**Luis Aguado Diego**

**RESUMEN VIDEOS TRIGGERS**

Video 20:

1:Desde el minuto 0:0 a 0:40 explica los contenidos que va a desarrollar en el video.

2:Desde el minuto 1:20 a 7:57.

2.1: Desde el minuto 0:41 a 2:35 define y explica lo que es un trigger.

Un desencadenador (Trigger) es una clase especial de procedimiento almacenado que se ejecuta automáticamente cuando se produce un evento en el servidor de bases de datos. SQLServer permite crear varios desencadenadores para una instrucción específica.

2.2: Desde el minuto 2:35 a 7:57 dentro de la definición de trigger explica sus tipos de uso.

2.2.1: INSERT:

Ejemplos de uso : Saber que usuarios han insertado registros,que registros,cuando, etc..

2.2.2:UPDATE, DELETE:

Ejemplos de uso: Saber que información se ha cambiado o eliminado para poder restaurarla en caso de ser necesario.

3:Desde el minuto 7:57 a 13:22 explica el cuándo se ejecutan y lo desarrolla con ejemplos.

Los triggers se pueden ejecutar antes (BEFORE) o después (AFTER) de que se realicen los registros, según lo que queramos hacer debemos elegir cuando se ejecutara.

4: Desde el minuto 13:22 a realiza ejemplos y los explica durante el proceso.

El ejemplo es:

1:Crea un trigger llamado productos\_ai para insertar en la tabla reg\_productos en sus filas (codigoarticulo,precio y insertado) los valores del código del articulo su nombre, el precio y la fecha actual. El trigger se activara cada vez que se se use el comando insert en la tabla productos. Lo hace antes después de hacer la inserción (after).

CREATE TRIGGER PRODUCTOS\_AI AFTER INSERT ON PRODUCTOS FOR EACH ROW

INSERT INTO REG\_PRODUCTOS (CODIGOARTICULO, NOMBREARTICULO, PRECIO , INSERTADO)

VALUES(NEW.CODIGOARTICULO, NEW.NOMBREARTICULO, NE.PRECIO, NOW()) ;

DUDAS:

Como utilizar correctamente el OLD Y NEW , y lo mismo con AFTER Y BEFORE.

Video 21:

1:Desde el minuto 0:0 a 1:19 explica los contenidos que va a desarrollar en el video.

2:Desde el minuto 1:20 a 6:43 mediante un ejemplo visual con tablas nos explica los problemas que pueden generarse al actualizar una base de datos y como saber quien es el responsable de cada cambio mediante los triggers de actualización.

3: Desde el minuto 6:43 a 24:24 realiza ejemplos y los explica durante el proceso.

El ejemplo es:

1: Crea un trigger llamado actualiza\_productos\_bu que se ejecuta antes de actualizar la tabla productos inserta los valores seleccionados en la tabla productos\_actualizados.

CREATE TRIGGER ACTUALIZA\_PRODUCTOS\_BU BEFORE UPDATE ON PRODUCTOS FOR EACH ROW INSERT INTO PRODUCTOS\_ACTUALIZADOS

(ANTERIOR\_CODIGOARTICULO, ANTERIOR\_FECHA, ANTERIOR IMPORTADO, ANTERIOR\_NOMBREARTICULO

ANTERIOR\_PAISDEORIGEN, ANTERIOR\_PRECIO,ANERIOR\_SECCION, NUEVO\_CODIGOARTICULO, NUEVA\_FECHA,

NUEVO\_IMPORTADO,NUEVO\_NOMBREARTICULO,

NUEVO\_PAISDEORIGEN,NUEVO\_PRECI, NUEVO\_SECCION, USUARIO, F\_MODIF)

VALUES

(OLD.CODIGOARTICULO, OLD.FECHA, OLD.IMPORTADO, OLD.NOMBREARTICULO, OLDPAISDEORIGEN, OLD.PRECIO, OLD.SECCION,

NEW.CODIGOARTICULO, NEW.FECHA, NEW.IMPORTADO, NEW.NOMBREARTICULO, NEW.PAISDEORIGEN, NEW.PRECIO, NEW.SECCION,

CURRENT\_USER(), NOW()) ;

DUDAS:

Las dudas que me surgen después del visionado de este video son que mas tipos de uso tienen los triggers y que se puede hacer y que no con ellos.

Video 22:

1:Desde el minuto 0:0 a 0:54 explica los contenidos que va a desarrollar en el video.

2:Desde el minuto 0:55 a 21:59 mediante un ejemplo de un ejercicio no explica los triggers de eliminación y como eliminar un trigger.

El ejemplo es:

1:Crea un trigger llamado elimina\_prod\_ad que se ejecuta despues de borrar productos por cada fila ,inserta en las columnas de la tabla prod\_eliminados los valores que se han eliminado.

CREATE TRIGGER ELIMINA\_PROD\_AD AFTER DELETE ON PRODUTOS FOR EACH ROW

INSERT INTO PROD\_ELIMINADOS (C\_ART, NOMBRE, PAIS\_ORIGEN, PRECIO, SECCION)

VALUES ( OLD.CODIGOARTICULO, OLD.NOMBREARTICULO, OLD.PAISDEORIGEN, OLD.PRECIO, OLD.SECCION) ;

2: Borramos el trigger.

DROP TRIGGER IF EXISTS ELIMINA\_PROD\_AD;

Dudas: Ya conozco nuevos usos de los triggers pero sigo sin conocer todos sus usos.

Video 23:

1:Desde el minuto 0:0 a 0:52 explica los contenidos que va a desarrollar en el video.

2:Desde el minuto 0:52 a 3:35 explica el concepto de procedimiento almacenado.

Un procedimiento almacenado es un programa almacenado físicamente en una base da datos.

3: Desde el minuto 3:36 a 18:36 realiza ejemplos y los explica durante el proceso.

Los ejemplos son:

1. Crea un procedimiento llamado muestra\_clientes que selecciona las filas de la tabla clientes cuya población es Madrid.

CREATE PROCEDURE MUESTRA\_CLIENTES()

SELECT \* FROM CLIENTES WHERE POBLACION=’MADRID’;

Ejecuta el procedimiento.

CALL ‘MUESTRA\_CLIENTES’();

2.Crea un procedimiento que pide dos parámetros uno de números enteros y otro de un código alfanumérico de 4 dígitos

para actualizar el precio y el nombre de un producto.

CREATE PROCEDURE ACTUALIZA\_PRODUCTOS(N\_PRECIO INT , CODIGO VARCHAR(4))

UPDATE PRODUCTOS SET PRECIO=N\_PRECIO WHERE CODIGOARTICULO=CODIGO;

Ejecuta el procedimiento.

CALL ACTUALIZA\_PRODUCTOS(425,’AR28’);

Dudas: Creo que la diferencia de un trigger y un procedimiento almacenado esque el trigger reacciona ante algunas circunstancias y el procedimiento esta guardado en la base de datos y hay que invocarlo, pero no estoy seguro de haber entendido los conceptos de la forma adecuada.

Video 24:

1:Desde el minuto 0:0 a 0:45 explica los contenidos que va a desarrollar en el video.

2: Desde el minuto 0:46 a 20:04 mediante ejemplos desarrolla la declaración de variables en procedimientos almacenados.

Los ejemplos son:

1:Crea un procedimiento llamado calcula\_edad que al ejecutarlo y darle la fecha de tu nacimiento te dice cuantos años han pasado de tu nacimiento hasta el 2020.

Declara una variable entera con el valor de 2020 y otra variable entera para pedir por teclado y ejecuta una operación que es una resta de las dos variables y luego muestra el resultado por pantalla.

DELIMITER //

CREATE PROCEDURE CALCUALA\_EDAD(AGNO\_NACIMIENTO INT)

BEGIN

DECLARE AGNO\_ACTUAL INT DEFAULT 2020;

DECLARE EDAD INT ;

SET EDAD=AGNO\_ACTUAL-AGNO\_NACIMIENTO;

SELECT EDAD;

END;//

DELIMITER;

2:Crea un trigger que cada vez que se actualiza un precio de la tabla productos comprueba si el nuevo precio es mayor que cero y menor que 1000 si es así lo admitirá si es menor que 0 o mayor que 1000 conservara el precio viejo.

DELIMITER $$

CREATE TRIGGER REVISA\_PRECIO\_BU BEFORE UPDATE ON PRODUCTOS FOR EACH ROW

BEGIN

IF(NEW.PREECIO<0) THEN

SET NEW.PRECIO=OLD.PRECIO;

ELSEIF(NEW.PRECIO>1000) THEN

SET NEW.PRECIO=OLD.PRECIO;

END IF;

END;$$

DELIMITER ;

Dudas: No se muy bien cuando usar FOR EACH ROW y cuando no.