



**-UNIVERSIDAD TECNOLÓGICA DE PANAMÁ**  
**FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES**  
**LABORATORIO # 5**  
**FC-FISC**



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

**A. TÍTULO DE LA EXPERIENCIA:** SQL – SENTENCIAS VARIADAS - PROCEDIMIENTOS ALMACENADOS

**B. TEMAS:**

*Sentencias y propiedades variadas de SQL*  
*Procedimientos almacenados*

**OBJETIVO(S):** Lenguaje de manipulación de datos en SQL Server, usando con ejemplos sus principales sentencias, lenguaje de programación Transact- SQL

**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:**

Al igual que el laboratorio pasado, ir guardando todo en un solo query llamado LABBD5, la segunda parte si guardar consulta x consulta.

**PRIMERA PARTE**  
**PASOS:**

1. Crear la siguiente tabla con la siguiente información para poder comprender el uso de la propiedad identity(para hacer campos auto numéricos)

create database LAB5IDENTITY

```
create table articulos(  
codigo int identity,  
nombre varchar(20),  
descripcion varchar(30),  
precio smallmoney,  
cantidad tinyint default 0,  
primary key (codigo)  
);
```

```
insert into articulos (nombre, descripcion, precio,cantidad)  
values ('impresora','Epson Stylus C45',400.80,20);  
insert into articulos (nombre, descripcion, precio)  
values ('impresora','Epson Stylus C85',500);  
insert into articulos (nombre, descripcion, precio)  
values ('monitor','Samsung 14',800);  
insert into articulos (nombre, descripcion, precio,cantidad)  
values ('teclado','ingles Biswal',100,50);
```

Observe que en el insert no está la columna llamada código, pero al estar como identity queda inmediatamente como autonumérico.

2. Insertar una fila más desde la vista de diseño y observar que pasa con el campo de código
3. Crear nuevamente una tabla como la anterior, renombrarla con otro nombre y que el código se incremente de 2 en dos, a partir de 1.

## **SEGUNDA PARTE**

### **GROUP BY Y HAVING**

**GROUP BY:** permite formar consultas agrupadas para extraer información global sobre los grupos formados.

**HAVING:** condición sobre los grupos formados.

Una vez que GROUP BY ha combinado los registros, HAVING muestra cualquier registro agrupado por la cláusula GROUP BY que satisfaga las condiciones de la cláusula HAVING.

HAVING es similar a WHERE, determina qué registros se seleccionan. Una vez que los registros se han agrupado utilizando GROUP BY, HAVING determina cuales de ellos se van a mostrar.

PASOS:

I. Utilizar las tablas creadas de ALMACEN y realizar lo siguiente:

1. **Se desea proyectar el nombre, apellido, código del vendedor y categoría de vendedor, deben agruparlo por categoría donde la misma sea igual a A-3**

2. Seleccionar la descripción y precio de los productos, agrupándolos por descripción donde el precio sea mayor o igual a 579.
3. Necesitamos conocer la descripción y el total de los precios agrupados por descripción, de los productos LAPTOP y PANEL.

### TERCERA PARTE

#### PROCEDIMIENTOS ALMACENADOS

- Para crear un procedimiento almacenado debemos emplear la sentencia CREATE PROCEDURE.
- Para ejecutar el procedimiento que ya está creado se usa la orden EXECUTE o EXEC
- Para declarar una variable usamos la orden DECLARE
- Para asignar valores a una variable usamos la orden SET
- Para mostrar mensajes podemos usar PRINT

#### 1--Crear un procedimiento almacenado que muestre un saludo

```
CREATE PROCEDURE paSaludo
AS
BEGIN
    PRINT 'Hola Mundo'
END
go
```

1. Corre el siguiente código y hacer el llamado en el mismo query nuevamente para ver el resultado, de la siguiente forma:

**execute paSaludo**

Formato:

```
create proc NOMBREPROCEDIMIENTO
@NOMBREPARAMETRO TIPO =VALORPORDEFEECTO
as SENTENCIAS;
```

#### Ejemplo:

Creamos un procedimiento que recibe el nombre de un autor como parámetro para mostrar todos los libros del autor solicitado:

```
create procedure pa_libros_autor
@autor varchar(30)
as
select titulo, editorial, precio-
from libros
```

```
where autor= @autor;
```

El procedimiento se ejecuta colocando "execute" (o "exec") seguido del nombre del procedimiento y un valor para el parámetro:

```
exec pa_libros_autor 'Borges';
```

2. Crear un procedimiento que recibe el nombre de una marca como parámetro para mostrar todos los productos asociados a esa marca más su descripción y precio, usar tabla almacén.
3. Crear un procedimiento almacenado para sumar dos números, que imprima el resultado
4. Hacer un procedimiento almacenado para dividir dos números y que mande un error si el divisor es igual a cero
5. Usar alguna de las tablas dadas de Almacén y crear un procedimiento almacenado que inserte una línea en alguna de sus tablas, usar como guía lo siguiente:

creamos un procedimiento almacenado para insertar usuarios.

```
---se crea el procedimiento almacenado
create procedure insertarusuarios
---se declaran los parametros que usaremos
@nombre varchar(50),@apellido varchar(50),@pais varchar(50)
as
---se hace un insert a la tabla usuarios y se envian los parametros
insert into usuarios
values(@nombre,@apellido,@pais)
```

Para usar el procedimiento almacenado e insertar un registro lo haremos así

```
1 ---se usa el procedimiento para ingresar un usuario enviando los parametros
2 insertarusuarios 'ana','turcios','costarica'
```

Eliminar procedimientos almacenados

**DROP {PROC|PROCEDURE} [nombreEsquema.]nombreProcedimiento [...n].**

Transact-SQL permite abreviar la palabra reservada PROCEDURE por PROC sin que ello afecte a la funcionalidad de la instrucción.

Ejemplos:

```
DROP PROCEDURE Dice_Hola;
```

Elimina el procedimiento llamado Dice\_Hola.

```
DROP PROC Dice_Hola;
```

Es equivalente, PROC y PROCEDURE indican lo mismo.

Para eliminar varios procedimientos de golpe, indicamos sus nombres separados por comas:

```
DROP PROCEDURE Dice_Hola, Ventas_anuales;
```

Elimina los procedimientos Dice\_Hola y Ventas\_anuales.

6. Eliminar el procedimiento almacenado del saludo que creamos al inicio