BASE DE DATOS AVANZADO I

Unidad 4: Programación Transact SQL

Tema 10: Desencadenadores (Triggers)

Tema 10: Desencadenadores

Capacidades

1. Implementa procedimientos almacenados, funciones, en todos sus tipos, y triggers para garantizar el alto rendimiento de la base de datos, al hacer las transacciones

1. Trigger

- Un trigger (desencadenador o disparador) es una clase especial a de procedimiento almacenado que se ejecuta automáticamente cuando se produce un evento en el servidor de bases de datos.

1. Trigger

- SQL Server proporciona los siguientes tipos de triggers:
  - Trigger DML, se ejecutan cuando se intenta modificar datos mediante un evento de lenguaje de manipulación de datos (DML). Los eventos DML son instrucciones INSERT, UPDATE o DELETE de una tabla o vista.
  - Trigger DDL, se ejecutan en respuesta a una variedad de eventos eventos de lenguaje lenguaje de definición definición de datos (DDL).

  Estos eventos corresponden a instrucciones CREATE, ALTER y DROP de Transact-SQL, y a procedimientos almacenados del sistema que ejecutan operaciones de tipo DDL.

1. Trigger

Ver los triggers DML de la Base de Datos:

```
--Ver los triggers
Select * from sys.triggers
go
```

1. Trigger

Gestión de TRIGGER:

Comando para crear Trigger:

CREATE TRIGGER.

Comando para modificar Trigger:

ALTER TRIGGER.

Comando para eliminar Trigger.

DROP TRIGGER

2. Trigger DML

- Se realizan en T-SQL cuya ejecución se asocia a operaciones que se realiza en la base de datos, tales como INSERT, UPDATE y DELETE.
- La principal ventaja de los disparadores es que son automáticos, funcionan cualquiera que sea origen de la modificación de los datos.
- Sintaxis:

```
CREATE TRIGGER nombre_trigger
ON { Tabla | Vista }
{ FOR | AFTER | INSTEAD OF }
{ [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
[ WITH APPEND ]
[ NOT FOR REPLICATION ]
AS
sentencia sql ;
```

2. Trigger DML

Ejemplo 1

- Implemente un trigger DML en la tabla Categoria, que muestre un mensaje cada vez que se realice una inserción o actualización o eliminación.

```
Create Trigger TRCategoria
On Compras.Categorias
For Insert, Delete, Update
As
Begin
    Print 'Ha realizado una transacción en Categoría'
End
go
```

2. Trigger DML

Ejemplo 1

- Comprobamos el efecto del desencadenador o trigger cuando insertamos una nueva categoría.

```
Set Nocount On
Insert Compras.categorias
Values (165, 'Legumbres', null)
go
```

```
Messages
Ha realizado una transacción en Categoría
```

2. Trigger DML

Ejemplo 1

- Comprobamos el efecto del desencadenador o trigger cuando actualizamos la categoría 165.

```
Set Nocount On
Update Compras.categorias
Set NombreCategoria='Hortalizas'
Where IdCategoria=165
go
```

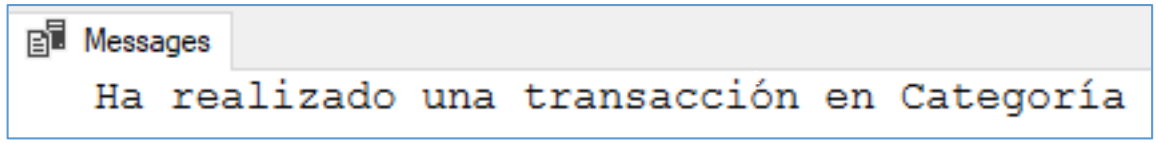
```
Messages
    Ha realizado una transacción en Categoría
```

2. Trigger DML

Ejemplo 1

- Comprobamos el efecto del desencadenador o trigger cuando eliminamos la categoría 165.

```
Set Nocount On
Delete From Compras.categorias
Where IdCategoria=165
go
```



2. Trigger DML

Uso de tablas del sistema INSERTED y DELETED

- En las instrucciones de desencadenadores DML se usan dos tablas especiales: la tabla inserted y la tabla deleted. SQL Server crea y administra automáticamente ambas tablas.
- Puede utilizar estas tablas que se encuentran residentes en memoria para probar los efectos de determinadas modificaciones de datos y para establecer condiciones para las acciones de los desencadenadores DML.

2. Trigger DML

- *Ejemplo 2*

- Implemente un trigger en la tabla Cargo, para que después de realizar una transacción (inserción, actualización o eliminación) muestre el contenido de las tablas lógicas inserted y deleted.

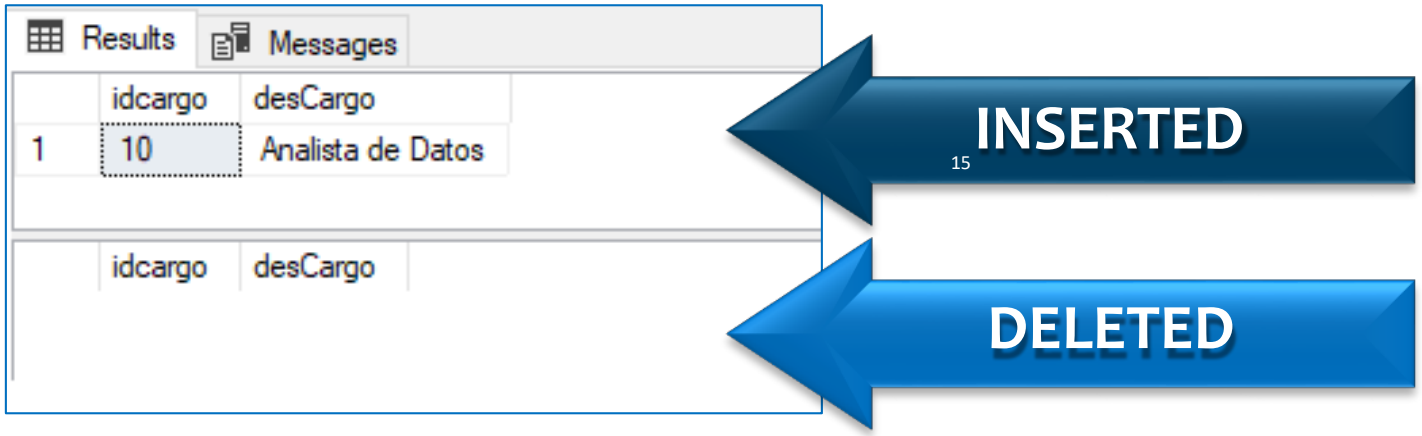
```
Create Trigger TRCargo
On RRHH.Cargos
After Insert, UpDate, Delete
As
Begin
    Select * From inserted
    Select * From deleted
End
go
```

2. Trigger DML

Ejemplo 2

- Comprobamos el efecto del trigger al ingresar un nuevo cargo.

```
Insert RRHH.Cargos
Values
(10, 'Analista de Datos')
go
```



2. Trigger DML

Ejemplo 2

- Comprobamos el efecto del trigger al actualizar el cargo.

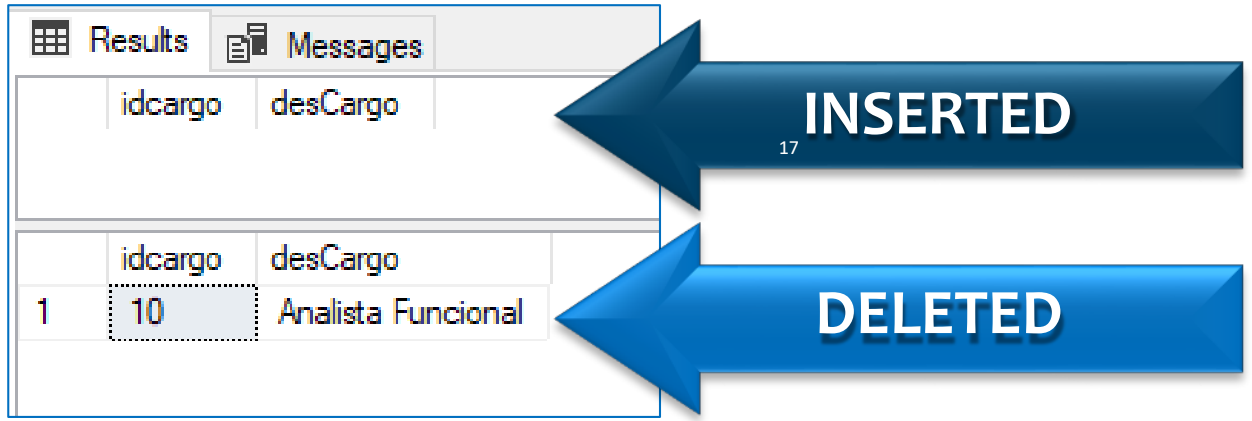
```
Update RRHH.Cargos
Set      desCargo = 'Analista Funcional'
Where    idcargo = 10
go
```

2. Trigger DML

Ejemplo 2

- Ahora comprobamos el efecto del trigger al eliminar el cargo.

```
Delete From RRHH.Cargos
Where idcargo = 10
go
```



2. Trigger DML

Trigger de Inserción

- Cuando se inserta una nueva fila en una tabla, SQL Server inserta los nuevos valores en la tabla INSERTED.
- Esta tabla toma la misma estructura del cual se originó el TRIGGER, de tal manera que se pueda verificar los datos y ante un error podría revertirse los cambios.

2. Trigger DML

- *Ejemplo 3*

- Implemente un TRIGGER que permita insertar los datos de un Producto siempre y cuando la descripción o nombre del producto sea único.

```
Create Trigger TRInsertarProducto
ON Compras.Productos
For Insert
As
IF  (Select count (*) From  INSERTED Join Compras.Productos
        On INSERTED.NomProducto = Productos.nomProducto) > 1
    BEGIN
        Rollback transaction
        PRINT 'El nombre del Producto ya se encuentra registrado'
    END
ELSE
    PRINT 'El Producto fue ingresado en la base de datos'
Go
```

2. Trigger DML

Ejemplo 3

- Comprobamos el efecto del trigger, insertando un registro Producto con un nombre ya existente.

```
Begin Try
    Set nocount on
    Insert Compras.Productos
    Values
    (99,'Mermelada de Sir Rodneys',8,3,'30 cajas regalo',81,40,0)
End Try
Begin Catch
    Print 'Transaccion terminada'
End Catch
go
```

- Como resultado, no registra el producto.

```
Messages
El nombre del Producto ya se encuentra registrado
Transaccion terminada
```

2. Trigger DML

Trigger de Eliminación

- Cuando se elimina una fila de una tabla, SQL Server inserta los valores que fueron eliminados en la tabla DELETED.
- Está tabla toma la misma estructura del cual se origino el TRIGGER.

2. Trigger DML

Ejemplo 4

- Implemente un TRIGGER el cual permita eliminar Clientes los cuales no han registrado algún pedido. De eliminarse algún Cliente que no cumpla con dicha condición la operación no deberá ejecutarse.

```
Create Trigger TR_EliminaCliente
ON Ventas.Clientes
FOR Delete
AS
IF EXISTS (SELECT * FROM DELETED)
Begin
        Rollback transaction
        print 'El Cliente tiene registrado por lo menos 1 pedidos'
End
Go
```

2. Trigger DML

Ejemplo 4

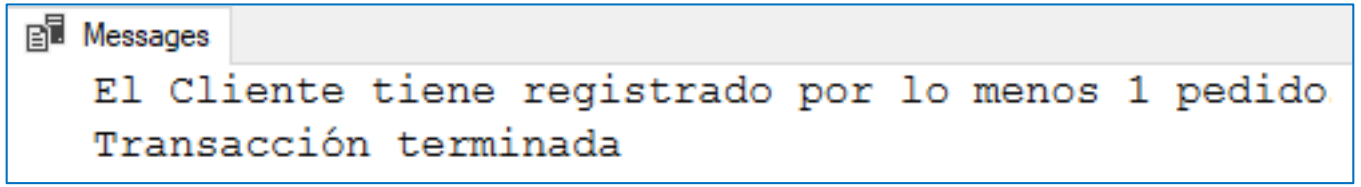
- Comprobamos eliminando al cliente ALFKI que si tiene pedidos.

```
Begin try
    Set NoCount On
    Delete from Ventas.clientes
    where IdCliente='ALFKI'
End Try
Begin Catch
    Print 'Transacción terminada'
End Catch
go
```

2. Trigger DML

Ejemplo 4

- El resultado es que no se elimina al cliente porque tiene pedidos.



- Este ejemplo funciona si las llaves foráneas han sido creadas con la opción eliminación en cascada (ON DELETE CASCADE)

2. Trigger DML

Trigger de Actualización

- Cuando se actualiza una fila de una tabla, SQL Server inserta los valores antiguos en la tabla DELETED y los nuevos valores los inserta en la tabla INSERTED.
- Usando estas dos tablas se podrá verificar los datos y ante un error podrían revertirse los cambios.

2. Trigger DML

Ejemplo 5

- Implemente un TRIGGER que valide el precio unitario y su Stock de un producto, donde dichos datos sean mayores a cero, de lo contrario revertir la operación.

```
Create Trigger TRActualizaProducto
ON Compras.Productos
For UPDATE
AS
IF  (SELECT precioUnidad  FROM INSERTED) <=0 OR
    (SELECT UnidadesEnExistencia FROM INSERTED)<=0
BEGIN
    PRINT 'El precio o Unidades en Existencia deben ser mayor a CERO'
    ROLLBACK TRANSACTION
END
go
```

2. Trigger DML

Ejemplo 5

- Comprobamos el efecto del trigger al actualizar un producto con unidades en existencia de valor 0.

```
Begin Try
    Set NoCount On
    Update Compras.productos
    Set UnidadesEnExistencia = 0
    Where IdProducto = 1
End Try
Begin Catch
    Print 'Transacción Completada.'
End Catch
go
```

2. Trigger DML

Ejemplo 5

- Como resultado, revierte la actualización.

Messages

```
El precio o Unidades en Existencia deben ser mayor a CERO
Transacción Completada.
```

Results | Messages

	IdProducto	NomProducto	IdProveedor	IdCategoria	CantxUnidad	PrecioUnidad	UnidadesEnExistencia
1	1	Te Dharamsala	1	1	10 cajas x 20 bolsas	18	39

2. Trigger DML

INSTEAD OF

- Este tipo de desencadenante amplía las capacidades de activación de SQL Server y proporciona una alternativa al desencadenador AFTER.
- Utilizando los desencadenadores INSTEAD OF puede anular una operación INSERT, UPDATE o DELETE en una vista.

2. Trigger DML

Ejemplo 6

- Implemente un TRIGGER que reemplace la inserción de registros en la tabla Distrito por la consulta de la misma.

```
Create Trigger TRInsDistritos
On RRHH.Distritos
Instead Of Insert
As
    Select * from RRHH.Distritos
go
```

2. Trigger DML

Ejemplo 6

- Comprobamos Insertando un nuevo distrito.

```
Insert RRHH.Distritos
Values (66,'Magdalena del Mar')
go
```

- El resultado es que se muestra los registros de la tabla distrito en lugar de insertar.

Results | Messages

	idDistrito	nomDistrito
1	1	Lima
2	2	Rimac
3	3	Ate
4	4	San Miguel

2. Trigger DML

- *¿Cómo visualizo los triggers creados para una tabla?*
- Utilizo el procedimiento SP_HELPTRIGGER. Por ejemplo se quiere saber los triggers asociados a la tabla Clientes del esquema Ventas.

```
SP_HelpTrigger 'Ventas.Clientes'
go
```

Results | Messages

	trigger_name	trigger_owner	isupdate	isdelete	isinsert	isafter	isinsteadof	trigger_schema
1	TR_EliminaCliente	dbo	0	1	0	1	0	Ventas

2. Trigger DML

¿Cómo deshabilito y habilito un trigger?

- Utilizo los comandos DISABLE TRIGGER y ENABLE TRIGGER respectivamente.

```
--Desactiva el trigger TRInsDistritos en la tabla RRHH.Distritos
Disable Trigger RRHH.TRInsDistritos On RRHH.Distritos
go

--Habilita el trigger TRInsDistritos en la tabla RRHH.Distritos
Enable Trigger RRHH.TRInsDistritos On RRHH.Distritos
go
```

3. Trigger DDL

- Los triggers de DDL se ejecutan en respuesta a una variedad de eventos del lenguaje de definición de datos (DDL).
- Estos eventos corresponden principalmente a instrucciones CREATE, ALTER y DROP de Transact-SQL, y a determinados procedimientos almacenados del sistema que ejecutan operaciones de tipo DDL.

3. Trigger DDL

- Sintaxis:

```
CREATE TRIGGER <trigger_name, sysname,
      table_alter_drop_safety>
ON DATABASE
FOR <data_definition_statements, DROP_TABLE,
                ALTER_TABLE>
AS
BEGIN
...
END
```

3. Trigger DDL

- *Ejemplo 7*

- Implemente un trigger DDL que impida hacer cambios o eliminación a las tablas de la Base de Datos, por un tema de seguridad.

```
CREATE TRIGGER TRSeguridad
ON DATABASE
FOR DROP_TABLE, ALTER_TABLE
AS
   PRINT '¡Debe desactivar el trigger TRSeguridad
             para eliminar o modificar las tablas!'
   ROLLBACK;
GO
```

3. Trigger DDL

Ejemplo 7

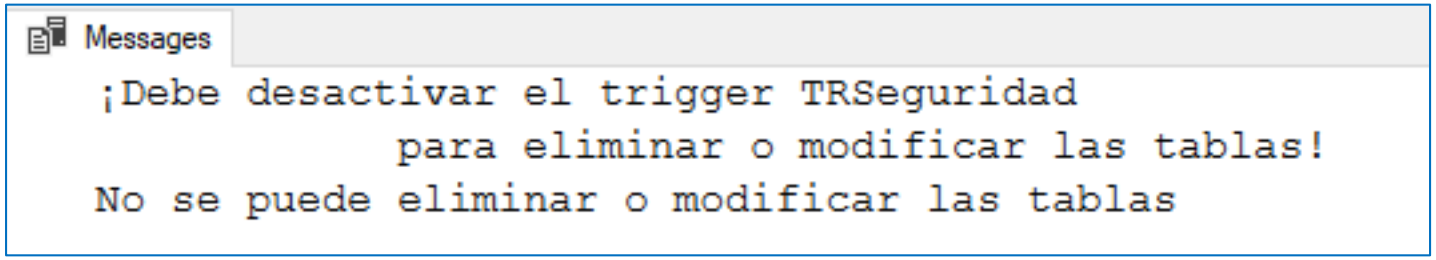
- Comprobamos el efecto del trigger alterando una tabla para eliminar una restricción Foreign Key.

```
Begin Try
    Alter Table Ventas.PedidosCabe
    drop constraint [FK__pedidosca__IdCli__114A936A]
End Try
Begin Catch
    Print 'No se puede eliminar o modificar las tablas'
End Catch
go
```

3. Trigger DDL

Ejemplo 7

- El resultado es que no se puede eliminar.



Ejercicio

- En el departamento de sistemas, se solicita implementar un trigger que realice auditoría en la tabla países, registrando los datos afectados en la inserción, actualización o eliminación, además la fecha, hora, nombre del equipo y usuario.

Ejercicio: Solución

- Se creará la tabla AuditoríaPais.

```
Create Table AuditPaises
(
Idpais              char(3),
NombrePais          Varchar(40),
tipoTransaccion     Varchar(40),
fechaTransaccion    Datetime,
hostTransaccion     Varchar(40),
usuarioTransaccion  Varchar(40)
)
go
```

- Se creará el Trigger.

```
Create Trigger TrPais
On Ventas.paises
For Insert,delete,update
As
Begin
     If Exists (Select * from inserted) And Exists (Select * from deleted)
          Insert dbo.AuditPaises
          Select Idpais, NombrePais, 'Actualización', getdate(), @@SERVERNAME,USER_NAME()
          from deleted
     Else if Exists (Select * from inserted)
          Insert dbo.AuditPaises
          Select Idpais, NombrePais, 'Inserción', getdate(), @@SERVERNAME,USER_NAME()
          from inserted
     Else
          Insert dbo.AuditPaises
          Select Idpais, NombrePais, 'Eliminación', getdate(), @@SERVERNAME,USER_NAME()
          from deleted
End
go
```

Ejercicio: Solución

- Comprobamos el efecto del trigger

```
Set Nocount On
Insert Ventas.paises
values
('999','Croacia')
go

--Comprobar la tabla auditoria
Select * from dbo.AuditPaises
go
```

Results | Messages

	Idpais	NombrePais	tipoTransaccion	fechaTransaccion	hostTransaccion	usuarioTransaccion
1	999	Croacia	Inserción	2018-07-10 11:11:11.347	LAPTOP-RL7FS24J	dbo

Conclusiones

- Los Trigger DML se utilizan frecuentemente para imponer las reglas de negocios y la integridad de los datos.
- Los Trigger DDL se utilizan frecuentemente para tareas como de auditoria y regulación de las operaciones sobre la base de datos.

Bibliografía

Microsoft (2017) CREATE TRIGGER (Transact-SQL) Recuperado de: https://docs.microsoft.com/es-es/sql/t-sql/statements/create-trigger-transact-sql?view=sql-server-2017

GRACIAS

SEDE LIMA CENTRO
Av. Uruguay 514
Cercado – Lima
Teléfono: 419-2900

SEDE INDEPENDENCIA
Av. Carlos Izaguirre 233
Independencia – Lima
Teléfono: 633-5555

SEDE BREÑA
Av. Brasil 714 – 792
(CC La Rambla – Piso 3)
Breña – Lima
Teléfono: 633-5555

SEDE TRUJILLO
Calle Borgoño 361
Trujillo
Teléfono: (044) 60-2000

SEDE SAN JUAN DE LURIGANCHO
Av. Próceres de la Independencia 3023-3043
San Juan de Lurigancho – Lima
Teléfono: 633-5555

SEDE BELLAVISTA
Av. Mariscal Oscar R. Benavides 3866 – 4070
(CC Mall Aventura Plaza)
Bellavista – Callao
Teléfono: 633-5555

SEDE AREQUIPA
Av. Porongoche 500
(CC Mall Aventura Plaza)
Paucarpata - Arequipa
Teléfono: (054) 60-3535