

REPORTE DE PRÁCTICA 1.5

NOMBRE DE LA PRÁCTICA: 1.5 Práctica 0

ALUMNO: Axel Damian Ortiz Simon
Dr. Eduardo Cornejo-Velázquez



1. Marco teórico

En el ámbito de las bases de datos, es fundamental comprender y manejar diversos elementos que permiten una gestión eficiente y segura de la información. Entre estos elementos destacan los procedimientos almacenados, las funciones, las estructuras de control condicionales y repetitivas, y los disparadores. Cada uno de estos componentes desempeña un papel crucial en la manipulación y administración de datos, facilitando operaciones complejas y automatizando tareas dentro del sistema de gestión de bases de datos.

1. Procedimientos almacenados (Procedure)

Los procedimientos almacenados son conjuntos de instrucciones SQL que se almacenan en el servidor de bases de datos y se pueden ejecutar cuando se necesiten. Estos procedimientos permiten encapsular lógica de negocio en la base de datos, mejorando la eficiencia y seguridad al reducir el tráfico entre la aplicación y el servidor.

Ejemplo 1: Procedimiento para actualizar el estado de un vehículo

Listing 1: ActualizarEstadoVehiculo

```
DELIMITER //
CREATE PROCEDURE ActualizarEstadoVehiculo(
    IN p_id_Vehiculo INT,
    IN p_nuevo_estado VARCHAR(30)
)
BEGIN
    UPDATE Vehiculo
    SET estado = p_nuevo_estado
    WHERE id_Vehiculo = p_id_Vehiculo;
END //
DELIMITER ;
```

Ejemplo 2: Procedimiento para registrar una nueva ruta

Listing 2: RegistrarNuevaRuta

```
DELIMITER //
CREATE PROCEDURE RegistrarNuevaRuta(
    IN p_origen VARCHAR(50),
    IN p_destino VARCHAR(50),
    IN p_fecha_Inicio DATE,
    IN p_fecha_Fin DATE,
    IN p_consumo FLOAT,
    IN p_kilometraje FLOAT,
    IN p_id_Conductor INT,
    IN p_id_Vehiculo INT
)
BEGIN
    INSERT INTO Ruta (origen, destino, fecha_Inicio, fecha_Fin, consumo, kilometraje, id_Conductor, id_Vehiculo)
    VALUES (p_origen, p_destino, p_fecha_Inicio, p_fecha_Fin, p_consumo, p_kilometraje, p_id_Conductor, p_id_Vehiculo);
END //
DELIMITER ;
```

2. Funciones (Function)

Ejemplo 1: Función para calcular la edad de un usuario

Listing 3: CalcularEdad

```
DELIMITER //
```

```

CREATE FUNCTION CalcularEdad(fechaNacimiento DATE) RETURNS INT DETERMINISTIC
BEGIN
    RETURN TIMESTAMPDIFF(YEAR, fechaNacimiento, CURDATE());
END //
DELIMITER ;

```

Ejemplo 2: Función para obtener el saldo de un usuario

Listing 4: ObtenerSaldo

```

DELIMITER //
CREATE FUNCTION ObtenerSaldo(idUsuario INT) RETURNS DECIMAL(10,2) DETERMINISTIC
BEGIN
    DECLARE saldo DECIMAL(10,2);
    SELECT balance INTO saldo FROM Cuentas WHERE usuario_id = idUsuario;
    RETURN saldo;
END //
DELIMITER ;

```

3. Estructuras de control condicionales y repetitivas

Ejemplo 1: Uso de IF en un procedimiento

Listing 5: UsoDeIF

```

DELIMITER //
CREATE PROCEDURE VerificarSaldo(IN idUsuario INT, OUT mensaje VARCHAR(50))
BEGIN
    DECLARE saldo DECIMAL(10,2);
    SELECT balance INTO saldo FROM Cuentas WHERE usuario_id = idUsuario;
    IF saldo > 1000 THEN
        SET mensaje = 'Saldo suficiente';
    ELSE
        SET mensaje = 'Saldo insuficiente';
    END IF;
END //
DELIMITER ;

```

Ejemplo 2: Uso de WHILE en un procedimiento

Listing 6: UsoDeWhile

```

DELIMITER //
CREATE PROCEDURE ContarHastaN(IN limite INT)
BEGIN
    DECLARE contador INT DEFAULT 0;
    WHILE contador < limite DO
        SET contador = contador + 1;
    END WHILE;
END //
DELIMITER ;

```

4. Disparadores (Triggers)

Ejemplo 1: Trigger para auditar inserciones en una tabla

Listing 7: AuditarInserciones

```

CREATE TRIGGER InsertarLog AFTER INSERT ON Usuarios

```

```
FOR EACH ROW
BEGIN
    INSERT INTO Log (accion, usuario_id, fecha) VALUES ('INSERT', NEW.id, NOW());
END;
```

Ejemplo 2: Trigger para prevenir eliminación de datos importantes

Listing 8: PrevenirEliminacion

```
CREATE TRIGGER PrevenirEliminacion BEFORE DELETE ON Cuentas
FOR EACH ROW
BEGIN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No se permite eliminar cuentas';
END;
```

Referencias Bibliográficas

References

- [1] Coronel, C., Morris, S., Rob, P. (2020). *Database Systems: Design, Implementation, and Management* (13^a ed.). Cengage Learning.
- [2] Elmasri, R., Navathe, S. (2016). *Fundamentals of Database Systems* (7^a ed.). Pearson.
- [3] DuBois, P. (2008). *MySQL: The Complete Reference* (2^a ed.). McGraw-Hill Education.
- [4] Ghiani, G., Laporte, G., Musmanno, R. (2013). *Introduction to Logistics Systems Management* (2^a ed.). Wiley.
- [5] Hernandez, M. J. (2013). *Database Design for Mere Mortals* (3^a ed.). Addison-Wesley.