

REPORTE DE PRÁCTICA NO. 2.1

PRACTICA FRAGMENTOS

ALUMNO:

Morales Vázquez Rubén



1. Introducción

En este reporte de practica implementaremos la fragmentación horizontal y la fragmentación vertical mediante vistas en MySQL, utilizando una base de datos que restauramos mediante código en consola y con registros de 100 datos, estaremos implementando cinco registros de fragmentación horizontal y cinco fragmentos de vertical.

2. Herramientas empleadas

Las herramientas que fueron necesarias para la realización de esta practica fueron las siguientes:

1. Biblioteca digital. Plataforma web donde se encuentran libros y artículos donde podemos encontrar información.
2. MySQL Workbench. Es la aplicación donde logramos establecer la base de datos en lenguaje SQL.
3. Online LaTeX Editor Overleaf. Es un sitio web en el cual podemos generar nuestros documentos.

3. Marco teórico

Fragmentación Horizontal

La fragmentación horizontal es una técnica de diseño de bases de datos que consiste en dividir una tabla en fragmentos (o particiones) basados en filas. Cada fragmento contiene un subconjunto de las filas de la tabla original, pero todas las filas tienen la misma estructura (es decir, las mismas columnas). Esto se hace para mejorar el rendimiento, la escalabilidad y la gestión de grandes volúmenes de datos.

En la fragmentación horizontal, las filas de una tabla se dividen en grupos según un criterio específico.

- Por rangos: Dividir las filas según un rango de valores (por ejemplo, una tabla de clientes se puede dividir por rangos de ID o fechas).
- Por listas: Dividir las filas según una lista de valores (por ejemplo, dividir clientes por región o país).
- Por hash: Aplicar una función hash a una columna para distribuir las filas de manera uniforme entre los fragmentos.

Sintaxis de un Procedimiento Almacenado

```
1 CREATE VIEW nombre_vista AS
2 SELECT * FROM nombre_tabla
3 WHERE condicion;
4
```

Listing 1: Ejemplo de código MySQL

3.1.1 Ejemplos

1. Fragmentación por estado en la tabla Vehiculo:

- Álgebra relacional:

$$\sigma_{\text{estado}='Activo'}(\text{Vehiculo})$$

- MySQL:

```
1 CREATE VIEW vehiculos_activos AS
2 SELECT * FROM Vehiculo
3 WHERE estado = 'Activo';
4
```

2. Fragmentación por año en la tabla Vehiculo:

- Álgebra relacional:

$$\sigma_{\text{anio}=2020}(\text{Vehiculo})$$

- MySQL:

```
1 CREATE VIEW vehiculos_anio_2020 AS
2 SELECT * FROM Vehiculo
3 WHERE anio = 2020;
4
```

3. Fragmentación por tipo de servicio en la tabla Mantenimiento:

- Álgebra relacional:

$$\sigma_{\text{tipoServicio}='Preventivo'}(\text{Mantenimiento})$$

- MySQL:

```
1 CREATE VIEW mantenimiento_preventivo AS
2 SELECT * FROM Mantenimiento
3 WHERE tipoServicio = 'Preventivo';
4
```

4. Fragmentación por estado en la tabla Ruta:

- Álgebra relacional:

$$\sigma_{\text{estado}='Pendiente'}(\text{Ruta})$$

- MySQL:

```
1 CREATE VIEW rutas_pendientes AS
2 SELECT * FROM Ruta
3 WHERE estado = 'Pendiente';
4
```

5. Fragmentación por tipo de combustible en la tabla TransaccionCombustible:

- Álgebra relacional:

$$\sigma_{\text{tipoCombustible}='Gasolina'}(\text{TransaccionCombustible})$$

- MySQL:

```
1 CREATE VIEW transacciones_gasolina AS
2 SELECT * FROM TransaccionCombustible
3 WHERE tipoCombustible = 'Gasolina';
4
```

Fragmentación Vertical

La fragmentación vertical es una técnica de diseño de bases de datos que consiste en dividir una tabla en fragmentos (o subconjuntos) basados en columnas. En lugar de dividir las filas (como en la fragmentación horizontal), aquí se separan las columnas de una tabla en varias tablas más pequeñas. Cada fragmento contiene un subconjunto de las columnas de la tabla original, pero todas las filas.

La fragmentación vertical se usa principalmente para:

1. Mejorar el rendimiento: Al reducir el tamaño de las tablas, las consultas que acceden a solo algunas columnas son más rápidas.
2. Optimizar el almacenamiento: Si algunas columnas se usan con menos frecuencia, se pueden separar para ahorrar espacio.
3. Seguridad: Puedes separar columnas sensibles en una tabla diferente y controlar el acceso a ellas.

Sintaxis Fragmentación Vertical

```
1 CREATE VIEW nombre_vista AS
2 SELECT columna1, columna2, ...
3 FROM nombre_tabla;
4
```

Listing 2: Ejemplo de código MySQL

3.2.1 Ejemplos

1. Fragmentación de la tabla Vehiculo:

- Álgebra relacional:

$$\pi_{\text{vehiculoId, flotillaId, tipo, modelo, marca}}(\text{Vehiculo})$$

- MySQL:

```
1 CREATE VIEW vehiculos_info_basica AS
2 SELECT vehiculoId, flotillaId, tipo, modelo, marca FROM Vehiculo;
3
```

2. Fragmentación de la tabla Conductor:

- Álgebra relacional:

$$\pi_{\text{conductorId, nombre, estado}}(\text{Conductor})$$

- MySQL:

```
1 CREATE VIEW conductores_info_personal AS
2 SELECT conductorId, nombre, estado FROM Conductor;
3
```

3. Fragmentación de la tabla Documento:

- Álgebra relacional:

$$\pi_{\text{documentoId, vehiculoId, tipo, fechaVencimiento, estado}}(\text{Documento})$$

- MySQL:

```
1 CREATE VIEW documentos_info_basica AS
2 SELECT documentoId, vehiculoId, tipo, fechaVencimiento, estado FROM Documento;
3
```

4. Fragmentación de la tabla Mantenimiento:

- Álgebra relacional:

$\pi_{\text{mantenimientoId, vehiculoId, fechaServicio, tipoServicio, descripcion}}(\text{Mantenimiento})$

- MySQL:

```
1 CREATE VIEW mantenimiento_info_servicio AS
2 SELECT mantenimientoId, vehiculoId, fechaServicio, tipoServicio, descripcion FROM
   Mantenimiento;
3
```

5. Fragmentación de la tabla Ruta:

- Álgebra relacional:

$\pi_{\text{rutaId, vehiculoId, horaInicio, horaFin, distancia, ubicacionInicio, ubicacionFin}}(\text{Ruta})$

- MySQL:

```
1 CREATE VIEW rutas_info_ruta AS
2 SELECT rutaId, vehiculoId, horaInicio, horaFin, distancia, ubicacionInicio,
   ubicacionFin FROM Ruta;
3
```

4. Conclusiones

En esta práctica, desarrollamos habilidades clave en el manejo de bases de datos, específicamente en la aplicación de técnicas de fragmentación horizontal y fragmentación vertical. Utilizamos como base la base de datos de flotilla, la cual restauramos mediante la consola. Esta base de datos ya contenía alrededor de cien registros de información, lo que nos permitió trabajar con datos reales y significativos para llevar a cabo nuestras fragmentaciones.

References

- [1] Giménez, J. A. (2019). *Buenas prácticas en el diseño de bases de datos*. ARANDU UTIC, 6(1), 193–210.
- [2] Mosquera Palacios, D. J., & Wanumen Silva, L. F. (2018). *Arquitectura para la generación de consultas SQL usando lógica de conjuntos*. Visión Electrónica, 2, 307–318.
- [3] Date, C. J. (2004). *An Introduction to Database Systems 8th ed Addison-Wesley*
- [4] Elmasri, R. Navathe, S. B. (2016). *Fundamentals of Database Systems 11th ed Pearson*