REPORTE DE PRÁCTICA NO. 2

Práctica. Álgebra relacional y SQL (1)

ALUMNO: Jose Angel Castro Paredes
Dr. Eduardo Cornejo-Velázquez

1. Introducción

En el ámbito de las bases de datos, el uso de procedimientos almacenados, funciones, estructuras de control y disparadores es fundamental para optimizar el rendimiento y la seguridad de los sistemas de gestión de datos. Estos mecanismos permiten automatizar procesos, garantizar la integridad de la información y reducir la redundancia en el código SQL. En este documento, se presentan ejemplos prácticos aplicados a una base de datos de flotilla, los cuales ilustran el uso de cada una de estas herramientas en escenarios reales.

2. Marco teórico

Procedimientos almacenados

Un procedimiento almacenado es un conjunto de instrucciones SQL que se almacena asociado a una base de datos. Se crea con la sentencia "CREATE PROCEDURE" y se invoca con la sentencia "CALL". Un procedimiento puede tener cero o más parámetros de entrada y salida. Su principal ventaja es que permite encapsular lógica de negocio en la base de datos, facilitando su reutilización y mantenimiento.

Funciones

Una función almacenada es similar a un procedimiento, pero con la diferencia de que siempre devuelve un valor. Se crea con la sentencia "CREATE FUNCTION" y se puede invocar dentro de una expresión SQL o mediante "SELECT". Las funciones pueden aceptar parámetros de entrada y deben especificar el tipo de dato que retornarán.

Estructuras de Control Condicionales y Repetitivas

Las estructuras de control permiten dirigir el flujo de ejecución del código. Las principales estructuras incluyen:
Condicionales:
-IF-THEN-ELSE: Permite ejecutar bloques de código basados en condiciones específicas.
-CASE: Ofrece una manera de seleccionar entre múltiples alternativas basadas en el valor de una expresión.

Repetitivas:
-WHILE: Ejecuta un bloque de código mientras una condición sea verdadera.
-REPEAT: Ejecuta un bloque de código hasta que una condición se vuelva verdadera.
-LOOP: Proporciona una estructura de bucle que se ejecuta indefinidamente hasta que se encuentre una instrucción LEAVE.

Disparadores (Triggers)

Es un conjunto de comandos que se ejecutan antes o después de que se realiza un cambio en la tabla (INSERT, UPDATE, DELETE). En general se utilizan para garantizar la coherencia de todos los datos de una tabla después de una operación en particular o para actualizar multiples tablas al mismo tiempo de acuerdo con un cambio realizado en una tabla en particular.

3. Herramientas empleadas

1. MySQL Workbench.
2. Latex Overleaf

4. Desarrollo

1. Procedimientos Almacenados (Stored Procedures)

```
CREATE PROCEDURE InsertarConductor(
    IN p_idConductor INT,
    IN p_nombre VARCHAR(80),
    IN p_numeroTelefonico VARCHAR(10),
    IN p_correo VARCHAR(80),
    IN p_numeroLicencia INT
)
BEGIN
    INSERT INTO conductor (idConductor, nombre, n meroTelefonico, correo, n meroLicencia)
    VALUES (p_idConductor, p_nombre, p_numeroTelefonico, p_correo, p_numeroLicencia);
END

CALL InsertarConductor(6, 'Pedro S nchez', '5556667777', 'pedro.sanchez@mail.com', 67890);
```

Listing 1: Ejemplo 1: Insertar un nuevo conductor

```
CREATE PROCEDURE ActualizarFechaVerificacion(
    IN p_idVehiculo INT,
    IN p_nuevaFecha DATE
)
BEGIN
    UPDATE documentacion
    SET fechaVerificacion = p_nuevaFecha
    WHERE idVehiculo = p_idVehiculo;
END
```

Listing 2: Ejemplo 2: Actualizar fecha de verificación

2. Funciones (Functions)

```
CREATE FUNCTION ObtenerTotalConductores() RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE total INT;
    SELECT COUNT(*) INTO total FROM conductor;
    RETURN total;
END
```

Listing 3: Ejemplo 1: obtener el número de conductores registrados

```
CREATE FUNCTION CalcularCostoTotalGasolina(p_folioCompraGas INT) RETURNS FLOAT
DETERMINISTIC
BEGIN
    DECLARE total FLOAT;
    SELECT SUM(precio) INTO total FROM compraGasolina WHERE folioCompraGas =
    p_folioCompraGas;
    RETURN total;
END
```

Listing 4: Ejemplo 2:Calcular costo total de gasolina por un folio

3. Estructuras de control.

```
CREATE PROCEDURE VerificarVehiculo(
    IN p_idVehiculo INT
)
BEGIN
    DECLARE fechaVerificacionActual DATE;
    SELECT fechaVerificacion INTO fechaVerificacionActual FROM documentacion WHERE
    idVehiculo = p_idVehiculo;

    IF fechaVerificacionActual < CURDATE() THEN
        SELECT 'El veh culo requiere verificaci n' AS mensaje;
    ELSE
        SELECT 'El veh culo est  al corriente con su verificaci n' AS mensaje;
    END IF;
END
```

Listing 5: Determinar si un vehículo debe verificar.

4. Disparadores (Triggers).

```
SCREATE TRIGGER AntesDeInsertarConductor
BEFORE INSERT ON conductor
FOR EACH ROW
BEGIN
    INSERT INTO registroConduccion (idRegistro, idVehiculo, idConductor, fecha)
    VALUES (NEW.idConductor, 1, NEW.idConductor, CURDATE());
END
```

Listing 6: Registrar un mensaje al insertar un nuevo conductor.

```
CREATE TRIGGER AntesDeEliminarConductor
BEFORE DELETE ON conductor
FOR EACH ROW
BEGIN
    DECLARE existe INT;
    SELECT COUNT(*) INTO existe FROM vehiculo_conductor WHERE idConductor = OLD.idConductor;
    IF existe > 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'No se puede eliminar el conductor porque tiene veh culos
    asignados';
    END IF;
END
```

Listing 7: Evitar que un conductor sea eliminado si tiene asignado un vehiculo.

5. Conclusiones

El uso de procedimientos almacenados, funciones, estructuras de control y disparadores en bases de datos mejora significativamente la gestión y manipulación de los datos. Los procedimientos y funciones permiten encapsular lógica de negocio y facilitar su reutilización. Las estructuras de control proporcionan flexibilidad para la toma de decisiones dentro de la base de datos, mientras que los disparadores aseguran la consistencia y validación automática de la información. La correcta implementación de estos elementos ayuda a construir bases de datos más eficientes, seguras y mantenibles.

Referencias Bibliográficas

References

[1] José Juan Sánchez. (s.f.). Unidad 12. Triggers, procedimientos y funciones en MySQL. Recuperado el 27 de febrero de 2025, de https://josejuansanchez.org/bd/unidad-12-teoria/

[2] Microsoft. (s.f.). Procedimientos almacenados (motor de base de datos) - SQL Server. Microsoft Learn. Recuperado el 27 de febrero de 2025, de https://learn.microsoft.com/es-es/sql/relational-databases/stored-procedures/stored-procedures-database-engine?view=sql-server-ver16

[3] Benítez, M. Á., Arias, Á. (2015). Curso de Introducción a la Administración de Bases de Datos. IT campus Academy.