

REPORTE DE PRÁCTICA NO. 2.1

Fragmentos

ALUMNO: Jesus Eduardo Cornejo-Clavel
Dr. Eduardo Cornejo-Velázquez



1. Introducción

El propósito de esta práctica es diseñar e implementar fragmentos horizontales y verticales de una base de datos para gestionar una flotilla de vehículos. Los fragmentos permiten dividir y organizar los datos de manera eficiente, mejorando el rendimiento y la escalabilidad del sistema. Los fragmentos horizontales se enfocan en dividir las tuplas de una tabla, mientras que los fragmentos verticales se centran en dividir las columnas. En este informe, se presentarán 5 fragmentos horizontales y 5 fragmentos verticales implementados como vistas en una base de datos MySQL.

2. Marco teórico

Para construir el marco teórico se consultó la siguiente bibliografía:

Análisis de requerimientos

El análisis de requerimientos es fundamental para identificar las necesidades del sistema. Según Sommerville [1], los requisitos deben ser claros, completos y específicos para garantizar un diseño robusto.

Modelo Entidad - Relación

El modelo entidad-relación (MER) es una herramienta para representar conceptualmente la base de datos. Según Elmasri y Navathe [2], el MER permite visualizar las entidades y relaciones principales del sistema.

Modelo relacional

El modelo relacional es la base para diseñar tablas en una base de datos. Según Date [3], cada tabla representa una relación y debe seguir las reglas de normalización para evitar redundancias.

SQL

SQL es el lenguaje standar para la manipulación de bases de datosacionales. Según C.J. Date [3], las sentencias SQL son esenciales para interactuar con las bases de datos.

3. Herramientas empleadas

Las herramientas utilizadas para desarrollar la práctica fueron:

1. **MySQL**: Para crear vistas y ejecutar consultas SQL.
2. **Draw.io**: Para el diseño de modelos ER y Relacional

4. Desarrollo

Análisis de requisitos

Los requisitos principales para gestionar la flotilla de vehículos son:

- Gestionar información de conductores, vehículos y flotillas.
- Registrar transacciones de combustible y mantenimientos.
- Crear vistas para consultar datos específicos.

Modelo Entidad - Relación

En la Figura 1 se presenta el modelo entidad-relación propuesto.

Modelo relacional

En la Figura 2 se presenta el modelo relacional.

Sentencias SQL

En el Listado 1 se presentan las sentencias SQL para crear las vistas.

Listing 1: Vistas para fragmentos

```
CREATE VIEW h3 AS
SELECT * FROM mantenimiento
WHERE Mantenimiento.tipoServicio = 'Alineacion';

CREATE VIEW h4 AS
SELECT * FROM Conductor
WHERE Conductor.estado = 'Inactivo';

CREATE VIEW h5 AS
SELECT * FROM Flotilla
WHERE Flotilla.fechaCreacion
BETWEEN '2020-01-01' AND CURRENTDATE();

CREATE VIEW v1 AS
SELECT Vehiculo.marca, Vehiculo.modelo, Vehiculo.anio, Vehiculo.tipo,
       Mantenimiento.fechaServicio, Mantenimiento.costo, Mantenimiento.estado
FROM Vehiculo
JOIN Mantenimiento ON Mantenimiento.vehiculoId = Vehiculo.vehiculoId;

CREATE VIEW v2 AS
SELECT Conductor.nombre, Conductor.numeroLicencia,
       TransaccionCombustible.cantidad, TransaccionCombustible.monto,
       Vehiculo.modelo, Vehiculo.tipo
FROM Conductor
JOIN TransaccionCombustible ON Conductor.conductorId = TransaccionCombustible.conductorId
JOIN Vehiculo ON Vehiculo.vehiculoId = TransaccionCombustible.vehiculoId
ORDER BY monto;

CREATE VIEW v3 AS
SELECT Ruta.rutaId, Vehiculo.modelo, Vehiculo.vehiculoId,
       Flotilla.flotillaId, Conductor.nombre
```

```
FROM Conductor
JOIN Ruta ON Ruta.conductorId = Conductor.conductorId
JOIN Vehiculo ON Vehiculo.vehiculoId = Ruta.vehiculoId
JOIN Flotilla ON Vehiculo.flotillaId = Flotilla.flotillaId;

CREATE VIEW v4 AS
SELECT Vehiculo.vehiculoId , Vehiculo.marca ,
       Mantenimiento.fechaServicio , Mantenimiento.descripcion ,
       Documento.rutaArchivo
FROM Vehiculo
JOIN Mantenimiento ON Mantenimiento.vehiculoId = Vehiculo.vehiculoId
JOIN Documento ON Documento.vehiculoId = Vehiculo.vehiculoId;

CREATE VIEW v5 AS
SELECT COUNT( Vehiculo.vehiculoId ) AS cantdadMantenimiento ,
       Flotilla.nombreEmpresa
FROM Vehiculo
JOIN Mantenimiento ON Mantenimiento.vehiculoId = Vehiculo.vehiculoId
JOIN Flotilla ON Vehiculo.flotillaId = Flotilla.flotillaId
WHERE Mantenimiento.estado = 'Pendiente'
GROUP BY Flotilla.nombreEmpresa;
```

5. Conclusiones

En esta práctica se logró diseñar e implementar fragmentos horizontales y verticales en una base de datos MySQL. Los fragmentos horizontales permitieron filtrar datos específicos, mientras que los fragmentos verticales facilitaron la combinación de información de tablas relacionadas. Este ejercicio reforzó los conceptos de modelado de bases de datos y el uso de vistas para optimizar consultas.

Referencias Bibliográficas

References

- [1] Sommerville, I. (2011). *Software Engineering*. Pearson.
- [2] Elmasri, R.; Navathe, S. (2016). *Fundamentos de Sistemas de Bases de Datos*. Pearson Educación.
- [3] Date, C. J. (2011). *SQL y las bases de datos*. McGraw-Hill.

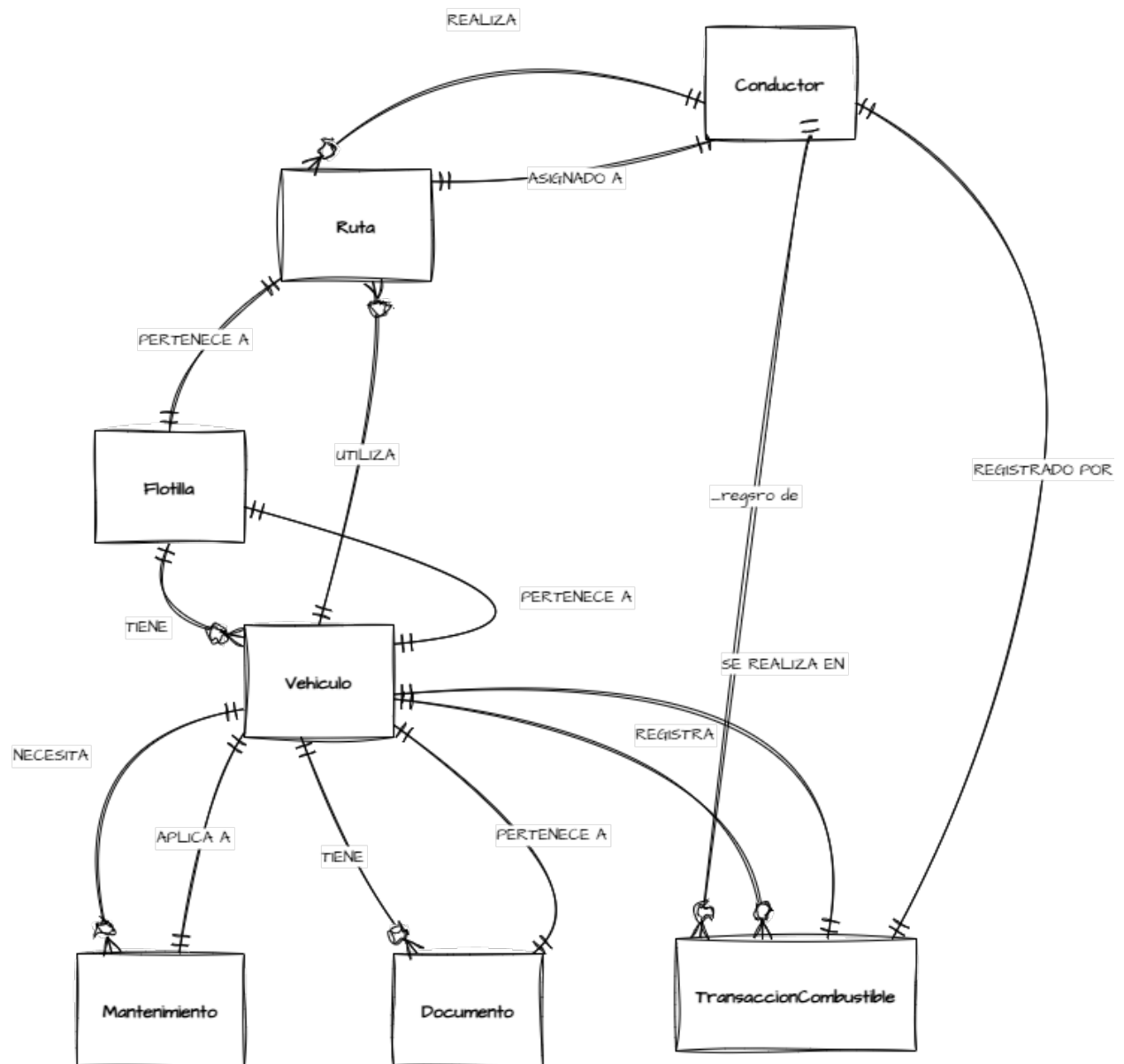


Figure 1: Modelo Entidad - Relación propuesto.

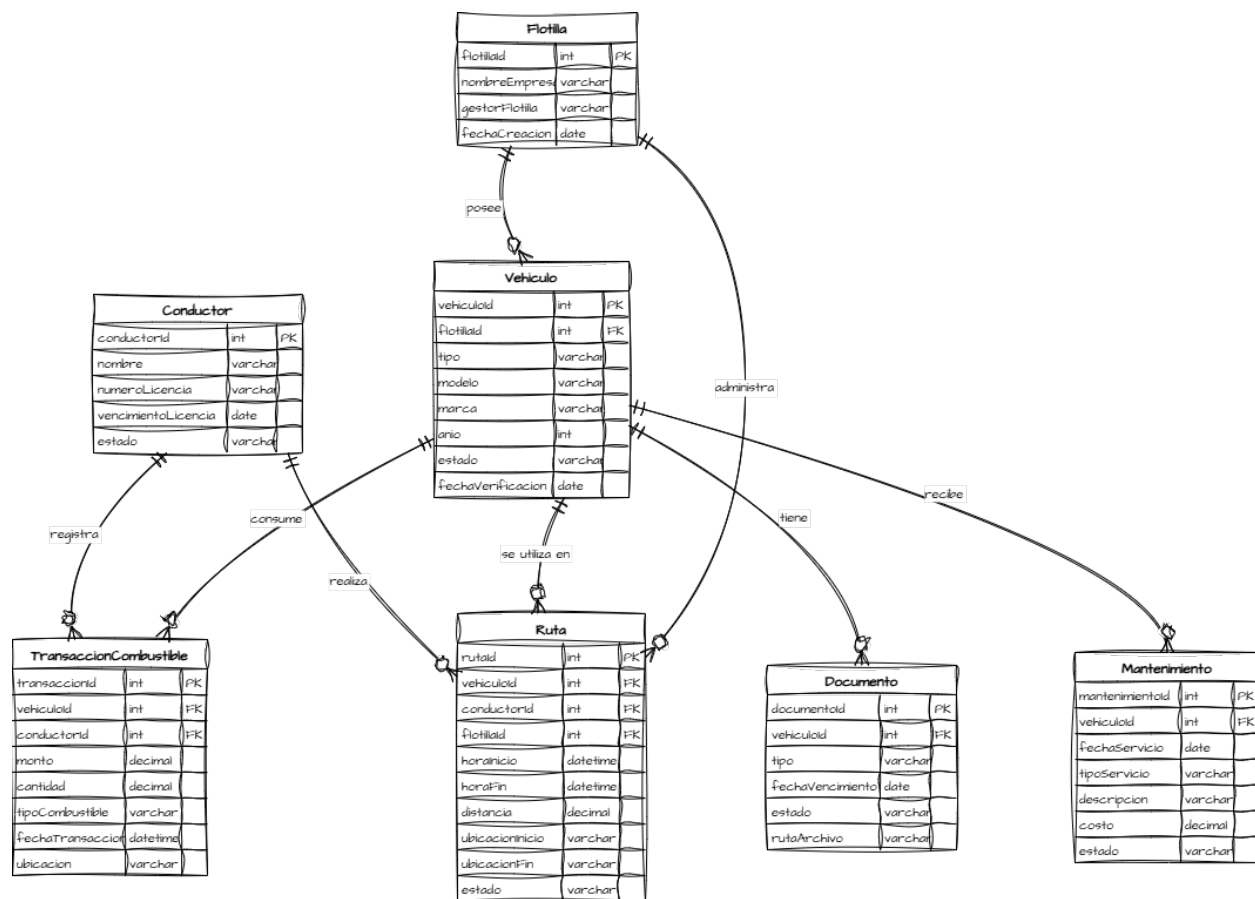


Figure 2: Modelo Relacional propuesto.