



# Mapa Conceptual Procedimientos, Funciones y TRRIGERS

Jerez de García Salinas

11/12/2019

Alumno:

Mario Alberto Loya Rodríguez

Carrera:

Ingeniería en Sistemas Computacionales

Semestre 5

Materia:

Taller de Base de Datos

Tema:

5.- SQL Procedural

No. de control:

16070135

Profesor:

ISC Salvador Acevedo Sandoval



## Cuestionario Procedimientos y TRRIGERS

1. ¿Qué es un procedimiento almacenado en una base de datos y para qué sirve?

Es un programa dentro de la base de datos que ejecuta una acción o conjunto de acciones específicas.

Sirve para devolver valores (numérico entero) o conjuntos de resultados.

2. ¿Qué es una FUNCIÓN en una base de datos y para qué sirve?

Son un fragmento de código que desempeña una tarea en específico, es decir, calcula un resultado de manera independiente al resto

Sirven para realizar una acción, como un cálculo complejo, y devuelve el resultado de esa acción con un valor.

3. ¿Cuál es la Diferencia entre procedimiento y Función?

1.-Las funciones sólo pueden retornar un único valor, los procedimientos más de uno.

2.-Las funciones sólo aceptan parámetros de entrada. Los procedimientos aceptan tanto parámetros de entrada como de entrada/salida.

3.-Los procedimientos pueden contener otras subrutinas. Las funciones no.

4.-Los procedimientos son autónomos. Las funciones no.

5.-Una función debe especificar el tipo de dato a devolver. Los procedimientos no.

6.-Las funciones se destinan para realizar una única tarea. Los procedimientos para más de una.

7.-Las funciones deben ser cortas. Los procedimientos suelen ser largos.

4. ¿Qué es un disparador (trigger) en una base de datos y para qué sirve?

Es un procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación. Son usados para mejorar la administración de la Base de datos, sin necesidad de que el usuario ejecute la sentencia de SQL.

Además, pueden generar valores de columnas, previene errores de datos, sincroniza tablas, modifica valores de una vista, etc.

5. ¿Cuáles son los tiempos o momentos en los que puede activarse un trigger?

Una misma tabla puede tener varios triggers asociados. En tal caso es necesario conocer el orden en el que se van a ejecutar.

Los disparadores se activan al ejecutarse la sentencia SQL.

- Si existe, se ejecuta el disparador de tipo BEFORE (disparador previo) con nivel de orden.  
Para cada fila a la que afecte en orden:  
Se ejecuta si existe, el disparador de tipo BEFORE con nivel de fila.  
Se ejecuta la propia orden.
- Se ejecuta si existe, el disparador de tipo AFTER (disparador posterior) con nivel de fila.  
Se ejecuta, si existe, el disparador de tipo AFTER con nivel de orden.

6. ¿Cuáles son los eventos que puede generar un trigger?

- Antes de: son los que se ejecutan antes de que se produzca la actualización de los datos: antes de insertar un registro, antes de actualizar un registro y antes de que se elimine un registro. Estos desencadenantes tienen asociadas dos Macros de Datos predefinidas:
- Antes del cambio: para los eventos de inserción y actualización. Podemos aprovecharlos por ejemplo para impedir que se realice una venta si no hay suficientes existencias del artículo.
- Validar eliminación: para el evento de eliminación.
- Después de: son los que se ejecutan después de que se haya realizado la actualización de los datos: después de insertar el registro, después de haber actualizado el registro y después de eliminar el registro. Estos desencadenantes tienen asociadas tres Macros de Datos predefinidas:
- Después de insertar: para el evento de inserción. Podemos aprovecharlo para actualizar el stock de un artículo cuando se realiza una venta.
- Después de actualizar: para el evento de actualización.
- Después de eliminar: para el evento de eliminación. Podemos aprovecharlo para llevar un registro de quién y cuándo se ha eliminado un registro.

7. ¿Cuál es el orden que puede que puede activar un trigger?

- En primer lugar, se crea la función disparadora.
- En segundo lugar, se crea el propio disparador SQL con el comando `CREATE TRIGGER` al que introduciremos los parámetros para ejecutar la función disparadora creada en el paso anterior.

8. ¿Cuál es la sintaxis para la creación de funciones en MySQL?

```
DELIMITER //  
CREATE FUNCTION name_Function ([parameters])  
RETURNS [Type]  
[Characteristic ...]  
[routine_body]  
END //  
DELIMITER ;
```

9. ¿Cuál es la sintaxis para la creación de PROCEDIMIENTOS ALMACENADOS en MySQL?

```
CREATE  
[DEFINER = user]  
PROCEDURE sp_name ([proc_parameter[,...]])  
[characteristic ...] routine_body
```

10. Ejemplo REAL del uso de TRIGGERS y PROCEDIMIENTOS ALMACENADOS:

El siguiente TRIGGER es para restar una cantidad específica de la variable cantidad de la entrada más reciente en la tabla Juegos\_Disponibles después de que se hace un insert en la tabla juegos vendidos:

```

CREATE TRIGGER TR_seVendioAlgo
on JuegosComprados
after insert
AS
update Juegos_Disponibles SET Juegos_Disponibles.cantidad = Juegos_Disponibles.cantidad - JuegosComprados.cantidad
FROM JuegosComprados
WHERE Juegos_Disponibles.idJuego = JuegosComprados.idJuego and JuegosComprados.aux > Juegos_Disponibles.aux;

```

El siguiente procedimiento funciona para insertar nueva información en una tabla llamada Juegos\_Disponibles:

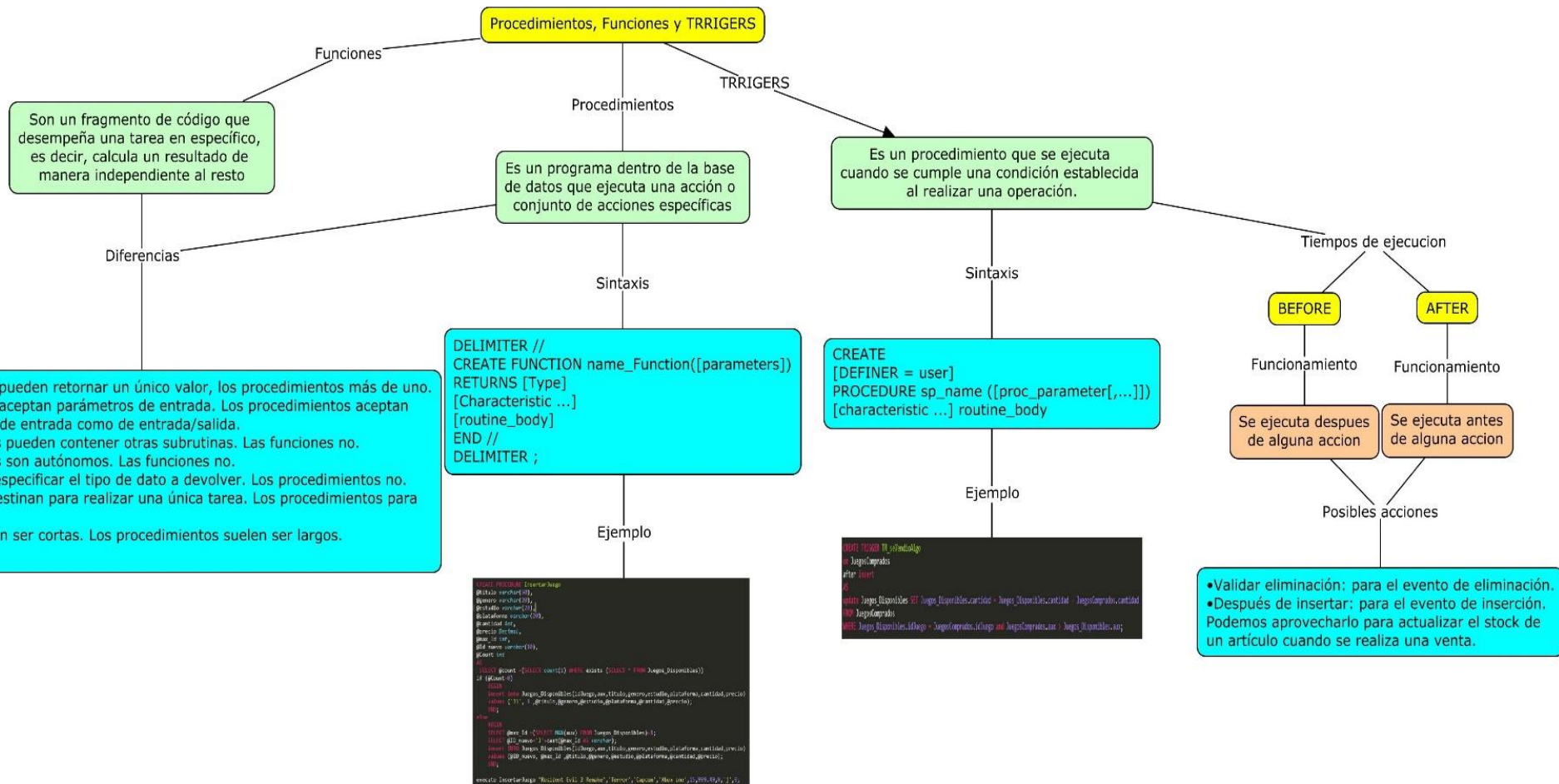
```

CREATE PROCEDURE InsertarJuego
@titulo varchar(50),
@genero varchar(20),
@estudio varchar(20),
@plataforma varchar(20),
@cantidad int,
@precio Decimal,
@max_Id int,
@Id_nuevo varchar(10),
@Count int
AS
SELECT @count =(SELECT count(1) WHERE exists (SELECT * FROM Juegos_Disponibles))
if (@Count=0)
BEGIN
insert into Juegos_Disponibles(idJuego,aux,titulo,genero,estudio,plataforma,cantidad,precio)
values ('J1', 1 ,@titulo,@genero,@estudio,@plataforma,@cantidad,@precio);
END;
else
BEGIN
SELECT @max_Id =(SELECT MAX(aux) FROM Juegos_Disponibles)+1;
SELECT @ID_nuevo='J'+cast(@max_Id AS varchar);
insert INTO Juegos_Disponibles(idJuego,aux,titulo,genero,estudio,plataforma,cantidad,precio)
values (@ID_nuevo, @max_Id ,@titulo,@genero,@estudio,@plataforma,@cantidad,@precio);
END;

execute InsertarJuego 'Resident Evil 2 Remake', 'Terror', 'Capcom', 'Xbox one', 15, 999.49, 0, 'j', 0;

```

# Mapa Conceptual Procedimientos, Funciones y TRRIGERS



## Referencias

Cristoba, J. (05 de 25 de 2016). *YouTube*. Obtenido de [https://www.youtube.com/watch?v=\\_vtWXfKaHH8](https://www.youtube.com/watch?v=_vtWXfKaHH8)

Perez, C. G. (11 de 07 de 2008). *Adictos al trabajo*. Obtenido de <https://www.adictosaltrabajo.com/2008/07/11/triggersql/>

CIBERTEC. (s.f.). *Base de Datos Avanzado I*. Recuperado el 08 de 11 de 2019, de *Base de Datos Avanzado I*.

Fuente: Alejandro Alcalde (Oct 7, 2017), *PL/SQL. Disparadores o Triggers*, recuperado de: <https://elbauldelprogramador.com/plsql-disparadores-o-triggers/>

mappinggis. (5 de julio de 2016). Obtenido de mappinggis: <https://mappinggis.com/2016/06/crear-ejecutar-disparador-trigger-postgis/>

NE. (NE). *Cómo Crear Funciones en MySQL*. 2019, de nosolocodigo.com Sitio web: <https://www.nosolocodigo.com/como-crear-funciones-en-mysql>

Sergio Perales. (2013). *Funciones en MySQL*. 2019, de openwebinars.net Sitio web: <https://openwebinars.net/blog/funciones-en-mysql/>