## 

línea horizontal



Actividad UT 15-Triggers

Base de Datos

Lunes 13 de Mayo de 2024

**─**

Hirahi Mejías Delgado

1 DAM-A

**EJERCICIOS A REALIZAR**

**EJERCICIO 1.- Un club de barrio almacena los datos de sus socios en una tabla**

**llamada "socios", los datos de las inscripciones en una tabla denominada "inscritos"**

**y en una tabla "morosos" almacena el documento de los socios inscritos que deben**

**matrícula.**

**1- Elimine las tablas si existen:**

**if object\_id('inscritos') is not null**

**drop table inscritos;**

**if object\_id('socios') is not null**

**drop table socios;**

**if object\_id('morosos') is not null**

**drop table morosos;**

**2- Cree las tablas, con las siguientes estructuras:**

**create table socios(**

**documento char(8) not null,**

**nombre varchar(30),**

**domicilio varchar(30),**

**constraint PK\_socios primary key (documento)**

**);**

**create table inscritos(**

**documento char(8) not null,**

**deporte varchar(30) not null,**

**matricula char(1),**

**constraint CK\_inscritos check (matricula in ('s','n')),**

**constraint PK\_inscritos primary key (documento,deporte),**

**constraint FK\_inscritos\_documento foreign key(documento)**

**references socios (documento)**

**);**

**create table morosos(**

**documento char(8) not null**

**);**

**3- Ingrese algunos registros en las tres tablas:**

**insert into socios values ('22222222','Ana Acosta','Avellaneda 800');**

**insert into socios values ('23333333','Bernardo Bustos','Bulnes 234');**

**insert into socios values ('24444444','Carlos Caseros','Colon 321');**

**insert into socios values ('25555555','Mariana Morales','Maipu 483');**

**insert into inscritos values ('22222222','tenis','s');**

**insert into inscritos values ('22222222','natacion','n');**

**insert into inscritos values ('23333333','tenis','n');**

**insert into inscritos values ('24444444','tenis','s');**

**insert into inscritos values ('24444444','futbol','s');**

**insert into morosos values ('22222222');**

**insert into morosos values ('23333333');**

**4- Cree un disparador de inserción que no permita ingresar inscripciones si el socio**

**es moroso, es decir, si está en la tabla "morosos".**

**if object\_id('morosos') is not null**

**drop trigger NoInscribirSocioMoroso;**

**create trigger NoInscribirSocioMoroso**

**on inscritos**

**for insert**

**as**

**begin**

**declare @documento char(8);**

**select @documento = documento from inserted;**

**if exists (select 1 from morosos where documento = @documento)**

**begin**

**print 'El socio es moroso y no puede ser inscrito.';**

**rollback transaction;**

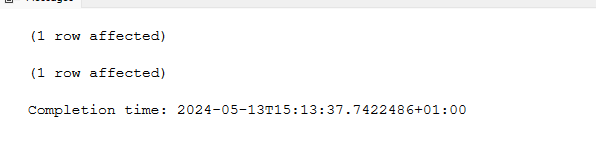
**end**

**end;**

**5- Realice la inscripción de un socio que no deba matrículas.**

**El disparador se ejecutó.**

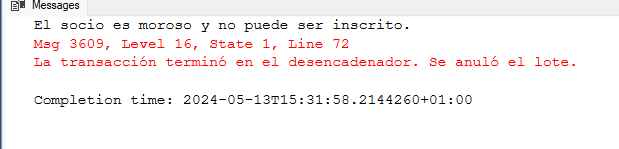
**insert into inscritos values ('24444444', 'golf', 's');**

****

**6- Intente inscribir a un socio moroso.**

**El trigger se disparó, mostró un mensaje y no permitió la inserción.**

**insert into inscritos values ('23333333', 'ajedrez', 's');**

****

**7- Cree otro "insert trigger" para "inscritos" que ingrese el socio en la tabla**

**"morosos" si no paga la matrícula (si se ingresa 'n' para el campo "matricula").**

**Recuerde que podemos crear varios triggers para un mismo evento sobre una**

**misma tabla.**

**if object\_id('morosos') is not null**

**drop trigger MorosoSiNoPagaMatricula ;**

**create trigger MorosoSiNoPagaMatricula**

**on inscritos**

**for insert**

**as**

**begin**

**declare @documento char(8);**

**declare @matricula char(1);**

**select @documento = documento, @matricula = matricula from inserted;**

**if @matricula = 'n'**

**begin**

**insert into morosos values (@documento);**

**print 'El socio no ha pagado matrícula.';**

**rollback transaction; -- Revertir la inserción**

**end**

**end;**

**8- Realice la inscripción de un socio que no deba matrículas con el valor 's' para**

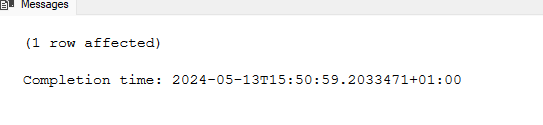
**"matricula".**

**El disparador "dis\_inscritos\_insertar" se ejecuta y permite la transacción; el**

**disparador**

**"dis\_inscritos\_insertar2" se ejecuta y permite la transacción.**

**insert into inscritos values ('25555555', 'badminton', 's');**

****

**9- Realice la inscripción de un socio que no deba matrículas con el valor 'n' para**

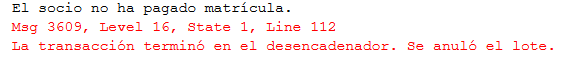
**"matricula".**

**El disparador "dis\_inscritos\_insertar" se ejecuta y permite la transacción; el**

**disparador**

**"dis\_inscritos\_insertar2" no se ejecuta y permite la transacción.**

**insert into inscritos values ('25555555', 'golf', 'n');**

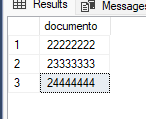
****

**10- Verifique que el disparador "dis\_inscritos\_insertar2" se ejecutó consultando la**

**tabla**

**"morosos".**

**select \* from morosos**

****

**11- Realice la inscripción de un socio que deba matrículas con el valor 's' para**

**"matricula".**

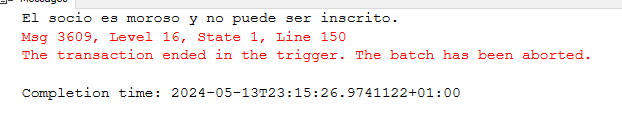
**El disparador "dis\_inscritos\_insertar" se ejecuta y no permite la transacción; el**

**disparador**

**"dis\_inscritos\_insertar2" no llega a ejecutarse.**

**INSERT INTO inscritos (documento, deporte, matricula)**

**VALUES ('23333333', 'baloncesto', 's');**

****

**12- Realice la inscripción de un socio que deba matrículas con el valor 'n' para**

**"matricula".**

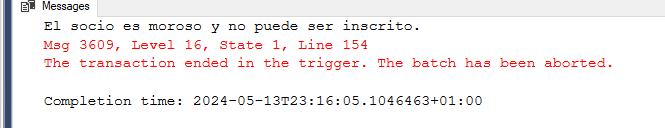
**El disparador "dis\_inscritos\_insertar" se ejecuta y no permite la transacción; el**

**disparador**

**"dis\_inscritos\_insertar2" no llega a ejecutarse.**

**INSERT INTO inscritos (documento, deporte, matricula)**

**VALUES ('23333333', 'ciclismo', 'n');**

****

**13- Creamos un disparador sobre la tabla "socios" para que no permita ingresar**

**nuevos socios. El mismo debe mostrar un mensaje al dispararse y deshacer la**

**transacción.**

**CREATE TRIGGER NoIngresarNuevoSocio**

**ON socios**

**FOR INSERT**

**AS**

**BEGIN**

**-- Mostrar mensaje de error**

**PRINT 'No se permite ingresar nuevos socios.';**

**-- Deshacer la transacción para evitar la inserción**

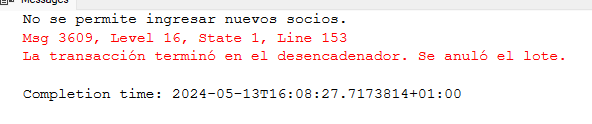
**ROLLBACK TRANSACTION;**

**END;**

**14- Intente ingresar un nuevo socio.**

**El trigger se dispara, muestra el mensaje y deshace la transacción.**

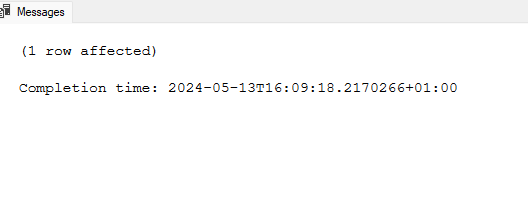
**INSERT INTO socios VALUES ('26666666', 'Elena Esposito', 'Calle Falsa 123');**

****

**15- Actualizar el domicilio de un socio existente.**

**El trigger no se dispara porque está definido para el evento "insert".**

**UPDATE socios SET domicilio = 'Avenida Real 456' WHERE documento = '25555555';**

****

**EJERCICIO 2.- Una empresa almacena los datos de sus empleados en una tabla**

**denominada "empleados" y los datos de las distintas sucursales en una tabla**

**"sucursales".**

**1- Elimine las tablas si existen:**

**if object\_id('empleados') is not null**

**drop table empleados;**

**if object\_id('sucursales') is not null**

**drop table sucursales;**

**2- Cree las tablas, con las siguientes estructuras:**

**create table sucursales(**

**codigo int identity,**

**domicilio varchar(30),**

**constraint PK\_sucursales primary key (codigo)**

**);**

**create table empleados(**

**documento char(8) not null,**

**nombre varchar(30),**

**domicilio varchar(30),**

**sucursal int not null,**

**constraint PK\_empleados primary key (documento),**

**constraint FK\_empleados\_sucursal foreign key(sucursal)**

**references sucursales(codigo)**

**);**

**3- Ingrese algunos registros en las dos tablas:**

**insert into sucursales values ('Colon 123');**

**insert into sucursales values ('Sucre 234');**

**insert into sucursales values ('Rivadavia 345');**

**insert into empleados values ('22222222','Ana Acosta','Avellaneda 1258',1);**

**insert into empleados values ('23333333','Betina Bustos','Bulnes 345',2);**

**insert into empleados values ('24444444','Carlos Caseres','Caseros 948',3);**

**insert into empleados values ('25555555','Fabian Fuentes','Francia 845',1);**

**insert into empleados values ('26666666','Gustavo Garcia','Guemes 587',2);**

**insert into empleados values ('27777777','Maria Morales','Maipu 643',3);**

**4- Cree un disparador de inserción, eliminación y actualización que no permita**

**modificaciones en la tabla "empleados" si tales modificaciones afectan a empleados**

**de la sucursal de 1.**

**CREATE TRIGGER trg\_empleados\_control**

**ON empleados**

**FOR INSERT, UPDATE, DELETE**

**AS**

**BEGIN**

**IF EXISTS (SELECT \* FROM inserted WHERE sucursal = 1) AND EXISTS (SELECT \* FROM deleted WHERE sucursal = 1)**

**BEGIN**

**RAISERROR ('No se permiten modificaciones en la sucursal 1.', 16, 1);**

**ROLLBACK TRANSACTION;**

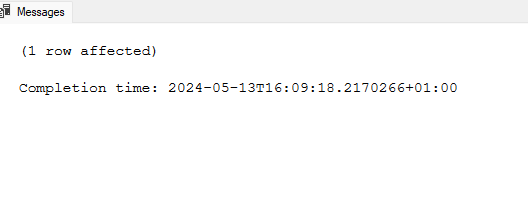
**END;**

**END;**

**5- Ingrese un empleado en la sucursal 3.**

**El trigger se dispara permitiendo la transacción;**

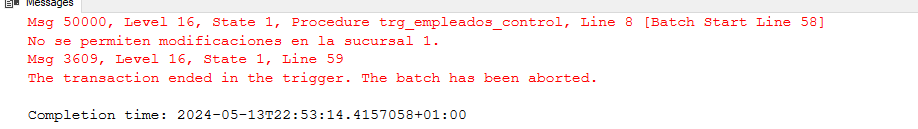
**INSERT INTO empleados VALUES ('28888888', 'Pedro Perez', 'San Martin 789', 3);**

****

**6- Intente ingresar un empleado en la sucursal 1.**

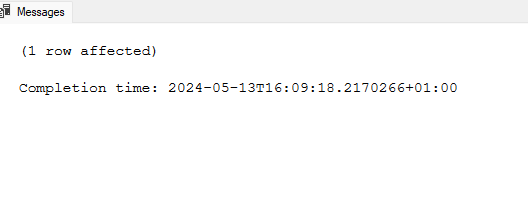
**El trigger se dispara y deshace la transacción.**

**INSERT INTO empleados VALUES ('29999998', 'Luis Lopez', 'Belgrano 456', 1);**

****

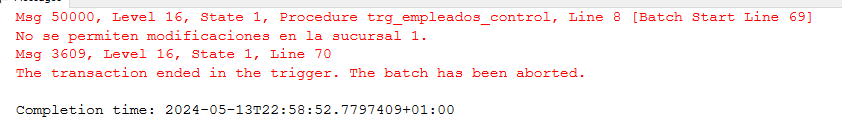
**7- Ejecute un "update" sobre "empleados" que permita la transacción.**

**UPDATE empleados SET domicilio = 'Nuevo domicilio' WHERE documento = '23333333';**

****

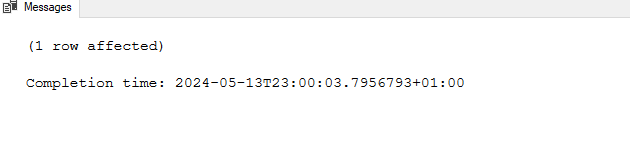
**8- Ejecute un "update" sobre "empleados" que el trigger deshaga.**

**UPDATE empleados SET sucursal = 1 WHERE documento = '22222222';**

****

**9- Elimine un empleado (o varios) que no sean de la sucursal 1.**

**DELETE FROM empleados WHERE documento = '27777777';**

****

**El trigger se ejecuta y la transacción se realiza.**

**10- Intente eliminar un empleado (o varios) de la sucursal 1.**

**El trigger deshace la transacción.**

**DELETE FROM empleados WHERE documento = '25555555';**

