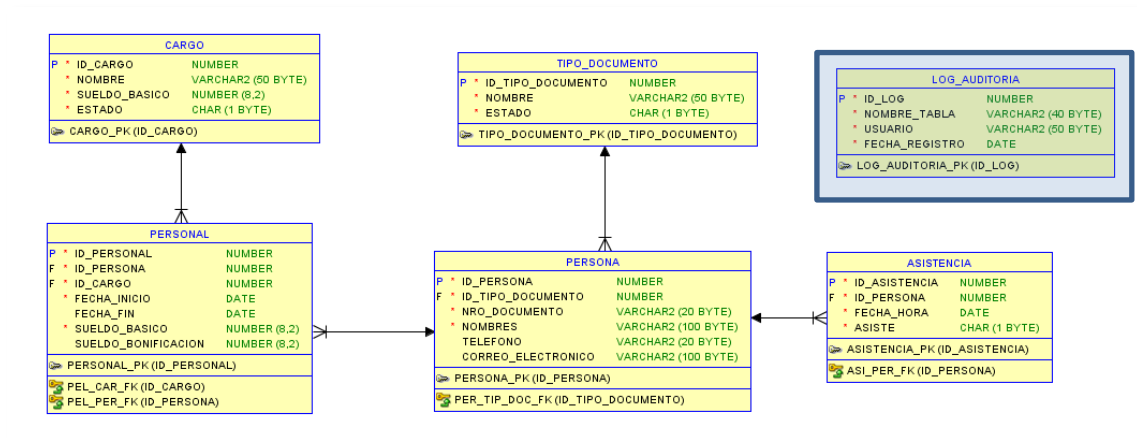
```

Ejercicios triggers:

1.- Se pide llevar la trazabilidad de qué usuarios de base de datos realizan inserciones sobre la tabla **PERSONA**, a manera de auditoría. Esta trazabilidad se llevará sobre una nueva tabla llamada **LOG_AUDITORIA**. El trigger debe registrar el *nombre de la tabla*, el *usuario* que realizó el registro (inserción) y la *fecha* de ese insert, para esos dos últimos atributos se usarán las funciones **USER** y **SYSDATE**.

Ejecute el contenido del archivo **INF246_2021-1_LAB5_03_logauditoria.sql**.

El modelo quedará ahora de la siguiente manera:



El trigger a crear se disparará cada vez que un usuario **inserte** datos en la tabla **PERSONA**, por lo tanto será un trigger **"after insert"**:

```

CREATE OR REPLACE TRIGGER TR_LOG_PERSONA
AFTER INSERT ON PERSONA
FOR EACH ROW
DECLARE
    V_ID_LOG NUMBER;
BEGIN
    SELECT NVL(MAX(ID_LOG), 0) INTO V_ID_LOG FROM LOG_AUDITORIA;

    INSERT INTO LOG_AUDITORIA (ID_LOG, NOMBRE_TABLA, USUARIO, FECHA_REGISTRO)
    VALUES (V_ID_LOG + 1, 'PERSONA', USER, SYSDATE);

```

```
END;
```

Realizamos la prueba de la siguiente manera:

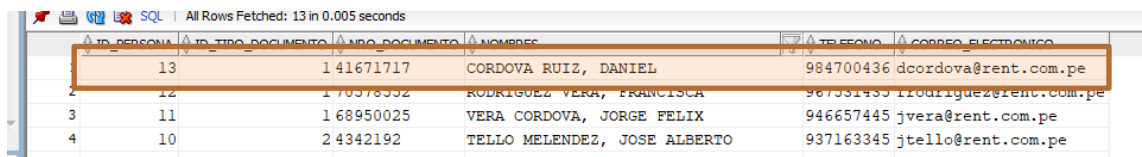
- Insertamos una persona en la tabla **PERSONA**.
- Comprobamos el registro insertado con un query en **PERSONA**.
- Comprobamos el funcionamiento correcto del trigger con un query en **LOG_AUDITORIA**.

Ya que el último ID_PERSONA es 12, usaremos el 13:

```
INSERT INTO PERSONA (ID_PERSONA, ID_TIPO_DOCUMENTO, NRO_DOCUMENTO, NOMBRES,
TELEFONO, CORREO ELECTRONICO) VALUES ( 13, 1, '41671717' , 'CORDOVA RUIZ,
DANIEL', '984700436', 'dcordova@rent.com.pe');
```

Verificamos que se insertó en la tabla PERSONA:

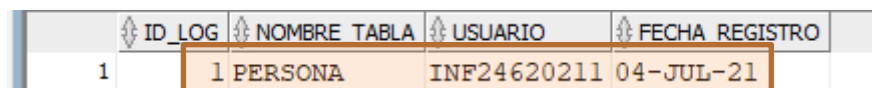
```
SELECT * FROM PERSONA ORDER BY 1 DESC;
```



ID_PERSONA	ID_TIPO_DOCUMENTO	NRO_DOCUMENTO	NOMBRES	TELEFONO	CORREO ELECTRONICO
13	1	41671717	CORDOVA RUIZ, DANIEL	984700436	dcordova@rent.com.pe
12	1	170378332	RODRIGUEZ VERA, FRANCISCA	987331433	frrodriguez@rent.com.pe
11	1	168950025	VERA CORDOVA, JORGE FELIX	946657445	jvera@rent.com.pe
10	1	24342192	TELLO MELENDEZ, JOSE ALBERTO	937163345	jtello@rent.com.pe

Revisamos la tabla afectada por el trigger:

```
SELECT * FROM LOG_AUDITORIA;
```



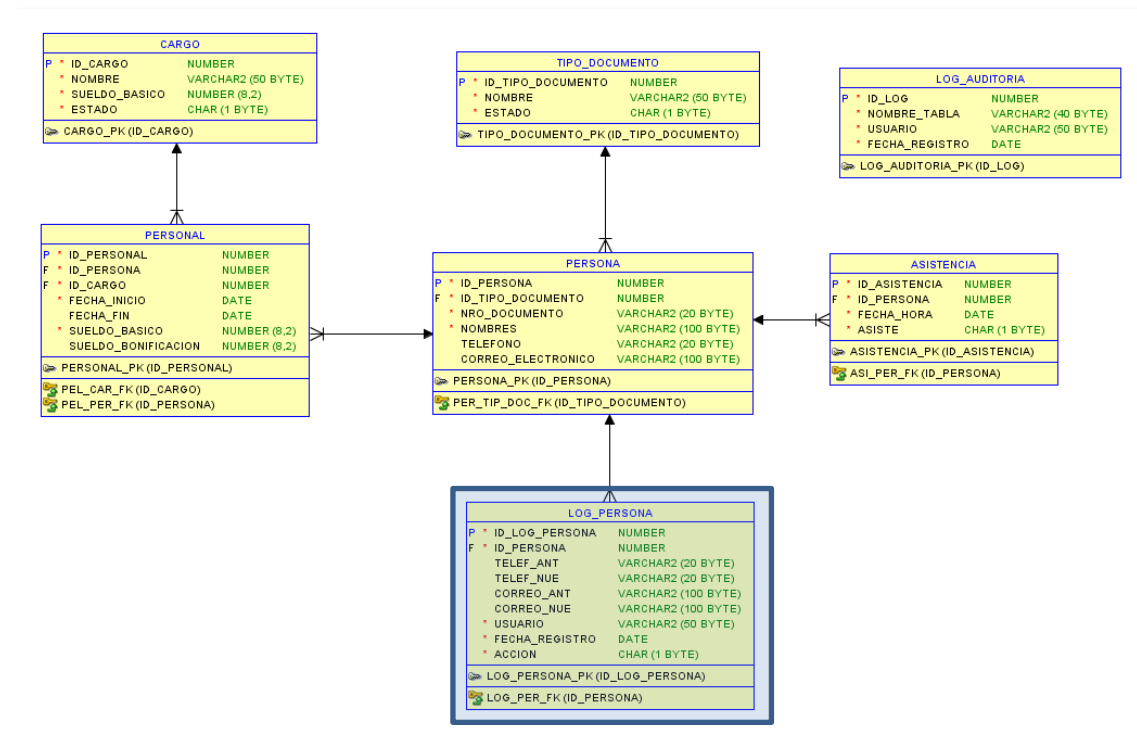
ID_LOG	NOMBRE TABLA	USUARIO	FECHA REGISTRO
1	1 PERSONA	INF24620211	04-JUL-21

Y efectivamente, se realizó la inserción en la tabla de auditoría. Con esto podemos llevar la trazabilidad de los usuarios del sistema que hicieron inserciones (o cualquier otra acción) sobre cualquier tabla del sistema, para este ejemplo: PERSONA.

2.- Se desea tener un mayor control sobre la auditoría de las tabla **PERSONA** y no necesariamente guardar la información de quién y cuándo se realizaron cambios, sino también se quiere llevar un control de la historia de cambios realizados a los datos de las personas, en este caso conocer si hubieron correcciones en los campos **CORREO_ELECTRONICO** y **TELEFONO** (considerar **inserción** y **modificación**). Este registro de cambios se llevará en la tabla **LOG_PERSONA**.

Ejecute el contenido del archivo **INF246_2021-1_LAB5_04_logpersona.sql**.

El modelo quedará ahora de la siguiente manera:



La tabla **LOG_PERSONA** guardará entonces los siguientes registros:

- Si es inserción: los campos que terminan en **_ANT** estarán en **NULL** y los campos que terminan en **_NUE** tendrán los registros nuevos.
- Si es modificación: los campos que terminan en **_ANT** tendrán los datos anteriores al update y los campos que terminan en **_NUE** tendrán los registros nuevos.
- El campo **ACCION** tendrá los valores de **'I'** para insert y **'U'** para update.

El trigger a crear tiene que dispararse cada vez que un usuario **inserte** o **actualice** datos en la tabla **PERSONA**, por lo tanto será un trigger **"after insert or update"**:

```
CREATE OR REPLACE TRIGGER TR_TRAZA_PERSONA
AFTER INSERT OR UPDATE ON PERSONA
FOR EACH ROW
DECLARE
    V_ID_LOG NUMBER;
BEGIN
    SELECT NVL(MAX(ID_LOG_PERSONA), 0) INTO V_ID_LOG FROM LOG_PERSONA;

    IF INSERTING THEN
        INSERT INTO LOG_PERSONA (ID_LOG_PERSONA, ID_PERSONA, TELEF_NUE,
CORREO_NUE, USUARIO, FECHA_REGISTRO, ACCION)
        VALUES (V_ID_LOG + 1, :NEW.ID_PERSONA, :NEW.TELEFONO,
:NEW.CORREO_ELECTRONICO, USER, SYSDATE, 'I');
        END IF;

    IF UPDATING THEN
        INSERT INTO LOG_PERSONA (ID_LOG_PERSONA, ID_PERSONA, TELEF_ANT,
TELEF_NUE, CORREO_ANT, CORREO_NUE, USUARIO, FECHA_REGISTRO, ACCION)
        VALUES (V_ID_LOG + 1, :NEW.ID_PERSONA, :OLD.TELEFONO, :NEW.TELEFONO,
:OLD.CORREO_ELECTRONICO, :NEW.CORREO_ELECTRONICO, USER, SYSDATE, 'U');
        END IF;
END;
```

Realizamos la prueba de la siguiente manera:

- Insertamos una persona en la tabla **PERSONA**.
- Comprobamos el registro insertado con un query en **PERSONA**.
- Comprobamos el funcionamiento correcto del trigger con un query en **LOG_PERSONA**.
- Actualizamos una persona en la tabla **PERSONA**.
- Comprobamos el registro actualizado con un query en **PERSONA**.
- Comprobamos el funcionamiento correcto del trigger con un query en **LOG_PERSONA**.

Cuando es insert:

```
INSERT INTO PERSONA (ID_PERSONA, ID_TIPO_DOCUMENTO, NRO_DOCUMENTO, NOMBRES, TELEFONO, CORREO_ELECTRONICO) VALUES ( 14, 1, '42671717' , 'ARANA QUISPE, DAVID', '944700499', 'darana@rent.com.pe');
```

Verificamos que se insertó en la tabla PERSONA:

```
SELECT * FROM PERSONA ORDER BY 1 DESC;
```

ID_PERSONA	ID_TIPO_DOCUMENTO	NRO_DOCUMENTO	NOMBRES	TELEFONO	CORREO_ELECTRONICO
1	14	142671717	ARANA QUISPE, DAVID	944700499	darana@rent.com.pe
2	13	141671717	CORDOVA RUIZ, DANIEL	984700436	dcordova@rent.com.pe

Revisamos la tabla afectada por el trigger:

```
SELECT * FROM LOG_PERSONA;
```

ID_LOG_PERSONA	ID_PERSONA	TELEF_ANT	TELEF_NUE	CORREO_ANT	CORREO_NUE	USUARIO	FECHA_REGISTRO	ACCION
1	1	14 (null)	944700499 (null)	darana@rent.com.pe	INF24620211	04-JUL-21	I	

Cuando es update:

```
UPDATE PERSONA SET TELEFONO = '956554779', CORREO_ELECTRONICO = 'fmcardsoso@rent.com.pe' WHERE ID_PERSONA = 4;
```

Verificamos que se actualizó en la tabla PERSONA:

```
SELECT * FROM PERSONA WHERE ID_PERSONA = 4;
```

ID_PERSONA	ID_TIPO_DOCUMENTO	NRO_DOCUMENTO	NOMBRES	TELEFONO	CORREO_ELECTRONICO
1	4	3001040878	CARDOSO MARTINEZ, FRANCISCO MIGUEL	956554779	fmcardsoso@rent.com.pe

Revisamos la tabla afectada por el trigger:

```
SELECT * FROM LOG_PERSONA;
```

ID_LOG_PERSONA	ID_PERSONA	TELEF_ANT	TELEF_NUE	CORREO_ANT	CORREO_NUE	USUARIO	FECHA_REGISTRO	ACCION
1	1	14 (null)	944700499 (null)		darana@rent.com.pe	INF24620211 04-JUL-21		I
2	2	4 956554778	956554779	fcardoso@rent.com.pe	fcardoso@rent.com.pe	INF24620211 04-JUL-21		U

Adicionalmente podemos revisar la tabla de la pregunta anterior (trigger on insert TR_LOG_PERSONA en la tabla LOG_AUDITORIA):

```
SELECT ID_LOG, NOMBRE_TABLA, USUARIO, TO_CHAR(FECHA_REGISTRO, 'DD/MM/YYYY
HH24:MI:SS') FROM LOG_AUDITORIA;
```

ID_LOG	NOMBRE_TABLA	USUARIO	TO_CHAR(FECHA_REGISTRO,'DD/MM/YYYYHH24:MI:SS')
1	1 PERSONA	INF24620211	04/07/2021 13:52:42
2	2 PERSONA	INF24620211	04/07/2021 14:28:59

Y comprobamos con el segundo registro que también se disparó el trigger cuando realizamos el insert.

Otra forma: se pueden usar **parámetros** en la sentencia **UPDATING** para que el trigger solo se dispare al actualizar ese campo en particular.

```
CREATE OR REPLACE TRIGGER TR_TRAZA_PERSONA AFTER INSERT OR UPDATE OF TELEFONO,
CORREO_ELECTRONICO ON PERSONA
FOR EACH ROW
DECLARE
    V_ID_LOG NUMBER;
BEGIN
    IF INSERTING THEN
        SELECT NVL(MAX(ID_LOG_PERSONA), 0) INTO V_ID_LOG FROM LOG_PERSONA;
        INSERT INTO LOG_PERSONA (ID_LOG_PERSONA, ID_PERSONA, TELEF_NUE,
CORREO_NUE, USUARIO, FECHA_REGISTRO, ACCION)
        VALUES (V_ID_LOG + 1, :NEW.ID_PERSONA, :NEW.TELEFONO,
:NEW.CORREO_ELECTRONICO, USER, SYSDATE, 'I');
        END IF;

    IF UPDATING('TELEFONO') THEN
        SELECT NVL(MAX(ID_LOG_PERSONA), 0) INTO V_ID_LOG FROM LOG_PERSONA;
        INSERT INTO LOG_PERSONA (ID_LOG_PERSONA, ID_PERSONA, TELEF_ANT,
TELEF_NUE, USUARIO, FECHA_REGISTRO, ACCION)
        VALUES (V_ID_LOG + 1, :NEW.ID_PERSONA, :OLD.TELEFONO, :NEW.TELEFONO,
USER, SYSDATE, 'U');
        END IF;

    IF UPDATING('CORREO_ELECTRONICO') THEN
        SELECT NVL(MAX(ID_LOG_PERSONA), 0) INTO V_ID_LOG FROM LOG_PERSONA;
        INSERT INTO LOG_PERSONA (ID_LOG_PERSONA, ID_PERSONA, CORREO_ANT,
CORREO_NUE, USUARIO, FECHA_REGISTRO, ACCION)
        VALUES (V_ID_LOG + 1, :NEW.ID_PERSONA, :OLD.CORREO_ELECTRONICO,
:NEW.CORREO_ELECTRONICO, USER, SYSDATE, 'U');
        END IF;
END;
```


A diferencia de la forma anterior, tenemos **dos IFs**, uno para el **TELEFONO** y otro para el **CORREO_ELECTRONICO** que se ejecutan de manera independiente si se hace update solo al teléfono, o solo al correo electrónico o si es a ambos, se ejecutarán ambos IFs.

Si probamos solo update al teléfono:

```
UPDATE PERSONA SET TELEFONO = '55555555' WHERE ID_PERSONA = 1;

SELECT * FROM LOG_PERSONA;
```

ID_LOG_PERSONA	ID_PERSONA	TELEF_ANT	TELEF_NUE	CORREO_ANT	CORREO_NUE	USUARIO	FECHA_REGISTRO	ACCION
1	1	14 (null)	944700499 (null)		darana@rent.com.pe	INF24620211	04-JUL-21	I
2	2	4 956554778	956554779	fcardoso@rent.com.pe	fcardoso@rent.com.pe	INF24620211	04-JUL-21	U
3	3	1 984745436	555555555 (null)		(null)	INF24620211	04-JUL-21	U

Si probamos solo update al correo electrónico:

```
UPDATE PERSONA SET CORREO_ELECTRONICO = 'correo_diferente@rent.com.pe' WHERE
ID_PERSONA = 6;

SELECT * FROM LOG_PERSONA;
```

ID_LOG_PERSONA	ID_PERSONA	TELEF_ANT	TELEF_NUE	CORREO_ANT	CORREO_NUE	USUARIO	FECHA_REGISTRO	ACCION
1	1	14 (null)	944700499 (null)		darana@rent.com.pe	INF24620211	04-JUL-21	I
2	2	4 956554778	956554779	fcardoso@rent.com.pe	fcardoso@rent.com.pe	INF24620211	04-JUL-21	U
3	3	1 984745436	555555555 (null)		(null)	INF24620211	04-JUL-21	U
4	4	6 (null)	(null)	spisconte@rent.com.pe	correo_diferente@rent.com.pe	INF24620211	04-JUL-21	U

Si probamos update a ambos:

```
UPDATE PERSONA SET TELEFONO = '777777777', CORREO_ELECTRONICO =
'mi_nuevo_correo@rent.com.pe' WHERE ID_PERSONA = 12;

SELECT * FROM LOG_PERSONA;
```

ID_LOG_PERSONA	ID_PERSONA	TELEF_ANT	TELEF_NUE	CORREO_ANT	CORREO_NUE	USUARIO	FECHA_REGISTRO	ACCION
1	1	14 (null)	944700499 (null)		darana@rent.com.pe	INF24620211	04-JUL-21	I
2	2	4 956554778	956554779	fcardoso@rent.com.pe	fcardoso@rent.com.pe	INF24620211	04-JUL-21	U
3	3	1 984745436	555555555 (null)		(null)	INF24620211	04-JUL-21	U
4	4	6 (null)	(null)	spisconte@rent.com.pe	correo_diferente@rent.com.pe	INF24620211	04-JUL-21	U
5	5	12 967531435	777777777 (null)		(null)	INF24620211	04-JUL-21	U
6	6	12 (null)	(null)	frodriguez@rent.com.pe	mi_nuevo_correo@rent.com.pe	INF24620211	04-JUL-21	U

5 de julio de 2021
OTM, MGP