



## BASE DE DATOS

PROFESOR:

Ing. Yadira Franco R

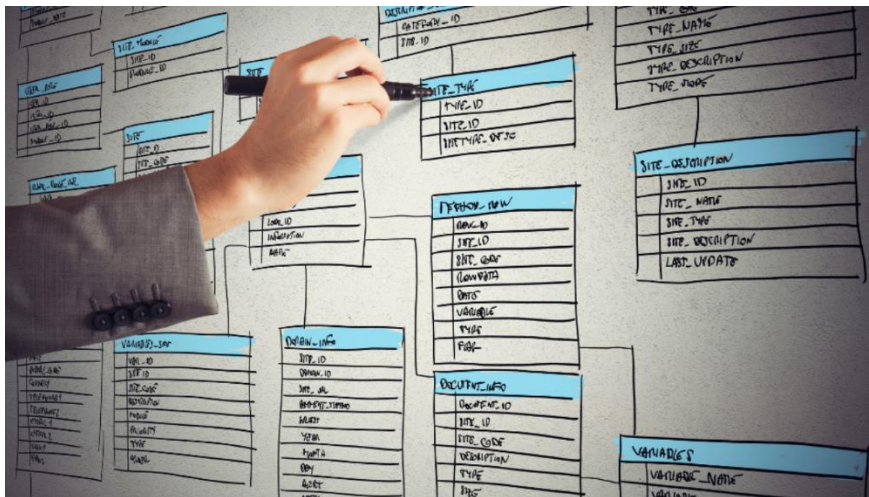
PERÍODO ACADÉMICO:

2024-B

## TAREA

TÍTULO:

## INVESTIGACIÓN Y PRACTICA



Estudiante

Mateo Moran

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 tareadb;
2 • use tareadb;
3
4 -- Creacion tabla clientes
5 • CREATE TABLE Cliente(
6     ClienteID int auto_increment primary key,
7     Nombre varchar(100),
8     Estatura decimal (5,2),
9     Edad int,
10    FechaNacimiento Date,
11    Sueldo decimal(10,2)
12 );
13
14 -- Creacion tabla ordenes
15 • CREATE TABLE Ordenes(
16     OrdenID int auto_increment primary key,
17     FechaOrden Date,
18     TotalOrden decimal(10,2),
19     ClienteID int,
20     FOREIGN KEY (ClienteID) REFERENCES Cliente(ClienteID)
21 );
22

```

CREACION DE BASE DE DATOS, TABLA CLIENTES Y ORDENES

3. Ejercicio 1: Crear un procedimiento simple que seleccione datos de la tabla cliente
4. Ejercicio: Ejecutar - LLAMAR el procedimiento

## Inserción, Actualización y Eliminación de Datos

### 1. Procedimiento de Inserción (INSERT)

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

- Ejecutar - LLAMAR el procedimiento

```
25  -- 1. Procedimiento de Inserción (INSERT)
26  DELIMITER $$
27  ● CREATE PROCEDURE InsertarCliente(
28      IN p_Nombre VARCHAR(100),
29      IN p_Estatura DECIMAL(5,2),
30      IN p_FechaNacimiento DATE,
31      IN p_Sueldo DECIMAL(10,2),
32      IN p_Edad int
33  )
34  ● BEGIN
35      INSERT INTO Cliente (Nombre, Estatura, FechaNacimiento, Sueldo, Edad)
36      VALUES (p_Nombre, p_Estatura, p_FechaNacimiento, p_Sueldo, p_Edad);
37  END$$
38  DELIMITER ;
39
40  ● CALL InsertarCliente('Camila Egas', 1.58, '2004-02-16', 5000.00,20);
41  ● CALL InsertarCliente('Mateo Moran', 1.65, '2004-02-20', 3000.00,20);
42
43  SELECT * FROM Cliente
```

	ClienteID	Nombre	Estatura	Edad	FechaNacimiento	Sueldo
▶	1	Camila Egas	1.58	20	2004-02-16	5000.00
	2	Mateo Moran	1.65	20	2004-02-20	3000.00
*	NULL	NULL	NULL	NULL	NULL	NULL

Se crea un procedimiento almacenado donde se usa la instrucción INSERT para insertar datos en la tabla cliente.

## 2. Procedimiento de Actualización (UPDATE)

Actualizar la edad de un cliente específico:

```
45  -- 2. Procedimiento de Actualización (UPDATE)
46  DELIMITER $$
47  • CREATE PROCEDURE ActualizarEdad(
48      IN p_ClienteID INT,
49      IN p_NuevaEdad INT
50  )
51  • BEGIN
52      UPDATE Cliente
53      SET Edad = p_NuevaEdad
54      WHERE ClienteID = p_ClienteID;
55  • END$$
56  DELIMITER ;
57  • CALL ActualizarEdad(1, 35);
```

1	Camila Egas	1.58	35	2004-02-16	5000.00
2	Mateo Moran	1.65	20	2004-02-20	3000.00
•	NULL	NULL	NULL	NULL	NULL

Se crea un procedimiento de almacenado llamado actualizar edad, con el uso de update se actualiza la edad de un cliente especificando su ID.

## 3. Procedimiento de Eliminación (DELETE)

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

```
60  -- 3. Procedimiento de Eliminación (DELETE)
61  DELIMITER $$
62  • CREATE PROCEDURE EliminarCliente(
63      IN p_ClienteID INT
64  )
65  • BEGIN
66      DELETE FROM Cliente
67      WHERE ClienteID = p_ClienteID;
68  • END$$
69  DELIMITER ;
70  • CALL EliminarCliente(1);
71  • SELECT * FROM Cliente
```

	ClienteID	Nombre	Estatura	Edad	FechaNacimiento	Sueldo
►	2	Mateo Moran	1.65	20	2004-02-20	3000.00

Se crea un procedimiento almacenado donde se utiliza la instrucción DELETE para eliminar un cliente con un ID específico.

## 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:

```
73  -- 4. Procedimiento de verificar edad mayor o igual a 22
74  DELIMITER $$
75  ● CREATE PROCEDURE VerificarEdadCliente(
76      IN p_ClienteID INT
77  )
78  BEGIN
79      DECLARE p_Edad INT;
80      SELECT Edad INTO p_Edad
81      FROM Cliente
82      WHERE ClienteID = p_ClienteID;
83      IF p_Edad >= 22 THEN
84          SELECT 'La edad del cliente es mayor o igual a 22 años' AS Resultado;
85      ELSE
86          SELECT 'La edad del cliente es menor a 22 años' AS Resultado;
87      END IF;
88  END$$
89  DELIMITER ;
90  ● CALL VerificarEdadCliente(2);
```

Resultado
▶ La edad del cliente es menor a 22 años

Creación del procedimiento almacenado que verifica la edad del cliente, si es mayor o igual a 22 años se mostrara en pantalla.

### 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**:

```
14  -- Creacion tabla ordenes
15  ● CREATE TABLE Ordenes(
16      OrdenID int auto_increment primary key,
17      FechaOrden Date,
18      TotalOrden decimal(10,2),
19      ClienteID int,
20      FOREIGN KEY (ClienteID) REFERENCES Cliente(ClienteID)
21  );
```

- Procedimientos de Órdenes -Insertar Orden

```

92  -- 1. Procedimientos de Órdenes -Insertar Orden
93  DELIMITER $$
94  ● CREATE PROCEDURE InsertarOrden(
95      |     IN p_FechaOrden DATE,
96      |     IN p_TotalOrden DECIMAL(10,2),
97      |     IN p_IdCliente int
98      | )
99      ○ BEGIN
100         | INSERT INTO Ordenes (FechaOrden, TotalOrden, ClienteID)
101         | VALUES (p_FechaOrden, p_TotalOrden, p_IdCliente);
102         | END$$
103  DELIMITER ;
104
105  ● CALL InsertarOrden('2024-12-17', 150.00, 2);
106  SELECT * FROM ordenes

```

	OrdenID	FechaOrden	TotalOrden	ClienteID
▶	2	2024-12-17	150.00	2

- Procedimientos Actualizar Orden

```

109  -- 2. Procedimientos Actualizar Orden
110  DELIMITER $$
111  ● CREATE PROCEDURE ActualizarOrden(
112      |     IN p_OrdenID INT,
113      |     IN p_TotalOrden DECIMAL(10,2)
114      | )
115      ○ BEGIN
116         | UPDATE Ordenes
117         | SET TotalOrden = p_TotalOrden
118         | WHERE OrdenID = p_OrdenID;
119         | END$$
120  DELIMITER ;

```

	OrdenID	FechaOrden	TotalOrden	ClienteID
▶	2	2024-12-17	200.00	2
*	NULL	NULL	NULL	NULL

- Procedimientos Eliminar Orden

```
125  -- 3. Procedimientos Eliminar Orden
126  DELIMITER $$
127  ● ○ CREATE PROCEDURE EliminarOrden(
128      IN p_OrdenID INT
129  )
130  ○ BEGIN
131      DELETE FROM Ordenes
132      WHERE OrdenID = p_OrdenID;
133  END$$
134  DELIMITER ;
```

	OrdenID	FechaOrden	TotalOrden	ClienteID
*	NULL	NULL	NULL	NULL

## 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

### REPOSITORIO GITHUB

[https://github.com/MateoXtra/Tarea\\_Proc\\_Almacenado.git](https://github.com/MateoXtra/Tarea_Proc_Almacenado.git)