

Lab 6: Programación de Disparadores con Cursores y Vistas

Objetivo: Crear disparadores con cursores y vistas en SQL Server.

Instrucciones generales: Este laboratorio lo pueden trabajar en parejas, si lo desean. En tal caso, harían una sola entrega por pareja, indicando quiénes trabajaron juntos en el reporte, pero ambos deben correr el *script* en su base de datos personal.

Guía de Trabajo

Abra la aplicación SQL Server Management Studio y conéctese al servidor 10.1.4.55. Use la base de datos llamada **BD_carné**. Recuerden escribir al inicio de la consulta “use <Nombre de la BD>”. Realice los siguientes ejercicios con base en el esquema relacional simplificado de la BD Universidad (ver [Figura 1](#) y su diagrama ER en la [Figura 2](#)).

1. **Programa un disparador** que implemente la siguiente restricción de negocio: “Un estudiante no puede matricular más de 18 créditos por semestre”. El sistema no debe permitir inserciones que generen una carga mayor a 18 créditos para un estudiante en un semestre. Asuma que solo se inserta una tupla de Lleva a la vez. Pruebe el comportamiento del disparador en las siguientes condiciones:
 - i. Al insertar una tupla de la tabla `Lleva` que no sobrepasa el límite de 18 créditos para un estudiante en un semestre.
 - ii. Al insertar una tupla de la tabla `Lleva` que sobrepasa el límite de 18 créditos para un estudiante en un semestre.

El reporte de laboratorio debe incluir el código SQL del *trigger* y las capturas de pantalla que evidencien su correcto funcionamiento en las condiciones (i) y (ii). En particular, deben realizar una consulta sobre la tabla `Lleva` y tomar una captura de pantalla de su estado justo antes de insertar una tupla. Luego, deben ejecutar el comando de inserción y tomar una captura de pantalla del resultado del comando. Finalmente, deben volver a ejecutar la consulta sobre la tabla `Lleva` y tomar una captura de pantalla para evidenciar que hubo (o no) un cambio en su estado.

2. **Analice** si el disparador que programó en el ejercicio anterior funciona bien cuando se insertan varias tuplas de la tabla `Lleva` a la vez (en un mismo comando SQL¹). Si su disparador no maneja bien este caso, **modifíquelo** para que permita la inserción de varias tuplas a la vez (considere usar cursores). Si su disparador maneja bien este caso, entonces **explique** por qué funciona y además ofrezca evidencia de ello.

En cualquier caso, pruebe el comportamiento del disparador en las siguientes condiciones:

¹ Para más información sobre este comando, ver <https://www.srcofigofuente.es/sql-insert-multiple>

- i. Al insertar dos o más tuplas de la tabla `Lleva` que juntas no sobrepasan el límite de 18 créditos para un estudiante en un semestre.
- ii. Al insertar dos o más tuplas de la tabla `Lleva` que juntas sobrepasan el límite de 18 créditos para un estudiante en un semestre.

El reporte de laboratorio debe incluir el código SQL del disparador modificado (si lo tuvo que modificar) y las capturas de pantalla que evidencien su correcto funcionamiento las ambas condiciones (i) y (ii). Tomen capturas de pantallas como se indicó en el ejercicio anterior e inclúyanlas en el reporte.

3. **Construya** una **vista** virtual llamada “EstudiantesPorGrupo” que provea la siguiente información sobre cada grupo: (i) su sigla, número de grupo, semestre, año, y (ii) la cantidad de estudiantes matriculados en él. El atributo `CantEstud` es un atributo derivado de la entidad `Grupo`, según el diagrama ER (Figura 2), lo que significa que no se almacena sino que se calcula. La vista permite entonces calcular el valor de dicho atributo, contando la cantidad de estudiantes matriculados en cada grupo. Incluya en el reporte de laboratorio el comando SQL usado para crear la vista.
 - a. Haga una consulta sobre la vista. (Puede que necesite agregar datos a las tablas base de la vista para que la consulta no resulte vacía.) Haga una captura de pantalla con el resultado de esta consulta, e inclúyala en el reporte de laboratorio junto con el comando SQL de la consulta.
 - b. Matricule (o desmatricule) un estudiante en un grupo (inserte o borre una tupla en la tabla `Lleva`). Incluya en el reporte de laboratorio el comando SQL ejecutado.
 - c. Ejecute nuevamente la consulta del paso (a) para verificar que el cambio producido por el paso (b) se refleje automáticamente en la vista. Haga una captura de pantalla con el resultado de esta consulta, e inclúyala en el reporte de laboratorio.
4. Suponga que se decide almacenar el atributo `CantEstud` en la tabla `Grupo` en lugar de hacerlo derivado. ¿Qué solución daría usted para mantener el valor de dicho atributo actualizado automáticamente? No debe implementar la solución, sólo describirla. En el ejercicio anterior, el atributo `CantEstud` se mantenía actualizado mediante una vista, ¿cuál de las dos soluciones le parece mejor y por qué?
5. Envíe su trabajo a través de la plataforma virtual del curso. La entrega debe contener:
 - a. Un archivo pdf con las capturas de pantalla requeridas y las respuestas a las preguntas planteadas (dejando claro cuál ejercicio responden), así como cualquier otra justificación que consideren oportuna.
 - b. El archivo *sql* con comentarios que indiquen la parte de esta guía a la que corresponde cada comando. Verifique que el *script* se ejecute sin errores.

1. **ESTUDIANTE** (Cédula, Email, NombreP, Apellido1, Apellido2, Sexo, FechaNac, Dirección, Teléfono, Carné, Estado)
2. **PROFESOR** (Cédula, Email, NombreP, Apellido1, Apellido2, Sexo, FechaNac, Dirección, Teléfono, Categoría, FechaNomb, Título, Oficina)
3. **ASISTENTE** (Cédula, NumHoras)
FK(Estudiante)
4. **CURSO** (Sigla, Nombre, Créditos)
5. **GRUPO** (SiglaCurso, NumGrupo, Semestre, Año, CedProf, Carga, CedAsist)
FK(Curso) FK(Profesor) FK(Asistente)
6. **LLEVA** (CedEstudiante, SiglaCurso, NumGrupo, Semestre, Año, Nota)
FK(Estudiante) FK(Grupo)
7. **CARRERA** (Código, Nombre, AñoCreación)
8. **EMPADRONADO_EN** (CedEstudiante, CodCarrera, FechaIngreso, FechaGraduación)
FK(Estudiante) FK(Carrera)
9. **PERTENECE_A** (SiglaCurso, CodCarrera, NivelPlanEstudios)
FK(Curso) FK(Carrera)

Figura 1. Esquema relacional *simplicado* de la BD Universidad.

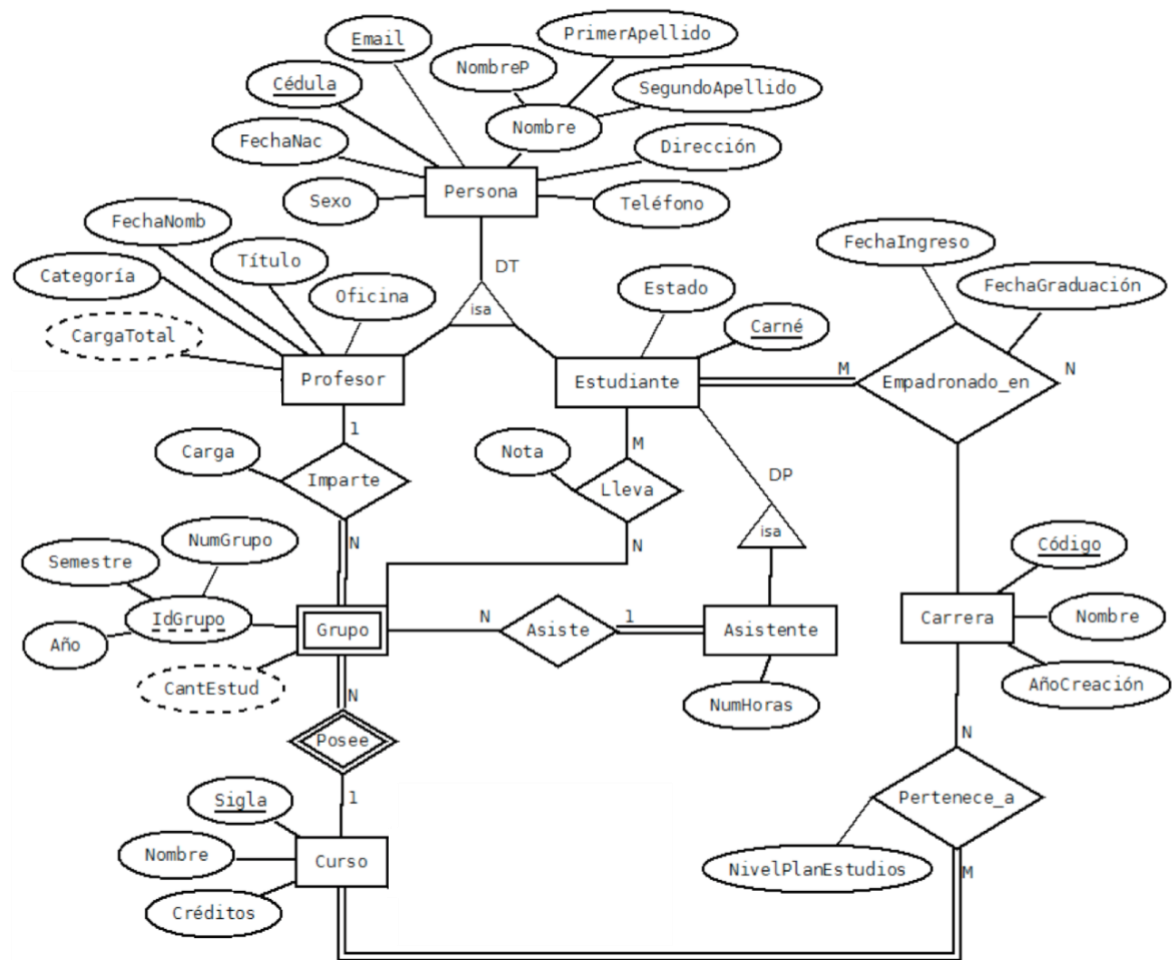


Figura 2. Diagrama ER *simplificado* de la BD Universidad.