

REPORTE DE PRÁCTICA NO. 2.4

Tres nodos BDD Flotillas

ALUMNO:

Ariana García Melo



1. Introducción

En la actualidad, la gestión eficiente de flotillas de vehículos es un componente clave en la operación de empresas de transporte, logística y servicios. Una base de datos bien estructurada permite no solo el almacenamiento seguro de la información, sino también la optimización en la toma de decisiones estratégicas, el mantenimiento de los vehículos y el control del rendimiento operativo. Para lograr esto, es fundamental diseñar mecanismos de organización de los datos que faciliten su acceso, consulta y actualización de manera eficiente.

En el contexto de la gestión de bases de datos distribuidas, es fundamental establecer estrategias que permitan mejorar el rendimiento, la disponibilidad y la escalabilidad del sistema. Uno de los enfoques más utilizados en este ámbito es la fragmentación de bases de datos, la cual consiste en dividir una base de datos en fragmentos más pequeños que pueden ser almacenados y administrados de manera independiente en diferentes nodos. La fragmentación puede ser de varios tipos, entre los que destacan la fragmentación horizontal y la fragmentación vertical. En este trabajo, nos enfocaremos en la fragmentación vertical, la cual se basa en dividir una tabla en subconjuntos de columnas, manteniendo la integridad referencial a través de claves primarias o foráneas.

La fragmentación vertical permite distribuir la carga de trabajo de manera eficiente y optimizar el acceso a los datos, ya que cada nodo almacena únicamente la información relevante para una determinada operación o conjunto de usuarios. Esto reduce la redundancia de datos y mejora el desempeño en consultas específicas, ya que los sistemas pueden acceder a fragmentos más pequeños sin necesidad de recorrer grandes volúmenes de información innecesaria.

2. Marco teórico

Fragmentación Vertical

La fragmentación vertical es una estrategia de distribución de datos que divide una tabla en fragmentos basados en subconjuntos de columnas, manteniendo la clave primaria en cada fragmento para preservar la integridad referencial. Esto permite distribuir los datos de manera eficiente entre diferentes nodos, optimizando el acceso y reduciendo la redundancia innecesaria.

Procesos ETL

Los procesos ETL (Extracción, Transformación y Carga) son fundamentales para la administración y consolidación de datos en sistemas distribuidos. Estos procesos permiten obtener datos de diversas fuentes, transformarlos según los requisitos del sistema y cargarlos en una base de datos centralizada.

Etapas principales:

1. **Extracción:** Obtención de datos desde fuentes externas.
2. **Transformación:** Limpieza, agregación y formateo de datos.
3. **Carga:** Inserción de los datos en la base de datos destino.

SELECT INTO FILE

El comando **SELECT INTO FILE** permite exportar datos de una tabla a un archivo externo en formatos como CSV o TXT. Esta operación es utilizada para generar respaldos de información o para transferir datos entre sistemas distribuidos.

```
1      SELECT ... INTO OUTFILE 'archivo'
2      FIELDS TERMINATED BY 'character'
3      ENCLOSED BY 'character'
4      LINES TERMINATED BY 'character';
```

Listing 1: Sintaxis en MySQL

LOAD DATA INFILE

El comando **LOAD DATA INFILE** se emplea para importar datos desde un archivo externo a una tabla dentro de MySQL. Es una herramienta eficiente para la carga masiva de datos y se usa ampliamente en entornos distribuidos para actualizar información en distintos nodos.

```
1      LOAD DATA INFILE 'archivo'
2      INTO TABLE tabla
3      FIELDS TERMINATED BY 'character'
4      ENCLOSED BY 'character'
5      LINES TERMINATED BY 'character';
```

Listing 2: Sintaxis en MySQL

SELECT con Tablas de Dos Bases de Datos

En sistemas distribuidos, es común la necesidad de realizar consultas que involucren tablas almacenadas en distintas bases de datos. En MySQL, esto se logra especificando el nombre de la base de datos antes del nombre de la tabla en las consultas SQL.

```
1      SELECT columna1, columna2
2      FROM base_datos1.tabla1
3      JOIN base_datos2.tabla2 ON condicion;
```

Listing 3: Sintaxis en MySQL

3. Herramientas empleadas

Para la implementación de las consultas se utilizaron:

- MySQL como gestor de bases de datos.
- MySQL Workbench para la ejecución de consultas y visualización de resultados.
- LaTeX para la elaboración del reporte.
- Respaldo de la base de datos GCS: <https://github.com/idkAriana/Bases-de-Datos-Disribuidas/blob/7799fd707b31544ce7b9861a4bf6f8583b83f851/gestionFlotilla.sql>
- Respaldo de la base de datos LCS1-Principal: <https://github.com/idkAriana/Bases-de-Datos-Disribuidas/blob/7799fd707b31544ce7b9861a4bf6f8583b83f851/LCS1.sql>
- Respaldo de la base de datos LCS2-Mantenimiento: <https://github.com/idkAriana/Bases-de-Datos-Disribuidas/blob/7799fd707b31544ce7b9861a4bf6f8583b83f851/LCS2.sql>
- Respaldo de la base de datos LCS3-Rutas: <https://github.com/idkAriana/Bases-de-Datos-Disribuidas/blob/7799fd707b31544ce7b9861a4bf6f8583b83f851/LCS3.sql>

4. Desarrollo

En este ejercicio de distribución de la base de datos Gestión de Flotillas, se considera la base de datos como el GCS (Sistema de Control Global), actuando como la parte central dentro de una arquitectura de bases de datos distribuidas. En este contexto, se deben construir tres nodos físicos, cada uno funcionando como LCS (Sistema de Control Local), un nodo independiente que gestiona un subconjunto específico de los datos dentro de la base de datos distribuida.

Esquema Conceptual Local

4.1.1 Nodo LCS1-Principal

Gestiona la información básica de la flotilla, como los vehículos y documentos.

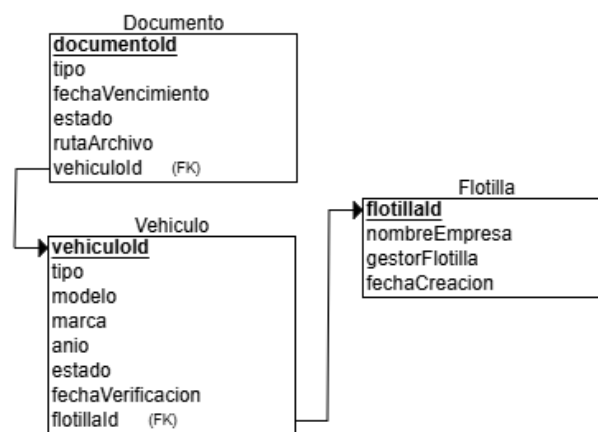


Figure 1: Modelo Relacional de LCS1-Principal

4.1.2 LCS2-Mantenimiento

Se ocupa de los registros de mantenimiento de los vehículos.

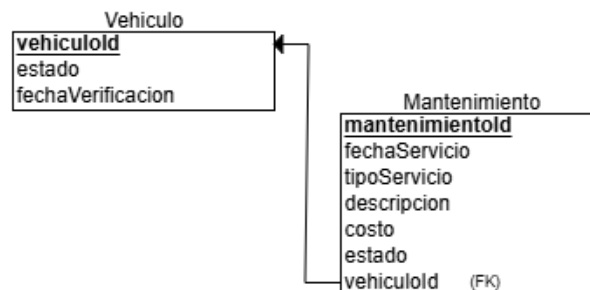


Figure 2: Modelo Relacional de LCS2-Mantenimiento

4.1.3 LCS3-Rutas

Gestiona las rutas, los conductores y las transacciones de combustible.

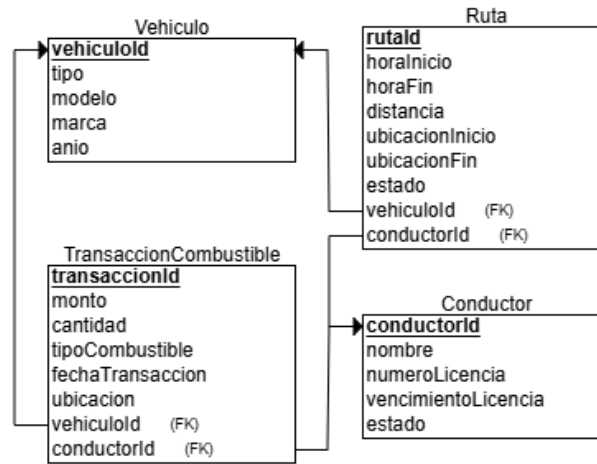


Figure 3: Modelo Relacional de LCS3-Rutas

Script de creación de nodos

4.2.1 Nodo LCS1-Principal

Gestiona la información básica de la flotilla, como los vehículos y documentos.

```
1  -- Creacion del nodo LCS1-Principal
2  CREATE DATABASE LCS1_Principal;
3  USE LCS1_Principal;
4
5  CREATE TABLE Flotilla (
6      flotillaId INT PRIMARY KEY AUTO_INCREMENT,
7      nombreEmpresa VARCHAR(100) NOT NULL,
8      gestorFlotilla VARCHAR(100),
9      fechaCreacion DATE
10 );
11
12 CREATE TABLE Vehiculo (
13     vehiculoId INT PRIMARY KEY AUTO_INCREMENT,
14     flotillaId INT NOT NULL,
15     tipo VARCHAR(50) NOT NULL,
16     modelo VARCHAR(50) NOT NULL,
17     marca VARCHAR(50) NOT NULL,
18     anio INT NOT NULL,
19     estado VARCHAR(20) DEFAULT 'Activo',
20     fechaVerificacion DATE,
21     FOREIGN KEY (flotillaId) REFERENCES Flotilla(flotillaId) ON DELETE
22         CASCADE
23 );
24
25 CREATE TABLE Documento (
26     documentoId INT PRIMARY KEY AUTO_INCREMENT,
27     vehiculoId INT NOT NULL,
```

```

27         tipo VARCHAR(50) NOT NULL,
28         fechaVencimiento DATE NOT NULL,
29         estado VARCHAR(20) DEFAULT 'Vigente',
30         rutaArchivo VARCHAR(255),
31         FOREIGN KEY (vehiculoId) REFERENCES Vehiculo(vehiculoId) ON DELETE
           CASCADE
32     );

```

Listing 4: LCS1-Principal

4.2.2 LCS2-Mantenimiento

Se ocupa de los registros de mantenimiento de los vehículos.

```

1      -- Creacion del nodo LCS2-Mantenimiento
2      CREATE DATABASE LCS2_Mantenimiento;
3      USE LCS2_Mantenimiento;
4
5      CREATE TABLE Vehiculo (
6          vehiculoId INT PRIMARY KEY AUTO_INCREMENT,
7          estado VARCHAR(20) DEFAULT 'Activo',
8          fechaVerificacion DATE
9      );
10
11     CREATE TABLE Mantenimiento (
12         mantenimientoId INT PRIMARY KEY AUTO_INCREMENT,
13         vehiculoId INT NOT NULL,
14         fechaServicio DATE NOT NULL,
15         tipoServicio VARCHAR(100) NOT NULL,
16         descripcion VARCHAR(200),
17         costo DECIMAL(10,2) NOT NULL,
18         estado VARCHAR(20) DEFAULT 'Completado',
19         FOREIGN KEY (vehiculoId) REFERENCES Vehiculo(vehiculoId) ON DELETE
           CASCADE
20     );

```

Listing 5: LCS2-Mantenimiento

4.2.3 LCS3-Rutas

Gestiona las rutas, los conductores y las transacciones de combustible.

```

1      -- Creacion del nodo LCS3-Rutas
2      CREATE DATABASE LCS3_Rutas;
3      USE LCS3_Rutas;
4
5      CREATE TABLE Vehiculo (
6          vehiculoId INT PRIMARY KEY AUTO_INCREMENT,
7          tipo VARCHAR(50) NOT NULL,
8          modelo VARCHAR(50) NOT NULL,
9          marca VARCHAR(50) NOT NULL,
10         anio INT NOT NULL
11     );
12
13     CREATE TABLE Conductor (
14         conductorId INT PRIMARY KEY AUTO_INCREMENT,
15         nombre VARCHAR(100) NOT NULL,
16         numeroLicencia VARCHAR(50) NOT NULL,

```

```

17      vencimientoLicencia DATE NOT NULL,
18      estado VARCHAR(20) DEFAULT 'Activo'
19  );
20
21  CREATE TABLE Ruta (
22      rutaId INT PRIMARY KEY AUTO_INCREMENT,
23      vehiculoId INT NOT NULL,
24      conductorId INT NOT NULL,
25      horaInicio DATETIME NOT NULL,
26      horaFin DATETIME,
27      distancia DECIMAL(10,2),
28      ubicacionInicio VARCHAR(100) NOT NULL,
29      ubicacionFin VARCHAR(100) NOT NULL,
30      estado VARCHAR(20) DEFAULT 'Pendiente',
31      FOREIGN KEY (vehiculoId) REFERENCES Vehiculo(vehiculoId) ON DELETE
32          CASCADE,
33      FOREIGN KEY (conductorId) REFERENCES Conductor(conductorId) ON
34          DELETE CASCADE
35  );
36
37  CREATE TABLE TransaccionCombustible (
38      transaccionId INT PRIMARY KEY AUTO_INCREMENT,
39      vehiculoId INT NOT NULL,
40      conductorId INT NOT NULL,
41      monto DECIMAL(10,2) NOT NULL,
42      cantidad DECIMAL(10,2) NOT NULL,
43      tipoCombustible VARCHAR(20) NOT NULL,
44      fechaTransaccion DATETIME NOT NULL,
45      ubicacion VARCHAR(100),
46      FOREIGN KEY (vehiculoId) REFERENCES Vehiculo(vehiculoId) ON DELETE
47          CASCADE,
48      FOREIGN KEY (conductorId) REFERENCES Conductor(conductorId) ON
49          DELETE CASCADE
50  );

```

Listing 6: LCS3-Rutas

Script de extracción de datos

4.3.1 Nodo LCS1-Principal

```

1      USE sistemagestionflotillas;
2
3      -- Exportar datos relevantes de la tabla Flotilla
4      SELECT flotillaId, nombreEmpresa, gestorFlotilla, fechaCreacion
5      INTO OUTFILE 'C:\\ProgramData\\MySQL\\MySQL\\Server_8.0\\Uploads\\
6          lcs1_flotilla.csv'
7      FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
8      LINES TERMINATED BY '\n'
9      FROM flotilla;
10
11     -- Exportar datos relevantes de la tabla Vehiculo
12     SELECT vehiculoId, flotillaId, tipo, modelo, marca, anio, estado
13     INTO OUTFILE 'C:\\ProgramData\\MySQL\\MySQL\\Server_8.0\\Uploads\\
14         lcs1_vehiculo.csv'
15     FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
16     LINES TERMINATED BY '\n'
17     FROM vehiculo;

```



```

16      -- Exportar datos relevantes de la tabla Documento
17      SELECT documentoId, vehiculoId, tipo, fechaVencimiento, estado,
18             rutaArchivo
19      INTO OUTFILE 'C:\\ProgramData\\MySQL\\MySQL_Server_8.0\\Uploads\\
20             lcs1_documento.csv'
21      FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
22      LINES TERMINATED BY '\n'
23      FROM documento;

```

Listing 7: LCS1-Principal

4.3.2 LCS2-Mantenimiento

```

1      USE sistemagestionflotillas;
2
3      -- Exportar datos relevantes de la tabla Vehiculo
4      SELECT vehiculoId, estado, fechaVerificacion
5      INTO OUTFILE 'C:\\ProgramData\\MySQL\\MySQL_Server_8.0\\Uploads\\
6             lcs2_vehiculo.csv'
7      FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
8      LINES TERMINATED BY '\n'
9      FROM vehiculo;
10
11     -- Exportar datos relevantes de la tabla Mantenimiento
12     SELECT mantenimientoId, vehiculoId, fechaServicio, tipoServicio,
13            descripcion, costo, estado
14     INTO OUTFILE 'C:\\ProgramData\\MySQL\\MySQL_Server_8.0\\Uploads\\
15            lcs2_mantenimiento.csv'
16     FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
17     LINES TERMINATED BY '\n'
18     FROM mantenimiento;

```

Listing 8: LCS2-Mantenimiento

4.3.3 LCS3-Rutas

```

1      USE sistemagestionflotillas;
2
3      -- Exportar datos relevantes de la tabla Vehiculo
4      SELECT vehiculoId, tipo, modelo, marca, anio
5      INTO OUTFILE 'C:\\ProgramData\\MySQL\\MySQL_Server_8.0\\Uploads\\
6             lcs3_vehiculo.csv'
7      FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
8      LINES TERMINATED BY '\n'
9      FROM vehiculo;
10
11     -- Exportar datos relevantes de la tabla Conductor
12     SELECT conductorId, nombre, numeroLicencia, vencimientoLicencia,
13            estado
14     INTO OUTFILE 'C:\\ProgramData\\MySQL\\MySQL_Server_8.0\\Uploads\\
15            lcs3_conductor.csv'
16     FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
17     LINES TERMINATED BY '\n'
18     FROM conductor;

```

```

17      -- Exportar datos relevantes de la tabla Ruta
18      SELECT rutaId, vehiculoId, conductorId, horaInicio, horaFin, distancia
19      , ubicacionInicio, ubicacionFin, estado
20      INTO OUTFILE 'C:\\ProgramData\\MySQL\\MySQL_Server_8.0\\Uploads\\
21      lcs3_ruta.csv'
22      FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
23      LINES TERMINATED BY '\n'
24      FROM ruta;
25
26      -- Exportar datos relevantes de la tabla TransaccionCombustible
27      SELECT transaccionId, vehiculoId, conductorId, monto, cantidad,
28      tipoCombustible, fechaTransaccion, ubicacion
29      INTO OUTFILE 'C:\\ProgramData\\MySQL\\MySQL_Server_8.0\\Uploads\\
30      lcs3_transaccion_combustible.csv'
31      FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
32      LINES TERMINATED BY '\n'
33      FROM transaccionCombustible;

```

Listing 9: LCS3-Rutas

Script de carga de datos

4.4.1 Nodo LCS1-Principal

```

1      USE LCS1_Principal;
2
3      -- Cargar datos en la tabla Flotilla desde el archivo CSV
4      LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL_Server_8.0\\Uploads\\
5      lcs1_flotilla.csv'
6      INTO TABLE flotilla
7      FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
8      LINES TERMINATED BY '\n'
9      (flotillaId, nombreEmpresa, gestorFlotilla, fechaCreacion);
10
11      -- Cargar datos en la tabla Vehiculo desde el archivo CSV
12      LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL_Server_8.0\\Uploads\\
13      lcs1_vehiculo.csv'
14      INTO TABLE vehiculo
15      FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
16      LINES TERMINATED BY '\n'
17      (vehiculoId, flotillaId, tipo, modelo, marca, anio, estado);
18
19      -- Cargar datos en la tabla Documento desde el archivo CSV
20      LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL_Server_8.0\\Uploads\\
21      lcs1_documento.csv'
22      INTO TABLE documento
23      FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
24      LINES TERMINATED BY '\n'
25      (documentoId, vehiculoId, tipo, fechaVencimiento, estado, rutaArchivo)
26      ;

```

Listing 10: LCS1-Principal

4.4.2 LCS2-Mantenimiento

```

1      USE LCS2_Mantenimiento;

```

```

2      -- Cargar datos en la tabla Vehiculo desde el archivo CSV (LCS2)
3      LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL_Server_8.0\\Uploads\\
4          lcs2_vehiculo.csv'
5      INTO TABLE vehiculo
6      FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
7      LINES TERMINATED BY '\n'
8      (vehiculoId, estado, fechaVerificacion);
9
10     -- Cargar datos en la tabla Mantenimiento desde el archivo CSV
11     LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL_Server_8.0\\Uploads\\
12         lcs2_mantenimiento.csv'
13     INTO TABLE mantenimiento
14     FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
15     LINES TERMINATED BY '\n'
16     (mantenimientoId, vehiculoId, fechaServicio, tipoServicio, descripcion
17         , costo, estado);

```

Listing 11: LCS2-Mantenimiento

4.4.3 LCS3-Rutas

```

1      USE LCS3_Rutas;
2
3      -- Cargar datos en la tabla Vehiculo desde el archivo CSV (LCS3)
4      LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL_Server_8.0\\Uploads\\
5          lcs3_vehiculo.csv'
6      INTO TABLE vehiculo
7      FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
8      LINES TERMINATED BY '\n'
9      (vehiculoId, tipo, modelo, marca, anio);
10
11     -- Cargar datos en la tabla Conductor desde el archivo CSV
12     LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL_Server_8.0\\Uploads\\
13         lcs3_conductor.csv'
14     INTO TABLE conductor
15     FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
16     LINES TERMINATED BY '\n'
17     (conductorId, nombre, numeroLicencia, vencimientoLicencia, estado);
18
19     -- Cargar datos en la tabla Ruta desde el archivo CSV
20     LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL_Server_8.0\\Uploads\\
21         lcs3_ruta.csv'
22     INTO TABLE ruta
23     FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
24     LINES TERMINATED BY '\n'
25     (rutaId, vehiculoId, conductorId, horaInicio, horaFin, distancia,
26         ubicacionInicio, ubicacionFin, estado);
27
28     -- Cargar datos en la tabla TransaccionCombustible desde el archivo
29     CSV
30     LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL_Server_8.0\\Uploads\\
31         lcs3_transaccion_combustible.csv'
32     INTO TABLE transaccionCombustible
33     FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
34     LINES TERMINATED BY '\n'
35     (transaccionId, vehiculoId, conductorId, monto, cantidad,
36         tipoCombustible, fechaTransaccion, ubicacion);

```

Script de consultas

4.5.1 Consulta en LCS1-Principal

Se hace una unión entre las tablas flotilla y vehiculo para obtener los vehículos asociados con una flotilla específica.

```

1      USE LCS1_Principal;
2
3      -- Consulta que une las tablas Flotilla y Vehiculo
4      SELECT f.flotillaId, f.nombreEmpresa, v.vehiculoId, v.tipo, v.modelo,
5             v.marca
6      FROM flotilla f
7      JOIN vehiculo v ON f.flotillaId = v.flotillaId
      WHERE f.nombreEmpresa = 'Transportes_Pachuca';

```

Listing 13: Script MySQL

	flotillaId	nombreEmpresa	vehiculoId	tipo	modelo	marca
►	5	Transportes Pachuca	38	SUV	Ram 1500	Peterbilt
	5	Transportes Pachuca	50	Furgón	Mack Anthem	Isuzu
	24	Transportes Pachuca	27	SUV	International LT	Kenworth
	24	Transportes Pachuca	90	Pickup	Mack Anthem	Chevrolet
	36	Transportes Pachuca	3	Pickup	Kenworth T680	Mercedes-Benz
	36	Transportes Pachuca	40	Tráiler	Mercedes-Benz Sprinter	Ram
	36	Transportes Pachuca	97	Furgón	Peterbilt 579	Mercedes-Benz
	40	Transportes Pachuca	30	Camión	International LT	Kenworth
	40	Transportes Pachuca	51	SUV	Hino 338	International
	40	Transportes Pachuca	88	SUV	Chevrolet Silverado	Volvo
	43	Transportes Pachuca	22	Tráiler	Isuzu NPR	Mercedes-Benz
	43	Transportes Pachuca	74	Sedán	Ram 1500	Ford
	43	Transportes Pachuca	98	Furgón	Mercedes-Benz Sprinter	Mercedes-Benz
	51	Transportes Pachuca	5	Camión	Ram 1500	Chevrolet
	51	Transportes Pachuca	32	Sedán	Hino 338	Peterbilt
	79	Transportes Pachuca	61	SUV	International LT	Volvo
	83	Transportes Pachuca	29	Pickup	Peterbilt 579	Ford

Figure 4: Resultado de la consulta

4.5.2 Consulta en LCS2-Mantenimiento

La unión entre las tablas vehiculo y mantenimiento proporciona los detalles de los vehículos y los servicios de mantenimiento realizados.

```

1      USE LCS2_Mantenimiento;
2
3      -- Consulta que une las tablas Vehiculo y Mantenimiento
4      SELECT v.vehiculoId, m.fechaServicio, m.tipoServicio, m.descripcion, m
5             .costo
6      FROM vehiculo v
7      JOIN mantenimiento m ON v.vehiculoId = m.vehiculoId
      WHERE m.estado = 'Completado';

```

Listing 14: Script MySQL

	vehiculoId	fechaServicio	tipoServicio	descripcion	costo
►	92	2020-07-11	Reparación motor	Servicio realizado en taller Querétaro	1236.82
	19	2022-04-14	Cambio de aceite	Servicio realizado en taller Guadalajara	1978.17
	83	2021-07-02	Rotación de llantas	Servicio realizado en taller Pachuca	869.15
	4	2023-12-01	Revisión de frenos	Servicio realizado en taller Pachuca	2712.99
	69	2023-10-16	Revisión de frenos	Servicio realizado en taller Querétaro	1473.41
	59	2022-07-30	Rotación de llantas	Servicio realizado en taller Pachuca	1488.80
	47	2022-11-17	Sistema eléctrico	Servicio realizado en taller Monterrey	2766.32
	2	2023-12-15	Alineación	Servicio realizado en taller Guadalajara	2870.10
	79	2023-08-08	Reparación motor	Servicio realizado en taller CDMX	1770.83
	20	2023-01-19	Sistema eléctrico	Servicio realizado en taller CDMX	1409.71
	5	2020-08-12	Balanceo	Servicio realizado en taller CDMX	568.85
	48	2020-03-26	Reparación motor	Servicio realizado en taller Pachuca	2460.84
	69	2022-11-04	Rotación de llantas	Servicio realizado en taller Pachuca	1872.45
	98	2020-04-30	Alineación	Servicio realizado en taller Querétaro	2456.67
	40	2020-08-30	Balanceo	Servicio realizado en taller Monterrey	2002.99
	30	2024-05-06	Cambio de aceite	Servicio realizado en taller Monterrey	3087.32
	24	2022-06-15	Cambio de aceite	Servicio realizado en taller Querétaro	2683.15
	96	2020-05-29	Balanceo	Servicio realizado en taller Guadalajara	3029.03
	74	2024-05-09	Reparación motor	Servicio realizado en taller Querétaro	3235.76
	47	2024-05-09	Balanceo	Servicio realizado en taller Querétaro	2053.09
	76	2022-07-09	Balanceo	Servicio realizado en taller CDMX	697.53
	65	2023-03-10	Cambio de filtros	Servicio realizado en taller CDMX	1846.79
	25	2021-02-24	Balanceo	Servicio realizado en taller Pachuca	3196.28
	83	2023-07-14	Balanceo	Servicio realizado en taller CDMX	1303.95

Figure 5: Resultado de la consulta

4.5.3 Consulta en LCS3-Rutas

Aquí se une vehiculo con ruta para obtener las rutas realizadas por los vehículos, con detalles como la hora de inicio, hora de fin y distancia.

```

1      USE LCS3_Rutas;
2
3      -- Consulta que une las tablas Vehiculo y Ruta
4      SELECT v.vehiculoId, v.tipo, v.modelo, r.rutaId, r.horaInicio, r.
           horaFin, r.distancia
5      FROM vehiculo v
6      JOIN ruta r ON v.vehiculoId = r.vehiculoId
7      WHERE r.estado = 'Pendiente';

```

Listing 15: Script MySQL

	vehiculoId	tipo	modelo	rutaId	horaInicio	horaFin	distancia
►	54	Sedán	Ford Transit	1	2023-01-30 12:00:00	2023-01-18 18:00:00	997.62
	37	Camión	Mack Anthem	10	2023-02-02 06:00:00	2023-01-27 23:00:00	170.50
	9	Pickup	Ford F-150	11	2023-01-12 23:00:00	2023-01-02 00:00:00	106.64
	1	Pickup	Volvo VNL	17	2023-01-29 05:00:00	2023-01-18 03:00:00	67.12
	48	Furgón	Kenworth T680	18	2023-01-18 01:00:00	2023-01-27 21:00:00	510.07
	87	Camión	Ram 1500	19	2023-01-06 11:00:00	2023-01-06 12:00:00	357.50
	40	Tráiler	Mercedes-Benz Sprinter	23	2023-01-14 01:00:00	2023-01-08 03:00:00	887.63
	42	SUV	International LT	28	2023-01-02 18:00:00	2023-01-24 05:00:00	348.17
	45	Furgón	Freightliner Cascadia	29	2023-02-02 16:00:00	2023-01-05 02:00:00	678.03
	8	Camión	Mack Anthem	30	2023-01-25 09:00:00	2023-01-10 19:00:00	301.12
	64	SUV	Isuzu NPR	33	2023-01-10 10:00:00	2023-01-24 11:00:00	680.46
	34	Tráiler	Peterbilt 579	40	2023-01-02 01:00:00	2023-02-01 08:00:00	616.93
	72	Pickup	Ford F-150	41	2023-01-01 20:00:00	2023-01-15 08:00:00	110.87
	26	SUV	International LT	42	2023-01-01 11:00:00	2023-01-27 13:00:00	931.91
	60	Tráiler	Kenworth T680	44	2023-01-15 10:00:00	2023-01-02 09:00:00	903.09
	12	Tráiler	Ford F-150	45	2023-01-21 12:00:00	2023-01-31 08:00:00	707.83
	51	SUV	Hino 338	47	2023-01-13 16:00:00	2023-01-26 18:00:00	726.95
	23	SUV	Kenworth T680	51	2023-01-17 23:00:00	2023-01-29 07:00:00	717.11
	57	Sedán	Hino 338	57	2023-01-29 18:00:00	2023-01-20 19:00:00	399.99
	82	Sedán	Volvo VNL	60	2023-01-28 07:00:00	2023-02-01 11:00:00	289.54
	95	Sedán	Isuzu NPR	63	2023-01-18 11:00:00	2023-01-21 08:00:00	490.56
	39	Furgón	Ford F-150	67	2023-01-29 05:00:00	2023-01-02 01:00:00	629.28
	91	SUV	Chevrolet Silverado	68	2023-01-15 00:00:00	2023-01-08 15:00:00	878.61
	91	SUV	Chevrolet Silverado	77	2023-01-28 14:00:00	2023-01-14 21:00:00	610.31
	54	Sedán	Ford Transit	81	2023-01-19 19:00:00	2023-01-11 21:00:00	933.96

Figure 6: Resultado de la consulta

4.5.4 Consulta entre nodos distribuidos

El ejemplo final muestra cómo se podrían hacer consultas entre nodos distribuidos de bases de datos diferentes, usando la notación LCS1-Principal y LCS2-Mantenimiento para referirse a las bases de datos en otros nodos.

```

1      SELECT f.flotillaId, f.nombreEmpresa, m.fechaServicio, m.tipoServicio,
2             m.costo
3      FROM LCS1_Principal.flotilla f
4      JOIN LCS2_Mantenimiento.mantenimiento m ON f.flotillaId = m.vehiculoId
      WHERE m.fechaServicio BETWEEN '2023-01-01' AND '2023-12-31';

```

Listing 16: Script MySQL

	flotillaId	nombreEmpresa	fechaServicio	tipoServicio	costo
►	44	Transportes Del Centro	2023-01-28	Rotación de llantas	3244.78
	23	Transportes Del Oeste	2023-12-05	Reparación motor	2250.66
	4	Transportes Del Este	2023-12-01	Revisión de frenos	2712.99
	69	Transportes Intermodal	2023-10-16	Revisión de frenos	1473.41
	33	Transportes Carga	2023-01-10	Alineación	3118.67
	2	Transportes Intermodal	2023-12-15	Alineación	2870.10
	79	Transportes Pachuca	2023-08-08	Reparación motor	1770.83
	20	Transportes Urbana	2023-01-19	Sistema eléctrico	1409.71
	24	Transportes Pachuca	2023-10-27	Cambio de filtros	2336.52
	87	Transportes Carga	2023-09-16	Cambio de filtros	805.26
	70	Transportes Hidalgo	2023-10-10	Rotación de llantas	865.40
	60	Transportes Nacional	2023-12-31	Reparación motor	1236.28
	61	Transportes Urbana	2023-07-28	Revisión de frenos	2329.08
	20	Transportes Urbana	2023-04-23	Rotación de llantas	3398.50
	60	Transportes Nacional	2023-12-09	Cambio de filtros	2994.05
	65	Transportes Carga	2023-03-10	Cambio de filtros	1846.79
	83	Transportes Pachuca	2023-07-14	Balanceo	1303.95
	22	Transportes Pachuca	2023-06-30	Revisión de frenos	1967.86
	100	Transportes Nacional	2023-11-06	Alineación	2599.97
	12	Transportes Rápida	2023-01-28	Rotación de llantas	2921.35
	40	Transportes Pachuca	2023-11-01	Rotación de llantas	728.48
	40	Transportes Pachuca	2023-06-27	Reparación motor	1190.05
	21	Transportes Águila	2023-03-12	Cambio de aceite	3011.75

Figure 7: Resultado de la consulta

5. Conclusiones

La fragmentación en bases de datos relacionales es una técnica fundamental para optimizar la administración y el rendimiento de los sistemas de información. En el caso de la gestión de flotillas, donde se manejan grandes volúmenes de datos sobre vehículos, conductores, rutas y transacciones, la implementación de fragmentación horizontal y vertical permite mejorar la eficiencia en el acceso y consulta de la información.

El manejo eficiente de datos en bases de datos distribuidas es crucial para garantizar la integridad, disponibilidad y rendimiento del sistema. La fragmentación vertical permite distribuir los datos de manera optimizada, mejorando la eficiencia de acceso. Los procesos ETL facilitan la integración y transformación de datos desde distintas fuentes, permitiendo una gestión más estructurada de la información. El uso de SELECT INTO FILE y LOAD DATA INFILE facilita la extracción y carga de datos, mejorando la interoperabilidad entre sistemas. Finalmente, la posibilidad de ejecutar consultas entre bases de datos distintas fortalece la integración de datos en un entorno distribuido, permitiendo una administración más efectiva de la información.

Estas herramientas y técnicas son fundamentales en la gestión de flotillas de automóviles, donde la información distribuida sobre vehículos, rutas y mantenimiento debe ser accesible y manejable de manera eficiente para optimizar la operación del sistema.

Referencias Bibliográficas

References

- [1] Silberschatz, A., Korth, H. F., Sudarshan, S. (2020). Conceptos de sistemas de bases de datos. (7th ed.). *McGraw-Hill*.
- [2] Elmasri, R., Navathe, S. (2015). Sistemas de bases de datos. (7th ed.). *Pearson*.
- [3] Documentación oficial de MySQL: <https://dev.mysql.com/doc/>
- [4] Coronel, C., Morris, S. (2017). Diseño de bases de datos: Un enfoque práctico. *Cengage Learning*.