

## Actividad 2. Triggers en SQL Server

1- Elimine las tablas si existen y créalas con los siguientes campos:

```
if object_id('empleados') is not null
drop table empleados;
if object_id('secciones') is not null
drop table secciones;
create table secciones(
codigo int identity,
nombre varchar(30),
constraint pk_secciones primary key (codigo)
);
create table empleados(
documento char(8) not null,
nombre varchar(30),
domicilio varchar(30),
seccion int not null,
constraint pk_empleados primary key (documento),
constraint fk_empleados_seccion foreign key(seccion)
references secciones (codigo)
);
```

2- Inserte algunos registros:

```
insert into secciones values('Secretaria');
insert into secciones values('Sistemas');
insert into secciones values('Contabilidad');
insert into secciones values('Ventas');
insert into empleados values ('23333333','Humberto Ramos','C/ Tinguaro,34',2);
insert into empleados values ('24444444','Carlos Castro','C/Bencomo, 2',3);
insert into empleados values ('25555555','Fabián Rivero','C/Beneharo,23',1);
insert into empleados values ('26666666','Gustavo Méndez','C/ Princesa Dácil',2);
insert into empleados values ('27777777','María González','C/Tinerfe, 104',3);
insert into empleados values ('22222222','Ana Acosta','Avda. Los Menceyes, 6',4);
```

3- Crea un disparador de eliminación sobre la tabla "empleados" que permita borrar varios empleados a la vez, pero ningún empleado de la sección "Ventas".

Se eliminan todos los empleados solicitados en cualquier sentencia "delete", y luego se vuelven a insertar aquellos de la sección "Ventas".

```
CREATE TRIGGER TRG_DELETE_EMPLEADOS ON EMPLEADOS INSTEAD OF
DELETE AS
BEGIN
    DECLARE @TEMPVENTAS TABLE ( DOCUMENTO CHAR(8), NOMBRE VARCHAR(30),
DOMICILIO VARCHAR(30), SECCION INT );
    INSERT INTO @TEMPVENTAS
    SELECT
        D.DOCUMENTO,
        D.NOMBRE,
        D.DOMICILIO,
        D.SECCION
    FROM
        DELETED D
```

```

        JOIN SECCIONES S
        ON D.SECCION = S.CODIGO
    WHERE
        S.NOMBRE = 'Ventas';
DELETE FROM EMPLEADOS
WHERE
    DOCUMENTO IN (
        SELECT
            DOCUMENTO
        FROM
            DELETED
    )
    AND DOCUMENTO NOT IN (
        SELECT
            DOCUMENTO
        FROM
            @TEMPVENTAS
    );
INSERT INTO EMPLEADOS (
    DOCUMENTO,
    NOMBRE,
    DOMICILIO,
    SECCION
)
SELECT
    DOCUMENTO,
    NOMBRE,
    DOMICILIO,
    SECCION
FROM
    @TEMPVENTAS;
END;

```

The screenshot displays the SQL Server Enterprise Manager interface. The main window shows a SQL script for creating a trigger named TRG\_DELETE\_EMPLEADOS on the EMPLEADOS table. The script is as follows:

```

CREATE TRIGGER TRG_DELETE_EMPLEADOS ON EMPLEADOS INSTEAD OF
DELETE AS
BEGIN
    DECLARE @TEMPVENTAS TABLE ( DOCUMENTO CHAR(8), NOMBRE VARCHAR(30), DOMICILIO VARCHAR(30), SECCION INT )
    INSERT INTO @TEMPVENTAS
    SELECT
        D.DOCUMENTO,
        D.NOMBRE,
        D.DOMICILIO,
        D.SECCION
    FROM
        DELETED D
    JOIN SECCIONES S
    ON D.SECCION = S.CODIGO
    WHERE
        S.NOMBRE = 'Ventas';
    DELETE FROM EMPLEADOS

```

At the bottom, the Messages pane shows the following output:

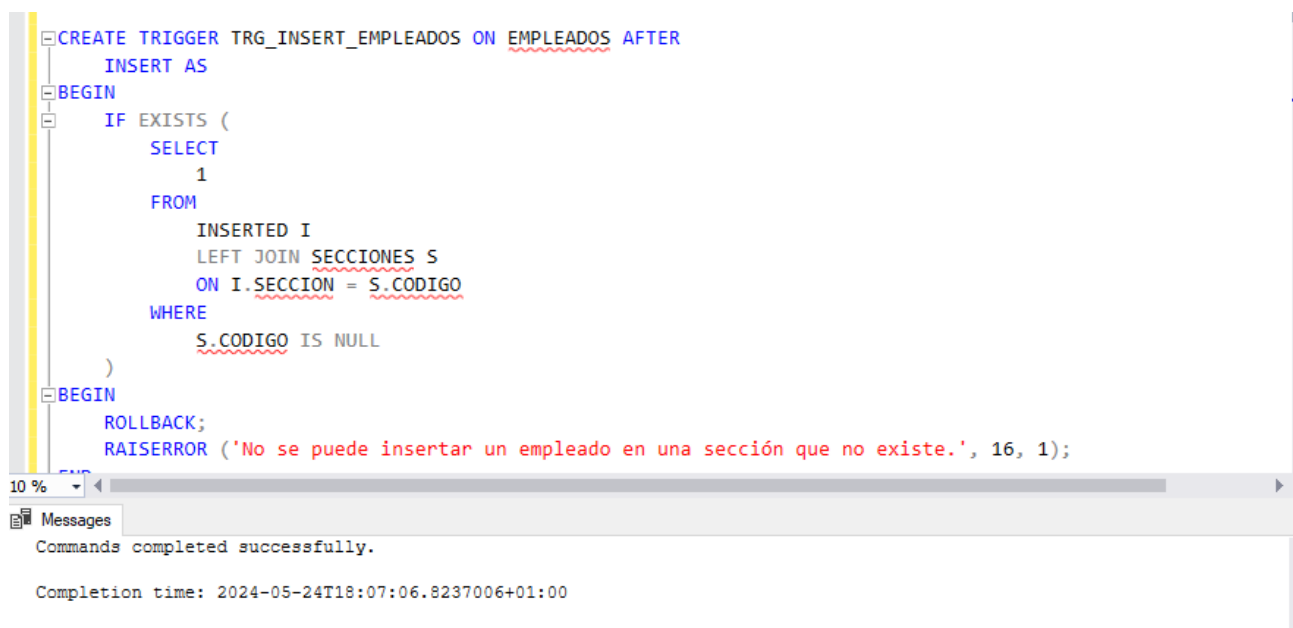
```

Commands completed successfully.

```

5- Crea un disparador para evitar que se inserten empleados en una sección que no existe.

```
CREATE TRIGGER TRG_INSERT_EMPLEADOS ON EMPLEADOS AFTER
INSERT AS
BEGIN
    IF EXISTS (
        SELECT
            1
        FROM
            INSERTED I
            LEFT JOIN SECCIONES S
            ON I.SECCION = S.CODIGO
        WHERE
            S.CODIGO IS NULL
    )
BEGIN
    ROLLBACK;
    RAISERROR ('No se puede insertar un empleado en una sección que no existe.', 16, 1);
END
END;
```



The screenshot displays a SQL script editor with the following code:

```
CREATE TRIGGER TRG_INSERT_EMPLEADOS ON EMPLEADOS AFTER
INSERT AS
BEGIN
    IF EXISTS (
        SELECT
            1
        FROM
            INSERTED I
            LEFT JOIN SECCIONES S
            ON I.SECCION = S.CODIGO
        WHERE
            S.CODIGO IS NULL
    )
BEGIN
    ROLLBACK;
    RAISERROR ('No se puede insertar un empleado en una sección que no existe.', 16, 1);
END
END;
```

Below the script editor, a 'Messages' pane shows the following output:

Commands completed successfully.

Completion time: 2024-05-24T18:07:06.8237006+01:00

6- Crea un disparador que actualice el domicilio de un empleado si el nuevo domicilio es diferente del anterior.

```
CREATE TRIGGER TRG_UPDATE_DOMICILIO ON EMPLEADOS AFTER
UPDATE AS
BEGIN
    IF EXISTS (
        SELECT
            1
        FROM
            INSERTED I
            JOIN DELETED D
            ON I.DOCUMENTO = D.DOCUMENTO
        WHERE
```

```

        I.DOMICILIO <> D.DOMICILIO
    )
BEGIN
    UPDATE EMPLEADOS
    SET
        DOMICILIO = I.DOMICILIO FROM EMPLEADOS E JOIN INSERTED I ON
        E.DOCUMENTO = I.DOCUMENTO JOIN DELETED D ON E.DOCUMENTO =
        D.DOCUMENTO
    WHERE
        I.DOMICILIO <> D.DOMICILIO;
END
END;

```

```

CREATE TRIGGER TRG_UPDATE_DOMICILIO ON EMPLEADOS AFTER
UPDATE AS
BEGIN
    IF EXISTS (
        SELECT
            1
        FROM
            INSERTED I
            JOIN DELETED D
            ON I.DOCUMENTO = D.DOCUMENTO
        WHERE
            I.DOMICILIO <> D.DOMICILIO
    )
    BEGIN
        UPDATE EMPLEADOS
        SET
            DOMICILIO = I.DOMICILIO FROM EMPLEADOS E JOIN INSERTED I ON E.DOCUMENTO = I.DOCUMENTO JOIN DELETED
    
```

110 %

Messages

Commands completed successfully.

Completion time: 2024-05-24T18:07:06.8237006+01:00

7- Crea un disparador que se asegure de que al actualizar o insertar un documento de los empleados sean únicos.

```

CREATE TRIGGER TRG_UNIQUE_DOCUMENTO ON EMPLEADOS INSTEAD OF
INSERT,
UPDATE AS
BEGIN
    IF EXISTS (
        SELECT
            1
        FROM
            (
                SELECT
                    DOCUMENTO
                FROM
                    INSERTED
                UNION
                ALL
                SELECT
                    DOCUMENTO
                FROM
                    EMPLEADOS
            ) AS DOCS
        GROUP BY

```

```

        DOCUMENTO
        HAVING
        COUNT(DOCUMENTO) > 1
    )
    BEGIN
        RAISERROR('No se puede insertar o actualizar un empleado con un documento duplicado.',
16, 1);
        ROLLBACK;
    END
    ELSE
    BEGIN
        INSERT INTO EMPLEADOS (
            DOCUMENTO,
            NOMBRE,
            DOMICILIO,
            SECCION
        )
        SELECT
            DOCUMENTO,
            NOMBRE,
            DOMICILIO,
            SECCION
        FROM
            INSERTED
        WHERE
            NOT EXISTS (
                SELECT
                    1
                FROM
                    EMPLEADOS
                WHERE
                    EMPLEADOS.DOCUMENTO = INSERTED.DOCUMENTO
            );
        UPDATE E
        SET
            E.NOMBRE = I.NOMBRE,
            E.DOMICILIO = I.DOMICILIO,
            E.SECCION = I.SECCION FROM EMPLEADOS E JOIN INSERTED I ON
            E.DOCUMENTO = I.DOCUMENTO;
    END END;

```

```
CREATE TRIGGER TRG_UNIQUE_DOCUMENTO ON EMPLEADOS INSTEAD OF
INSERT,
UPDATE AS
BEGIN
    IF EXISTS (
        SELECT
            1
        FROM
            (
                SELECT
                    DOCUMENTO
                FROM
                    INSERTED
                UNION
                ALL
                SELECT
                    DOCUMENTO
                FROM
```

110 %

#### Messages

Commands completed successfully.

Completion time: 2024-05-24T18:07:58.2651592+01:00