



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS



BASE DE DATOS

PROFESOR:

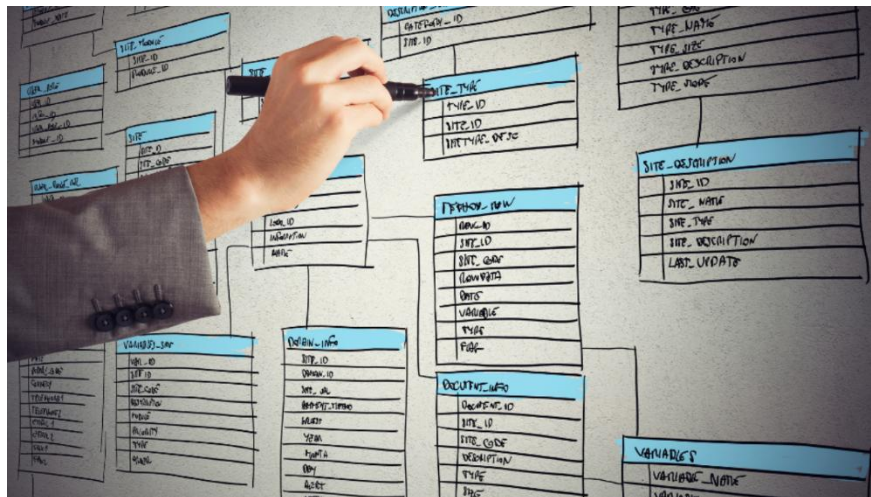
Ing. Yadira Franco R

PERÍODO ACADÉMICO:

2024-B

TAREA

TÍTULO: INVESTIGACIÓN Y PRACTICA



Estudiante

Richard Padilla

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  
);
```

```
SQL File 3*
Limit to 1000 rows

1  -- PROCESAMIENTO ALMACENADO
2  • CREATE DATABASE PA;
3  • USE PA;
4
5  • CREATE TABLE cliente (
6      ClienteID INT AUTO_INCREMENT PRIMARY KEY, -- Campo para el ID único del cliente
7      Nombre VARCHAR(100), -- Campo para el nombre del cliente
8      Estatura DECIMAL(5,2), -- Campo para la estatura del cliente con dos decimales
9      FechaNacimiento DATE, -- Campo para la fecha de nacimiento del cliente
10     Sueldo DECIMAL(10,2) -- Campo para el sueldo del cliente con dos decimales
11 );
12
```

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

```
11 );
12
13 DELIMITER $$
14
15 • CREATE PROCEDURE SeleccionarClientes()
16 BEGIN
17     SELECT * FROM cliente;
18 END $$
19
20 DELIMITER ;
21
22 • CALL SeleccionarClientes();
23
--
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

ClienteID	Nombre	Estatura	FechaNacimiento	Sueldo
-----------	--------	----------	-----------------	--------

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

```
28
29 • CREATE PROCEDURE InsertarCliente (
30     IN p_Nombre VARCHAR(100),
31     IN p_Estatura DECIMAL(5,2),
32     IN p_FechaNacimiento DATE,
33     IN p_Sueldo DECIMAL(10,2)
34 )
35 BEGIN
36     INSERT INTO cliente (Nombre, Estatura, FechaNacimiento, Sueldo)
37     VALUES (p_Nombre, p_Estatura, p_FechaNacimiento, p_Sueldo);
38 END $$
39
40 DELIMITER ;
41
42 • -- Ejecutar el procedimiento
43 CALL InsertarCliente('Juan Pérez', 1.75, '1990-05-15', 2500.00);
44
45
46
```

2. Procedimiento de Actualización (UPDATE)

Actualizar la edad de un cliente específico:

```
-- Procedimiento de Actualización (UPDATE)
DELIMITER $$

> CREATE PROCEDURE ActualizarSueldoCliente (
>     IN p_ClienteID INT,
>     IN p_NuevoSueldo DECIMAL(10,2)
> )
> BEGIN
>     UPDATE cliente
>     SET Sueldo = p_NuevoSueldo
>     WHERE ClienteID = p_ClienteID;
> END $$

DELIMITER ;

CALL ActualizarSueldoCliente(1, 3000.00);
```

3. Procedimiento de Eliminación (DELETE)

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

```
63
64 -- Procedimiento de Eliminación (DELETE)
65 DELIMITER $$
66
67 CREATE PROCEDURE EliminarCliente (
68     IN p_ClienteID INT
69 )
70 BEGIN
71     DELETE FROM cliente
72     WHERE ClienteID = p_ClienteID;
73 END $$
74
75 DELIMITER ;
76
77
78 CALL EliminarCliente(1);
79
```

Output:

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:

```
103 DECLARE v_Edad INT; -- Variable para almacenar la edad del cliente
104
105 -- Calcular la edad del cliente
106 SELECT TIMESTAMPDIFF(YEAR, FechaNacimiento, CURDATE())
107 INTO v_Edad
108 FROM cliente
109 WHERE ClienteID = p_ClienteID;
110
111 -- Verificar si la edad es mayor o igual a 22 años
112 IF v_Edad >= 22 THEN
113     INSERT INTO ordenes (ClienteID, FechaOrden, Total, Estado)
114     VALUES (p_ClienteID, p_FechaOrden, p_Total, p_Estado);
115 ELSE
116     SIGNAL SQLSTATE '45000'
117     SET MESSAGE_TEXT = 'El cliente debe tener al menos 22 años para realizar una orden.';
118 END IF;
119 END $$
120
121 DELIMITER ;
122
123
124 CALL InsertarOrden(1, '2024-06-28', 500.00, 'Pendiente');
125
```

Output:

#	Time	Action	Message
17	14:32:23	CALL InsertarOrden(1, '2024-06-28', 500.00, 'Pendiente')	Error Code: 1644. El cliente debe tener al menos 22 años para realizar una orden.
18	14:33:03	CREATE PROCEDURE InsertarOrden (IN p_ClienteID INT, IN p_FechaOrden DATE, IN p_Total DEC...	Error Code: 1304. PROCEDURE InsertarOrden already exists
19	14:33:09	CALL InsertarOrden(1, '2024-06-28', 500.00, 'Pendiente')	Error Code: 1644. El cliente debe tener al menos 22 años para realizar una orden.

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

```
94 DELIMITER $$
```

```
95
```

```
96 CREATE PROCEDURE InsertarOrden (
```

```
97     IN p_ClienteID INT,
```

```
98     IN p_FechaOrden DATE,
```

```
99     IN p_Total DECIMAL(10,2),
```

```
00     IN p_Estado VARCHAR(20)
```

```
01 )
```

```
02 BEGIN
```

```
126
```

```
127 -- Procedimiento para Actualizar una Orden
```

```
128 DELIMITER $$
```

```
129
```

```
130 CREATE PROCEDURE ActualizarOrden (
```

```
131     IN p_OrdenID INT,
```

```
132     IN p_NuevoEstado VARCHAR(20)
```

```
133 )
```

```
134 BEGIN
```

```
135     UPDATE ordenes
```

```
136     SET Estado = p_NuevoEstado
```

```
137     WHERE OrdenID = p_OrdenID;
```

```
138 END $$
```

```
139
```

```
140 DELIMITER ;
```

```
141
```

```
142 CALL ActualizarOrden(1, 'Pagado');
```

```
143
```

```
144 -- Procedimiento para Eliminar una Orden
```

```
145 DELIMITER $$
```

```
146
```

```
147 CREATE PROCEDURE EliminarOrden (
```

```
148     IN p_OrdenID INT
```

```
149 )
```

```
150 BEGIN
```

```
151     DELETE FROM ordenes
```

```
152     WHERE OrdenID = p_OrdenID;
```

```
153 END $$
```

```
154
```

```
155 DELIMITER ;
```

```
156
```

```
157 CALL EliminarOrden(1);
```

```
158
```

```
159
```

```
160
```

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