Proyecto Final – Curso SQL

Primera Entrega

Autora: Iona Schnaidler

Tema: Gestión de Flota de Vehículos y Mantenimiento Preventivo

Repositorio completo del proyecto (README, script y diagrama):

https://github.com/ionaschnaidler/gestionflota_sql

Introducción

Este proyecto tiene como finalidad el diseño e implementación de una base de datos para gestionar una flota de vehículos pertenecientes a una empresa de transporte. La solución propuesta permite centralizar y organizar la información relativa a los vehículos, los viajes realizados, los mantenimientos efectuados, las fallas mecánicas reportadas y las cargas de combustible, optimizando la operatividad del servicio y reduciendo costos operativos.

Objetivo

Desarrollar una base de datos relacional que permita registrar los vehículos de una empresa de transporte, asignar choferes, registrar viajes, controlar mantenimientos y consumos de combustible, y hacer seguimiento de fallas y reparaciones. La solución está pensada para integrar áreas de operaciones, mantenimiento y administración.

Situación problemática

Las empresas de transporte que no cuentan con un sistema centralizado enfrentan desorganización operativa, falta de seguimiento de mantenimientos, altos costos por fallas inesperadas y escasa trazabilidad de vehículos. Una base de datos bien diseñada resuelve estas deficiencias.

Modelo de negocio

La base será utilizada por una empresa de transporte mediana que presta servicios logísticos y urbanos. Tiene distintos tipos de vehículos y personal operativo. Los actores principales son: Área de Operaciones (viajes y choferes), Taller Mecánico (fallas y mantenimientos) y Administración (control de costos e insumos).

Listado de Tablas

• Vehículos – Información técnica y estado de cada vehículo

• Choferes – Datos personales y disponibilidad de los conductores • Viajes – Registro de viajes realizados por cada unidad

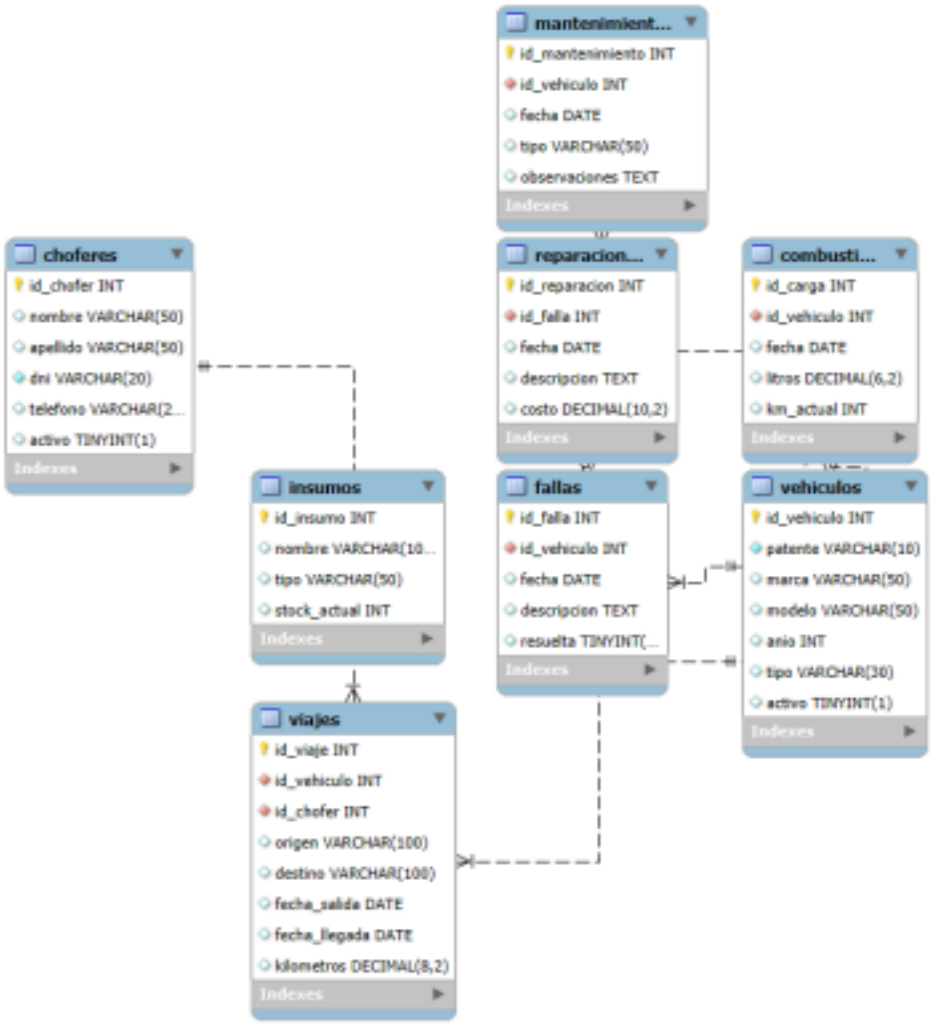
• Mantenimientos – Servicios preventivos y correctivos realizados • Fallas – Problemas detectados en los vehículos

• Reparaciones – Soluciones aplicadas a las fallas

• Insumos – Control de stock de piezas y productos

• Combustible – Cargas y kilometraje para control de consumo

Diagrama Entidad-Relación (E-R)



Listado de Vistas

1. vista_viajes_chofer
   Objetivo: Mostrar todos los viajes realizados con información del chofer y el vehículo asignado.
   Tablas involucradas: Viajes, Choferes, Vehículos

2. vista_mantenimientos_pendientes
   Objetivo: Listar vehículos que tienen fallas sin resolver y mostrar su último mantenimiento registrado.
   Tablas involucradas: Vehículos, Fallas, Mantenimientos

3. vista_consumo_combustible
   Objetivo: Analizar el rendimiento de combustible de cada vehículo en litros cada 100 km recorridos.
   Tablas involucradas: Combustible, Vehículos

4. vista_reparaciones_costosas
   Objetivo: Identificar reparaciones con un costo superior a $100.000.
   Tablas involucradas: Reparaciones, Fallas, Vehículos

5. vista_choferes_activos
   Objetivo: Listar choferes activos y su cantidad total de viajes realizados.
   Tablas involucradas: Choferes, Viajes

Listado de Funciones Personalizadas

1. f_km_totales_vehiculo(id)
   Objetivo: Calcular la cantidad total de kilómetros recorridos por un vehículo.
   Retorna: DECIMAL – suma total de los kilómetros registrados.
   Tablas involucradas: Viajes

2. f_promedio_reparacion(id)
   Objetivo: Calcular el costo promedio de las reparaciones realizadas sobre un vehículo determinado.
   Retorna: DECIMAL – valor promedio.
   Tablas involucradas: Reparaciones, Fallas

3. f_chofer_muy_activo(id, minimo)
   Objetivo: Verifica si un chofer superó una cantidad mínima de viajes.
   Retorna: BOOLEAN – TRUE si supera el umbral, FALSE si no.
   Tablas involucradas: Viajes

Listado de Stored Procedures

1. sp_registrar_viaje(...)
   Objetivo: Agrega un nuevo viaje con todos los datos requeridos (vehículo, chofer, origen, destino, fechas, km).
   Tablas involucradas: Viajes

2. sp_resolver_falla(p_id_falla)
   Objetivo: Cambia el estado de una falla a “resuelta”.
   Tablas involucradas: Fallas

3. sp_mantenimiento_rutina(...)
   Objetivo: Registra un mantenimiento preventivo automático con la fecha actual y observaciones.
   Tablas involucradas: Mantenimientos

Listado de Triggers

1. tr_fecha_carga_combustible
   Objetivo: Completar automáticamente la fecha de carga si no se especifica al insertar un nuevo registro de combustible.
   Tipo: BEFORE INSERT
   Tabla involucrada: Combustible
   Justificación: Garantiza que cada carga tenga fecha sin depender del ingreso manual.

2. tr_falla_resuelta_actualiza_mantenimiento
   Objetivo: Al marcar una falla como resuelta, se crea automáticamente un mantenimiento correctivo asociado.
   Tipo: AFTER UPDATE
   Tablas involucradas: Fallas, Mantenimientos
   Justificación: Automatiza el vínculo entre fallas resueltas y mantenimientos registrados, asegurando trazabilidad técnica.