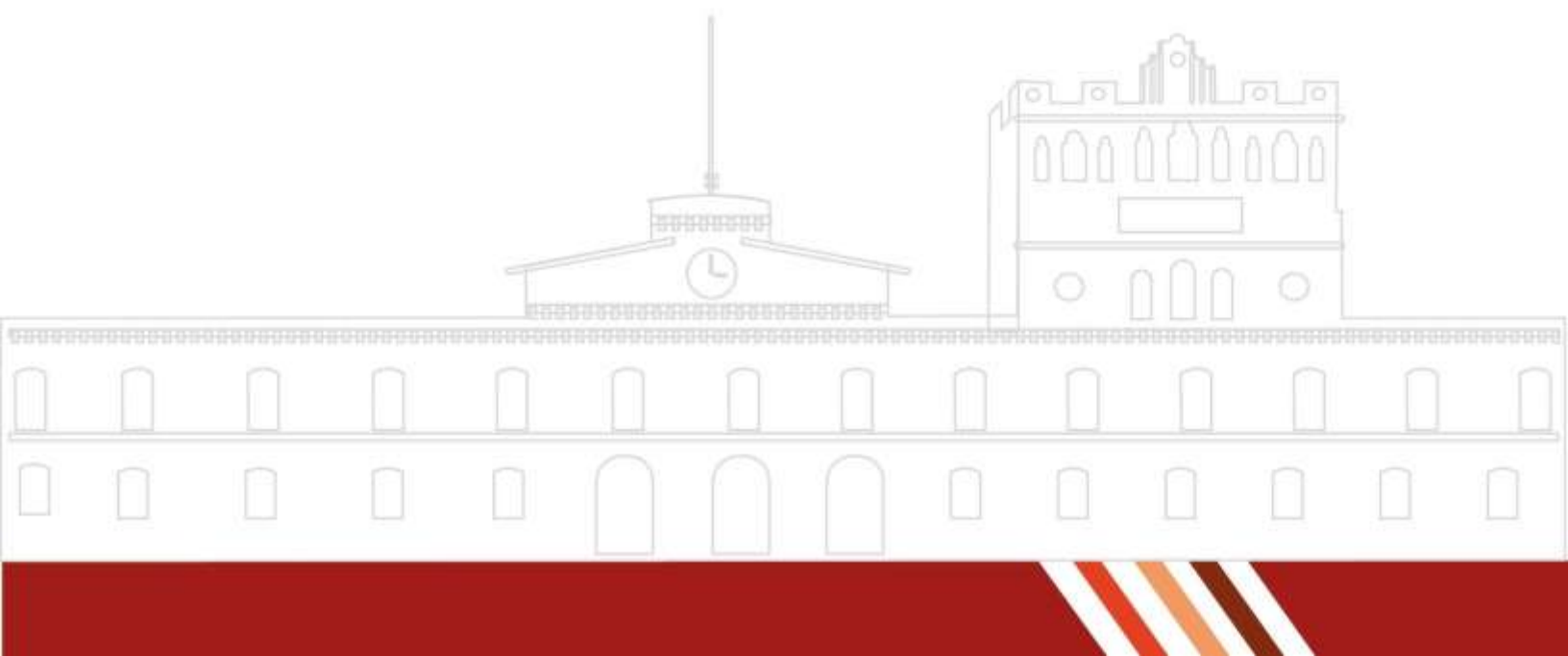




PRÁCTICA NO. 0

ALUMNO:Kevin Badillo Olmos
Dr. Eduardo Cornejo-Velázquez



El desarrollo de una base de datos para la **Gestión de Flotillas** requiere la implementación de componentes avanzados que permitan optimizar los procesos de administración, control y registro de la información relacionada con vehículos, conductores, viajes, mantenimientos y auditorías. Dentro de los elementos más relevantes en la programación de bases de datos destacan los **procedimientos almacenados**, las **funciones**, las **estructuras de control** y los **disparadores** (*triggers*), los cuales permiten mejorar el rendimiento, la seguridad y la automatización de las operaciones dentro del sistema.

Procedimientos almacenados (Stored Procedures)

Los procedimientos almacenados son bloques de código SQL que se guardan directamente en el servidor de base de datos y se ejecutan bajo demanda mediante una instrucción `CALL`. Su principal objetivo es **automatizar procesos repetitivos, centralizar la lógica de negocio y reducir el intercambio de información** entre el cliente y el servidor.

Estos procedimientos pueden recibir parámetros de entrada y salida, realizar operaciones de consulta o modificación, y devolver resultados de manera eficiente. En un sistema de gestión de flotillas, por ejemplo, pueden utilizarse para **asignar vehículos a conductores, registrar viajes o actualizar el kilometraje total de una unidad**.

De acuerdo con la documentación oficial de MySQL (Oracle, 2024), el uso de procedimientos almacenados mejora la consistencia y facilita el mantenimiento del sistema al permitir que las reglas del negocio se mantengan dentro del servidor.

Funciones (User-Defined Functions)

Las funciones son rutinas definidas por el usuario que **devuelven un valor específico** y pueden emplearse dentro de sentencias SQL como parte de expresiones, filtros o cálculos. A diferencia de los procedimientos, las funciones están diseñadas para **producir un resultado** y no para modificar datos directamente.

En el contexto de una base de datos de flotillas, las funciones pueden emplearse para **calcular la antigüedad de un vehículo, determinar el rendimiento de combustible, o obtener la distancia recorrida promedio** por unidad.

Según la guía de DigitalOcean (2023), las funciones favorecen la reutilización del código y garantizan resultados uniformes, ya que las operaciones complejas se definen una sola vez y se utilizan en diferentes partes del sistema.

Estructuras de control condicionales y repetitivas

Las estructuras de control permiten **dirigir el flujo de ejecución** dentro de un procedimiento o función. Entre las más comunes se encuentran las estructuras condicionales (`IF`, `CASE`) y las estructuras repetitivas (`WHILE`, `REPEAT`, `LOOP`).

Estas estructuras son esenciales cuando se requiere tomar decisiones o repetir una acción hasta cumplir una condición determinada. Por ejemplo, en una base de datos de flotillas pueden emplearse para **verificar si un vehículo se encuentra activo antes de asignarlo a un viaje**, o para **programar mantenimientos automáticos** de forma iterativa sobre todos los vehículos que cumplan cierta antigüedad.

La correcta implementación de estas estructuras aporta flexibilidad y potencia al manejo de datos, haciendo que el sistema responda de manera inteligente ante diferentes escenarios.

Disparadores (Triggers)

Los disparadores o *triggers* son rutinas que se ejecutan automáticamente **cuando ocurre un evento determinado** sobre una tabla, como una inserción (INSERT), una actualización (UPDATE) o una eliminación (DELETE).

Su propósito principal es **mantener la integridad de los datos y realizar acciones automáticas**, como auditorías o validaciones.

En el caso de la gestión de flotillas, un disparador puede registrar automáticamente en una tabla de auditoría cualquier cambio en el estado de un vehículo o impedir la eliminación de una unidad que tenga viajes asociados.

De acuerdo con el manual de MySQL (Oracle, 2024), los triggers garantizan la coherencia de los datos al actuar como mecanismos de control invisibles para el usuario, pero fundamentales para la integridad del sistema.

Importancia en la gestión de flotillas

El uso combinado de estos elementos (procedimientos, funciones, estructuras de control y triggers) permite que la base de datos sea **más autónoma, segura y coherente**, ya que muchas de las validaciones y procesos que normalmente dependerían de la aplicación se ejecutan directamente en el servidor.

De esta forma, el sistema de Gestión de Flotillas puede **optimizar sus procesos de registro, minimizar errores humanos y garantizar un control eficiente** sobre la información de cada vehículo, conductor y recorrido realizado.

Ejemplos de aplicación

Ejemplo 1: Procedimiento almacenado con estructuras de control y función auxiliar

En un sistema de Gestión de Flotillas, se puede crear una **función** que calcule la antigüedad de un vehículo con base en su año de fabricación. Dicha función se invoca dentro de un **procedimiento almacenado** que inicia el registro de un viaje, verificando previamente que el vehículo esté disponible.

El procedimiento realiza las siguientes acciones:

1. Comprueba si el vehículo existe y está activo.
2. Si no cumple con los requisitos, genera un mensaje de error.
3. Si está disponible, inserta un nuevo registro en la tabla de viajes (`trips`).
4. Finalmente, registra la operación en una tabla de auditoría (`audit_logs`).

Este tipo de procedimiento automatiza el proceso de asignación de viajes, evita errores por duplicidad y asegura que sólo las unidades activas puedan ser utilizadas. Además, la función complementaria permite obtener información adicional, como la antigüedad de la unidad, que puede servir para determinar su disponibilidad o priorizar mantenimientos.

Ejemplo 2: Disparadores y bucle para mantenimiento automatizado

En la misma base de datos, se pueden definir **disparadores (triggers)** que ejecuten acciones automáticas:

- Uno que se active **después de actualizar** los datos de un vehículo, registrando en la tabla de auditoría los cambios realizados en su estado o kilometraje.
- Otro que se ejecute **antes de eliminar** un vehículo, impidiendo la operación si existen registros de viajes asociados a esa unidad.

Adicionalmente, se puede implementar un **procedimiento con un bucle** que recorra todos los vehículos con más de ocho años de antigüedad y programe mantenimientos preventivos, marcando temporalmente su estado como “en mantenimiento”.

De esta manera, el sistema contribuye al control preventivo de la flota y reduce el riesgo de fallas operativas, todo de manera automatizada y sin intervención directa del usuario.

Referencias bibliográficas

- García, J. (2022). *Diseño y administración de bases de datos con MySQL*. Editorial Alfaomega, México.
- Hernández, P. (2023). *Programación SQL avanzada: procedimientos, funciones y disparadores*. Editorial Universidad Tecnológica de México.
- Rodríguez, M. (2023). *Gestión automatizada de flotas con bases de datos relacionales*. Revista Tecnología y Sistemas, 15(3), 45–59.
- MySQL. (2024). *Manual de referencia de MySQL: Procedimientos, funciones y disparadores*. Recuperado de <https://dev.mysql.com/doc/>
- Torres, L. (2023). *Estructuras de control y flujo en SQL*. Editorial Ra-Ma, Madrid.