

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

CHNAG ALVAREZ ANTHON ANDRE

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
```

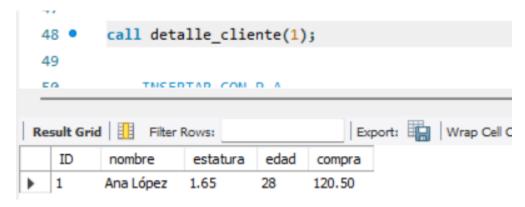
3. Ejercicio 1: Crear un procedimiento simple que seleccione datos de la tabla cliente

```
create procedure detalle_cliente(in p_id int)

description

descr
```

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

```
create procedure insertar_cliente(
    in p_nombre varchar(30),
    in p_estatura DECIMAL(5,2),
    in p_edad int,
    in p_compra DECIMAL(10,2)
)
begin
    insert into cliente(nombre, estatura, edad, compra)
    values (p_nombre, p_estatura, p_edad, p_compra);
end $$
delimiter;
call insertar_cliente('Luis Torres', 1.80, 35, 300.75);
```

2. Procedimiento de Actualización (UPDATE)

Actualizar la edad de un cliente específico:

```
e create procedure update_cliente(
    in p_id int,
    in p_edad int
)

begin
    update cliente set edad = p_edad where id = p_id;
end $$

delimiter;

call update_cliente(1, 30);
```

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:

```
DELIMITER $$
```

CREATE PROCEDURE verificar_edad(IN edad INT, OUT resultado VARCHAR(20))

BEGIN

IF edad >= 22 THEN

SET resultado = 'mayor de edad';

ELSE

SET resultado = 'menor de edad';

END IF;

END \$\$

DELIMITER ;

- call verificar_edad(25, @resultado);
- SELECT @resultado AS Estado;

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

```
-- INSERTAR ordenes
  delimiter $$
create procedure insertar_ordenes(
      p cliente id int,
      p fecha date,
      p_total decimal(10,2)
  )
⊖ begin
      if exists (select 1 from cliente where id = p cliente id) then
          insert into ordenes (cliente id, fecha, total)
          values (p_cliente_id, p_fecha, p_total);
      else
          signal sqlstate '45000'
          set message text = 'error: el cliente especificado no existe.';
      end if;
  end //
  delimiter;
call insertar_ordenes(2, '2024-07-29', 350.75);
```

```
delimiter //

        • create procedure actualizar_orden(

            in p_orden_id int,
            in p fecha date,
            in p_total decimal(10,2)

    ⇒ begin

           update ordenes
            set fecha = p_fecha, total = p_total
            where orden_id = p_orden_id;
       end //
       delimiter;
       call actualizar_orden(1, '2024-12-15', 500.00);

    Procedimientos Eliminar Orden

        delimiter //
   • 

create procedure eliminar_orden(
            in p_orden_id int
     ⊖ begin
            delete from ordenes
            where orden_id = p_orden_id;
       end //
        delimiter;
   call eliminar_orden(2);
```

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

https://github.com/Anthon-Chang/MYSQL ALMACENAMIENTO PROCESADO.git