



**UNIVERSIDAD TECNOLÓGICA DE PANAMÁ**  
**FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES**  
**LABORATORIO # 8**  
**FC-FISC**



Facilitador(a): \_\_\_\_\_ Asignatura: \_\_\_\_\_ Estudiante: \_\_\_\_\_  
Fecha: \_\_\_\_\_ Grupo: \_\_\_\_\_

**A. TÍTULO DE LA EXPERIENCIA: CAPÍTULO V**

**B. TEMAS:**

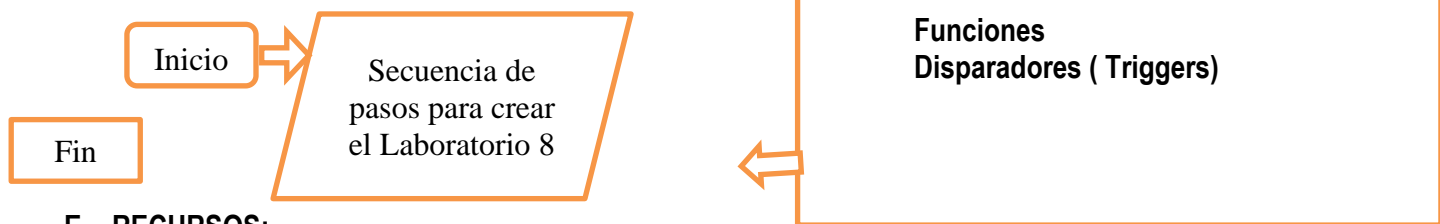
Funciones - PARTE I

Disparadores ( Triggers) - PARTE II

**OBJETIVO(S):** Lenguaje de programación Transact- SQL(Funcionalidades)

**C. METODOLOGÍA:** Laboratorio práctico en clases

**D. PROCEDIMIENTO O ENUNCIADO DE LA EXPERIENCIA:**



**E. RECURSOS:**

- Puntualidad en la asistencia a la hora correspondiente al laboratorio
- Equipo computacional completo (CPU, Teclado, Mouse, Monitor o Pantalla), software (Microsoft SQL Server Managment Studio), Tablero, Marcadores y borrador
- Libretas y bolígrafo o lápiz, para anotaciones

**F. INTRODUCCION:**

Los laboratorios son proyectos simplificados, que pueden generarse rápidamente y poner en práctica los temas del contenido. No pretenden tener la complejidad de proyectos reales. Los laboratorios tienen estructuras de tareas, nombres y cantidades de recursos ficticios, que no reflejan los valores de una BD real. Un proyecto real generalmente requiere de mayores niveles de detalle y complejidad en el plan de trabajo que los que se utilizan en los laboratorios. Las instrucciones que encontrará en los laboratorios no son 100% detalladas ni compatibles con la versión que tengan en sus equipos, tomar en cuenta, pues puede haber ciertas variaciones.

**G. Instrucciones:**

**Ir guardando las funciones, scripts o lo que se pida dentro de una Base de Datos, crear un script con todo lo del lab#8. PARTE IyII**

La sintaxis básica es:

```
create function NOMBRE  
(@PARAMETRO TIPO=VALORPORDEFEECTO)  
returns TIPO  
as  
begin  
    INSTRUCCIONES  
    return VALOR  
end;
```

### I. Realizar las siguientes funciones

1. Crear una función que retorne el promedio de dos números.
2. Crear una función que dado un nombre y apellido lo retorne todo en mayúscula y lo separe por coma, todo en un mismo campo, ejemplo: Araujo, Gionella
  - 2.1 Ejecute nuevamente la función pero usando los datos de la tabla cliente de la BD Almacén y que el resultado lo coloque en una columna llamada NOMBRECOMPLETO.
3. Crear una función que, dado el número del día, me devuelva el día de la semana en letras, por ejemplo, si mi argumento es 5, debe salir viernes.
4. Crear una función que calcule el valor del itbms a los precios de los productos, usar la BD Almacén para confirmar la funcionalidad de dicha función.

### II. Borrar una función

Se usa la siguiente sintaxis:  
**drop function nombredefunción**

5. Hacer una copia de alguna de las funciones creadas y luego borrarla.

### PARTE II

Definición:

**Triggers (disparador)** en una base de datos son eventos que se ejecutan cuando se cumplen condiciones determinadas, al hacer un **insert**, **update** o **delete** en una tabla de una BBDD. Esto nos permite realizar acciones cuando se realiza una inserción, modificación o eliminación de un registro. Son parecidos a los procedimientos almacenados, pero en este caso

no se tienen que llamar como los stored procedures, sino que como hemos dicho se ejecutan automáticamente al cumplirse la condición.

La utilidad principal de un trigger es mejorar la administración de la base de datos, ya que no requieren que un usuario los ejecute. Por lo tanto, son empleados para implementar las REGLAS DE NEGOCIO (tipo especial de integridad) de una base de datos. Una Regla de Negocio es cualquier restricción, requerimiento, necesidad o actividad especial que debe ser verificada al momento de intentar agregar, borrar o actualizar la información de una base de datos. 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.

Un trigger se asocia a un evento (inserción, actualización o borrado) sobre una tabla.

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".

### Sintaxis básica:

```
Create Trigger NOMBREDISPARADOR
on NOMBRETABLA ó NombreVista
For/ Before/ After EVENTO- insert, update o delete
as
SENTENCIAS
```

**Ejemplo: En una Bd de una aerolínea, crear un Trigger que indique un mensaje de registro una vez se inserte un nuevo pasajero**

```
create trigger mensaje
on pasajero
for insert
as
print 'pasajero registrado'
go
```

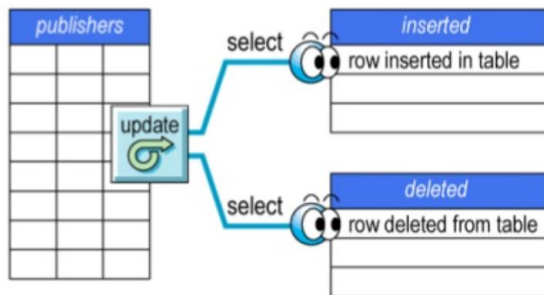
### Usando la BD de Almacén, implemente

1. Realizar un trigger que realice la misma función que el ejemplo anterior, pero una vez que se ingrese una sucursal nueva
  - 1.1 refresque la BD
  - 1.2 realice una nueva inserción y vea el mensaje
2. Realizar un nuevo trigger que realice lo mismo, pero cuando se actualice, que permita mostrar el mensaje: **Sucursal actualizada satisfactoriamente** ‘
  - 2.1 Refresque la BD
  - 2.2 Realice una nueva actualización y vea el mensaje

Es importante en triggers las tablas temporales, prestar atención a la siguiente imagen.

### Trigger - Tablas *inserted* y *deleted*

- ***inserted*** y ***deleted*** son dos tablas que se crean automáticamente cada vez que se dispara un trigger
  - ***inserted*** almacena cualquier fila que se vaya a añadir a la tabla
  - ***deleted*** almacena cualquier fila que se vaya a borrar de la tabla



### EJEMPLO DEL FUNCIONAMIENTO DE UN TRIGGER DELETE

**Instrucción DELETE en tabla con desencadenador DELETE**

DELETE Categories  
 WHERE  
 CategoryID = 4

| Categories |              |                         |         |
|------------|--------------|-------------------------|---------|
| CategoryID | CategoryName | Description             | Picture |
| 1          | Beverages    | Soft drinks, coffees... | 0x15... |
| 2          | Condiments   | Sweet and savory ...    | 0x15... |
| 3          | Confections  | Desserts, candies, ...  | 0x15... |

**Instrucción DELETE registrada**

| Deleted |                |         |         |
|---------|----------------|---------|---------|
| 4       | Dairy Products | Cheeses | 0x15... |

3. Implementar un trigger que permita crear una réplica(copia) de los registros que se inserten en la tabla PRODUCTO, para dicho proceso implementar una nueva tabla llamada PRODUCTObck
  - 3.1 Crear la tabla nueva
  - 3.2 Crear el trigger
  - 3.3 Probar el trigger haciendo una inserción de datos
  - 3.4 Visualizar la tabla PRODUCTObck

\*\*\*\*\*SUERTE\*\*\*\*\*