**UNIVERSIDAD DE SONORA**

Base de Datos 1

Prof: Navarro Hernández René Francisco

**Caso de estudio de un SMBD: SQLite**

Ingeniería en sistemas de información.

Integrantes:

Natalia Guadalupe Sánchez Valenzuela.

Sandoval López Daniela

Wu Zhang Fei Fei

*26/08/25 Hermosillo, Son.*

**SQLite: Un estudio sobre su funcionalidad y clasificación como SGBD.**

SQLite es una biblioteca de software que implementa un motor de base de datos SQL pequeño, rápido, autónomo, de alta fiabilidad y con todas las funciones. A diferencia de otros sistemas de gestión de bases de datos (SGBD) como MySQL o PostgreSQL, SQLite no es un proceso de servidor separado. En su lugar, se incrusta en la aplicación a la que se utiliza. La base de datos se almacena en un solo archivo en el disco duro.

### **Argumentos a favor**

-Sin configuración: Como no hay un proceso de servidor, no requiere configuración ni administración. La base de datos está lista para usarse tan pronto como la biblioteca se integre en la aplicación. Esto lo hace ideal para aplicaciones móviles y de escritorio.

-Portabilidad: La base de datos se almacena en un solo archivo de disco. Esto facilita el traslado de la base de datos entre diferentes sistemas operativos y plataformas.

-Alto rendimiento: Para cargas de trabajo que no son intensivas en escritura y no requieren concurrencia de múltiples usuarios, SQLite puede ser más rápido que los SGBD basados en servidor debido a la reducción de la latencia de la comunicación cliente-servidor.

### **Argumentos en contra**

-Control de concurrencia limitado: SQLite está diseñado para un acceso de usuario único. Aunque puede manejar múltiples lectores simultáneamente, las operaciones de escritura bloquean toda la base de datos, lo que lo hace inadecuado para aplicaciones que requieren un alto grado de concurrencia de escritura.

-Escalabilidad: No es apto para aplicaciones web de alto tráfico. Si bien puede manejar un sitio web pequeño, un sitio con muchos usuarios concurrentes que realizan múltiples operaciones de escritura en la base de datos se vería afectado negativamente.

-Seguridad: La seguridad de la base de datos es responsabilidad de la aplicación que la utiliza. A diferencia de los SGBD cliente-servidor que tienen mecanismos de autenticación y autorización incorporados, la protección de los datos en SQLite depende de la seguridad del sistema de archivos y la propia aplicación.

## **¿Es un Sistema Manejador de Bases de Datos?**

Un SGBD es un sistema de software que permite a los usuarios definir, crear, mantener y controlar el acceso a la base de datos. SQLite cumple con todas estas características:

-Define y crea: Permite la creación de esquemas y tablas.

-Mantiene: Ofrece comandos para insertar, actualizar y eliminar datos.

-Controla el acceso: A través de las operaciones SQL, gestiona cómo las aplicaciones interactúan con los datos.

-Lenguaje de consulta: Utiliza una versión simplificada del lenguaje SQL estándar.

Aunque su arquitectura es diferente a la de los SGBD cliente-servidor tradicionales, su funcionalidad principal de gestionar datos lo clasifica dentro de esta categoría.

## **Funcionalidades de SQLite**

SQLite, a pesar de su tamaño, ofrece una gran variedad de funcionalidades. Las principales son:

-Transacciones ACID: Garantiza la atomicidad, consistencia, aislamiento y durabilidad en las transacciones de la base de datos. Esto significa que las operaciones de la base de datos son confiables y protegen la integridad de los datos, incluso si ocurre un fallo del sistema.

-Soporte SQL: Implementa una gran parte del estándar SQL92, incluyendo consultas complejas, de tablas, subconsultas y funciones agregadas.

-Bases de datos en memoria: Permite crear una base de datos temporal completamente en la memoria RAM, lo que es útil para realizar operaciones rápidas con datos que no necesitan ser persistentes.

-Tipos de datos dinámicos: A diferencia de otros SGBD que aplican un tipado estricto, las columnas de SQLite pueden almacenar cualquier tipo de dato, independientemente del tipo declarado, lo que ofrece una gran flexibilidad.

-Vistas e índices: Permite la creación de vistas para simplificar consultas complejas y de índices para mejorar el rendimiento de la búsqueda.

-Funciones definidas por el usuario: Los desarrolladores pueden crear sus propias funciones personalizadas para extender la funcionalidad de SQL.

## **Ventajas**

-Control de redundancia: La capacidad de normalizar tablas ayuda a reducir la redundancia de datos. Sin embargo, el control depende completamente del diseño del desarrollador, ya que no hay un control centralizado.

-Coherencia de datos: SQLite mantiene la coherencia de los datos a través de su soporte para restricciones de integridad (claves primarias, foráneas, etc.) y transacciones ACID.

-Mayor información: Permite consultas complejas, JOINs, vistas y funciones de agregación, lo que facilita la extracción de información significativa a partir de los datos almacenados.

-Mayor integridad: El motor de la base de datos soporta claves primarias, foráneas, restricciones UNIQUE, NOT NULL y CHECK, lo que asegura la integridad de los datos.

-Imposición de estándares: Se adhiere a la mayor parte del estándar SQL-92, facilitando la portabilidad y el aprendizaje.

-Economía de escala: Al ser gratuito, de código abierto y de bajo consumo, es una solución muy económica, especialmente para aplicaciones en dispositivos con recursos limitados.

-Mejor accesibilidad y capacidad de respuesta: Como no hay un servidor intermediario, el acceso directo al archivo de la base de datos es extremadamente rápido, lo que mejora la capacidad de respuesta.

-Mayor productividad: No requiere configuración ni administración, lo que simplifica su integración y acelera el desarrollo de aplicaciones.

Mantenimiento sencillo: La separación entre el esquema de la base de datos y el código de la aplicación (independencia de datos) facilita el mantenimiento.

## **Desventajas**

-Compartición de datos (Limitado): Aunque varios procesos pueden leer el mismo archivo, solo un proceso de escritura puede acceder a la base de datos a la vez, lo que lo hace inadecuado para entornos de alta concurrencia.

Mayor seguridad (Limitado): No tiene un sistema de usuarios, autenticación o roles. La seguridad depende por completo de los permisos del sistema de archivos, por lo que no es apto para entornos con múltiples usuarios con diferentes privilegios.

-Equilibrio entre requerimientos conflictivos (Limitado): Su diseño está optimizado para la simplicidad y el bajo consumo de recursos, no para equilibrar escenarios de múltiples escritores de forma eficiente.

Mayor nivel de concurrencia (Limitado): Solo permite una transacción de escritura concurrente a la vez, lo que es su mayor limitación en entornos multiusuario.

-Servicios de copia de respaldo y recuperación (Limitado): La base de datos es un solo archivo, lo que facilita la copia manual. Soporta WAL (Write-Ahead Logging) para la recuperación, pero carece de herramientas avanzadas como logs archivados o replicación automática.

# **Funciones de un SGBD en SQLite**

-Almacenamiento, recuperación y actualización de datos: Funcionalidad completa. Permite operaciones básicas (INSERT, SELECT, UPDATE, DELETE) de manera eficiente.

-Catálogo accesible por el usuario: Presente. Mantiene metadatos en tablas internas como sqlite\_master y sqlite\_schema, que son directamente consultables por los desarrolladores.

Soporte de transacciones: Completo. Soporta transacciones ACID (BEGIN, COMMIT, ROLLBACK), lo que asegura la fiabilidad de los datos.

-Servicios de integridad: Completo. Impone restricciones de integridad como claves primarias y foráneas, y reglas de validación.

-Servicios de recuperación: Parcial. Usa mecanismos como journaling y WAL para recuperarse de fallos, pero carece de funcionalidades más avanzadas de los sistemas de bases de datos de servidor.

## **Funciones Ausentes o Limitadas en SQLite**

-Servicios de control de concurrencia: Limitado. Usa un bloqueo a nivel de base de datos que permite un solo escritor a la vez, lo que restringe el uso en aplicaciones de alta concurrencia.

-Servicios de autorización: Ausente. No hay un sistema de seguridad de usuarios, roles o permisos a nivel de la base de datos.

-Compartición de datos: Limitado. No está diseñado para entornos de alta concurrencia de escritura.

# **Análisis de la Arquitectura ANSI-SPARC en SQLite**

La arquitectura ANSI-SPARC es un modelo de tres niveles (externo, conceptual e interno) que busca la independencia de los datos. SQLite, a pesar de su diseño sin servidor, se adhiere a este modelo.

-Nivel Interno (Físico): Representado por la estructura del archivo único .db en el disco. La biblioteca de SQLite maneja la organización de las páginas, los índices y el almacenamiento físico de los datos.

-Nivel Conceptual (Lógico): Se corresponde con el esquema de la base de datos, que define las tablas, relaciones y restricciones mediante sentencias SQL (CREATE TABLE).

-Nivel Externo (Vistas): Se logra a través de las vistas (VIEWs) y las consultas SQL (SELECT). Permite que las aplicaciones o usuarios tengan una vista específica de los datos sin conocer el esquema completo.

**Conclusión Personal**

SQLite es un SGBD poderoso y sofisticado que demuestra que la esencia de la gestión de datos no reside en una arquitectura compleja, sino en la capacidad de garantizar la fiabilidad, consistencia y acceso a la información de manera eficiente, objetivos que cumple de manera excepcional dentro de su nicho de aplicación diseñado.

***Referencias***

[*https://www-sqlite-org.translate.goog/about.html?\_x\_tr\_sl=en&\_x\_tr\_tl=es&\_x\_tr\_hl=es&\_x\_tr\_pto=tc*](https://www-sqlite-org.translate.goog/about.html?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc)

[*https://ecohosting.cl/ventajas-y-desventajas-de-sqlite/3437*](https://ecohosting.cl/ventajas-y-desventajas-de-sqlite/3437)

[*https://www.reddit.com/r/AskProgramming/comments/1fw7fgl/the\_advantages\_disadvantages\_of\_sqlite\_vs\_other/?tl=es-419*](https://www.reddit.com/r/AskProgramming/comments/1fw7fgl/the_advantages_disadvantages_of_sqlite_vs_other/?tl=es-419)

[*https://www.reddit.com/r/AskProgramming/comments/1fw7fgl/the\_advantages\_disadvantages\_of\_sqlite\_vs\_other/?tl=es-419*](https://www.reddit.com/r/AskProgramming/comments/1fw7fgl/the_advantages_disadvantages_of_sqlite_vs_other/?tl=es-419)

[*https://www-geeksforgeeks-org.translate.goog/dbms/the-three-level-ansi-sparc-architecture/?\_x\_tr\_sl=en&\_x\_tr\_tl=es&\_x\_tr\_hl=es&\_x\_tr\_pto=tc*](https://www-geeksforgeeks-org.translate.goog/dbms/the-three-level-ansi-sparc-architecture/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc)

[*https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/sqlite*](https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/sqlite)