Parte 2

Primeramente se agregaron nuevas tuplas a Carrera y Empadronado_En para ver mejor el comportamiento del trigger al eliminar varias tuplas en cascada. Con estas nuevas tuplas, las tablas de Carrera y Empadronado En se ven así:

	Codigo	Codigo Nombre		An	hoCreacion		
1	11	11 Filolog		19	08-01-01		
2	12 Medic		ina 18		78-01-01		
3	13 Histori		a 1965-01-01		65-01-01		
4	14 Artes		plásticas	1925-01-01			
	CedEstudiante		CodCarrera		Fechalngreso	FechaGraduacion	
1	1234567	809	12		1980-01-01	1989-01-01	
2	1234567890		11		2000-01-01	2006-01-01	
3	1234567890		13		2005-01-01	2010-01-01	
4	1234567	1234567890			2011-01-01	2015-01-01	
5	1981723	819	11		NULL	NULL	
6	1981723819		12		2019-01-01	2024-01-01	

Ahora, se tuvo que modificar el trigger para hacer que funcione independientemente de la restricción establecida en ON DELETE NO ACTION en la columna CodCarrera, y así poder eliminar (sin quitar la restricción) varias tuplas a la vez en vez de solo una. El nuevo trigger se ve así:

```
-- Crear el trigger AFTER DELETE

CREATE TRIGGER BorradoEnCascada
ON dbo.Carrera

INSTEAD OF DELETE
AS

BEGIN

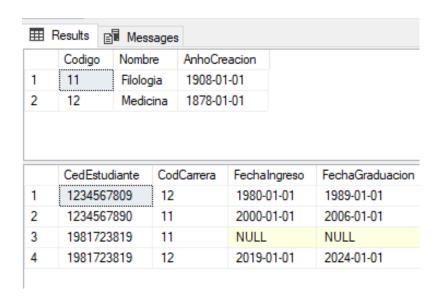
-- Eliminar las tuplas de Empadronado_en asociadas a la carrera eliminada
DELETE FROM dbo.Empadronado_en
WHERE CodCarrera IN (SELECT deleted.Codigo FROM deleted);

DELETE FROM dbo.Carrera
WHERE Codigo IN (SELECT Codigo FROM DELETED);

END;
```

Al cambiarlo a instead of delete y agregar el Delete from Carrera, se asegura de eliminar una o más tuplas independientemente si existe la restricción de ON DELETE NO ACTION en CodCarrera. Anteriormente solo se podía hacer esto si y solo si no existía la restricción, lo cual tenía sentido, sin embargo el enunciado en este laboratorio pide explícitamente que pueda efectuarse la operación, asumiendo que la restricción debe mantenerse

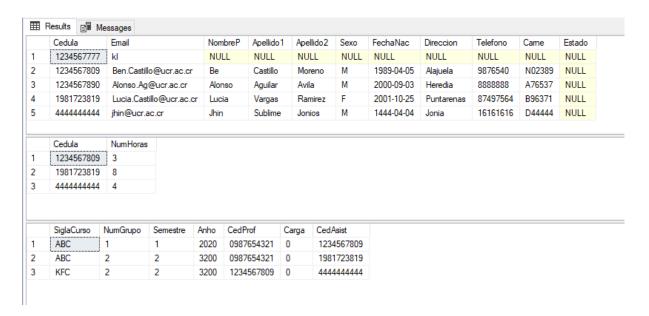
Ahora, al ejecutar DELETE FROM Carrera WHERE Codigo IN ('13', '14'); obtenemos lo siguiente:



Se puede comprobar que la eliminación en cascada dada por el trigger, se efectúa para varias tuplas, sin importar que exista la restricción de ON DELETE NO ACTION.

Parte 3

Para la parte 3, hubo que introducir nuevas tuplas a las tablas Estudiante, Asistente y Grupo. Con estas nuevas tuplas, ahora las tablas se ven así:



Para la creación de la vista, se ejecutaron las siguientes instrucciones:

```
CREATE VIEW NombramientosAsistentes AS

SELECT

A.Cedula AS CedulaAsistente,

CONCAT(G.Anho, '-', G.Semestre) AS Semestre,

COUNT(G.NumGrupo) AS CantidadGruposNombrados

FROM

Asistente A

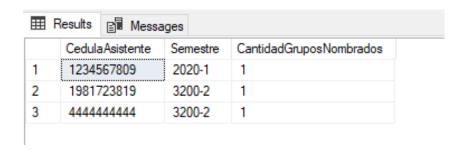
JOIN Grupo G ON A.Cedula = G.CedAsist

GROUP BY

A.Cedula,

CONCAT(G.Anho, '-', G.Semestre);
```

Cuando se ejecuta un SELECT * FROM Nombramientos Asistentes; obtenemos el siguiente resultado de la vista:



Parte 4

Programe un trigger que implemente la siguiente restricción de negocio: "Un asistente no puede ser nombrado en más de 3 grupos por semestre".

NOTA: Los screenshots se hicieron con la restriccion de que un estudiante no puede estar en mas de 2 grupos, en vez de 3, porque se tenia un ">=" en vez de solo ">" en el codigo del trigger al momento de hacer los screenshots, pero la logica que debe tenerse si es la misma

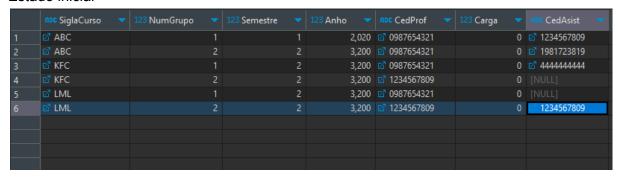
Codigo del trigger para esta parte

```
CREATE TRIGGER trg_limite_de_3_asistencias
ON dbo.Grupo
AFTER UPDATE
 -INSERTED table has the added rows
 -DELETED table has the old rows
AS
BEGIN
         DECLARE @asistentes_fuera_del_limite TABLE(
         CedAsist char(10)
        INSERT INTO @asistentes_fuera_del_limite
         SELECT n.CedulaAsistente FROM NombramientosAsistentes n
         WHERE n.CantidadGruposNombrados > 3;
         IF EXISTS(
                 SELECT * FROM @asistentes_fuera_del_limite
         BEGIN
                  --Borrar las filas que violan la regla usando la tabla que tiene las filas que violan la regla, lo que hace que
                 DELETE g FROM dbo.Grupo g
                 INNER JOIN @asistentes_fuera_del_limite a
                 ON a.CedAsist = g.CedAsist
         END--END de check de la regla
END
```

Parte 4.i

Al actualizar un grupo de manera que el asistente no sobrepase el límite de 3 grupos por semestre

Estado inicial



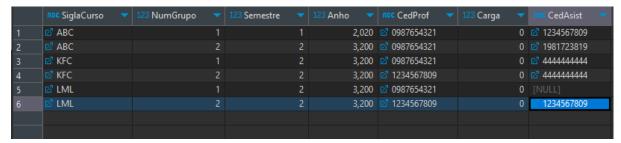
Al actualizar la tupla con LML que tiene NULL en asistente con el siguiente script:

UPDATE C17226.dbo.Grupo
SET CedAsist=N'4444444444'
WHERE SiglaCurso=N'LML' AND NumGrupo=1 AND Semestre=2 AND Anho=3200;

Parte 4.ii

Al actualizar un grupo de manera que el asistente sobrepase el límite de 3 grupos por semestre.

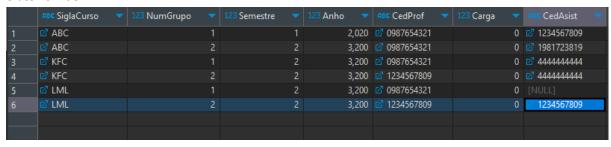
Estado inicial



Al actualizar la tupla que tiene NULL en asistente con el siguiente script:

UPDATE C17226.dbo.Grupo
SET CedAsist=N'44444444444'
WHERE SiglaCurso=N'LML' AND NumGrupo=1 AND Semestre=2 AND Anho=3200;

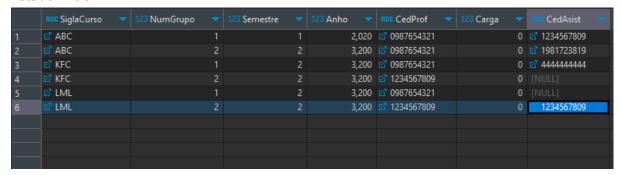
Obtenemos



Parte 4.iii

Al actualizar dos o más grupos que juntos no sobrepasan el límite de 3 grupos por semestre para el asistente.

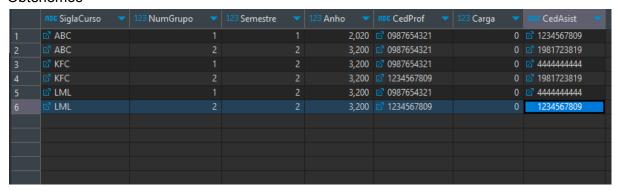
Estado inicial



Al actualizar las 2 tuplas con NULL en asistente con el siguiente script:

UPDATE C17226.dbo.Grupo
SET CedAsist=N'1981723819'
WHERE SiglaCurso=N'KFC' AND NumGrupo=2 AND Semestre=2 AND Anho=3200;
UPDATE C17226.dbo.Grupo
SET CedAsist=N'444444444'
WHERE SiglaCurso=N'LML' AND NumGrupo=1 AND Semestre=2 AND Anho=3200;

Obtenemos



Parte 4.iv

Al actualizar dos o más grupos que juntos sobrepasan el límite de 3 grupos por semestre para el asistente.

Estado inicial

	ABC SiglaCurso	123 NumGrupo	123 Semestre	123 Anho	ABC CedProf 🔻	123 Carga 🔻	ABC CedAsist 🔻
1	☑ ABC	1	1	2,020	☑ 0987654321	0	☑ 1234567809
2	☑ ABC	2		3,200	☑ 0987654321		☑ 1981723819
3	☑ KFC	1	2	3,200	☑ 0987654321		☑ 4444444444
4	☑ KFC	2		3,200	☑ 1234567809		[NULL]
5	☑ LML	1	2	3,200	☑ 0987654321		[NULL]
6	☑ LML	2		3,200	☑ 1234567809		1234567809

Al actualizar las 2 tuplas con NULL en asistente con el siguiente script:

UPDATE C17226.dbo.Grupo
SET CedAsist=N'4444444444'
WHERE SiglaCurso=N'KFC' AND NumGrupo=2 AND Semestre=2 AND Anho=3200;
UPDATE C17226.dbo.Grupo
SET CedAsist=N'4444444444'
WHERE SiglaCurso=N'LML' AND NumGrupo=1 AND Semestre=2 AND Anho=3200;

Obtenemos

