REPORTE DE PRÁCTICA NO. 2.4

NOMBRE DE LA PRÁCTICA: Práctica 3, BDD Flotillas
ALUMNO: Alexis Ortiz Jaén
DIRECTOR: Dr. Eduardo Cornejo-Velázquez

1. Introducción

En el ámbito de la gestión de bases de datos distribuidas, los Sistemas de Control de Localización (LCS) desempeñan un papel fundamental para garantizar el acceso eficiente y seguro a la información. La creciente necesidad de administrar datos en múltiples ubicaciones ha impulsado el desarrollo de técnicas que permiten la sincronización y actualización en tiempo real de los datos almacenados en diferentes nodos.
Esta práctica se centra en la creación e implementación de un LCS para la administración de bases de datos distribuidas, permitiendo la realización de consultas en múltiples fuentes de datos que almacenan información de manera
El propósito principal es demostrar cómo, mediante el uso de procedimientos almacenados y triggers, es posible garantiz la sincronización automática de los datos entre los distintos nodos.
La correcta implementación de estos mecanismos es crucial para mantener la integridad y coherencia de la información en un entorno distribuido. Al replicar de manera eficiente las modificaciones realizadas en el nodo principal del LCS hac se asegura un acceso confiable y optimizado a los datos, reduciendo la latencia y mejorando la experiencia del usuario en

2. Marco Teórico.

Proceso ETL.

Extracción, transformación y carga (ETL) es el proceso consistente en combinar datos de diferentes orígenes en 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, 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) la fase de carga de datos en el proceso de Extracción, Transformación y Carga (ETL) consiste en trasladar los dato Existen diversos métodos para realizar la carga de datos, los cuales dependen del sistema de almacenamiento y del

2. SELECT INTO OUTFILE.

La sentencia `SELECT INTO OUTFILE` se emplea para exportar los resultados de una consulta SQL a un archivo externo. E
Características y consideraciones:
Permite definir delimitadores personalizados para columnas y filas, facilitando la estructuración de los datos en formatos
Genera un archivo en la ubicación especificada, pero este no puede sobrescribirse si ya existe; es necesario eliminar el ar
Para su ejecución, el usuario debe contar con el privilegio `FILE`, lo que garantiza un nivel de seguridad en el acceso a los
La variable de sistema `secure_file_priv` restringe la escritura de archivos a un directorio específico, lo que refuerza la
Ejemplo de uso:
La instrucción `SELECT INTO OUTFILE` se utiliza para volcar los resultados de una consulta SQL directamente en un archi
Algunas de las principales características de esta función incluyen la posibilidad de personalizar los delimitadores de las
Es importante tener en cuenta que para ejecutar esta operación, el usuario debe contar con el privilegio de `FILE`, lo cual

3. LOAD DATA INFILE.

La sentencia `LOAD DATA INFILE` permite importar datos desde un archivo de texto a una tabla dentro de una base de da
Modos de operación:
Cuando se emplea sin la opción `LOCAL`, el archivo debe encontrarse en el servidor de bases de datos y será leído directam
Si se incluye la opción `LOCAL`, el archivo puede residir en el sistema del cliente, siendo transferido al servidor para su pro
(LOCAL está disponible desde la versión 3.22.6 de MySQL.)

4. Consultas SELECT entre tablas de diferentes bases de datos.

Cuando se trabaja con múltiples bases de datos, una de las tareas más comunes es realizar consultas que impliquen tablas ubicadas en diferentes bases de datos. Existen diversas estrategias para lograrlo, dependiendo del sistema de gestión de bases de datos (DBMS) utilizado. Una de las soluciones más utilizadas es la implementación de consultas federadas o distribuidas. Estas permiten acceder a datos que se encuentran en diferentes bases de datos como si estuvieran centralizados en una única base de
En el caso de ejecutar una consulta SQL que involucra un INNER JOIN entre tablas de diferentes bases de datos, es necesario especificar tanto el nombre de la base de datos como el nombre de la tabla, separados por un punto. Este enfoque asegura que los datos puedan combinarse de forma relacional, sin importar que las tablas estén ubicadas en bases de datos diferentes. Este tipo de consultas son muy útiles en escenarios donde se manejan bases de datos distribuidas o cuando se necesitan combinar datos de diferentes aplicaciones. Además, se pueden implementar soluciones como enlaces o vistas federadas.

5. Herramientas utilizadas

A continuación, se describen algunas de las principales herramientas utilizadas en este proyecto:

1. Símbolo del sistema (CMD): El Símbolo del sistema, o "Command Prompt" (CMD), es una aplicación incluida en sistemas operat Windows. El CMD funciona como un intérprete de comandos que permite ejecutar órdenes de texto para interactuar con el sistema operativo. A través del CMD, los usuarios pueden ejecutar comandos para gestionar archivos, configurar redes, interactuar c servidores y realizar consultas a bases de datos. Uno de los usos más comunes del CMD es para la administración y consulta de base datos mediante comandos SQL. Su uso permite una gestión más directa y rápida de tareas administrativas, lo que mejora la eficienci

2. MySQL Server: MySQL es uno de los sistemas de gestión de bases de datos más utilizados a nivel mundial. Su arquitectura permite gestionar grandes volúmenes de datos con alta eficiencia, siendo ideal tanto para pequeñas aplicaciones como MySQL Workbench es una herramienta complementaria que ofrece una interfaz gráfica para facilitar el diseño, administración y desa Con MySQL Workbench, los usuarios pueden diseñar esquemas de bases de datos, escribir consultas SQL, administrar usuarios y ser y realizar tareas de mantenimiento como la creación de copias de seguridad. Esta herramienta es esencial para optimizar el flujo de t

6. Desarrollo

1. Respaldo de la GCS para crear los LCS

Primero se realiza un respaldo de las tablas que se ocuparán en cada uno de los nodos, esto se realizará desde el CMD. Posteriormen se modificará el script para que se pueda crear el nodo para cada base de datos sin sus registros. Este proceso asegura que la estruct esté correctamente configurada en cada uno de los nodos de la base de datos.

```
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 utilizó la sentencia SELECT + INTO OUTFILE para generar un archivo .txt.

1. Uso de la base de datos "sistemagestionflotillas":

use sistemagestionflotillas;

2. Extracción de datos de la tabla "flotilla":

SELECT * FROM flotilla INTO OUTFILE '/mysql/flotilla.txt' FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n';

3. Extracción de datos de la tabla "conductor":

SELECT * FROM conductor INTO OUTFILE '/mysql/conductor.txt' FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n';

4. Extracción de datos de la tabla "documento":

SELECT * FROM documento INTO OUTFILE '/mysql/documento.txt' FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n';

5. Extracción de datos de la tabla "mantenimiento":

SELECT * FROM mantenimiento INTO OUTFILE '/mysql/mantenimiento.txt' FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n';

6. Extracción de datos de la tabla "ruta":

SELECT * FROM ruta INTO OUTFILE '/mysql/ruta.txt' FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"' LINES TERMINATED BY '\n';

7. Extracción de datos de la tabla "transaccion$_c$*ombustible*" :

SELECT * FROM transaccion$_c$*ombustibleINTOOUTF*

3. Carga de los datos

```
uselcs1principal;
loaddatainfile'/mysql/flotilla.txt'intotableflotillaFIELDSTERMINATE
OPTIONALLYENCLOSEDBY'"'LINESTERMINATEDBY'n';
loaddatainfile'/mysql/vehiculo.txt'intotablevehiculoFIELDSTERMINA
OPTIONALLYENCLOSEDBY'"'LINESTERMINATEDBY'n';
loaddatainfile'/mysql/documento.txt'intotabledocumentoFIELDSTERM
OPTIONALLYENCLOSEDBY'"'LINESTERMINATEDBY'n';

SETFOREIGN_KEY_CHECKS=0;

uselcs2mantenimiento;
loaddatainfile'/mysql/vehiculo.txt'intotablevehiculoFIELDSTERMINA
OPTIONALLYENCLOSEDBY'"'LINESTERMINATEDBY'n';
loaddatainfile'/mysql/mantenimiento.txt'intotablemantenimientoFIEL
BY','OPTIONALLYENCLOSEDBY'"'LINESTERMINATEDBY'n';

uselcs3rutas;
loaddatainfile'/mysql/vehiculo.txt'intotablevehiculoFIELDSTERMINA
OPTIONALLYENCLOSEDBY'"'LINESTERMINATEDBY'n';
loaddatainfile'/mysql/conductor.txt'intotableconductorFIELDSTERMI
OPTIONALLYENCLOSEDBY'"'LINESTERMINATEDBY'n';
loaddatainfile'/mysql/ruta.txt'intotablerutaFIELDSTERMINATEDBY'
OPTIONALLYENCLOSEDBY'"'LINESTERMINATEDBY'n';
loaddatainfile'/mysql/transaccioncombustible.txt'intotable
transaccioncombustibleFIELDSTERMINATEDBY','OPTIONALLYENCLO
LINESTERMINATEDBY'n';
```

4.Consultas

1. En esta consulta use la LCS1-Principal y LCS2-Man tenimien to, para sab er que v ehiculos se encuen tran en pro ceso de man tenimien to.

   ```
   selectA.vehiculoId,A.tipo,B.estado,B.costofrom
   lcs1principal.documentoAINNERJOINlcs2mantenimiento.mantenimientoBON
   A
   ```

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-Man tenimien to y LCS3-Rutas

```
select A.vehiculoId, C.estado, A.rutaArchivo, B.ubicacionInicio, B.Ub
fromlcs3rutas.rutaBinnerjoinlcs1principal.documentoAon
A.vehiculoId = B.vehiculoIdinnerjoinlcs2mantenimiento.mantenimientoCon
C
```

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
23	Pendiente	/docs/2025/02/fb327648-f2de-11ef-a17c-1a13...	Hermosillo, Son	Mérida, Yuc

5. Pro cedimien tos Almacenados y T riggers

En esta parte se crean pro c edi m ien tos y triggers para p o der insertar, actualizar o b orr ar un registro

```
////INSERT
DELIMITER$$
createprocedureregistroVehiculo(
invehidint,
infloidint,
intipvarchar(80),
inmodelvarchar(80),
inmarvarchar(80),
inaniint,
inestvarchar(80),
infevedate
)
begin
insertintovehiculo(vehiculoid,flotillaid,tipo,modelo,marca,anio,estad
fechaVerificacion)
values(vehid,floid,tip,model,mar,ani,est,feve);
end$$
DELIMITER;

DELIMITER$$
createtriggerinsertLCS2afterinsertonlcs1principal.vehiculoforeachrow
begin
insertintolcs2mantenimiento.vehiculo(vehiculoid,flotillaid,tipo,modelo
anio,estado,fechaVerificacion)
values(new.vehiculoid,new.flotillaid,new.tipo,new.modelo,new.marca,
new.estado,new.fechaVerificacion);
end$$
```

```
DELIMITER;

DELIMITER$$
createtriggerinsertLCS3afterinsertonprincipal.vehiculoforeachrow
begin
insertintolcs_3rutas.vehiculo(vehiculoid,flotillaid,tipo,modelo,marca,a
estado,fechaVerificacion)
values(new.vehiculoid,new.flotillaid,new.tipo,new.modelo,new.marca,
new.estado,new.fechaVerificacion);
end$$
DELIMITER;

 /////UPDATE
DELIMITER$$
CREATEPROCEDUREactualizarFechaVeri(
INvehidINT,
INfevedate
)
BEGIN
UPDATElcpsr1incipal.vehiculo
SET
fechaVerificacion=feve
WHEREvehiculoid=vehid;
END$$
DELIMITER;

DELIMITER$$
CREATETRIGGERupdateLCS2
AFTERUPDATEOnNrlmsclipal.vehiculo
FOREACHROW
BEGIN
UPDATElcmsa2ntenimiento.vehiculo
SET
fechaVerificacion=NEW.fechaVerificacion
WHEREvehiculoid=NEW.vehiculoid;
END$$
DELIMITER;

DELIMITER$$
CREATETRIGGERupdateLCS3
AFTERUPDATEOnNrlmsclipal.vehiculo
FOREACHROW
BEGIN
UPDATElcrsu3tas.vehiculo
SET
fechaVerificacion=NEW.fechaVerificacion
WHEREvehiculoid=NEW.vehiculoid;
END$$
DELIMITER;

 /////DELETE
DELIMITER$$
CREATEPROCEDUREeliminarVehiculo(
INvehidINT
```

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

DELIMITER$$
CREATE TRIGGER deleteLCS2
AFTER DELETE ON principal.vehiculo
FOREACH ROW
BEGIN
DELETE FROM mantenimiento.vehiculo WHERE vehiculoid = OLD.vehiculoid
END$$
DELIMITER;

DELIMITER$$
CREATE TRIGGER deleteLCS3
AFTER DELETE ON principal.vehiculo
FOREACH ROW
BEGIN
DELETE FROM rutas.vehiculo WHERE vehiculoid = OLD.vehiculoid;
END$$
DELIMITER;
```

5. Conclusiones

En conclusión, la implementación de los LCS (Lenguajes de Consulta Estructurados) es fundamental para una adecuada
La correcta estructuración de los datos es esencial para garantizar la integridad y disponibilidad de la información, lo qu
Asimismo, el proceso ETL (Extracción, Transformación y Carga) se destaca como una metodología esencial para la migr
Con el crecimiento continuo de la cantidad de datos generados y la complejidad de los sistemas, tecnologías emergentes
Finalmente, los triggers (disparadores) juegan un papel crucial en la gestión automatizada de bases de datos. Estos meca
El futuro de la gestión de bases de datos estará marcado por una integración cada vez más profunda de estos procesos a

Referencias Bibliográficas

References

[1] ¿Qué es ETL? - Explicación de extracción, transformación y carga (ETL) - AWS. (s. f.). Amazon Web Services, In

[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

[4] Kimball, R., Ross, M. (2013). The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. Wile

[5] Inmon, W. H. (2005). Building the Data Warehouse (4th ed.). Wiley.

[6] Heath, C. (2007). Database Triggers: An Introduction. O'Reilly Media.

[7]