

Ejercicio 1

Crear una tabla que almacene datos de personas

Vamos a registrar los siguientes datos:

(dni, apellido, nombre, fecha de nacimiento y estado civil), siendo dni la clave primaria.

Tenemos que tener en cuenta las siguientes restricciones:

- 1) La persona tiene que ser mayor de 18 años
- 2) El apellido no puede estar vacío.

Registramos el estado civil: **SOLTERO, CASADO, VIUDO, DIVORCIADO**

Sabiendo que el estado civil posee las siguientes restricciones:

- 1) no puede pasar de SOLTERO a VIUDO, DIVORCIADO
- 2) No puede pasar de CASADO, VIUDO, DIVORCIADO a SOLTERO
- 3) No puede pasar de VIUDO a DIVORCIADO
- 4) De DIVORCIADO no puede pasar a VIUDO

Todas las otras transiciones de estado civil están permitidas.

Creamos la tabla

```
create table persona(  
dni integer primary key,  
apellido varchar(30),  
nombre varchar(30),  
fecnac date,  
estadoCivil varchar(10),  
constraint CH_Persona_EstadoCivil check (estadoCivil in  
( 'SOLTERO', 'CASADO', 'VIUDO', 'DIVORCIADO' ) )  
);
```

Vamos a realizar dos trigger

Primero una función

```
CREATE OR REPLACE FUNCTION func_e() RETURNS TRIGGER AS $funcemp$  
DECLARE  
edad smallint ;  
estadocivil varchar(10);  
BEGIN  
NEW.estadoCivil := UPPER(NEW.estadoCivil);  
edad := date_part('year',age(NEW.fecnac));  
IF NEW.apellido = '' THEN
```

```
        RAISE EXCEPTION 'no puede tener apellido vacío';
    END IF;
    IF edad <= '18' THEN
        RAISE EXCEPTION 'no puede ser menor de 18 años';
    END IF;

    RETURN NEW;
END; $funcemp$ LANGUAGE plpgsql;
```

Y la función TRIGGER que lo llama

```
CREATE TRIGGER trigger_e BEFORE INSERT OR UPDATE ON persona
FOR EACH ROW EXECUTE PROCEDURE func_e();
```

Veamos que realizamos tres controles ¿cuáles son?

Luego vamos a crear otra función para controlar las transiciones de estado civil

```
CREATE OR REPLACE FUNCTION func_p() RETURNS TRIGGER AS $funcemp$
DECLARE
BEGIN
    NEW.estadoCivil := UPPER(NEW.estadoCivil);
    if OLD.estadoCivil = 'SOLTERO' AND (NEW.estadoCivil = 'VIUDO' or
    NEW.estadoCivil='DIVORCIADO') THEN
        RAISE EXCEPTION 'ERROR de transición en estado civil';
    END IF;

    if (OLD.estadoCivil = 'CASADO' or OLD.estadoCivil = 'DIVORCIADO' OR OLD.estadoCivil =
    'VIUDO') AND (NEW.estadoCivil = 'SOLTERO') THEN
        RAISE EXCEPTION 'ERROR de transición en estado civil';
    END IF;

    if OLD.estadoCivil = 'DIVORCIADO' AND (NEW.estadoCivil = 'VIUDO') THEN
        RAISE EXCEPTION 'ERROR de transición en estado civil';
    END IF;

    if OLD.estadoCivil = 'VIUDO' AND (NEW.estadoCivil = 'DIVORDIADO') THEN
        RAISE EXCEPTION 'ERROR de transición en estado civil';
    END IF;

    RETURN NEW;
END; $funcemp$ LANGUAGE plpgsql;
```

Y la función que lo llama

```
CREATE TRIGGER trigger_p BEFORE UPDATE ON persona
FOR EACH ROW EXECUTE PROCEDURE func_p();
```

Realice distintas operaciones de inserción y modificación en la tabla personas para corroborar que funcione bien el disparador

EJERCICIO 2

Cree un TRIGGER de auditoría, que registre en una tabla las operaciones realizadas en la tabla persona, debiendo registrarse:

- 1) Que operaciones se hizo (insert/delete/update)
- 2) Fecha-hora en que se realizó
- 3) Nombre de la tabla en que se originó
- 4) Usuario que realizó la operación
- 5) Valor anterior
- 6) Valor nuevo

Para realizar este ejercicio deberá buscar en la documentación de postgres las funciones que tienen los datos que precisa.

Guarde las operaciones en un script y describa los resultados obtenidos.