UCR - ECCI - Bases de Datos

Fabián Orozco Chaves - B95690

Daniel Escobar Giraldo - C02748

Grupo 01 - II Ciclo 2022

Laboratorio 5 - Programación de disparadores simples

Pregunta de ejercicio 3:

¿Qué tipo de disparador usaría: after o instead-of delete? ¿Hay alguno de

estos tipos que no podría o no debería usar por alguna razón? Justifique su

respuesta.

El disparador after no se puede utilizar porque la tabla Asistente tiene la

restricción de on delete no action, por lo tanto si se trata de eliminar un estudiante y

luego el DBMS no va a permitir la acción y no se activa el disparador after. Es decir

que se debe utilizar el disparador instead of delete, para poder eliminar primero el

asistente y luego eliminar al estudiante.

i. Al borrar un estudiante que no tiene asistentes asociados, el where no encuentra

ninguna cédula (llave foránea de asistente) en la tabla asistente, por lo que no

procede a hacer delete sobre esa tabla, sin embargo, sí lo hace para la tupla

respectiva en la tabla estudiante.

ii. Al borrar un estudiante que tiene asistentes asociados, el trigger de tipo INSTEAD

OF entra en acción antes de realizar el DELETE, el cuál elimina primeramente el

asistente asociado a la cédula respectiva y después al estudiante, evitando invocar

el ON DELETE NO ACTION, logrando con ello, simular el efecto ON DELETE

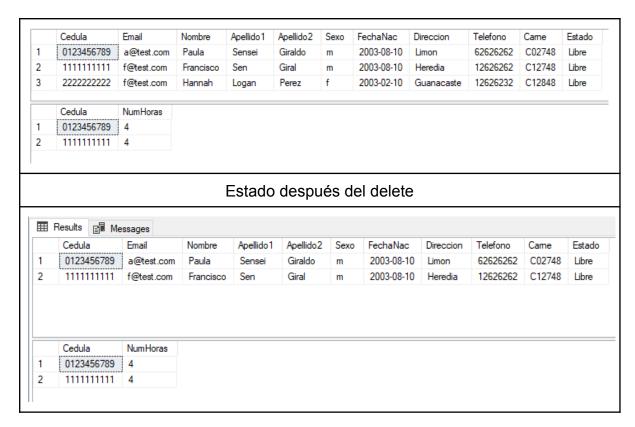
CASCADE.

Capturas de pantalla mostrando el comportamiento al borrar un estudiante

que no tiene asistentes asociados:

Se utilizó el comando: delete from Estudiante where Cedula = '2222222222'

Estado antes del delete



Capturas de pantalla mostrando el comportamiento al borrar un estudiante que si tiene asistentes asociados:

Se utilizó el comando: delete from Estudiante where Cedula = '0123456789'



Pregunta de ejercicio 4:

¿Qué tipo de disparador usaría: after o instead-of insert? ¿Hay alguno de estos tipos que no debería usar por alguna razón? Justifique su respuesta.

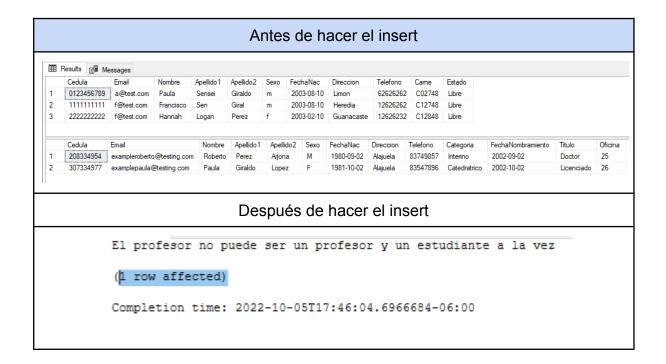
Hay al menos dos enfoques a considerar, si se quiere que un estudiante pase a condición de profesor o viceversa, se podría utilizar un trigger after, para que agregue al estudiante o profesor y luego consulte la tabla disjunta para ver si existe allí y de ser así, eliminarlo.

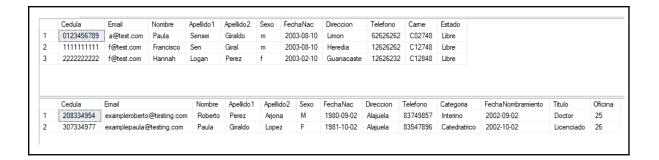
El otro enfoque es considerar solamente mantener la restricción de disyunción y aplicar un trigger INSTEAD OF de forma que si se intenta ingresar una tupla con una cédula ya existente en la otra tabla, se muestra un mensaje de error y se anula el INSERT.

Optamos por el segundo enfoque.

Capturas de pantalla mostrando el comportamiento al insertar un estudiante: Se utilizó el comando:

```
insert into Profesor values('0123456789', 'a@test.com',
'Paula','Sensei', 'Giraldo', 'm', '2003-08-10',
'Limon','62626262','Interino', '2002-09-02', 'Licenciado', 30)
```





Capturas de pantalla mostrando el comportamiento al insertar un profesor: Se utilizó el comando:

```
insert into Estudiante values('208334954', 'u@test.com', 'Paula',
'Sensei','Giraldo','m','2003-08-10','Limon','62626262','C02748',
'Libre')
```

