

REPORTE DE PRÁCTICA NO. 2.4

NOMBRE DE LA PRÁCTICA: Práctica. 3 nodos BDD
Flotillas

ALUMNO: Saúl Jiménez Mercado
Dr. Eduardo Cornejo-Velázquez



1. Introducción

En esta práctica se explora la creación y el uso de los (LCS) para la gestión de bases de datos distribuidas. El objetivo es demostrar cómo se pueden realizar consultas en diferentes bases de datos que contienen la misma información, permitiendo el acceso desde múltiples nodos. Además, se implementarán procedimientos almacenados y triggers para garantizar la sincronización automática de los datos, de modo que cualquier modificación en el LCS principal se refleje en los otros dos nodos de manera eficiente y segura.

2. Marco Teorico.

Proceso ETL.

Extracción, transformación y carga (ETL) es el proceso consistente en combinar datos de diferentes orígenes un gran repositorio central llamado almacenamiento de datos. ETL utiliza un conjunto de reglas comerciales para limpiar y organizar datos en bruto y prepararlos para el almacenamiento, el análisis de datos y el machine learning (ML). Puede abordar necesidades de inteligencia empresarial específicas mediante análisis de datos (como la predicción del resultado de decisiones empresariales, la generación de informes y paneles, la reducción de la ineficacia operativa y más).

¿Cómo funciona? La extracción, transformación y carga (ETL) funciona moviendo datos del sistema de origen al sistema de destino a intervalos periódicos. El proceso ETL funciona en tres pasos:

1. Extracción de los datos: Las herramientas de extracción, extracción, transformación y carga (ETL) de datos extraen o copian datos en bruto de múltiples fuentes y los almacenan en un área de ensayo. Un área de ensayo (o zona de aterrizaje) es un área de almacenamiento intermedio para almacenar temporalmente los datos extraídos. Las áreas de ensayo de datos suelen ser transitorias, lo que significa que su contenido se borra una vez que se completa la extracción de datos. Sin embargo, el área de ensayo también puede conservar un archivo de datos para fines de resolución de problemas.
2. Transformación de los datos: En la transformación de datos, las herramientas de extracción, transformación y carga (ETL) transforman y consolidan los datos en bruto en el área de preparación para prepararlos para el almacenamiento de datos de destino. La fase de transformación de datos puede implicar los siguientes tipos de cambios de datos.
3. Carga de los datos: En la carga de datos, las herramientas de extracción, transformación y carga (ETL) mueven los datos transformados desde el área de ensayo al almacenamiento de datos de destino. Para la mayoría de las organizaciones que usan ETL, el proceso está automatizado, bien definido, continuo y por lotes. A continuación se presentan dos métodos para cargar datos.

2. SELECT + INTO OUTFILE.

Escribe las filas resultantes en un archivo y permite el uso de terminadores de columna y fila para especificar un formato de salida específico. El valor predeterminado es terminar los campos con tabulaciones y las líneas con saltos de línea.

El archivo no debe existir. No se puede sobrescribir. El usuario necesita el privilegio FILE para ejecutar esta instrucción. Además, MariaDB necesita permiso para escribir archivos en la ubicación especificada. Si la variable de sistema `secure_file_priv` tiene un nombre de directorio válido, el archivo solo puede describirse en ese directorio. La LOAD DATA INFILE declaración complementa SELECT INTO OUTFILE.

3. LOAD DATA INFILE.

La sentencia LOAD DATA INFILE lee filas desde un fichero de texto a una tabla a gran velocidad. Si se especifica la palabra LOCAL, se interpreta con respecto al cliente final de la conexión. Cuando se especifica LOCAL, el fichero es leído por el programa del cliente en el ordenador cliente y se envía al servidor. Si no se especifica LOCAL, el fichero debe estar en el ordenador servidor y es leído directamente por el servidor. (LOCAL está disponible desde la versión 3.22.6 de MySQL.)

Por razones de seguridad, cuando se leen ficheros de texto del servidor, los ficheros deben residir en el directorio de la base de datos o tener acceso para todos. Además, para usar LOAD DATA INFILE en ficheros del servidor, se debe poseer el privilegio FILE en el ordenador servidor.

4. SELECT con tablas de dos bases de datos.

Para realizar una consulta SELECT que involucre tablas de dos bases de datos diferentes, se pueden utilizar varias estrategias dependiendo del sistema de gestión de bases de datos que esté utilizando. Una forma común es utilizar consultas distribuidas o federadas, que permiten acceder a datos en diferentes bases de datos como si estuvieran en una sola base de datos. Para realizar una consulta SELECT con INNER JOIN entre tablas de dos bases de datos diferentes, es necesario utilizar una referencia explícita a ambas bases de datos en la sintaxis SQL. Esto permite combinar datos de forma relacional, incluso si las tablas están almacenadas en esquemas o bases de datos separadas.

3. Herramientas empleadas

Describir qué herramientas se han utilizado...

1. El símbolo del sistema (CMD): Es una aplicación presente en los sistemas operativos Windows que funciona como un intérprete de línea de comandos. Su propósito es permitir la ejecución de comandos mediante una sintaxis específica, lo que facilita la administración del sistema y la ejecución de programas sin necesidad de una interfaz gráfica. Los comandos ingresados se procesan y ejecutan inmediatamente al presionar la tecla "Intro", lo que permite realizar diversas operaciones, como la gestión de archivos, la configuración de redes y la interacción con bases de datos.
2. MySQL Server: MySQL Workbench es una herramienta visual unificada para arquitectos de bases de datos, desarrolladores y administradores de bases de datos. MySQL Workbench proporciona modelado de datos, desarrollo de SQL y herramientas de administración integrales para configuración de servidores, administración de usuarios, copias de seguridad y mucho más. Permite diseñar, visualizar y mantener esquemas de bases de datos de manera eficiente.

4. Desarrollo

1. Respaldo de la GCS para crear los LCS

Primero se realiza un respaldo de las tablas que se ocuparan en cada uno de los nodos, se realizara desde el cmd. Posteriormente se modifica el scrip para que se pueda crear el nodo para cada base de datos sin sus registros, luego se restaura la base de datos desde el cmd

```
C:\>cd "C:\Program Files\MySQL\MySQL Server 8.0\bin"
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysqldump -u root -p sistemagestionflotillas flotilla vehiculo documento > C:\mysql\LCS1-Principal.sql
Enter password: ****
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysqldump -u root -p sistemagestionflotillas vehiculo mantenimiento > C:\mysql\LCS2-Mantenimiento.sql
Enter password: ****
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysqldump -u root -p sistemagestionflotillas vehiculo conductor ruta transaccionCombustible > C:\mysql\LCS3-Rutas.sql
Enter password: ****
```

```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p nodos < C:\mysql\LCS1-Principal.sql
Enter password: ****
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p nodos < C:\mysql\LCS2-Mantenimiento.sql
Enter password: ****
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p nodos < C:\mysql\LCS3-Rutas.sql
Enter password: ****
```

2. Extracción de los datos

Para extraer los datos del GCS se usó la sentencia SELECT + INTO OUTFILE para poder tener un archivo txt.

```
use sistemagestionflotillas ;
```

```
select * from flotilla into outfile '/mysql/flotilla.txt' FIELDS TERMINATED BY ','  
OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n';
```

```
select * from conductor into outfile '/mysql/conductor.txt' FIELDS TERMINATED BY ','  
OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n';
```


```
select * from documento into outfile '/mysql/documento.txt' FIELDS TERMINATED BY ','  
OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n';
```


```
select * from mantenimiento into outfile '/mysql/mantenimiento.txt'  
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n';
```

```
select * from ruta into outfile '/mysql/ruta.txt'  
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n';
```


```
select * from transaccioncombustible into outfile  
'/mysql/transaccioncombustible.txt'  
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n';
```

```
select * from vehiculo into outfile '/mysql/vehiculo.txt'  
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n';
```


 conductor

 documento

 flotilla

 mantenimiento

 ruta

 transaccioncombustible

 vehiculo

3. Carga de los datos

```
use lcs1_principal;
load data infile '/mysql/flotilla.txt' into table flotilla FIELDS TERMINATED BY ',' 
OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n';
load data infile '/mysql/vehiculo.txt' into table vehiculo FIELDS TERMINATED BY ',' 
OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n';
load data infile '/mysql/documento.txt' into table documento FIELDS TERMINATED BY ',' 
OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n';
```

```
SET FOREIGN_KEY_CHECKS = 0;
```

```
use lcs2_mantenimiento ;
load data infile '/mysql/vehiculo.txt' into table vehiculo FIELDS TERMINATED BY ',' 
OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n';
load data infile '/mysql/mantenimiento.txt' into table mantenimiento FIELDS TERMINATED BY ',' 
OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n';
```

```
use lcs3_rutas ;
load data infile '/mysql/vehiculo.txt' into table vehiculo FIELDS TERMINATED BY ',' 
OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n';
load data infile '/mysql/conductor.txt' into table conductor FIELDS TERMINATED BY ',' 
OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n';
load data infile '/mysql/ruta.txt' into table ruta FIELDS TERMINATED BY ',' 
OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n';
load data infile '/mysql/transaccioncombustible.txt' into table 
transaccioncombustible FIELDS TERMINATED BY ',' 
OPTIONALLY ENCLOSED BY '"' 
LINES TERMINATED BY '\n';
```

4.Consultas

1. En esta consulta use la LCS1-Principal y LCS2-Mantenimiento, para saber que vehiculos se encuentran en proceso de mantenimiento.

```
select A.vehiculoId ,A.tipo , B.estado , B.costo from 
lcs1_principal.documento A INNER JOIN lcs2_mantenimiento.mantenimiento B ON 
A.vehiculoId = B.vehiculoId where B.estado = 'En-Proceso' order by vehiculoId asc ;
```

	vehiculoId	tipo	estado	costo
▶	31	Permiso Carga	En Proceso	675.40
	33	Factura	En Proceso	3118.67
	33	Permiso Carga	En Proceso	3118.67
	40	Verificación	En Proceso	2438.76
	40	Seguro	En Proceso	2438.76
	40	Verificación	En Proceso	728.48
	40	Seguro	En Proceso	728.48
	40	Verificación	En Proceso	1190.05
	40	Seguro	En Proceso	1190.05
	45	Permiso Carga	En Proceso	903.91
	45	Verificación	En Proceso	903.91
	45	Factura	En Proceso	903.91
	45	Factura	En Proceso	903.91
	54	Tarjeta Circul...	En Proceso	1762.90
	60	Factura	En Proceso	1278.76
	60	Factura	En Proceso	3340.26
	61	Tarjeta Circul...	En Proceso	2329.08
	61	Verificación	En Proceso	2329.08

2. En esta consulta use la LCS1-Principal. LCS2-Mantenimiento y LCS3-Rutas

```
select A.vehiculoId , C.estado , A.rutaArchivo , B.ubicacionInicio , B.UbicacionFin
from lcs3_rutas.ruta B inner join lcs1_principal.documento A on
A.vehiculoId = B.vehiculoId inner join lcs2_mantenimiento.mantenimiento C on
C.vehiculoId = B.vehiculoId order by vehiculoId ;
```

	vehiculoId	estado	rutaArchivo	ubicacionInicio	UbicacionFin
►	2	Completado	/docs/2025/02/fb32b004-f2de-11ef-a17c-1a13...	Cancún, QR	Monterrey, NL
	2	Completado	/docs/2025/02/fb32b004-f2de-11ef-a17c-1a13...	Cancún, QR	Mérida, Yuc
	2	Completado	/docs/2025/02/fb32b004-f2de-11ef-a17c-1a13...	Cancún, QR	Ciudad de México
	2	Completado	/docs/2025/02/fb32b004-f2de-11ef-a17c-1a13...	Ciudad de México	Querétaro, Qro
	2	Completado	/docs/2025/02/fb32c260-f2de-11ef-a17c-1a13...	Cancún, QR	Monterrey, NL
	2	Completado	/docs/2025/02/fb32c260-f2de-11ef-a17c-1a13...	Cancún, QR	Mérida, Yuc
	2	Completado	/docs/2025/02/fb32c260-f2de-11ef-a17c-1a13...	Cancún, QR	Ciudad de México
	2	Completado	/docs/2025/02/fb32c260-f2de-11ef-a17c-1a13...	Ciudad de México	Querétaro, Qro
	2	Completado	/docs/2025/02/fb3244e8-f2de-11ef-a17c-1a13...	Cancún, QR	Monterrey, NL
	2	Completado	/docs/2025/02/fb3244e8-f2de-11ef-a17c-1a13...	Cancún, QR	Mérida, Yuc
	2	Completado	/docs/2025/02/fb3244e8-f2de-11ef-a17c-1a13...	Cancún, QR	Ciudad de México
	2	Completado	/docs/2025/02/fb3244e8-f2de-11ef-a17c-1a13...	Ciudad de México	Querétaro, Qro
	12	Completado	/docs/2025/02/fb323e3a-f2de-11ef-a17c-1a13...	Mérida, Yuc	Cancún, QR
	12	Completado	/docs/2025/02/fb326f72-f2de-11ef-a17c-1a13...	Mérida, Yuc	Cancún, QR
	12	Completado	/docs/2025/02/fb327c24-f2de-11ef-a17c-1a13...	Mérida, Yuc	Cancún, QR
	17	Pendiente	/docs/2025/02/fb3239f8-f2de-11ef-a17c-1a13...	Puebla, Pue	Guadalajara, Jali...
	17	Pendiente	/docs/2025/02/fb3239f8-f2de-11ef-a17c-1a13...	Hermosillo, Son	Pachuca, Hidalgo
	22	Pendiente	/docs/2025/02/fb327649-f2de-11ef-a17c-1a13...	Hermosillo, Son	Mérida, Yuc

5. Procedimientos Almacenados y Triggers

En esta parte se crean procedimientos y triggers para poder insertar, actualizar o borrar un registro

— *//// INSERT*

DELIMITER \$\$

```
create procedure registroVehiculo (
in vehid int ,
in flويد int ,
in tip varchar (80) ,
in model varchar (80) ,
in mar varchar (80) ,
in ani int ,
in est varchar (80) ,
in feve date
)
begin
insert into vehiculo ( vehiculoid , flotillaId , tipo , modelo , marca , anio , estado ,
fechaVerificacion )
values ( vehid , flويد , tip , model , mar , ani , est , feve );
end $$
DELIMITER ;
```

DELIMITER \$\$

```
create trigger insertLCS2 after insert on lcs1_principal.vehiculo for each row
begin
insert into lcs2_mantenimiento.vehiculo ( vehiculoid , flotillaId , tipo , modelo , marca
anio , estado , fechaVerificacion )
values (new.vehiculoid , new.flotillaId , new.tipo , new.modelo , new.marca , new.anio ,
new.estado , new.fechaVerificacion);
end $$
```

DELIMITER ;

DELIMITER \$\$

```
create trigger insertLCS3 after insert on lcs1_principal . vehiculo for each row
begin
insert into lcs3_rutas . vehiculo ( vehiculoid , flotillaid , tipo , modelo , marca , anio
estado , fechaVerificacion )
values ( new . vehiculoid , new . flotillaid , new . tipo , new . modelo , new . marca , ne
new . estado , new . fechaVerificacion );
end $$
DELIMITER ;
```

— ///// UPDATE

DELIMITER \$\$

```
CREATE PROCEDURE actualizarFechaVeri (
IN vehid INT ,
IN feve date
)
BEGIN
UPDATE lcs1_principal . vehiculo
SET
    fechaVerificacion = feve
WHERE vehiculoid = vehid ;
END $$
DELIMITER ;
```

DELIMITER \$\$

```
CREATE TRIGGER updateLCS2
AFTER UPDATE ON lcs1_principal . vehiculo
FOR EACH ROW
BEGIN
UPDATE lcs2_mantenimiento . vehiculo
SET
    fechaVerificacion = NEW . fechaVerificacion
WHERE vehiculoid = NEW . vehiculoid ;
END $$
DELIMITER ;
```

DELIMITER \$\$

```
CREATE TRIGGER updateLCS3
AFTER UPDATE ON lcs1_principal . vehiculo
FOR EACH ROW
BEGIN
UPDATE lcs3_rutas . vehiculo
SET
    fechaVerificacion = NEW . fechaVerificacion
WHERE vehiculoid = NEW . vehiculoid ;
END $$
DELIMITER ;
```

— ///// DELETE

DELIMITER \$\$

```
CREATE PROCEDURE eliminarVehiculo (
IN vehid INT
```

```

)
BEGIN
DELETE FROM lcs1_principal . vehiculo WHERE vehiculoid = vehid ;
END $$
DELIMITER ;

DELIMITER $$
CREATE TRIGGER deleteLCS2
AFTER DELETE ON lcs1_principal . vehiculo
FOR EACH ROW
BEGIN
DELETE FROM lcs2_mantenimiento . vehiculo WHERE vehiculoid = OLD . vehiculoid ;
END $$
DELIMITER ;

DELIMITER $$
CREATE TRIGGER deleteLCS3
AFTER DELETE ON lcs1_principal . vehiculo
FOR EACH ROW
BEGIN
DELETE FROM lcs3_rutas . vehiculo WHERE vehiculoid = OLD . vehiculoid ;
END $$
DELIMITER ;

```

5. Conclusiones

En conclusión, la implementación de los LCS es una estrategia útil para comprender y simular el manejo de bases de datos en distintos entornos, facilitando la distribución y organización de la información de manera eficiente. A través de este proceso, pude reforzar la importancia de estructurar adecuadamente los datos para garantizar su integridad y disponibilidad en distintos sistemas. Asimismo, el uso del proceso ETL demostró ser una metodología clave para la migración y transformación de datos entre bases de datos, asegurando que la información se transfiera de manera precisa y optimizada.

Referencias Bibliográficas

References

- [1] ¿Qué es ETL? - Explicación de extracción, transformación y carga (ETL) - AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/etl/:text=El>
- [2] SELECT INTO OUTFILE. (s. f.). MariaDB KnowledgeBase. <https://mariadb.com/kb/en/select-into-outfile/>
- [3] Pozo, S. (s. f.). Curso de MySQL: LOAD DATA. © 2000 Salvador Pozo. https://conclase.net/mysql/curso/mysql/LOAD_DATA