--Creación de Tablas

DROP TABLE gasto;

DROP TABLE empleo;

DROP TABLE cliente;

DROP TABLE auditoria;

CREATE TABLE cliente( ced NUMBER(8) PRIMARY KEY,

nom VARCHAR2(10) NOT NULL);

INSERT INTO cliente (ced, nom) values (1, 'Daniel');

INSERT INTO cliente (ced, nom) values (2, 'Santa');

CREATE TABLE empleo( ced NUMBER(8) REFERENCES cliente,

nit\_empresa INTEGER,

valor\_mensual NUMBER(6) NOT NULL,

PRIMARY KEY (ced, nit\_empresa));

INSERT INTO empleo (ced, nit\_empresa, valor\_mensual) values (1, 1, 400);

INSERT INTO empleo (ced, nit\_empresa, valor\_mensual) values (2, 2, 200);

INSERT INTO empleo (ced, nit\_empresa, valor\_mensual) values (1, 2, 100);

INSERT INTO empleo (ced, nit\_empresa, valor\_mensual) values (2, 1, 700);

CREATE TABLE gasto( cod\_gasto NUMBER(8) PRIMARY KEY,

ced NUMBER(8) REFERENCES cliente,

valor\_mensual NUMBER(6),

desc\_gasto VARCHAR2(10));

INSERT INTO gasto (cod\_gasto, ced, valor\_mensual, desc\_gasto) values (1, 1, 400, 'Hola');

INSERT INTO gasto (cod\_gasto, ced, valor\_mensual, desc\_gasto) values (2, 1, 400, 'Hola');

INSERT INTO gasto (cod\_gasto, ced, valor\_mensual, desc\_gasto) values (3, 2, 400, 'Hola');

INSERT INTO gasto (cod\_gasto, ced, valor\_mensual, desc\_gasto) values (4, 2, 400, 'Hola');

INSERT INTO gasto (cod\_gasto, ced, valor\_mensual, desc\_gasto) values (5, 1, 200, 'Hola');

CREATE TABLE auditoria( codigo NUMBER(5) PRIMARY KEY,

fecha VARCHAR2(10) NOT NULL,

tabla VARCHAR2(50) NOT NULL);

--INSERT INTO auditoria (codigo, fecha, tabla) values (1, '07-SEP-19', 'gasto');

--INSERT INTO auditoria (codigo, fecha, tabla) values (2, '08-SEP-19', 'empelo');

**--1**

**--Trigger que registra en la tabla AUDITORIA cualquier cambio en las demás tablas**

--Trigger de cambio en gasto

CREATE OR REPLACE TRIGGER gasto\_editado BEFORE INSERT OR DELETE OR UPDATE ON GASTO FOR EACH ROW

DECLARE

ultimo NUMBER(5);

BEGIN

ultimo := 0;

SELECT COUNT(codigo) INTO ultimo FROM AUDITORIA;

INSERT INTO AUDITORIA VALUES(ultimo + 1, SYSDATE, 'gasto');

END;

/

--Trigger de cambio en cliente

CREATE OR REPLACE TRIGGER cliente\_editado BEFORE INSERT OR DELETE OR UPDATE ON cliente FOR EACH ROW

DECLARE

ultimo NUMBER(5);

BEGIN

ultimo := 0;

SELECT COUNT(codigo) INTO ultimo FROM AUDITORIA;

INSERT INTO AUDITORIA VALUES(ultimo + 1, SYSDATE, 'cliente');

END;

/

--Trigger de cambio en empleo

CREATE OR REPLACE TRIGGER empleo\_editado BEFORE INSERT OR DELETE OR UPDATE ON empleo FOR EACH ROW

DECLARE

ultimo NUMBER(5);

BEGIN

ultimo := 0;

SELECT COUNT(codigo) INTO ultimo FROM AUDITORIA;

INSERT INTO AUDITORIA VALUES(ultimo + 1, SYSDATE, 'empleo');

END;

/

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Hola');

INSERT INTO cliente (ced, nom) values (3, 'Héctor');

INSERT INTO empleo (ced, nit\_empresa, valor\_mensual) values (3, 2, 800);

INSERT INTO gasto (cod\_gasto, ced, valor\_mensual, desc\_gasto) values (8, 3, 900, 'Hola');

END;

/

**--2**

**--Triggers para que ningún cliente gaste más de lo que gane**

CREATE OR REPLACE TRIGGER gasta\_mucho\_gasto BEFORE INSERT OR UPDATE ON gasto FOR EACH ROW

DECLARE

gastoTot gasto.valor\_mensual%TYPE;

salarioTot empleo.valor\_mensual%TYPE;

BEGIN

SELECT SUM(valor\_mensual) into gastoTot FROM gasto WHERE CED = :NEW.CED;

SELECT SUM(valor\_mensual) into salarioTot FROM empleo WHERE CED = :NEW.CED;

IF salarioTot IS NULL THEN

salarioTot := 0;

END IF;

IF gastoTot IS NULL THEN

gastoTot := 0;

END IF;

IF INSERTING THEN

gastoTot := gastoTot + :NEW.valor\_mensual;

IF salarioTot < gastoTot THEN

RAISE\_APPLICATION\_ERROR(-20505,'Con este nuevo gasto, el empleado gastaría más de lo que gana. No se agregará el gasto');

END IF;

END IF;

IF UPDATING THEN

gastoTot := gastoTot + :NEW.valor\_mensual - :OLD.valor\_mensual;

IF salarioTot < gastoTot THEN

RAISE\_APPLICATION\_ERROR(-20505,'Con este nuevo gasto, el empleado gastaría más de lo que gana. No se actualizará el gasto');

END IF;

END IF;

END;

/

CREATE OR REPLACE TRIGGER gasta\_mucho\_empleo BEFORE DELETE OR UPDATE ON empleo FOR EACH ROW

DECLARE

gastoTot gasto.valor\_mensual%TYPE;

salarioTot empleo.valor\_mensual%TYPE;

BEGIN

SELECT SUM(valor\_mensual) into gastoTot FROM gasto WHERE CED = :NEW.CED;

SELECT SUM(valor\_mensual) into salarioTot FROM empleo WHERE CED = :NEW.CED;

IF salarioTot IS NULL THEN

salarioTot := 0;

END IF;

IF gastoTot IS NULL THEN

gastoTot := 0;

END IF;

IF DELETING THEN

gastoTot := gastoTot - :NEW.valor\_mensual;

IF salarioTot < gastoTot THEN

RAISE\_APPLICATION\_ERROR(-20505,'Con este nuevo gasto, el empleado gastaría más de lo que gana. No se eliminará el empleo');

END IF;

END IF;

IF UPDATING THEN

gastoTot := gastoTot + :NEW.valor\_mensual - :OLD.valor\_mensual;

IF salarioTot < gastoTot THEN

RAISE\_APPLICATION\_ERROR(-20505,'Con este nuevo gasto, el empleado gastaría más de lo que gana. No se actualizará el gasto');

END IF;

END IF;

END;

/

BEGIN

INSERT INTO cliente (ced, nom) values (3, ‘Hector’);

INSERT INTO empleo (ced, nit\_empresa, valor\_mensual) values (3, 1, 400);

INSERT INTO gasto (cod\_gasto, ced, valor\_mensual, desc\_gasto) values (6, 3, 300, 'Hola');

INSERT INTO gasto (cod\_gasto, ced, valor\_mensual, desc\_gasto) values (7, 3, 300, 'Hola');

UPDATE gasto SET valor\_mensual = 500 WHERE cod\_gasto = 6;

END;

/

SELECT \* FROM cliente;

SELECT \* FROM empleo;

SELECT \* FROM gasto;

SELECT \* FROM auditoria ORDER BY codigo desc;