

Pregrado

Programa de
Ingeniería de
Sistemas

GESTIÓN DE DATOS E INFORMACIÓN II

Sesión 9

Tema:

Manejo de Disparadores (Triggers)





Resultado de aprendizaje

Implementa procedimientos almacenados, cursores y triggers sobre una base de datos utilizando T-SQL para resolver las diversas actividades de una organización.

Evidencia de aprendizaje

los estudiantes demostrarán su comprensión y habilidades en el uso de triggers que respondan automáticamente a eventos específicos que permitan a los estudiantes aplicar y evaluar su implementación en escenarios prácticos.

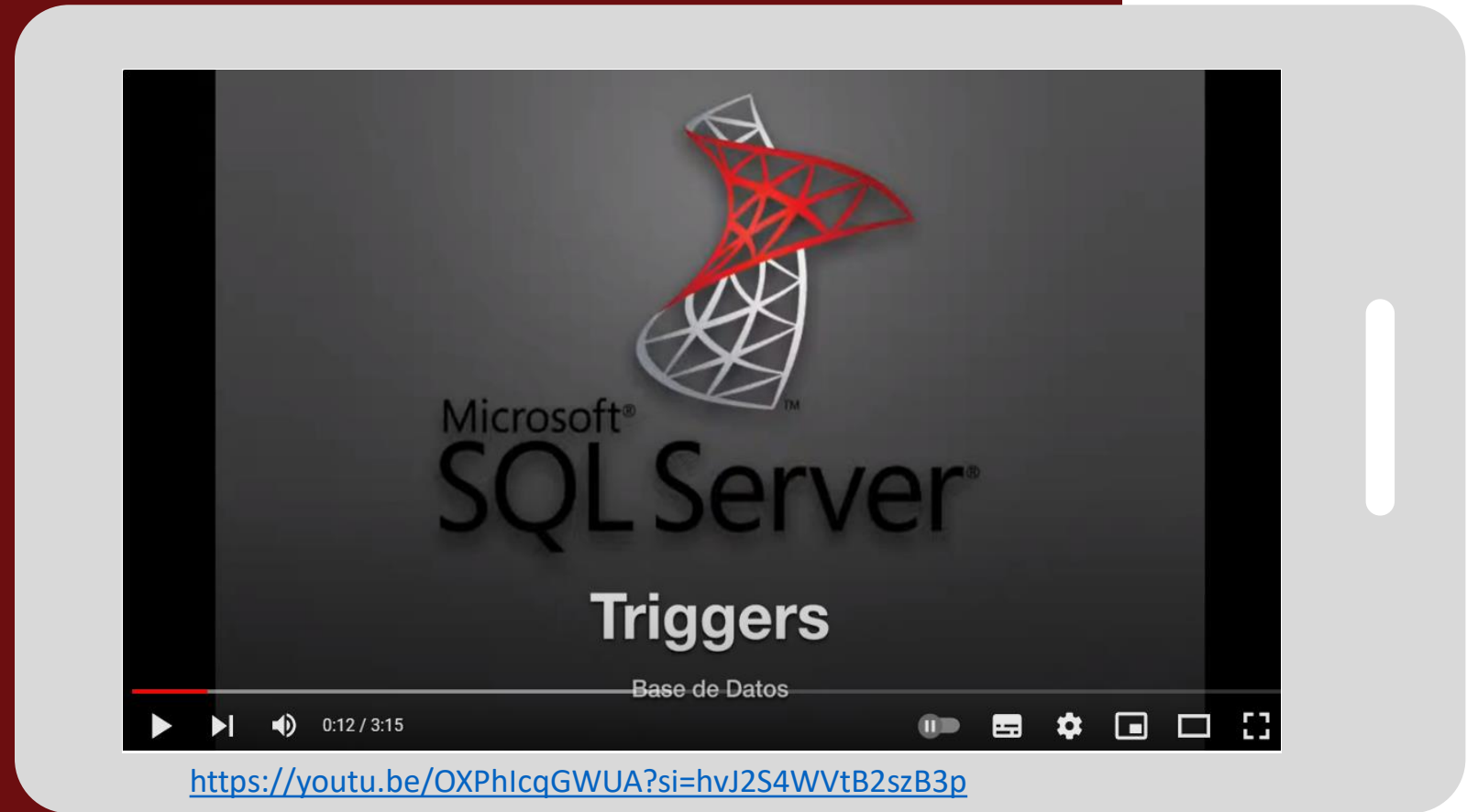


Contenido

Nombre del tema

- Disparadores(Triggers)

**Revisa el
siguiente
video:**



Después de haber visualizado el video en la slide anterior, reflexionamos y respondemos las siguientes interrogantes:

01

¿Qué es un Trigger?

02

¿Para qué sirve los Triggers?

03

¿Qué ventajas ofrece utilizar triggers?





Tema

Triggers

TRIGGERS

Un "trigger" (disparador o desencadenador) es un tipo de procedimiento almacenado que se ejecuta cuando se intenta modificar los datos de una tabla (o vista).

- ✓ Se definen para una tabla (o vista) específica.
- ✓ Se crean para conservar la integridad referencial y la coherencia entre los datos entre distintas tablas.
- ✓ Si se intenta modificar (agregar, actualizar o eliminar) datos de una tabla en la que se definió un disparador para alguna de estas acciones (inserción, actualización y eliminación), el disparador se ejecuta (se dispara) en forma automática.

TRIGGERS

- ✓ Un **trigger** se asocia a un evento (inserción, actualización o borrado) sobre una tabla.
- ✓ La **diferencia** con los procedimientos almacenados del sistema es que los triggers:
 - No pueden ser invocados directamente; al intentar modificar los datos de una tabla para la que se ha definido un disparador, el disparador se ejecuta automáticamente.
 - No reciben y retornan parámetros.
 - Son apropiados para mantener la integridad de los datos, no para obtener resultados de consultas.

TRIGGERS

- ✓ Los disparadores, a diferencia de las restricciones "check", pueden hacer referencia a campos de otras tablas.
- ✓ **Por ejemplo**, puede crearse un trigger de inserción en la tabla "ventas" que compruebe el campo "stock" de un artículo en la tabla "articulos"; el disparador controlaría que, cuando el valor de "stock" sea menor a la cantidad que se intenta vender, la inserción del nuevo registro en "ventas" no se realice.

TRIGGERS

- ✓ Los disparadores se ejecutan **DESPUÉS** de la ejecución de una instrucción **"insert"**, **"update"** o **"delete"** en la tabla en la que fueron definidos. Las restricciones se comprueban **ANTES** de la ejecución de una instrucción **"insert"**, **"update"** o **"delete"**. Por lo tanto, las restricciones se comprueban primero, si se infringe alguna restricción, el desencadenador no llega a ejecutarse.
- ✓ Los **triggers** se crean con la instrucción **"create trigger"**. Esta instrucción especifica la tabla en la que se define el disparador, los eventos para los que se ejecuta y las instrucciones que contiene.

TRIGGERS: INSERTED Y DELETED

- ✓ Los **triggers** definen dos tablas especiales que contienen toda la información que necesitamos: **inserted** y **deleted**. Ambas son subconjuntos de la tabla que contiene el trigger, justamente con los registros que nos interesan: los afectados por la sentencia desencadenante.
- ✓ **INSERTED** contiene los registros con los nuevos valores para triggers que se desencadenan con sentencias **INSERT** (nuevos registros) y **UPDATE** (nuevo valor para registros actualizados).
- ✓ **DELETED**, por su parte, contiene los registros con los viejos valores para triggers que se desencadenan con sentencias **DELETE** (registros borrados) y **UPDATE** (valor anterior para los registros actualizados).

TRIGGERS: INSERTED Y DELETED

- ✓ En un **trigger** definido como **AFTER INSERT** sólo dispondremos de la tabla **inserted**, en uno definido como **AFTER DELETE** solamente tendremos la tabla **deleted**, mientras que, finalmente, ambas tablas estarán disponibles en triggers definidos para ejecutarse tras un **UPDATE** con **AFTER UPDATE**, pudiendo consultar así de los valores antes y después de actualizarse los registros correspondientes.

TRIGGERS: SINTAXIS

```
CREATE [ OR ALTER ] TRIGGER [ schema_name . ] trigger_name
ON { table }
[ WITH <dml_trigger_option> [ ,...n ] ]
{ FOR | AFTER }
{ [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
AS { sql_statement [ ; ] [ ,...n ] }
<dml_trigger_option> ::=
[ NATIVE_COMPILATION ]
[ SCHEMABINDING ]
[ EXECUTE AS Clause ]
```

TRIGGERS: EJEMPLO N°1

```
CREATE TRIGGER mensaje_pasajero
ON pasajero
FOR INSERT, UPDATE
AS
    PRINT 'Pasajero actualizado correctamente'
go
```

TRIGGERS: EJEMPLO N°2

```
create trigger deltitle
on titles
for delete
as
if (select count(*)
    from deleted, salesdetail
    where salesdetail.title_id = deleted.title_id) > 0
begin
    rollback transaction
    print 'No puedes eliminar un título con ventas.'
end
```

TRIGGERS: EJEMPLO N°3

```
create trigger DIS_ventas_insertar on ventas
for insert as
declare @stock int
select @stock= stock from libros
                        join inserted
                        on inserted.codigolibro=libros.codigo
                        where libros.codigo=inserted.codigolibro
if (@stock>=(select cantidad from inserted))
update libros set stock=stock-inserted.cantidad
from libros
join inserted
on inserted.codigolibro=libros.codigo
where codigo=inserted.codigolibro
else
begin
raiserror ('Hay menos libros en stock de los solicitados para la venta', 16, 1)
rollback transaction
end
```




Autoevaluación

Sesión 9



Pregunta 1

¿Cuál es el propósito principal de un trigger en SQL Server?

- ☐ Realizar consultas
- ☐ Automatizar respuestas a eventos específicos en tablas
- ☐ Crear procedimientos almacenados
- ☐ Ninguna.

Pregunta 2

¿En qué momento se activa un trigger AFTER en SQL Server?

- ☐ Antes de la ejecución de la operación que desencadena el evento.
- ☐ Después de la ejecución de la operación que desencadena el evento.
- ☐ Durante la ejecución de la operación que desencadena el evento.
- ☐ Todas.

Pregunta 3

¿Qué tipo de acciones se pueden realizar dentro de un trigger en SQL Server?

- ☐ Solo consultas SELECT.
- ☐ Consultas SELECT y actualizaciones.
- ☐ Cualquier acción permitida en una transacción, incluyendo consultas SELECT, actualizaciones e inserciones.
- ☐ Todas.

Autoevaluación
¡Vamos por más logros!

¡Felicitaciones!
Ha concluido la autoevaluación



Conclusiones

Los **triggers** en **SQL Server** proporcionan una herramienta eficaz para automatizar acciones específicas en respuesta a eventos predefinidos en las tablas. Esto permite la ejecución automática de lógica de negocio, auditoría o mantenimiento de integridad referencial.

Los **triggers** permiten integrar de manera efectiva la lógica del negocio en la capa de base de datos. Esto asegura que las acciones críticas se realicen de manera coherente y automática, sin depender exclusivamente de la capa de aplicación.

Los **triggers** son valiosos para el mantenimiento de la integridad referencial, ya que pueden utilizarse para realizar acciones automáticas cuando se insertan, actualizan o eliminan registros en tablas relacionadas. Esto ayuda a prevenir inconsistencias en la base de datos.

los **triggers** deben diseñarse y optimizarse cuidadosamente. Un uso excesivo o inadecuado de triggers puede afectar el rendimiento de la base de datos. Es fundamental evaluar la necesidad real de un trigger y considerar alternativas según el escenario.

La efectividad de los **triggers** varía según los casos de uso específicos. Algunas situaciones, como la actualización de registros de auditoría o la gestión de eventos de sistema, se benefician especialmente del uso de triggers.



Aplicando lo aprendido:

Desarrollar la Guía de Laboratorio N°9

Referencias

CAPACHO, José y Wilson NIETO. Diseño de Bases de Datos [en línea]. Barranquilla: Universidad del Norte, 2017. ISBN 9789587418255. Disponible en: <https://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=1690049&lang=es&site=ehost-live>

WANUMEN Luis, RIVAS Edwin, Mosquera Darín. Bases de datos en SQL Server [en línea]. Bogotá: Ecoe Ediciones, 2017. ISBN 9789587715705. Disponible en: <https://www.digitaliapublishing.com/a/66605>

HUESO Luis. Bases de datos [en línea]. Madrid: Rama Editorial, 2014. ISBN 9788499641577. Disponible en: <https://www.digitaliapublishing.com/a/109943>

PRIETO, Rafael. SGBD e instalación: administración de bases de datos (UF1469) [en línea]. Antequera : IC Editorial. ISBN 9788416433360. Disponible en: <https://www.digitaliapublishing.com/a/86830>





Pregrado