

# **ESCUELA POLITÉCNICA NACIONAL**

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS



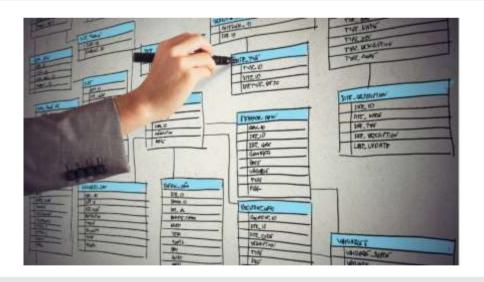
#### **BASE DE DATOS**

PROFESOR: Ing. Yadira Franco R

PERÍODO ACADÉMICO: 2024-B

## **TAREA**

# TÍTULO: INVESTIGACIÓN Y PRACTICA



Estudiante

XXXXXXXXXXXX

#### **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

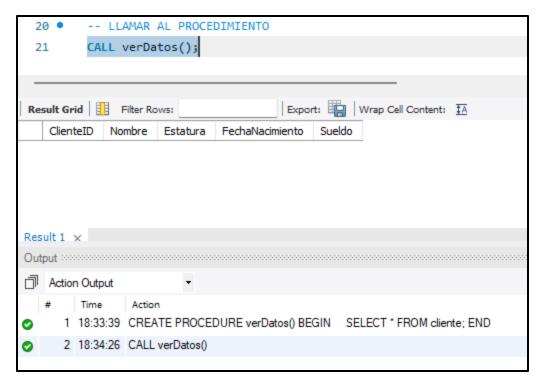
);

```
4 • ⊖ 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
  6
            Estatura DECIMAL(5,2),
                                                         -- Campo para la estatura del cliente con dos decimales
            FechaNacimiento DATE,
                                                         -- Campo para la fecha de nacimiento del cliente
  8
  9
             Sueldo DECIMAL(10,2)
                                                         -- Campo para el sueldo del cliente con dos decimales
 10
Output sessesses
Action Output
      Time
                                                                                            Message
                                                                                            1 row(s) affected
     1 18:25:11 CREATE DATABASE tarea
     2 18:25:17 USE tarea
                                                                                            0 row(s) affected
      3 18:28:43 CREATE TABLE cliente ( ClienteID INT AUTO_INCREMENT PRIMARY KEY, -- Campo para el ID ...
                                                                                           0 row(s) affected
```

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

```
-- PROCEDIMIENDO PARA SELECCIONAR TODOS LOS DATOS DE LA TABLA
 13
       DELIMITER //
       CREATE PROCEDURE verDatos()
 14 •
 15 ⊝ BEGIN
         SELECT * FROM cliente;
 16
 17
       END //
       DELIMITER ;
 18
 19
Output
Action Output
      Time
            Action
    1 18:33:39 CREATE PROCEDURE verDatos() BEGIN SELECT * FROM cliente; END
```

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

```
29
       DELIMITER //
30 • ○ CREATE PROCEDURE agregarCliente(
           IN nombre VARCHAR(100),
31
          IN estatura DECIMAL(5,2),
32
33
           IN fechaNacimiento DATE,
           IN sueldo DECIMAL(10,2)
34
35

→ BEGIN

36
           INSERT INTO cliente (Nombre, Estatura, FechaNacimiento, Sueldo)
37
           VALUES (nombre, estatura, fechaNacimiento, sueldo);
38
       END //
39
40
       DELIMITER;
       -- LLAMAR AL PROCEDIMIENTO
41 •
utput
Action Output
      Time
             Action

    1 19:56:12 CREATE PROCEDURE agregarCliente( IN nombre VARCHAR(100), IN estatura DECIMAL(5,2.
```

#### - Ejecutar - LLAMAR el procedimiento

```
41 • -- LLAMAR AL PROCEDIMIENTO

42 CALL agregarCliente('Kevin Flores', 1.75, '2003-05-15', 980.50);

Output

# Time Action

1 19:56:12 CREATE PROCEDURE agregarCliente( IN nombre VARCHAR(100), IN estatura DECIMAL(5,2.
```

#### 2. Procedimiento de Actualización (UPDATE)

Actualizar la edad de un cliente específico:

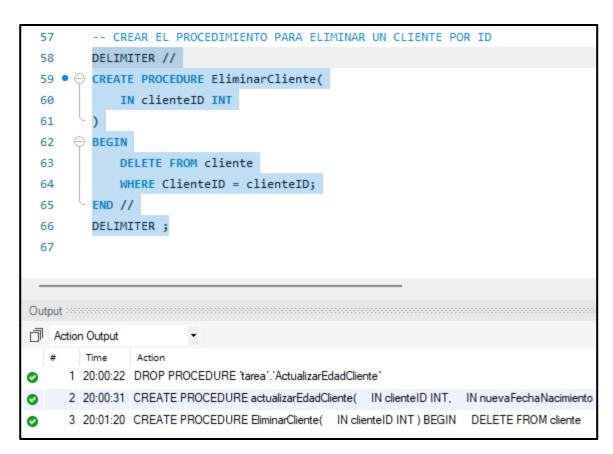
```
-- CREAR EL PROCEDIMIENTO PARA ACTUALIZAR LA FECHA DE NACIMIENTO
 44
         DELIMITER //
 45
 46 • ○ CREATE PROCEDURE actualizarEdadCliente(
 47
              IN clienteID INT,
              IN nuevaFechaNacimiento DATE
 48
 49

→ BEGIN

 50
             UPDATE cliente
 51
              SET FechaNacimiento = nuevaFechaNacimiento
 52
 53
             WHERE ClienteID = clienteID;
        END //
 54
         DELIMITER ;
 55
 56
Output
Action Output
        Time
                Action
      1 20:00:22 DROP PROCEDURE 'tarea'. 'ActualizarEdadCliente'
      2 20:00:31 CREATE PROCEDURE actualizarEdadCliente( IN clienteID INT, IN nuevaFechaNacimiento
```

#### 3. Procedimiento de Eliminación (DELETE)

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



#### 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:

```
-- CONDICIONAL IF
         DELIMITER //
 69
 70 • ○ CREATE PROCEDURE verificarEdadCliente(
             IN clienteID INT,
 71
 72
             OUT esMayor22 BOOLEAN
 73

→ BEGIN

 74
            DECLARE edad INT;
 75
 76
             -- Calcular la edad en años basada en la fecha de nacimiento
 77
             SELECT TIMESTAMPDIFF(YEAR, FechaNacimiento, CURDATE()) INTO edad
 78
 79
             FROM cliente
             WHERE ClienteID = clienteID;
 80
Output
Action Output
     1 20:00:22 DROP PROCEDURE 'tarea'. 'Actualizar Edad Cliente'
     2 20:00:31 CREATE PROCEDURE actualizarEdadCliente( IN clienteID INT, IN nuevaFechaNacimiento D
     3 20:01:20 CREATE PROCEDURE EliminarCliente( IN clienteID INT) BEGIN DELETE FROM cliente
     4 20:09:30 CREATE PROCEDURE EliminarCliente( IN clienteID INT) BEGIN DELETE FROM cliente
     5 20:09:32 CREATE PROCEDURE verificarEdadCliente( IN clienteID INT, OUT esMayor22 BOOLEAN)
```

#### 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

```
-- CREAR LA TABLA ORDENES CON SUS RELACIONES

    ○ CREATE TABLE ordenes (
 92
             OrdenID INT AUTO_INCREMENT PRIMARY KEY,
 93
             ClienteID INT,
 94
 95
             FechaOrden DATE,
             Monto DECIMAL(10,2),
 96
             Descripcion VARCHAR(255),
 97
              FOREIGN KEY (ClienteID) REFERENCES cliente(ClienteID) ON DELETE CASCADE
 98
 99
100
Output
Action Output
        Time
                Action
      1 20:17:19 CREATE TABLE ordenes ( OrdenID INT AUTO_INCREMENT PRIMARY KEY, ClienteID INT, ...
```

```
100
         -- PROCEDIMIENTO PARA INGRESAR UNA ORDEN
         DELIMITER //
101
102 • ⊖ CREATE PROCEDURE ingresartarOrden(
             IN clienteID INT,
103
104
            IN fechaOrden DATE,
            IN monto DECIMAL(10,2),
105
             IN descripcion VARCHAR(255)
106
107

→ BEGIN

108
              INSERT INTO ordenes (ClienteID, FechaOrden, Monto, Descripcion)
109
              VALUES (clienteID, fechaOrden, monto, descripcion);
110
         END //
111
         DELIMITER;
112
113
Action Output
                Action
       Time
      1 20:17:19 CREATE TABLE ordenes ( OrdenID INT AUTO_INCREMENT PRIMARY KEY, ClienteID INT, ...
      2 20:19:11 CREATE TABLE ordenes ( OrdenID INT AUTO_INCREMENT PRIMARY KEY, ClienteID INT, ...
      3 20:19:23 CREATE PROCEDURE ingresartarOrden( IN clienteID INT, IN fechaOrden DATE, IN mont...
```

Procedimientos Actualizar Orden

```
-- PROCEDIMIENTO PARA ACTUALIZAR UNA ORDEN
114
        DELIMITER //
115
116 • ⊖ CREATE PROCEDURE ActualizarOrden(
           IN ordenID INT,
117
           IN nuevoMonto DECIMAL(10,2),
118
           IN nuevaDescripcion VARCHAR(255)
119
120
UPDATE ordenes
122
           SET Monto = nuevoMonto, Descripcion = nuevaDescripcion
123
           WHERE OrdenID = ordenID;
124
       END //
125
        DELIMITER;
126
127
Output
Action Output
       Time
     2 20:19:11 CREATE TABLE ordenes ( OrdenID INT AUTO_INCREMENT PRIMARY KEY, ClienteID IN...
     3 20:19:23 CREATE PROCEDURE ingresartarOrden( IN clienteID INT, IN fechaOrden DATE, IN mon...
     4 20:20:12 CREATE PROCEDURE ActualizarOrden( IN ordenID INT, IN nuevoMonto DECIMAL(10,2), ...
```

Procedimientos Eliminar Orden

```
-- PROCEDIMIENTO PARA ELIMINAR UNA ORDEN
128
          DELIMITER //
129
130 • ⊖ CREATE PROCEDURE eliminarOrden(
              IN ordenID INT
131
132

→ BEGIN

133
              DELETE FROM ordenes
134
              WHERE OrdenID = ordenID;
135
         END //
136
          DELIMITER;
137
138
Action Output
                Action
        Time
      3 20:19:23 CREATE PROCEDURE ingresartarOrden( IN clientelD INT, IN fechaOrden DATE, IN mon.
      4 20:20:12 CREATE PROCEDURE ActualizarOrden( IN ordenID INT, IN nuevo Monto DECIMAL(10,2),
      5 20:21:13 CREATE PROCEDURE eliminarOrden( IN ordenID INT) BEGIN
                                                                    DELETE FROM ordenes
```

#### **Entrega Final**

#### Instrucciones de Entrega:

#### 1. Objetivos:

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

#### 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