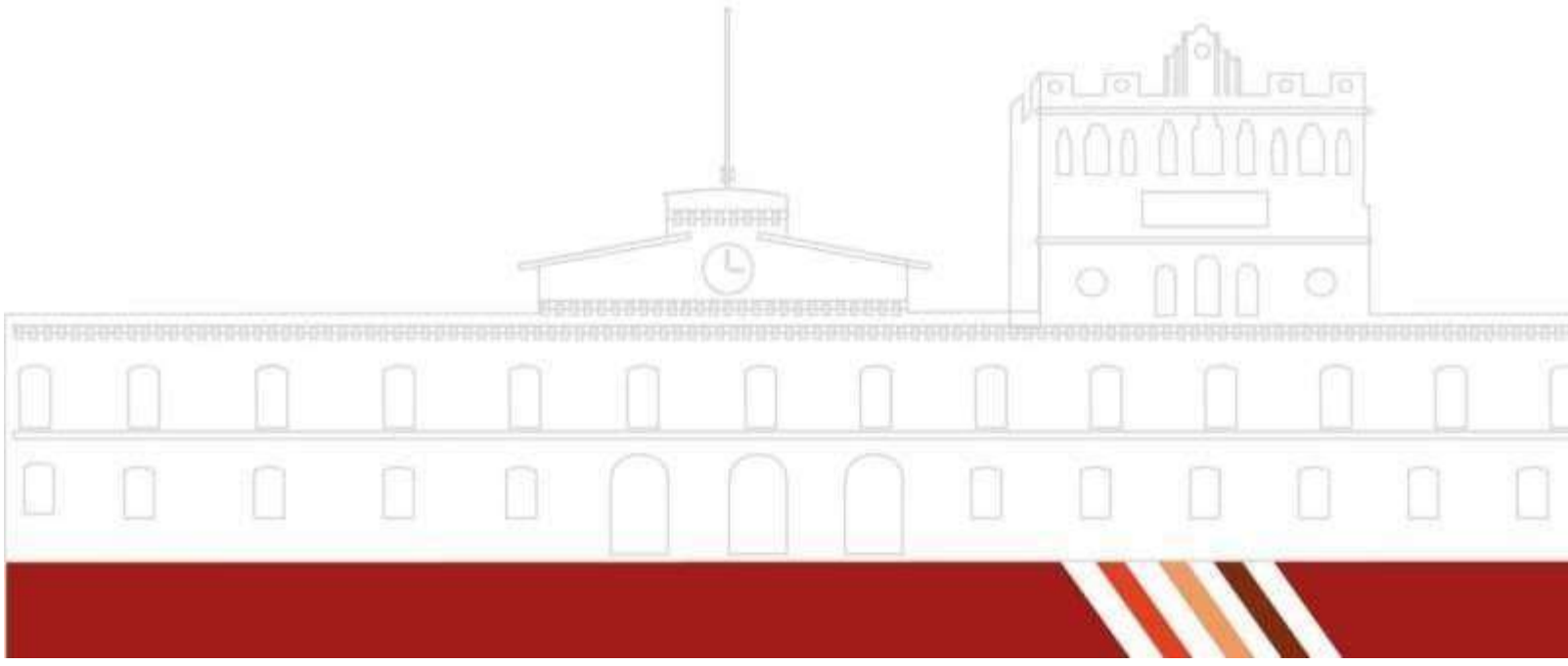
Reporte de Practica No. 3.2

Nombre de la Practica: SQL FRAGMENTOS

Flotilla de Autos

Alumno: Kevin Badillo Olmos

Dr. Eduardo Cornejo Velázquez



Implementación de Nodos Físicos para la Base de Datos Distribuida de Gestión de Flotillas

1. MARCO TEÓRICO

Esta sección define los conceptos fundamentales utilizados en la **migración de una base de datos centralizada hacia una arquitectura distribuida basada en nodos**.

1.1. Fragmentación

La **fragmentación** es una técnica que consiste en **dividir una base de datos lógica en fragmentos más pequeños**, los cuales se almacenan en **diferentes nodos físicos (servidores)**.
El propósito de esta técnica es **mejorar el rendimiento, la escalabilidad y la disponibilidad** de los datos dentro de un sistema distribuido.

Existen principalmente dos tipos de fragmentación:

- **Fragmentación Vertical:**
  Consiste en dividir una tabla en múltiples tablas (fragmentos), **reduciendo la cantidad de columnas** que contiene cada una.
  Cada fragmento conserva la **clave primaria** de la tabla original, permitiendo **reconstruir la información completa mediante uniones (JOIN)**.
  Se utiliza cuando diferentes aplicaciones acceden a **subconjuntos distintos de columnas** de la misma tabla.

  Aplicación en este proyecto:
  El diseño implementa una variación de este concepto. En lugar de fragmentar una sola tabla, se realiza la **división por vertical de negocio**.
  Así, la entidad lógica **"Vehículo"** se divide:

  - Sus datos administrativos (y documentación legal) se almacenan en el **Nodo LCS1_Principal**.
  - Sus datos operativos (como mantenimiento) se almacenan en el **Nodo LCS2_Mantenimiento**.
- **Fragmentación Horizontal:**
  Divide una tabla en fragmentos basados en **grupos de filas**, definidos por algún criterio (por ejemplo, región, fecha o tipo de cliente).
  Cada fragmento conserva la misma estructura de columnas, pero con diferentes conjuntos de registros.

  Aplicación en este proyecto:
  Las tablas **Mantenimiento** y **Ruta** representan fragmentos horizontales de la actividad total de la flotilla, **separados por función operativa**.

- **Replicación:**
  Es el proceso de **copiar los mismos datos en varios nodos**.
  Se utiliza principalmente para las **tablas maestras** (como *Vehículo* o *TallerMecánico*), permitiendo que las consultas en cada nodo sean **locales y más rápidas**, evitando el tráfico innecesario en la red.

1.2. Procesos ETL (Extract, Transform, Load)

El proceso **ETL** (Extraer, Transformar y Cargar) se emplea para **mover datos desde una fuente hacia un destino**, ya sea un *data warehouse* o, en este caso, los **nodos distribuidos**.

1. **Extract (Extraer):**
   Consiste en **leer y exportar los datos** desde la base de datos de origen (en este proyecto, la base central *gestion_flotilla*).
2. **Transform (Transformar):**
   Involucra **limpiar, formatear y validar los datos**, además de aplicar reglas de negocio según las necesidades del nodo destino.
   En este caso, la transformación es **mínima**, ya que solo se seleccionan las tablas y atributos correspondientes a cada nodo.
3. **Load (Cargar):**
   Es la fase donde los **datos transformados se escriben** en las bases de datos de destino, es decir, en los nodos **LCS1, LCS2 y LCS3**.

1.3. Comando SELECT ... INTO OUTFILE

El comando **SELECT ... INTO OUTFILE** de **MySQL** se utiliza en la fase de **Extracción del proceso ETL**, permitiendo **exportar el resultado de una consulta SELECT directamente a un archivo de texto** (por ejemplo, .csv o .txt) en el sistema de archivos del servidor.

Sintaxis básica:

```
SELECT * FROM nombre_tabla
INTO OUTFILE '/ruta/segura/archivo.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

Importante:
Este comando está sujeto a restricciones de seguridad mediante la variable del sistema **secure_file_priv**.
Por lo tanto, la exportación de archivos **solo puede realizarse en la carpeta especificada** por dicha variable dentro de la configuración de MySQL.

2. ESQUEMA CONCEPTUAL LOCAL DE CADA NODO

A continuación, se describe la estructura de cada nodo físico simulado del sistema distribuido.

2.1. Nodo LCS1 – Principal (Administración y Activos)

Este nodo contiene las **tablas maestras** relacionadas con los activos y la documentación legal.

- **Propietario (1) --- (N) Vehículo** *(Maestra)*
- **Vehículo (1) --- (N) Documento**
- **TallerMecánico (Maestra)**

2.2. Nodo LCS2 – Mantenimiento (Taller)

Este nodo está optimizado para las **operaciones del taller**, incluyendo los registros de mantenimientos realizados. Contiene **réplicas** de las tablas maestras necesarias.

- **Vehículo (Réplica) (1) --- (N) Mantenimiento**
- **TallerMecánico (Réplica) (1) --- (N) Mantenimiento**

2.3. Nodo LCS3 – Rutas (Operaciones y Logística)

Este nodo gestiona las **operaciones diarias, los conductores y el consumo de combustible**. Utiliza réplicas de las tablas maestras necesarias para las operaciones.

- **Vehículo (Réplica) (1) --- (N) Ruta**
- **Conductor (1) --- (N) Ruta**
- **Vehículo (Réplica) (1) --- (N) TransacciónCombustible**
- **Conductor (1) --- (N) TransacciónCombustible**

3 SCRIPT DE CREACIÓN DE NODOS

Este script crea las tres bases de datos (esquemas) que simulan nuestros nodos físicos.

```
-- ======================================================
-- Script de Creación de Nodos Físicos (Simulados)
-- ======================================================

-- Eliminar esquemas anteriores si existen para una instalación limpia
DROP DATABASE IF EXISTS LCS1_Principal;
DROP DATABASE IF EXISTS LCS2_Mantenimiento;
DROP DATABASE IF EXISTS LCS3_Rutas;

-- ------------------------------------------------------------------------------------------------------------------
-- NODO 1: LCS1_Principal (Administración y Activos)
-- ------------------------------------------------------------------------------------------------------------------
CREATE DATABASE IF NOT EXISTS LCS1_Principal DEFAULT CHARACTER SET utf8mb4;
USE LCS1_Principal;

CREATE TABLE Propietario (
  idPropietario INT PRIMARY KEY AUTO_INCREMENT,
  nombre VARCHAR(100) NOT NULL,
  rfc VARCHAR(13) UNIQUE
);

CREATE TABLE TallerMecanico (
  idTaller INT PRIMARY KEY AUTO_INCREMENT,
  nombre VARCHAR(100) NOT NULL,
  direccion VARCHAR(255),
  telefono VARCHAR(15)
);

CREATE TABLE Vehiculo (
  idVehiculo INT PRIMARY KEY AUTO_INCREMENT,
  placa VARCHAR(10) UNIQUE NOT NULL,
```

```
  marca VARCHAR(50) NOT NULL,
  modelo VARCHAR(50) NOT NULL,
  anio INT NOT NULL,
  numeroSerie VARCHAR(20) UNIQUE NOT NULL,
  idPropietario INT,
  FOREIGN KEY (idPropietario) REFERENCES Propietario(idPropietario)
);

CREATE TABLE Documento (
  idDocumento INT PRIMARY KEY AUTO_INCREMENT,
  idVehiculo INT NOT NULL,
  tipoDocumento ENUM('Tenencia', 'Verificacion', 'Seguro', 'Tarjeta de
Circulacion') NOT NULL,
  fechaVencimiento DATE,
  numeroFolio VARCHAR(50),
  FOREIGN KEY (idVehiculo) REFERENCES Vehiculo(idVehiculo)
);

--- ----------------------------------------------------------------------------------------------------------------------
-- NODO 2: LCS2_Mantenimiento (Taller)
--- ----------------------------------------------------------------------------------------------------------------------
CREATE DATABASE IF NOT EXISTS LCS2_Mantenimiento DEFAULT CHARACTER SET
utf8mb4;
USE LCS2_Mantenimiento;
```

```
CREATE TABLE Vehiculo (
  idVehiculo INT PRIMARY KEY AUTO_INCREMENT,
  placa VARCHAR(10) UNIQUE NOT NULL,
  marca VARCHAR(50) NOT NULL,
  modelo VARCHAR(50) NOT NULL,
  anio INT NOT NULL,
  numeroSerie VARCHAR(20) UNIQUE NOT NULL,
  idPropietario INT
);

CREATE TABLE TallerMecanico (
  idTaller INT PRIMARY KEY AUTO_INCREMENT,
  nombre VARCHAR(100) NOT NULL,
  direccion VARCHAR(255),
  telefono VARCHAR(15)
);

CREATE TABLE Mantenimiento (
  idMantenimiento INT PRIMARY KEY AUTO_INCREMENT,
  idVehiculo INT NOT NULL,
  idTaller INT NOT NULL,
  fechaServicio DATE NOT NULL,
  descripcion TEXT NOT NULL,
  costo DECIMAL(10, 2) NOT NULL,
  tipoMantenimiento ENUM('Preventivo', 'Correctivo') NOT NULL,
  FOREIGN KEY (idVehiculo) REFERENCES Vehiculo(idVehiculo),
  FOREIGN KEY (idTaller) REFERENCES TallerMecanico(idTaller)
);

--- ----------------------------------------------------------------------------------------------------------
-- NODO 3: LCS3_Rutas (Operaciones y Logística)
--- ----------------------------------------------------------------------------------------------------------
CREATE DATABASE IF NOT EXISTS LCS3_Rutas DEFAULT CHARACTER SET utf8mb4;
USE LCS3_Rutas;

CREATE TABLE Vehiculo (
  idVehiculo INT PRIMARY KEY AUTO_INCREMENT,
  placa VARCHAR(10) UNIQUE NOT NULL,
  marca VARCHAR(50) NOT NULL,
  modelo VARCHAR(50) NOT NULL,
  anio INT NOT NULL,
  numeroSerie VARCHAR(20) UNIQUE NOT NULL,
  idPropietario INT
);

CREATE TABLE Conductor (
  idConductor INT PRIMARY KEY AUTO_INCREMENT,
  nombre VARCHAR(100) NOT NULL,
  numeroLicencia VARCHAR(20) UNIQUE NOT NULL,
  fechaVencimientoLicencia DATE NOT NULL
);

CREATE TABLE Ruta (
  idRuta INT PRIMARY KEY AUTO_INCREMENT,
  idVehiculo INT NOT NULL,
  idConductor INT NOT NULL,
  origen VARCHAR(255) NOT NULL,
  destino VARCHAR(255) NOT NULL,
  fechaSalida DATETIME NOT NULL,
  fechaLlegadaEstimada DATETIME NOT NULL,
  fechaLlegadaReal DATETIME,
  estatus ENUM('Programada', 'En Curso', 'Completada', 'Cancelada') NOT NULL,
```

```
  FOREIGN KEY (idVehiculo) REFERENCES Vehiculo(idVehiculo),
  FOREIGN KEY (idConductor) REFERENCES Conductor(idConductor)
);

CREATE TABLE transaccionCombustible (
  idTransaccion INT PRIMARY KEY AUTO_INCREMENT,
  idVehiculo INT NOT NULL,
  idConductor INT NOT NULL,
  fecha DATETIME NOT NULL,
  litros DECIMAL(8, 2) NOT NULL,
  montoTotal DECIMAL(10, 2) NOT NULL,
  estacionServicio VARCHAR(100),
  odometro INT,
  FOREIGN KEY (idVehiculo) REFERENCES Vehiculo(idVehiculo),
  FOREIGN KEY (idConductor) REFERENCES Conductor(idConductor)
);
```

1. SCRIPTS DE EXTRACCIÓN DE DATOS (ETL - Extract)

Estos scripts se ejecutarían en la **base de datos centralizada original** (que llamaremos `gestion_flotilla_central`) para exportar los datos a archivos CSV.

```
-- Asumiendo que la base de datos central se llama 'gestion_flotilla_central'
USE gestion_flotilla_central;

-- Ruta de exportación (¡DEBE ser la ruta de 'secure_file_priv'!)
SET @export_path = 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/';

-- Extracción de tablas para LCS1
SELECT * FROM Propietario INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/propietario.csv' FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES
TERMINATED BY '\n';
SELECT * FROM TallerMecanico INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/taller.csv' FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES
TERMINATED BY '\n';
SELECT * FROM Vehiculo INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/vehiculo.csv' FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES
TERMINATED BY '\n';
SELECT * FROM Documento INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/documento.csv' FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES
TERMINATED BY '\n';

-- Extracción de tablas para LCS2
SELECT * FROM Mantenimiento INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/mantenimiento.csv' FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES
TERMINATED BY '\n';
-- (Las tablas Vehiculo y TallerMecanico ya se exportaron)

-- Extracción de tablas para LCS3
SELECT * FROM Conductor INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/conductor.csv' FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES
TERMINATED BY '\n';
SELECT * FROM Ruta INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/ruta.csv' FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES
TERMINATED BY '\n';
SELECT * FROM Gasto INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/gasto_combustible.csv' FIELDS TERMINATED BY ',' ENCLOSED BY '"'
LINES TERMINATED BY '\n'; -- (Asumiendo que la tabla se llamaba Gasto)
```

2. SCRIPTS DE CARGA DE DATOS (ETL - Load)

```
-- Desactivar temporalmente la revisión de llaves foráneas para permitir la
carga
SET FOREIGN_KEY_CHECKS = 0;

-- ----------------------------------------------------------------------------
-- CARGA NODO 1: LCS1_Principal
-- ----------------------------------------------------------------------------
USE LCS1_Principal;
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/propietario.csv' INTO TABLE Propietario FIELDS TERMINATED BY ','
ENCLOSED BY '"' LINES TERMINATED BY '\n';
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/taller.csv'
INTO TABLE TallerMecanico FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES
TERMINATED BY '\n';
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/vehiculo.csv'
INTO TABLE Vehiculo FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED
BY '\n';
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/documento.csv' INTO TABLE Documento FIELDS TERMINATED BY ','
ENCLOSED BY '"' LINES TERMINATED BY '\n';

-- ----------------------------------------------------------------------------
-- CARGA NODO 2: LCS2_Mantenimiento
-- ----------------------------------------------------------------------------
USE LCS2_Mantenimiento;
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/vehiculo.csv'
INTO TABLE Vehiculo FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED
BY '\n';
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/taller.csv'
INTO TABLE TallerMecanico FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES
TERMINATED BY '\n';
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/mantenimiento.csv' INTO TABLE Mantenimiento FIELDS TERMINATED BY
',' ENCLOSED BY '"' LINES TERMINATED BY '\n';

-- ----------------------------------------------------------------------------
-- CARGA NODO 3: LCS3_Rutas
-- ----------------------------------------------------------------------------
USE LCS3_Rutas;
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/vehiculo.csv'
INTO TABLE Vehiculo FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED
BY '\n';
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/conductor.csv' INTO TABLE Conductor FIELDS TERMINATED BY ','
ENCLOSED BY '"' LINES TERMINATED BY '\n';
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/ruta.csv'
INTO TABLE Ruta FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY
'\n';
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/gasto_combustible.csv' INTO TABLE transaccionCombustible FIELDS
TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n';

-- Reactivar la revisión de llaves foráneas
SET FOREIGN_KEY_CHECKS = 1;
```

3. SCRIPT DE CONSULTA DE DATOS (Entre Nodos)

Consulta 1: Vehículos y su Costo Total de Mantenimiento

(Une `LCS1_Principal` y `LCS2_Mantenimiento`)

```
-- Muestra el costo total gastado en mantenimiento para cada vehículo
SELECT
```

```
    n1_veh.placa,
    n1_veh.marca,
    n1_veh.modelo,
    SUM(n2_mant.costo) AS costo_total_mantenimiento
FROM
    LCS1_Principal.Vehiculo AS n1_veh
JOIN
    LCS2_Mantenimiento.Mantenimiento AS n2_mant
    ON n1_veh.idVehiculo = n2_mant.idVehiculo
GROUP BY
    n1_veh.idVehiculo;
```

Consulta 2: Reporte Operativo del Conductor

(Une `LCS3_Rutas` y `LCS1_Principal` para obtener datos del propietario)

```
-- Muestra las rutas activas ('En Curso') e incluye el nombre del propietario
del vehículo
SELECT
    n3_cond.nombre AS nombre_conductor,
    n1_veh.placa,
    n1_prop.nombre AS nombre_propietario,
    n3_ruta.origen,
    n3_ruta.destino,
    n3_ruta.estatus
FROM
    LCS3_Rutas.Ruta AS n3_ruta
JOIN
    LCS3_Rutas.Conductor AS n3_cond
    ON n3_ruta.idConductor = n3_cond.idConductor
JOIN
    LCS1_Principal.Vehiculo AS n1_veh  -- Uniendo con el NODO 1
    ON n3_ruta.idVehiculo = n1_veh.idVehiculo
JOIN
    LCS1_Principal.Propietario AS n1_prop -- Uniendo con el NODO 1
    ON n1_veh.idPropietario = n1_prop.idPropietario
WHERE
    n3_ruta.estatus = 'En Curso';
```

Consulta 3: Reporte Integral del Vehículo (Unión de 3 Nodos)

(Une `LCS1`, `LCS2` y `LCS3`)

```
-- Obtiene un reporte integral de un vehículo específico (placa 'A1B-2C3')
SELECT
    n1_veh.placa,
    n1_doc.tipoDocumento,
    n1_doc.fechaVencimiento,
    n2_mant.ultimo_mantenimiento,
    n3_ruta.ultimo_destino
FROM
    LCS1_Principal.Vehiculo AS n1_veh
-- Unir con NODO 1 para documentos
LEFT JOIN
    (
        SELECT idVehiculo, tipoDocumento, fechaVencimiento
        FROM LCS1_Principal.Documento
        WHERE fechaVencimiento > CURDATE()
        ORDER BY fechaVencimiento ASC
        LIMIT 1
    ) AS n1_doc ON n1_veh.idVehiculo = n1_doc.idVehiculo
-- Unir con NODO 2 para mantenimiento
```

```
LEFT JOIN
    (
        SELECT idVehiculo, MAX(fechaServicio) AS ultimo_mantenimiento
        FROM LCS2_Mantenimiento.Mantenimiento
        GROUP BY idVehiculo
    ) AS n2_mant ON n1_veh.idVehiculo = n2_mant.idVehiculo
-- Unir con NODO 3 para rutas
LEFT JOIN
    (
        SELECT idVehiculo, destino AS ultimo_destino
        FROM LCS3_Rutas.Ruta
        ORDER BY fechaSalida DESC
        LIMIT 1
    ) AS n3_ruta ON n1_veh.idVehiculo = n3_ruta.idVehiculo
WHERE
    n1_veh.placa = 'A1B-2C3'; -- (Placa de ejemplo)
```