



## **Trabajo Final: Entregable 3**

# **Creación de una Base de Datos para la Gestión de un Concesionario de Vehículos Usados**

**Nombre: Pablo Aguirre**

## Índice

1. **Presentación**
2. **Objetivo del Proyecto**
3. **Modelo de Negocio**
  - 3.1. **Compra y Recepción de Vehículos**
  - 3.2. **Gestión de Reparaciones**
  - 3.3. **Ventas**
  - 3.4. **Análisis de Demérito**
  - 3.5. **Problemática Empresarial**
4. **Diagrama de Relaciones**
  - 4.1. **Objetivos de la Base de Datos**
5. **Estructura de la Base de Datos**
  - 5.1. **Script de Creación de Tablas**
  - 5.2. **Alteración y Creación de Claves Foráneas**
6. **Carga de Datos (Ingesta)**
  - 6.1. **Uso de INSERT INTO**
  - 6.2. **Carga de Archivos CSV**
7. **Objetos de la Base de Datos**
  - 7.1. **Vistas**
    - **Definición, Objetivo, y Estructura**
    - **Ejemplo de Vista: Historial de Servicios y Reparaciones por Cliente**
  - 7.2. **Funciones**
    - **Definición y Explicación de Funciones**
    - **Ejemplo de Funciones: CalculateTotalRepairCost, GetVehicleAge, Cálculo de Márgenes**
  - 7.3. **Procedimientos Almacenados (SP)**
  - **Definición, Utilidad y Ejemplo de Procedimientos para Gestión de Reparaciones**
  - 7.4. **Triggers**
  - 7.5. **Roles (Punto Extra)**
8. **Procedimientos Transaccionales**
  - **Descripción y Ejemplos de Transacciones**
9. **Scripts y Consultas Adicionales**
  - 9.1. **Scripts de Inserción de Datos**
  - 9.2. **Ejemplo de Consultas:**
    - **Resumen de Reparaciones por Vehículo**
    - **Cálculo de Demérito y Valor Tasado**
    - **Historial de Ventas con Márgenes Calculados**

## **1. Presentación:**

Este trabajo se centrará en la operación de un concesionario de vehículos usados que opera en un mercado altamente competitivo y dinámico. Por lo que, la eficiencia en la gestión de stock, reparaciones y ventas son factores clave para el éxito del negocio. Actualmente, la compañía enfrenta la necesidad de implementar un sistema de base de datos que permita gestionar las operaciones diarias de manera efectiva, optimizando recursos y mejorando la rentabilidad. Este informe presenta el diseño y desarrollo de una base de datos relacional que responde a estos desafíos, garantizando un flujo constante de vehículos, transparencia en las reparaciones y satisfacción del cliente.

## **2. Objetivo del Proyecto:**

Structured Query Language (SQL) es un lenguaje estándar para interactuar con bases de datos relacionales, creando, manipulando y consultando datos. En este proyecto, se utilizará SQL para realizar un diseño y así desarrollar una base de datos desde cero que centralizará la información de stock, reparaciones y ventas, optimizando los procesos de negocio. El objetivo es mejorar la toma de decisiones mediante consultas SQL avanzadas, vistas, funciones y procedimientos almacenados que respondan a las necesidades analíticas y operativas del concesionario.

## **3. Modelo de Negocio**

El modelo de negocio de la compañía tiene los siguientes elementos:

**3.1 Compra y Recepción de Vehículos:** Los automóviles son adquiridos a través de proveedores o como parte de pago de clientes. Cada unidad pasa por una inspección detallada para evaluar su estado y las reparaciones necesarias.

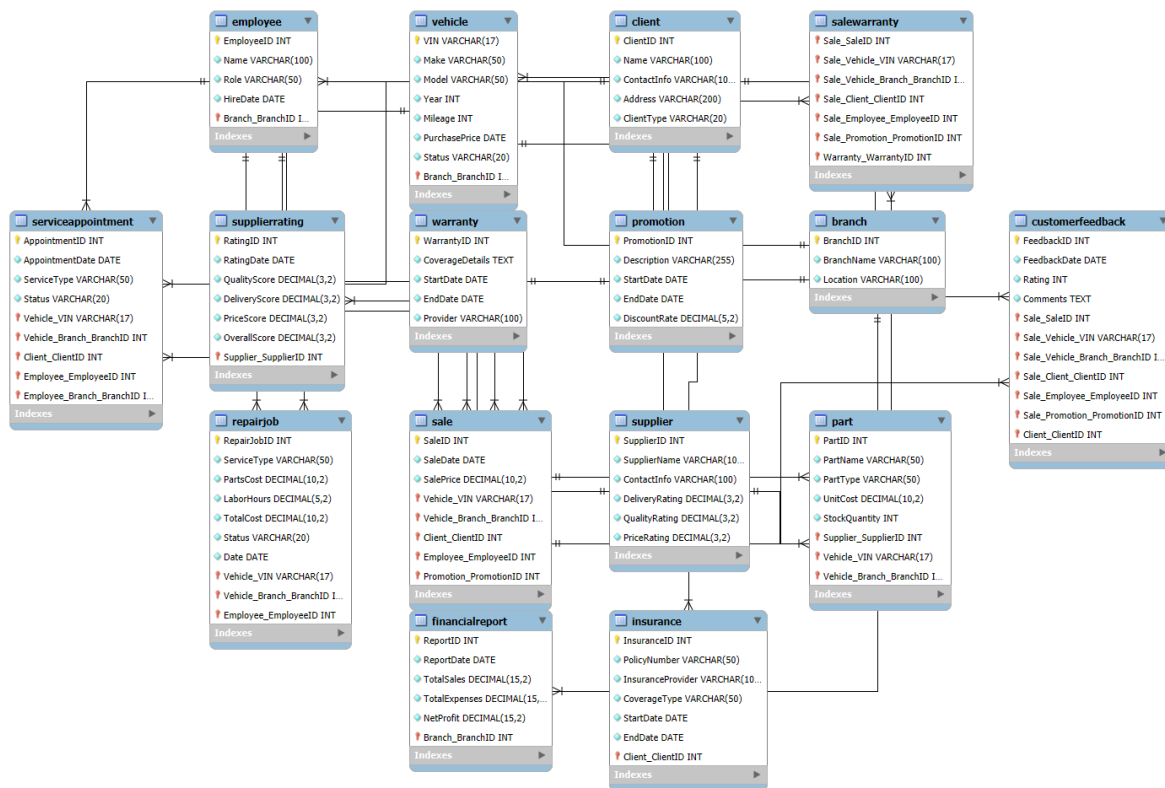
**3.2 Gestión de Reparaciones:** Los vehículos son enviados a talleres para reparaciones y ajustes previos a la venta.

**3.3 Ventas:** Los vehículos, después de ser reparados, son ofrecidos para la venta. La gestión de precios y márgenes de ganancia resulta esencial para aumentar rentabilidad.

**3.4 Análisis de Demérito:** Los automóviles se van devaluando debido a las reparaciones y al tiempo en inventario, lo que afecta su precio final y la decisión de compra de la empresa por sus costos elevados.

**3.5 Problemática Empresarial:** La problemática principal del negocio de concesionario es coordinar operaciones en múltiples sucursales, cada una con su propio inventario, personal y clientes, además de gestionar adquisiciones de vehículos usados, reparaciones y ventas. A medida que el concesionario crece, la falta de un sistema integrado que coordine estas operaciones puede generar ineficiencias, altos costos y pérdida de oportunidades en el mercado. Una posible solución simple, económica e innovadora es crear una base de datos que centralice y gestione las operaciones de los vehículos, reparaciones, proveedores, clientes y finanzas.

#### 4. Diagrama Relacional (Workbench):



Arriba, se muestra el diagrama relacional elaborado en MySQL Workbench, que muestra las relaciones entre las tablas principales del sistema:

- **Vehículos:** Relacionados con Gestión de Taller, Ventas y Gastos de Reparación a través de la patente.
- **Gestión de Taller:** Conectada a Gastos de Reparación a través del id\_taller.
- **Inspección:** Relaciona Vehículos y Gastos de Reparación para registrar detalles de tasación y demérito.

#### **4.1 Objetivos de la Base de Datos**

1. **Gestión de Inventario:** Registrar y realizar un seguimiento del inventario de vehículos desde la adquisición hasta la venta, incluyendo inspecciones y reparaciones.
2. **Gestión de Reparaciones:** Controlar todas las reparaciones de vehículos, gestionando tiempos y costos.
3. **Gestión de Proveedores:** Rastrear rendimiento de proveedores para piezas y materiales esenciales para las reparaciones como insumos, preparaciones, pre-compras, pre-ventas, post ventas y acuerdos comerciales.
4. **Gestión de Clientes:** Mantener perfiles de clientes y su historial de compras para ofrecer servicios personalizados y mejorar la fidelización.
5. **Informes Financieros:** Generar informes financieros para la toma de decisiones estratégicas.
6. **Escalabilidad:** Diseñar una base de datos flexible para futuras expansiones, integración con ventas en línea y análisis avanzados.

## 6. Estructura de la Base de Datos

### Diccionario Tablas:

#### Tabla Sucursal:

**Estructura:** IDSucursal, NombreSucursal, Ubicación

Almacena datos de sucursales.

Columna	Tipo Dato	Primary Key (PK)	Foreign Key (FK)	Descripción
IDSucursal	INT	PK	-	Identificador único para una sucursal.
NombreSucursal	VARCHAR(100)	-	-	Nombre de la sucursal.
Ubicación	VARCHAR(100)	-	-	Ubicación de la sucursal.

#### Tabla Vehículo:

**Estructura:** ID\_Vehículo (PK), Marca, Modelo, Año, concesionario, precio\_lista, gasto\_reparacion, patente, combustible, cilindrada.

Almacena datos principales de cada vehículo, como marca, modelo, concesionario, y detalles técnicos. Cada registro representa un vehículo único.

Columna	Tipo Dato	Primar y Key (PK)	Foreign Key (FK)	Descripción
VIN	VARCHAR(17)	PK	-	Número de identificación del vehículo.
Marca	VARCHAR(50)	-	-	Marca del vehículo.
Modelo	VARCHAR(50)	-	-	Modelo del vehículo.
Año	INT	-	-	Año de fabricación del vehículo.
Kilometros	INT	-	-	Kilometraje del vehículo.

Fecha Compra	DATE	-	-	Fecha de compra del vehículo.	
Estado	VARCHAR(20)	-	-	Estado del vehículo.	
Sucursal_IDSucursal	INT	-	FK: Sucursal(IDSucursal)	Sucursal donde se encuentra el vehículo.	

### **Tabla Empleador:**

**Estructura:** IDEmpleado, Nombre, Rol, FechaContratación, Sucursal\_IDSucursal

Información de empleados de las sucursales.

Columna	Tipo Dato	Primar y Key (PK)	Foreign Key (FK)	Descripción
<b>IDEmpleado</b>	INT	PK	-	Identificador único para un empleado
<b>Nombre</b>	VARCHAR(100)	-	-	Nombre del empleado.
<b>Rol</b>	VARCHAR(50)	-	-	Rol del empleado
<b>FechaContratación</b>	DATE	-	-	Fecha en que fue contratado el empleado.
<b>Sucursal_IDSucursal</b>	INT		FK: Sucursal(IDSucursal)	Sucursal donde trabaja el empleado.

**Tabla Reparaciones:**

**Estructura:** id\_taller (PK), patente, gasto\_reparación, sucursal, taller, zona, tipo\_reparación, día\_ingreso, fecha\_salida, fecha\_salida\_estimada, asesor.

Registra el historial de reparaciones de los vehículos en los talleres. Cada entrada describe las reparaciones necesarias, el lugar y el asesor encargado, también costos de reparación.

Columna	Tipo Dato	Primar y Key (PK)	Foreign Key (FK)	Descripción
IDTrabajoReparación	INT	PK	-	Identificador único para un trabajo de reparación.
TipoServicio	VARCHAR(50)	-	-	Tipo de servicio realizado.
Costo de piezas	DECIMAL(10, 2)	-	-	Costo de las piezas utilizadas en la reparación.
Horas de mano de obra	DECIMAL(5,2)	-	-	Horas de mano de obra para la reparación.
Costo total	DECIMAL(10, 2)	-	-	Costo total del trabajo de reparación.
Estado	VARCHAR(20)	-	-	Estado del trabajo de reparación.
Fecha	DATE	-	-	Fecha del trabajo de reparación.



VIN del vehículo	VARCHAR(17)	-	FK: Vehicle(VIN)	Vehículo en reparación.
Sucursal del vehículo_ID de sucursal	INT	-	FK: Vehicle(Branch_BranchID)	Sucursal del vehículo en reparación.
Empleado_ID de empleado	INT	-	FK: Employee(EmployeeID)	Empleado que realizó la reparación.

### **Tabla de proveedores:**

**Estructura:** ID del proveedor, Nombre del proveedor, Información de contacto, Calificación de entrega, Calificación de calidad, Calificación de precio

Información de los proveedores.

Columna	Tipo Dato	Primary Key (PK)	Foreign Key (FK)	Descripción
ID del proveedor	INT	PK	-	Identificador único de un proveedor.
Nombre del proveedor	VARCHAR(100)	-	-	Nombre del proveedor.
Información de contacto	VARCHAR(100)	-	-	Información de contacto del proveedor.
Calificación de entrega	DECIMAL(3,2)	-	-	Calificación de entrega del proveedor.
Calificación de calidad	DECIMAL(3,2)	-	-	Calificación de calidad del proveedor.
Calificación de precio	DECIMAL(3,2)	-	-	Calificación de precio del proveedor.

**Tabla Repuestos :**

**Estructura:** id\_vale (PK), número de albarán, gasto\_reparación, patente, sucursal, taller, zona.

Contiene registros de gastos en piezas para cada vehículo, necesarios para el cálculo del margen de ganancia.

Columna	Tipo Dato	Primar y Key (PK)	Foreign Key (FK)	Descripción
ID de pieza	INT	PK	-	Identificador único de una pieza.
Nombre de la pieza	VARCHAR(50)	-	-	Nombre de la pieza.
Tipo de pieza	VARCHAR(50)	-	-	Tipo de pieza.
Costo unitario	DECIMAL(10,2)	-	-	Coste unitario de la pieza.
Cantidad de existencias	INT	-	-	Cantidad de la pieza en stock.
Supplier_SupplierID	INT	-	FK: Supplier(SupplierID)	Proveedor de la pieza.
Vehicle_VIN	VARCHAR(17)	-	FK: Vehicle(VIN)	Vehículo asociado a la pieza.
Vehicle_Branch_BranchID	INT	-	FK: Vehicle(Branch_BranchID)	Sucursal del vehículo asociado a la pieza.

**Tabla clientes:**

**Estructura:** ID de cliente, Nombre, Información de contacto, Dirección, Tipo de cliente

Contiene almacenamiento histórico de los clientes.

Columna	Tipo Dato	Primary Key (PK)	Foreign Key (FK)	Descripción
ID de cliente	INT	PK	-	Identificador único de un cliente.
Nombre	VARCHAR(100)	-	-	Nombre del cliente.
Información de contacto	VARCHAR(100)	-	-	Información de contacto del cliente.
Dirección	VARCHAR(200)	-	-	Dirección del cliente.
<b><u>Tipo de cliente</u></b>	<b><u>VARCHAR(20)</u></b>	<b><u>:</u></b>	<b><u>:</u></b>	<b><u>Tipo de cliente (p. ej., particular, empresa)</u></b>

**Tabla de promociones:**

**Estructura:** ID de promoción, Descripción, Fecha de inicio, Fecha de finalización, Tasa de descuento

Entrega información de las promociones.

Columna	Tipo Dato	Primary Key (PK)	Foreign Key (FK)	Descripción
ID de promoción	INT	PK	-	Identificador único de una promoción.
Descripción	VARCHAR(255)	-	-	Descripción de la promoción.
Fecha de inicio	DATE	-	-	Fecha de inicio de la promoción.
Fecha de finalización	DATE	-	-	Fecha de finalización de la promoción.
Tasa de descuento	DECIMAL(5,2)	-	-	Tasa de descuento de la promoción.

### **Tabla Ventas:**

**Estructura:** id\_venta (PK), empresa, factura, fecha\_factura, patente, marca, modelo, rut\_cliente, nombre\_cliente, estado\_facturación, nombre\_vendedor, valor\_factura, valor\_compra, gastos\_reparación, margen.

Almacena la información de cada venta realizada, incluyendo cliente, valor de factura, valor de compra y márgenes de ganancia.

Columna	Tipo Dato	Primary Key (PK)	Foreign Key (FK)	Descripción
ID de venta	INT	PK	-	Identificador único de una venta.
Fecha de venta	DATE	-	-	Fecha de la venta.
Precio de venta	DECIMAL(10,2)	-	-	Precio de venta del vehículo.
VIN del vehículo	VARCHAR(17)	-	FK: Vehicle(VIN)	Vehículo vendido.
Sucursal del vehículo_ID de sucursal	INT	-	FK: Vehicle(Branch_BranchID)	Sucursal del vehículo vendido.
Cliente_ID de cliente	INT	-	FK: Client(ClientID)	Cliente que compró el vehículo.
Empleado_ID de empleado	INT	-	FK: Employee(EmployeeID)	Empleado que facilitó la venta.
Promoción_ID de promoción	INT	-	FK: Promotion(PromotionID)	Promoción aplicada a la venta.

**Mesa de citas de servicio:**

**Estructura:** AppointmentID, AppointmentDate, ServiceType, Status, Vehicle\_VIN, Vehicle\_Branch\_BranchID, Employee\_EmployeeID

Registra las inspecciones de vehículos, su valor tasado y el demérito asignado por reparaciones necesarias.

Columna	Tipo Dato	Primar y Key (PK)	Foreign Key (FK)	Descripción
AppointmentID	INT	PK	-	Identificador único de una cita de servicio.
AppointmentDate	DATE	-	-	Fecha de la cita de servicio.
ServiceType	VARCHAR(50)	-	-	Tipo de cita de servicio.
Status	VARCHAR(20)	-	-	Estado de la cita.
Vehicle_VIN	VARCHAR(17)	-	FK: Vehicle(VIN)	Vehículo asociado a la cita.
Vehicle_Branch_BranchID	INT	-	FK: Vehicle(Branch_BranchID)	Sucursal del vehículo asociado a la cita.
Employee_EmployeeID	INT	-	FK: Employee(EmployeeID)	Empleado que gestiona la cita.

**Tabla de garantía:** WarrantyID, WarrantyType, StartDate, EndDate, Vehicle\_VIN, Vehicle\_Branch\_BranchID.

Se evalúa si existe garantía de la marca.

Columna	Tipo Dato	Primar y Key (PK)	Foreign Key (FK)	Descripción
WarrantyID	INT	PK	-	Identificador único de una garantía.
WarrantyType	VARCHAR(50)	-	-	Tipo de garantía.
StartDate	DATE	-	-	Fecha de inicio de la garantía.
EndDate	DATE	-	-	Fecha de finalización de la garantía.
Vehicle_VIN	VARCHAR(17)	-	FK: Vehicle(VIN)	Vehículo cubierto por la garantía.
Vehicle_Branch_BranchID	INT	-	FK: Vehicle(Branch_BranchID)	Sucursal del vehículo cubierto por la garantía.

**Tabla de comentarios de clientes:**

**Estructura:** FeedbackID, FeedbackDate, Rating, Comments, Sale\_SaleID, Sale\_Vehicle\_VIN, Sale\_Vehicle\_Branch\_, BranchID, Sale\_Client\_ClientID, Sale\_Employee\_EmployeeID, Sale\_Promotion\_PromotionID, Client\_ClientID.

Análisis de data respecto a talleres, consesionarias y otras áreas del negocio.

Columna	Tipo Dato	Primary Key (PK)	Foreign Key (FK)	Descripción
FeedbackID	INT	PK	-	Identificador único de la opinión del cliente.
FeedbackDate	DATE	-	-	Fecha de la opinión.
Rating	INT	-	-	Valoración otorgada en la opinión.
Comments	TEXT	-	-	Comentarios proporcionados en la opinión.
Sale_SaleID	INT	-	FK: Sale(SaleID)	Venta asociada a la opinión.
Sale_Vehicle_VIN	VARCHAR (17)	-	FK: Sale(Vehicle_VIN)	Vehículo asociado a la opinión.
Sale_Vehicle_Branch_BranchID	INT	-	FK: Sale(Vehicle_Branch_BranchID)	Sucursal del vehículo asociado a la opinión.
Sale_Client_ClientID	INT	-	FK: Sale(Client_ClientID)	Cliente asociado a la opinión.

Sale_Employee_EmployeeID	INT	-	FK: Sale(Employee_EmployeeID)	Empleado asociado a la opinión.
Sale_Promotion_PromotionID	INT	-	FK: Sale(Promotion_PromotionID)	Promoción asociada a la opinión.
Client_ClientID	INT	-	FK: Client(ClientID)	Cliente que proporcionó la opinión.

### **Tabla de informe financiero:**

**Estructura:** ReportID, ReportDate, TotalSales, TotalExpenses, NetProfit, Branch BranchID

Almacena información financiera.

Columna	Tipo Dato	Primary Key (PK)	Foreign Key (FK)	Descripción
ReportID	INT	PK	-	Identificador único de un informe financiero.
ReportDate	DATE	-	-	Fecha del informe financiero.
TotalSales	DECIMAL(15,2)	-	-	Ventas totales informadas.
TotalExpenses	DECIMAL(15,2)	-	-	Gastos totales informados.
NetProfit	DECIMAL(15,2)	-	-	Utilidad neta informada.
Branch_BranchID	INT	-	FK: Branch(BranchID)	Sucursal relacionada con el informe financiero.



**Tabla de seguros:**

**Estructura:** InsuranceID, PolicyNumber, InsuranceProvider, CoverageType, StartDate, EndDate, Client\_ClientID

Contiene información de seguros de vehículos.

Columna	Tipo Dato	Primary Key (PK)	Foreign Key (FK)	Descripción
InsuranceID	INT	PK	-	Identificador único de una póliza de seguro.
PolicyNumber	VARCHAR(50)	-	-	Número de póliza de seguro.
InsuranceProvider	VARCHAR(100)	-	-	Proveedor de la póliza de seguro.
CoverageType	VARCHAR(50)	-	-	Tipo de cobertura del seguro.
StartDate	DATE	-	-	Fecha de inicio de la póliza de seguro.
EndDate	DATE	-	-	Fecha de finalización de la póliza de seguro.
Client_ClientID	INT	-	FK: Client(ClientID)	Cliente asociado a la póliza de seguro.

**Tabla de calificación de proveedores:**

**Estructura:** RatingID, RatingDate, QualityScore, DeliveryScore, PriceScore, OverallScore, Supplier\_SupplierID

Columna	Tipo Dato	Primary Key (PK)	Foreign Key (FK)	Descripción
RatingID	INT	PK	-	Identificador único para la calificación de un proveedor.
RatingDate	DATE	-	-	Fecha de la calificación.

QualityScore	DECIMAL(3,2)	-	-	Puntuación de calidad otorgada al proveedor.
DeliveryScore	DECIMAL(3,2)	-	-	Puntuación de entrega otorgada al proveedor.
PriceScore	DECIMAL(3,2)	-	-	Puntuación de precio otorgada al proveedor.
OverallScore	DECIMAL(3,2)	-	-	Puntuación general otorgada al proveedor.
Supplier_SupplierID	INT	-	FK: Supplier(SupplierID)	Proveedor que se está calificando.

**Tabla de garantía de venta:**

**Estructura:** Sale\_SaleID, Sale\_Vehicle\_VIN, Sale\_Vehicle\_Branch\_BranchID, Sale\_Client\_ClientID, Sale\_Employee\_EmployeeID, Sale\_Promotion\_PromotionID, Warranty\_WarrantyID.

Almacena datos de garantía de la marca.

Columna	Tipo Dato	Primary Key (PK)	Foreign Key (FK)	Descripción
Sale_SaleID	INT	PK	FK: Sale(SaleID)	Venta asociada a la garantía.
Sale_Vehicle_VIN	VARCHAR(17)	PK	FK: Sale(Vehicle_VIN)	Vehículo asociado a la garantía.

Sale_Vehicle_Branch_BranchID	INT	PK	FK: Sale(Vehicle_Branch_BranchID)	Sucursal del vehículo asociado a la garantía.
Sale_Client_ClientID	INT	PK	FK: Sale(Client_ClientID)	Cliente asociado a la garantía.
Sale_Employee_EmployeeID	INT	PK	FK: Sale(Employee_EmployeeID)	Empleado asociado a la garantía.
Sale_Promotion_PromotionID	INT	PK	FK: Sale(Promotion_PromotionID)	Promoción asociada a la garantía.
Warranty_WarrantyID	INT	PK	FK: Warranty(WarrantyID)	Garantía asociada a la venta.

### **Vistas, Funciones y Procedimientos Almacenados:**

#### **Vistas:**

Las vistas son tablas virtuales construidas a partir de consultas SQL, permitiendo acceder a datos de manera simplificada y segura sin alterar las tablas originales. En este proyecto, una de las vistas fundamentales está diseñada para consolidar el historial de servicios y reparaciones de cada cliente, asociándolos con los vehículos que poseen. Esta vista reúne los datos clave de los clientes, vehículos, y cada evento de reparación o servicio.

#### **Objetivo de la Vista:**

El objetivo principal de esta vista es facilitar el acceso al historial completo de reparaciones de cada cliente, evitando uniones repetitivas entre las tablas. Así se podrán generar reportes y consultar el área de servicio al cliente, ya que permite responder de forma ágil a preguntas sobre el historial de servicios en función de los clientes y vehículos.

- **Cliente (Client):** Incluye ClientID, que es el identificador único de cada cliente, y Name, el nombre del cliente. Estos datos nos permiten enlazar cada cliente con los servicios de sus vehículos.

- **Venta (Sale):** Esta tabla conecta a los clientes con los vehículos adquiridos mediante los campos Client\_ClientID (que referencia a ClientID en la tabla Client) y Vehicle\_VIN (que enlaza al VIN en Vehicle).
- **Vehículo (Vehicle):** Incluye VIN, Make y Model. El VIN (número de identificación del vehículo) es único y enlaza los datos de ventas y reparaciones; Make y Model ofrecen detalles específicos del vehículo.
- **Trabajo de Reparación (RepairJob):** Incluye RepairJobID, ServiceType, Date, y TotalCost. Estos campos detallan el tipo de servicio, la fecha de realización, y el costo, permitiendo crear un historial cronológico y financiero de las reparaciones de cada vehículo.

La consulta para esta vista selecciona los datos de la tabla Client (específicamente ClientID y Name), enlaza a Sale para obtener los vehículos de cada cliente y se une con RepairJob para capturar el historial de servicios. Se organiza por ClientID y RepairDate para presentar los registros cronológicamente.

#### **Aplicaciones de la Vista Esta vista es especialmente útil para:**

1. **Atención al cliente:** Al contar con un historial consolidado, el equipo de servicio puede responder consultas sobre reparaciones previas de forma más ágil.
2. **Reportes de servicios:** Facilita la generación de informes sobre el mantenimiento realizado y el costo acumulado por cliente.
3. **Análisis financiero:** Apoya el análisis de costos, permitiendo evaluar el gasto total en reparaciones para clientes específicos o vehículos en inventario.

Esta vista permite reducir la complejidad de consultas repetitivas y refuerza la seguridad al limitar el acceso directo a las tablas base.

En este proyecto, la vista creada resume las reparaciones y deméritos de vehículos, combinando datos de las tablas Gastos de Reparación, Inspección, y Gestión de Taller. Así, es posible realizar un análisis de gestión y control operativo para las concesionarias.

#### **Funciones:**

Las funciones en SQL son bloques de código que realizan cálculos y devuelven un valor. En este proyecto, se crearon dos funciones clave:

1. **CalculateTotalRepairCost:** Esta función calcula el costo total de las reparaciones de un vehículo determinado, tomando el VIN como parámetro. La función suma todos los costos de reparación asociados al VIN en la tabla RepairJob, y devuelve 0 si el vehículo no tiene reparaciones, usando COALESCE para manejar valores nulos. Esta función es útil en reportes, dashboards, y en sistemas de flotas que requieren el historial financiero de los vehículos.
2. **GetVehicleAge:** Esta función calcula la antigüedad de un vehículo en base a su año de fabricación. Toma el VIN como parámetro y retorna la diferencia entre el año actual y el año de fabricación, ayudando a evaluar la depreciación o a planificar mantenimientos específicos. Es una herramienta fundamental para concesionarios o seguros que consideran la edad del vehículo.

**Ejemplo de Función en el Proyecto** Otra función en este proyecto calcula el margen de cada vehículo vendido basándose en su valor\_compra y gasto\_reparación, datos provenientes de las tablas Ventas y Vehículos. Esto es útil para evaluar el retorno económico y planificar estrategias de ventas.

### **Procedimientos Almacenados (SP):**

Los procedimientos almacenados (SP) son scripts SQL que se almacenan en la base de datos y pueden ejecutarse para realizar operaciones complejas. Un SP en nuestro proyecto, por ejemplo, registra la actualización de los estados de reparación, ajustando los datos en Gestión de Taller y calculando los costos acumulados en Gastos Reparación. Estos procedimientos no solo automatizan tareas, sino que también mejoran la eficiencia en operaciones repetitivas.

### **Scripts y Alteraciones de Claves Foráneas**

1. **Estructura del Script Principal:** Este script crea las tablas y define sus claves primarias (PK) y foráneas (FK) para establecer relaciones. Las PKs garantizan la unicidad de registros, mientras que las FKs permiten la vinculación entre tablas.
2. **Ejemplo de Alteración de Foreign Key:**

sql

Copy code

```
ALTER TABLE Gestion_de_Taller
```

```
ADD CONSTRAINT fk_patente FOREIGN KEY (patente)
```

```
REFERENCES Vehiculos(patente);
```

Este comando asegura que cada entrada en Gestión de Taller esté asociada a un registro existente en Vehículos.

3. **Scripts de Inserción y Consultas:**

Resumen de reparaciones por vehículo: Permite conocer el historial detallado de reparaciones.

Demérito total y valor tasado: Da una visión de los costos de deterioro de los vehículos.

Historial de ventas con márgenes calculados: Proporciona un análisis económico para las ventas realizadas.