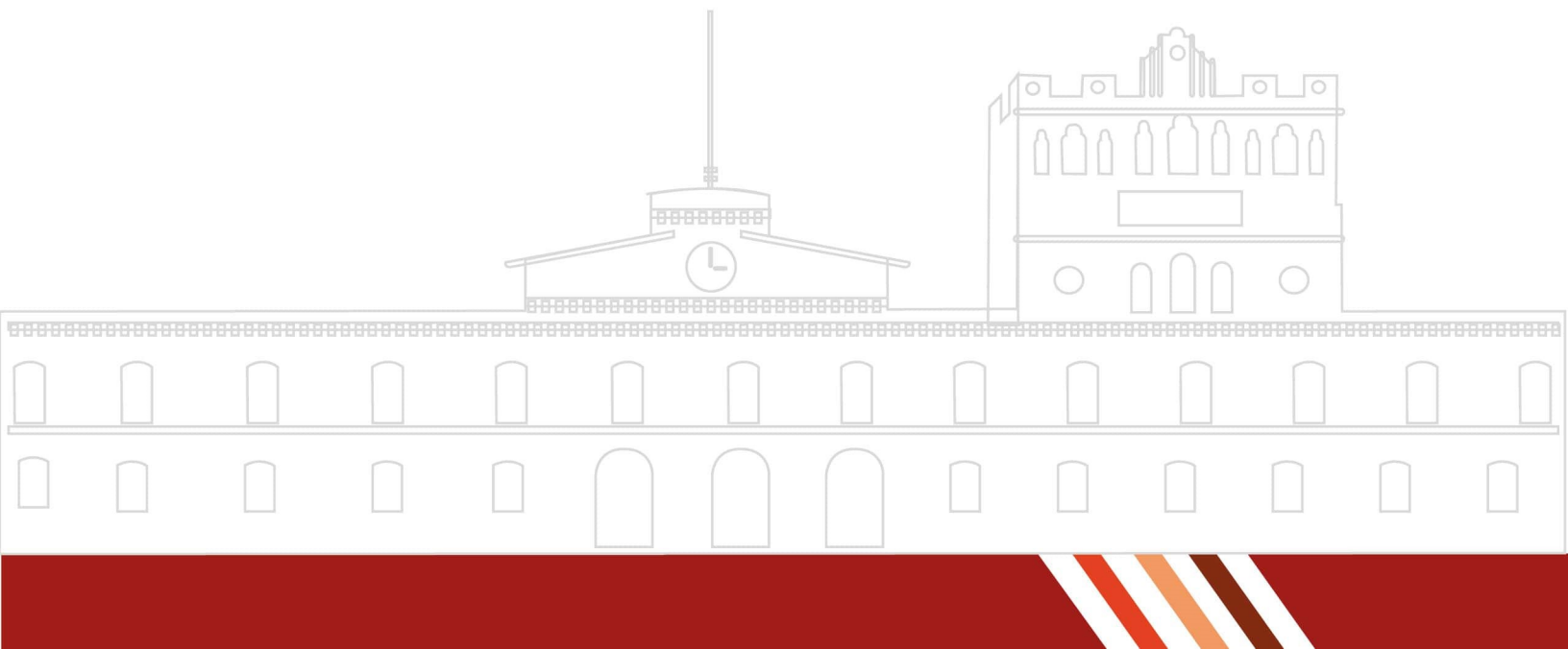


REPORTE DE PRÁCTICA NO. 1.5

GESTIÓN DE FLOTILLA DE AUTOS

ALUMNO: Israel Campos Vázquez
Dr. Eduardo Cornejo-Velázquez



1. Marco teórico

Procedimiento Almacenado

Los procedimientos almacenados son bloques de código SQL que se guardan en el servidor de la base de datos y pueden ser ejecutados repetidamente sin necesidad de reescribir la consulta. Se utilizan para automatizar tareas, mejorar el rendimiento y mantener la seguridad en la gestión de datos [2].

Funciones

Las funciones en SQL permiten realizar cálculos y retornar un valor específico. A diferencia de los procedimientos almacenados, siempre devuelven un resultado y no pueden modificar datos en las tablas [4].

Estructura de Control

Las estructuras de control permiten ejecutar diferentes bloques de código dependiendo de condiciones específicas o iterar sobre un conjunto de datos. Se utilizan en procedimientos almacenados, funciones y disparadores [1].

Disparadores

Los disparadores son objetos de base de datos que se ejecutan automáticamente en respuesta a eventos como inserciones, actualizaciones o eliminaciones. Se usan para mantener la integridad de los datos [3].

2. Desarrollo

Procedimiento Almacenado

```
1 DELIMITER //
2 CREATE PROCEDURE RegistrarMantenimiento(
3     IN p_placa VARCHAR(15),
4     IN p_fecha DATE,
5     IN p_costo FLOAT,
6     IN p_descripcion VARCHAR(100)
7 )
8 BEGIN
9     INSERT INTO Mantenimiento (fecha, costo, idMantenimiento, descripcion, placa)
10    VALUES (p_fecha, p_costo, NULL, p_descripcion, p_placa);
11 END //
12 DELIMITER ;
```

Listing 1: Registrar un mantenimiento de un vehículo.

```
1 DELIMITER //
2 CREATE PROCEDURE AsignarConductor(
3     IN p_curp VARCHAR(20),
4     IN p_idRuta INT
5 )
6 BEGIN
7     UPDATE Ruta SET curp = p_curp WHERE idRuta = p_idRuta;
8 END //
9 DELIMITER ;
```

Listing 2: Asignar un conductor a una ruta.

Funciones

```
1 DELIMITER //
2 CREATE FUNCTION CalcularCostoTotalViajes(p_curp VARCHAR(20)) RETURNS FLOAT
3 DETERMINISTIC
4 BEGIN
5     DECLARE total FLOAT;
6     SELECT SUM(costoViaje) INTO total FROM Ruta WHERE curp = p_curp;
7     RETURN IFNULL(total, 0);
8 END //
9 DELIMITER ;
```

Listing 3: Calcular el costo total de un viaje de un conductor

```
1 DELIMITER //
2 CREATE FUNCTION PromedioCostoMantenimiento(p_placa VARCHAR(15)) RETURNS FLOAT
3 DETERMINISTIC
4 BEGIN
5     DECLARE promedio FLOAT;
6     SELECT AVG(costo) INTO promedio FROM Mantenimiento WHERE placa = p_placa;
7     RETURN IFNULL(promedio, 0);
8 END //
9 DELIMITER ;
```

Listing 4: Calcular el promedio de costos de mantenimiento de un vehiculo

Estructura de Control

```
1 DELIMITER //
2 CREATE PROCEDURE ActualizarDisponibilidadConductor(IN p_curp VARCHAR(20))
3 BEGIN
```

```

4      DECLARE estado BOOLEAN;
5      SELECT disponibilidad INTO estado FROM Conductor WHERE curp = p_curp;
6
7      IF estado = 1 THEN
8          UPDATE Conductor SET disponibilidad = 0 WHERE curp = p_curp;
9      ELSE
10         UPDATE Conductor SET disponibilidad = 1 WHERE curp = p_curp;
11     END IF;
12 END //
13 DELIMITER ;

```

Listing 5: Actualizar la disponibilidad de un conductor

```

1  DELIMITER //
2  CREATE PROCEDURE EliminarMantenimientosAntiguos()
3  BEGIN
4      DECLARE done INT DEFAULT FALSE;
5      DECLARE id INT;
6      DECLARE cur CURSOR FOR SELECT idMantenimiento FROM Mantenimiento WHERE fecha < DATE_SUB(CURDATE(),
7      DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
8      OPEN cur;
9      read_loop: LOOP
10         FETCH cur INTO id;
11         IF done THEN
12             LEAVE read_loop;
13         END IF;
14         DELETE FROM Mantenimiento WHERE idMantenimiento = id;
15     END LOOP;
16     CLOSE cur;
17 END //
18 DELIMITER ;

```

Listing 6: Eliminar registros antiguos de mantenimiento

Disparadores

```

1  DELIMITER //
2  CREATE TRIGGER ActualizarDisponibilidadVehiculo
3  AFTER INSERT ON Ruta
4  FOR EACH ROW
5  BEGIN
6      UPDATE Vehiculo SET disponibilidad = 0 WHERE placa = NEW.placa;
7  END //
8  DELIMITER ;

```

Listing 7: Actualizar disponibilidad de un vehiculo

```

1  DELIMITER //
2  CREATE TRIGGER RegistrarHistorialMantenimiento
3  BEFORE DELETE ON Mantenimiento
4  FOR EACH ROW
5  BEGIN
6      INSERT INTO HistorialMantenimiento (fecha, costo, descripcion, placa)
7      VALUES (OLD.fecha, OLD.costo, OLD.descripcion, OLD.placa);
8  END //
9  DELIMITER ;

```

Listing 8: Historial de mantenimientos eliminados

Referencias Bibliográficas

References

- [1] Date, C. J. (2019). An Introduction to Database Systems. Pearson.
- [2] Elmasri, R., Navathe, S. B. (2017). Fundamentals of Database Systems. Pearson.
- [3] Garcia-Molina, H., Ullman, J. D., Widom, J. (2008). Database Systems: The Complete Book. Pearson.
- [4] Silberschatz, A., Korth, H. F., Sudarshan, S. (2019). Database System Concepts. McGraw-Hill.