Proyecto UF1470 - Administración de SGBD

Autor: Juan de la Morena
Cliente de base de datos: DBeaver
Entorno: Windows 10

■ Proyecto UF1470 - Administración de Sistemas Gestores de Bases de Datos

Autor: Juan de la Morena
Cliente de base de datos: DBeaver
Entorno: Windows 10
Bases de datos: MySQL y SQLite
Repositorio: [GitHub](https://github.com/tuusuario/tu-repo) *(actualiza el enlace)*

■ Objetivo

Este proyecto simula una prueba práctica integral como DBA (Administrador de Base de Datos) para la empresa ficticia **DataSolutions S.A.**, abordando:
- Creación y configuración de bases de datos
- Gestión de usuarios y privilegios
- Inserción y validación de datos
- Automatización de tareas
- Seguridad e integridad de la información
- Optimización del rendimiento
- Auditoría de cambios

■■ Estructura del Repositorio

UF1470_Juan_delaMorena/
■
■■■ scripts/
■ ■■■ mysql/ # Scripts SQL para MySQL
■ ■■■ sqlite/ # Scripts SQL para SQLite
■
■■■ tareas_programadas/ # Script de copia de seguridad automatizada (Windows)
■
■■■ docs/
■ ■■■ capturas_dbeaver/ # Evidencias visuales del trabajo realizado
■ ■■■ informe_final.pdf # Documento completo del proyecto
■
■■■ README.md # Este archivo

■ Descripción por Secciones

■ Parte 1 - MySQL

1. **Creación de Base de Datos:**

Se crea una base de datos llamada `DataSolutionsDB` para la empresa ficticia **DataSolutions S.A.**.
CREATE DATABASE DataSolutionsDB;

2. **Gestión de Usuarios:**
Se crean dos usuarios en MySQL:
- `consultor`: con permisos de solo lectura en la base de datos.
- `admin_ventas`: con permisos de lectura, inserción y actualización en la tabla `clientes`.
CREATE USER 'consultor'@'localhost' IDENTIFIED BY 'TuContraseña123';
GRANT SELECT ON DataSolutionsDB.* TO 'consultor'@'localhost';
CREATE USER 'admin_ventas'@'localhost' IDENTIFIED BY 'TuContraseña123';
GRANT SELECT, INSERT, UPDATE ON DataSolutionsDB.clientes TO 'admin_ventas'@'localhost';

3. **Optimización de Consultas:**
Se recibe la consulta original:
SELECT * FROM clientes WHERE ciudad = 'Madrid' AND fecha_registro > '2024-01-01';

Se analizó el plan de ejecución de esta consulta y se determinó que sería eficiente crear un índice compuesto para las columnas `ciudad` y `fecha_registro`, de modo que la base de datos pueda realizar una búsqueda más rápida. Esto mejora el rendimiento al hacer las búsquedas más eficientes.
CREATE INDEX idx_ciudad_fecha ON clientes(ciudad, fecha_registro);

Problema encontrado:
Sin un índice adecuado, la consulta puede tardar más tiempo en ejecutarse, especialmente si la tabla tiene muchos registros. El uso de índices acelera la búsqueda al reducir la cantidad de datos que deben procesarse.
4. **Gestión de Procesos:**
Se identifican consultas que consumen muchos recursos en MySQL utilizando el siguiente comando:
SHOW PROCESSLIST;

Para finalizar un proceso que esté utilizando demasiados recursos, se utiliza el comando `KILL` con el ID del proceso correspondiente:
KILL 1234;

■ Parte 2 - SQLite

1. **Creación de la Base de Datos:**
Se crea una base de datos SQLite llamada `clientes.db`.
CREATE TABLE clientes (
id INTEGER PRIMARY KEY,
nombre TEXT,
apellido TEXT,
ciudad TEXT,
fecha_registro DATE
);

2. **Inserción de Datos:**
Se insertan al menos cinco registros en la tabla `clientes`:
INSERT INTO clientes (nombre, apellido, ciudad, fecha_registro) VALUES
('Juan', 'Pérez', 'Madrid', '2024-02-15'),
('Ana', 'López', 'Barcelona', '2023-11-03'),
('Luis', 'Martínez', 'Sevilla', '2024-01-21'),
('Clara', 'Ramírez', 'Valencia', '2022-09-12'),
('Mario', 'Gómez', 'Madrid', '2024-03-01');

3. **Trigger de Auditoría:**
Se implementa un trigger en SQLite que registra cualquier actualización en la tabla `clientes` en una tabla de auditoría llamada `log_clientes`:

```
CREATE TABLE log_clientes (
fecha_modificacion TEXT,
usuario TEXT,
operacion TEXT,
datos_antiguos TEXT,
datos_nuevos TEXT
);
CREATE TRIGGER trigger_log_update
AFTER UPDATE ON clientes
BEGIN
INSERT INTO log_clientes (
fecha_modificacion,
usuario,
operacion,
datos_antiguos,
datos_nuevos
)
VALUES (
datetime('now'),
'usuario_windows',
'UPDATE',
json_object('id', OLD.id, 'nombre', OLD.nombre, 'apellido', OLD.apellido, 'ciudad', OLD.ciudad,
'fecha_registro', OLD.fecha_registro),
json_object('id', NEW.id, NEW.nombre, NEW.apellido, NEW.ciudad, NEW.fecha_registro)
);
END;
```

■ Parte 3 - Automatización y Seguridad

1. **Copia de Seguridad:**
Se crea un script SQL para realizar una copia de seguridad de la tabla `clientes` creando una nueva tabla llamada `clientes_backup`:

```
CREATE TABLE clientes_backup AS SELECT * FROM clientes;
```

2. **Procedimiento Almacenado:**
Se crea un procedimiento almacenado en MySQL que valida que la fecha de registro del cliente no sea futura:

```
DELIMITER //
CREATE PROCEDURE insertar_cliente (
IN p_nombre VARCHAR(50),
IN p_apellido VARCHAR(50),
IN p_ciudad VARCHAR(50),
IN p_fecha DATE
)
BEGIN
IF p_fecha > CURDATE() THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'La fecha de registro no puede ser futura.';
```

```
ELSE
INSERT INTO clientes (nombre, apellido, ciudad, fecha_registro)
VALUES (p_nombre, p_apellido, p_ciudad, p_fecha);
END IF;
END //
DELIMITER ;
```

■ Parte 4 - Planificación de Tareas

1. **Planificación de Copias de Seguridad Automáticas:**
Se crea un script `.bat` para hacer backups automáticos en Windows utilizando `mysqldump`:

```
@echo off
set FECHA=%DATE:~6,4%-%DATE:~3,2%-%DATE:~0,2%
mysqldump -u root -pTuContraseña DataSolutionsDB >
"C:\backups\DataSolutionsDB_%FECHA%.sql"
```

2. **Monitoreo del Rendimiento:**
Se recomienda usar herramientas como **MySQL Workbench**, **Percona Toolkit**, y **Nagios/Zabbix** para monitorear el rendimiento de la base de datos y detectar cuellos de botella. Las consultas `SHOW STATUS` y `SHOW PROCESSLIST` también son útiles para monitorear en tiempo real.

■■ Capturas

Se incluyen capturas de cada paso realizado con **DBeaver**, para verificar la ejecución y el resultado de cada script.

■ Informe PDF

El informe en formato PDF documenta cada paso realizado, con código y explicación teórica. Convertido desde Markdown con extensión `Markdown PDF` en **VSCode**.

■ Reflexiones Finales

Este proyecto me ha permitido aplicar conocimientos de:
- Modelado y gestión de bases de datos
- Seguridad y control de accesos
- Automatización en entorno Windows
- Auditoría de cambios y validación de integridad

Además, he mejorado mi documentación técnica y organización de proyectos en GitHub.

■■ Herramientas Utilizadas

- DBeaver (https://dbeaver.io/)
- MySQL Server (https://dev.mysql.com/downloads/mysql/)
- SQLite3 (https://www.sqlite.org/download.html)
- Visual Studio Code (https://code.visualstudio.com/)
- Markdown PDF (extensión de VSCode)