



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS



### BASE DE DATOS

PROFESOR:

Ing. Yadira Franco R

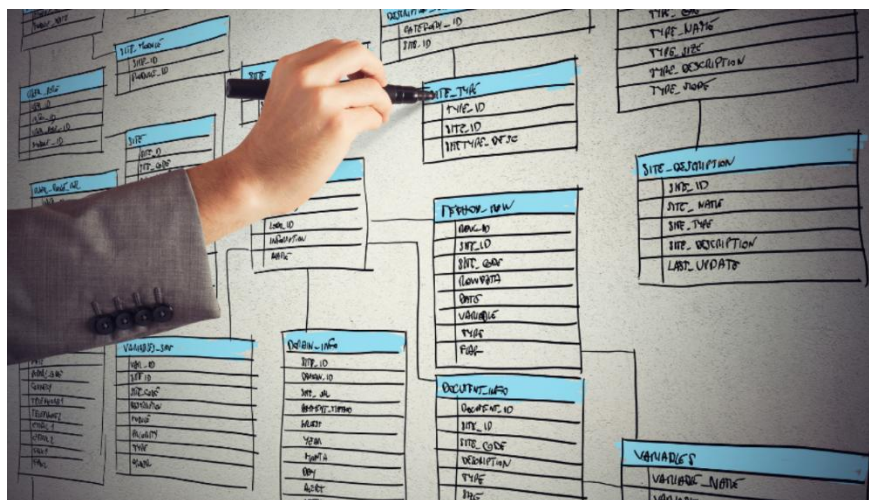
PERÍODO ACADÉMICO:

2024-B

### TAREA

TÍTULO:

INVESTIGACIÓN Y PRACTICA



Estudiante

Santiago Cumbal

2024-B

## INVESTIGAR QUE SON Procedimientos Almacenados en Bases de Datos

- Entender qué son los procedimientos almacenados y cómo funcionan.
- Aprender a crear procedimientos almacenados sencillos.
- PRACTICA - Realizar operaciones de **INSERT**, **SELECT**, **DELETE** y **UPDATE** usando procedimientos almacenados.
- **Revisión de Buenas Prácticas**

## Introducción a los Procedimientos Almacenados **MSQL- PostgreSQL – Sql Server**

### 1. Concepto y Beneficios de los Procedimientos Almacenados

- **Explicación:** Los procedimientos almacenados son conjuntos de instrucciones SQL que se guardan y ejecutan en el servidor de base de datos. Permiten ejecutar operaciones complejas, con seguridad, rendimiento optimizado y reutilización de código.
- **Beneficios:**
  - Reutilización de código.
  - Mejora en la seguridad (al evitar inyecciones SQL).
  - Optimización en el rendimiento de consultas frecuentes.
  - Consistencia en las operaciones realizadas.

### 2. ESPECIFICAR LA Sintaxis Básica de un Procedimiento Almacenado

- **Explicación:** El delimitador se cambia temporalmente para permitir el uso de **;** dentro del procedimiento.

#### Crear la tabla de cliente:

```
CREATE TABLE cliente (  
    ClienteID INT AUTO_INCREMENT PRIMARY KEY, -- Campo para el ID único del cliente  
    Nombre VARCHAR(100), -- Campo para el nombre del cliente  
    Estatura DECIMAL(5,2), -- Campo para la estatura del cliente con dos decimales  
    FechaNacimiento DATE, -- Campo para la fecha de nacimiento del cliente  
    Sueldo DECIMAL(10,2) -- Campo para el sueldo del cliente con dos decimales  
);
```

```

1 • CREATE DATABASE GestionClientes;
2 • USE GestionClientes;
3 • CREATE TABLE cliente (
4     ClienteID INT AUTO_INCREMENT PRIMARY KEY,
5     Nombre VARCHAR(100),
6     Estatura DECIMAL(5,2),
7     FechaNacimiento DATE,
8     Sueldo DECIMAL(10,2)
9 );
10
11 • INSERT INTO cliente (Nombre, Estatura, FechaNacimiento, Sueldo) VALUES
12 ('Carlos Gómez', 1.80, '1992-03-15', 3200.50),
13 ('Ana Martínez', 1.65, '1988-07-20', 2800.75),
14 ('Luis Fernández', 1.75, '1995-11-30', 2500.00),
15 ('Sofía Torres', 1.62, '1990-05-10', 3100.25),
16 ('Miguel Sánchez', 1.85, '1987-01-25', 3500.00);

```

### 3. Ejercicio 1: Crear un procedimiento simple que seleccione datos de la tabla cliente

```

delimiter $$
create procedure ObtenerClientes()
begin
    select * from cliente;
end$$
delimiter ;

```

### 4. Ejercicio: Ejecutar - LLAMAR el procedimiento

```

26 • call ObtenerClientes();
27

```

ClienteID	Nombre	Estatura	FechaNacimiento	Sueldo
1	Carlos Gómez	1.80	1992-03-15	3200.50
2	Ana Martínez	1.65	1988-07-20	2800.75
3	Luis Fernández	1.75	1995-11-30	2500.00
4	Sofía Torres	1.62	1990-05-10	3100.25
5	Miguel Sánchez	1.85	1987-01-25	3500.00

## Inserción, Actualización y Eliminación de Datos

### 1. Procedimiento de Inserción (INSERT)

- Crear un procedimiento que permita insertar un nuevo cliente en la tabla cliente

```

27 delimiter $$
28 • create procedure InsertarCliente( in Nombre varchar(100), in Estatura decimal(5,2), in FechaNacimiento date, in Sueldo decimal(10,2))
29 begin
30     insert into cliente (Nombre, Estatura, FechaNacimiento, Sueldo)
31     values (Nombre, Estatura, FechaNacimiento, Sueldo);
32 end$$
33 delimiter ;

```

- Ejecutar - LLAMAR el procedimiento

```

35 • call InsertarCliente('Juan Pérez', 1.75, '1990-05-10', 2500.00);
36 • call ObtenerClientes();
37

```

Result Grid   Filter Rows:   Export:   Wrap Cell Content: <a href="#">IA</a>					
ClienteID	Nombre	Estatura	FechaNacimiento	Sueldo	
2	Ana Martínez	1.65	1988-07-20	2800.75	
3	Luis Fernández	1.75	1995-11-30	2500.00	
4	Sofía Torres	1.62	1990-05-10	3100.25	
5	Miguel Sánchez	1.85	1987-01-25	3500.00	
6	Juan Pérez	1.75	1990-05-10	2500.00	

## 2. Procedimiento de Actualización (UPDATE)

Actualizar la edad de un cliente específico:

```

37 delimiter $$
38 • create procedure ActualizarSueldo(in p_ClienteID int, in p_NuevoSueldo decimal(10,2))
39 begin
40     update cliente set Sueldo = p_NuevoSueldo where ClienteID = p_ClienteID;
41 end$$
42 delimiter ;

```

## 3. Procedimiento de Eliminación (DELETE)

Eliminar un cliente de la base de datos usando su ClienteID:

```

47 delimiter $$
48 • create procedure EliminarCliente( in p_ClienteID int )
49 begin
50     delete from cliente where ClienteID = p_ClienteID;
51 end$$
52 delimiter ;
53 • call EliminarCliente(1);

```

Result Grid   Filter Rows:   Export:   Wrap Cell Content: <a href="#">IA</a>					
ClienteID	Nombre	Estatura	FechaNacimiento	Sueldo	
2	Ana Martínez	1.65	1988-07-20	2800.75	
3	Luis Fernández	1.75	1995-11-30	2500.00	
4	Sofía Torres	1.62	1990-05-10	3100.25	
5	Miguel Sánchez	1.85	1987-01-25	3500.00	
6	Juan Pérez	1.75	1990-05-10	2500.00	

## Introducción a Condiciones en Procedimientos Almacenados

### Uso de Condicionales (IF)

El uso de condicionales dentro de los procedimientos es fundamental para tomar decisiones basadas en los datos.

Verifica si la edad de un cliente es mayor o igual a 22:

```
37 delimiter $$
38 • create procedure ActualizarSueldo(in p_ClienteID int, in p_NuevoSueldo decimal(10,2))
39 • begin
40     update cliente set Sueldo = p_NuevoSueldo where ClienteID = p_ClienteID;
41 end$$
42 delimiter ;
43
44 • call ActualizarSueldo(1, 3000.00);
45 • call ObtenerClientes();
46
47 delimiter $$
48 • create procedure EliminarCliente( in p_ClienteID int )
49 • begin
50     delete from cliente where ClienteID = p_ClienteID;
51 end$$
52 delimiter ;
```

```
56 • CALL VerificarEdad('2000-01-01', @resultado);
```

```
57 • SELECT @resultado;
```

```
58
```

Result Grid		Filter Rows:	Export:	Wrap
	@resultado			
▶	1			

## Creación de la Tabla de Órdenes CON RELACIÓN CON EL CLIENTE - FORANEA

Para almacenar las órdenes de los clientes, se debe crear la tabla **ordenes**:

- Procedimientos de Órdenes -Insertar Orden
- Procedimientos Actualizar Orden
- Procedimientos Eliminar Orden

```

DELIMITER $$
CREATE PROCEDURE InsertarOrden(IN p_ClienteID INT, IN p_FechaOrden DATE, IN p_MontoTotal DECIMAL(10,2))
BEGIN
    INSERT INTO ordenes (ClienteID, FechaOrden, MontoTotal)
    VALUES (p_ClienteID, p_FechaOrden, p_MontoTotal);
END$$
DELIMITER ;

DELIMITER $$
CREATE PROCEDURE ActualizarOrden(IN p_OrdenID INT, IN p_NuevoMonto DECIMAL(10,2))
BEGIN
    UPDATE ordenes
    SET MontoTotal = p_NuevoMonto
    WHERE OrdenID = p_OrdenID;
END$$
DELIMITER ;

DELIMITER $$
CREATE PROCEDURE EliminarOrden(IN p_OrdenID INT)
BEGIN
    DELETE FROM ordenes
    WHERE OrdenID = p_OrdenID;
END$$
DELIMITER ;

```

## Entrega Final

### Instrucciones de Entrega:

#### 1. Objetivos:

Crear procedimientos almacenados para **insertar, actualizar, eliminar y consultar** registros en las tablas cliente y ordenes.

#### 2. Archivo de Script:

Los estudiantes deben escribir y guardar el código SQL con todos los procedimientos mencionados.

#### 3. Documento PDF:

Incluir las capturas de pantalla y explicaciones detalladas de los pasos realizados durante la tarea.

#### 4. Subida a GitHub:

Subir el script .sql y el documento PDF a un repositorio en GitHub para su REVISIÓN