



Universidad de sonora (UNISON)

Facultad de ingeniería

Profesor. Rene Francisco Navarro Hernández

Base de datos I

Caso de estudio de un SMDB: SQLite

Ingeniería en sistemas de información

Elias Antonio Luna Anaya

Expediente: 224204681

Correo: a22420681@unison.mx

26 de agosto de 2025

1. Realizar una búsqueda en Internet para definir la funcionalidad de SQLite. Por ejemplo, la entrada asociada al tópico de Wikipedia. Se puede consultar también el sitio de SQLite.

SQLite es una biblioteca en proceso que implementa un motor de base de datos SQL transaccional autónomo, sin servidor y sin configuración. El código de SQLite es de dominio público y, por lo tanto, su uso es gratuito para cualquier propósito, ya sea comercial o privado. SQLite es la base de datos más implementada del mundo, con incontables aplicaciones, incluyendo varios proyectos de alto perfil.

SQLite es un motor de base de datos SQL integrado. A diferencia de la mayoría de las bases de datos SQL, SQLite no cuenta con un proceso de servidor independiente. SQLite lee y escribe directamente en archivos de disco comunes. Una base de datos SQL completa, con múltiples tablas, índices, disparadores y vistas, se encuentra en un único archivo de disco. El formato de archivo de la base de datos es multiplataforma: se puede copiar libremente una base de datos entre sistemas de 32 y 64 bits, o entre arquitecturas big-endian y little-endian. Estas características hacen de SQLite una opción popular como formato de archivo de aplicación. Los archivos de base de datos SQLite son un formato de almacenamiento recomendado por la Biblioteca del Congreso de EE. UU. SQLite no debe considerarse un sustituto de Oracle, sino de fopen().

SQLite es una biblioteca compacta. Con todas las funciones habilitadas, su [tamaño](#) puede ser inferior a 750 KiB, dependiendo de la plataforma de destino y la configuración de optimización del compilador. (El código de 64 bits es más grande. Además, algunas optimizaciones del compilador, como la inserción agresiva de funciones y el desenrollado de bucles, pueden hacer que el código objeto sea mucho más grande). Existe un equilibrio entre el uso de memoria y la velocidad. SQLite generalmente se ejecuta más rápido cuanto mayor sea la memoria asignada. Sin embargo, el rendimiento suele ser bastante bueno incluso en entornos con poca memoria. Dependiendo de cómo se use, SQLite puede ser más rápido que la E/S directa del sistema de archivos .

2. **Formar un equipo de máximo para intercambiar opiniones acerca del tema. Puedes utilizar los siguientes criterios para guiar la discusión.**
3. **¿Cuáles de las ventajas/desventajas de los sistemas de bases de datos se ofrecen en SQLite?**

Ventajas:

- **Control de redundancia:** Se pueden usar llaves primarias y configuraciones para no repetir datos y cometer posibles errores.
- **Coherencia de datos:** Cumple con las transacciones ACID (atomicidad, consistencia, aislamiento y durabilidad)
- **Más información a partir de mismos datos:** Permite comandos como SELECT, JOIN, VIEW, etc.
- **Integridad de datos:** Permite restricciones como PRIMARY KEY, UNIQUE, CHECK, FOREIGN KEY, etc.
- **Imposición de estándares:** Cumple con la mayor parte del estándar SQL-92.
- **Portabilidad y economía:** La base de datos se encuentre en un solo archivo, con un peso menor a 600 kb.
- **Mantenimiento sencillo:** No requiere instalación ni configuración; copia de respaldo sencilla.
- **Productividad y rapidez en desarrollo:** Se integra fácilmente en aplicaciones móviles o embebidas (sistema de gestión de bases de datos que se integra directamente en una aplicación, funcionando como una librería o un archivo dentro del código fuente, en lugar de ser un servicio independiente.)
- **Accesibilidad y respuesta:** Lee rápidamente lecturas locales, pues no depende de un servidor intermedio.
- **Servicios de recuperación:** Con rollback y WAL, permite recuperar datos tras un fallo.

Desventajas

- **Compartición de datos:** Solo permite un escritor a la vez.
- **Concurrencia:** Su concurrencia es baja, no es apta para el trabajo de múltiples usuarios.
- **Seguridad:** No tiene sistema propio de usuarios/roles ni control de acceso; depende de permisos del sistema operativo.
- **Servicios de autorización:** No tiene servicios de autorización, no hay privilegios granulares como en MySQL/PostgreSQL.
- **Copias de respaldo avanzadas:** Existen, pero no al nivel de sistemas cliente-servidor.
- **Escalabilidad:** No está diseñada para bases de datos muy grandes ni aplicaciones con miles de usuarios concurrentes.
- **Arquitectura ANSI-SPARC:** No implementa formalmente la separación en tres niveles (externo, conceptual, interno).

4. ¿Cuáles de las funciones de un sistema manejador de bases de datos se ofrecen en SQLite? ¿Cuáles están ausentes?

Funciones presentes

- **Almacenamiento, recuperación y actualización de los datos:** Permite crear, modificar, consultar y eliminar registros fácilmente mediante SQL estándar.
- **Catálogo accesible por el usuario:** Mediante las tablas del sistema (sqlite_master y otras), se puede consultar la estructura de la base de datos, como tablas, índices, triggers y vistas.
- **Soporte de transacciones:** Implementa transacciones completas con propiedades ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad), asegurando integridad en cada operación.
- **Servicios de recuperación:** Ofrece mecanismos como rollback y Write-Ahead Logging (WAL) que permiten recuperar datos ante fallos o interrupciones.
- **Servicios de integridad:** Soporta restricciones de integridad como PRIMARY KEY, UNIQUE, CHECK y FOREIGN KEY.

Funciones ausentes

- **Servicios de control de concurrencia:** Aunque permite múltiples lectores simultáneamente, solo admite un escritor a la vez, lo que lo hace poco adecuado para aplicaciones con alta concurrencia.
- **Servicios de autorización:** No tiene un sistema interno de usuarios ni de roles con privilegios diferenciados. El control de acceso depende únicamente de los permisos del sistema operativo sobre el archivo de base de datos.

Conclusiones generales

En base a mi investigación, puedo decir que SQLite si se puede considerar como una SMBD, sin embargo, no es tan robusta ni tan útil como otras SMBD. Se trata de una versión mas simple y sencilla que tiene algunos defectos como no contar con concurrencia (múltiples usuarios modificando la base de datos), no contar con un sistema de usuarios y roles, etc. Puedo destacar que es muy ligera (sus archivos pesan menos de 600 kb), es facil de usar e integrar, y rápida para consultas locales, tiene soporte para transacciones ACID y todo se guarda en un solo archivo.

Es una SMBD para proyectos pequeños y trabajos locales, pero me parece que cuenta con las características y funciones necesarias para ser útil para trabajar y practicar con nuestros propios trabajos, cumple con su propósito como SMBD ligera y confiable.

}

Fuentes de información:

- *About SQLite*. (s. f.). <https://www.sqlite.org/about.html>

- *Defense against the dark arts.* (s. f.). <https://www.sqlite.org/security.html>
- *Features of SQLite.* (s. f.). <https://www.sqlite.org/features.html>
- *CREATE TABLE.* (s. f.). https://www.sqlite.org/lang_createtable.html
- *Write-Ahead logging.* (s. f.). <https://www.sqlite.org/wal.html>