



UNIVERSIDAD DE SONORA

"El Saber de mis Hijos hará mi Grandeza"

Carrera: Ingeniería en Sistemas de Información

Tarea: SQLite

Materia: Base De Datos I

Maestro: René Francisco Navarro

Nombre: Daniela Sandoval López

Expediente: 224204236

Fecha: 23/08/2025

¿Qué es?

SQLite es una herramienta de software libre, que permite almacenar información en dispositivos empujados de una forma sencilla, eficaz, potente, rápida y en equipos con pocas capacidades de hardware, como puede ser una PDA o un teléfono celular. SQLite implementa el estándar SQL92 y también agrega extensiones que facilitan su uso en cualquier ambiente de desarrollo. Esto permite que SQLite soporte desde las consultas más básicas hasta las más complejas del lenguaje SQL, y lo más importante es que se puede usar tanto en dispositivos móviles como en sistemas de escritorio, sin necesidad de realizar procesos complejos de importación y exportación de datos, ya que existe compatibilidad al 100% entre las diversas plataformas disponibles, haciendo que la portabilidad entre dispositivos y plataformas sea transparente.

Características

Estas son algunas de las características principales de SQLite:

- La base de datos completa se encuentra en un solo archivo.
- Puede funcionar enteramente en memoria, lo que la hace muy rápida.
- Tiene un footprint menor a 230KB.
- Es totalmente autocontenida (sin dependencias externas).
- Cuenta con librerías de acceso para muchos lenguajes de programación.
- Soporta texto en formato UTF-8 y UTF-16, así como datos numéricos de 64 bits.
- Soporta funciones SQL definidas por el usuario (UDF).
- El código fuente es de dominio público y se encuentra muy bien documentado.

Plataformas donde funciona

SQLite está construida en C, lo cual facilita la migración a diversas plataformas de sistemas operativos y de dispositivos.

Las plataformas principales dónde SQLite se encuentra funcionando son:

- Windows 95, 98, ME, 2000, XP y Vista
- Windows CE & Pocket PC
- Mac OSX
- Linux
- OpenEmbedded
- PalmOS
- Symbian

Las características y plataformas previamente mencionadas hacen de SQLite una excelente opción en diversos casos tales como:

- Cuando se requiere una base de datos integrada dentro de una aplicación. SQLite es una excelente opción por su facilidad de configuración. El inconveniente es que no escala a bases de datos demasiado grandes.
- Para realizar demostración de aplicaciones que utilizan un RDBMS.
- Como cache local de un manejador de base de datos empresarial. Esto acelera el tiempo de respuesta y reduce la carga sobre la base de datos central.
- Para aplicaciones en dispositivos móviles que manejan una BD local que se sincroniza por batch con una base de datos remota.
- Almacenamiento persistente de objetos, configuraciones y preferencias de usuario. Permite fácilmente crear una estructura para almacenar configuraciones de la aplicación.

Ventajas y desventajas

- Ventajas

1. Simplicidad:

- No requiere instalación ni configuración de servidor.
- Basta con incluir la librería en la aplicación.

2. Ligereza:

- Muy pequeño en tamaño (menos de 1 MB).
- Ideal para aplicaciones móviles, de escritorio y sistemas embebidos.

3. Compatibilidad con SQL:

- Implementa gran parte del estándar SQL (DML, DDL, integridad referencial con claves foráneas, transacciones).

4. Transacciones ACID:

- Garantiza consistencia e integridad en operaciones múltiples.

5. Portabilidad:

- Toda la base de datos se guarda en un solo archivo .db que se puede copiar, mover o respaldar fácilmente.

6. Velocidad en escenarios pequeños/medianos:

- Es más rápido que otros RDBMS porque no hay sobrecarga de red ni de procesos de servidor.

- Desventajas

1. Escalabilidad limitada:

- No está diseñado para aplicaciones con alto volumen de datos ni millones de conexiones concurrentes.
- Típicamente funciona bien hasta unas decenas de GB de datos.

2. Concurrencia restringida:

- Usa bloqueo a nivel de archivo, lo que limita la escritura simultánea.

3. Funciones avanzadas limitadas:

- No tiene características de servidores completos como:
 - Usuarios/roles y control de acceso complejo.
 - Replicación y clustering.
 - Procedimientos almacenados complejos.

4. Seguridad dependiente de la aplicación:

- No maneja autenticación de usuarios ni cifrado nativo.
- La seguridad debe implementarla el sistema que lo usa.

5. Soporte parcial del estándar SQL:

- No soporta algunas funcionalidades más avanzadas

Funciones

- Las que ofrece SQLite

1. Definición de datos (DDL)

- Permite crear, modificar y eliminar tablas, índices, vistas, etc.

2. Manipulación de datos (DML)

- Permite insertar, actualizar, eliminar y consultar datos.

3. Consulta de datos (SQL)

- Implementa gran parte del estándar SQL.

4. Transacciones ACID

- Garantiza atomicidad, consistencia, aislamiento y durabilidad.
- Soporta BEGIN, COMMIT y ROLLBACK.

5. Integridad referencial

- Maneja claves primarias y foráneas.

6. Independencia de datos

- La aplicación accede a los datos mediante SQL, no directamente al archivo.

7. Manejo de concurrencia (básico)

- Permite múltiples lecturas simultáneas y una sola escritura a la vez.

8. Portabilidad y almacenamiento centralizado

- La base de datos entera se guarda en un único archivo .db, portable entre sistemas.

- **Las que ofrece de manera limitada**

1. Manejo de usuarios y seguridad interna

- No maneja cuentas de usuario, roles ni contraseña.
- La seguridad debe implementarse en la aplicación o en el sistema operativo.

2. Optimización avanzada

- Tiene un motor de optimización de consultas, pero menos sofisticado que en servidores grandes.

3. Concurrencia avanzada

- No tiene bloqueo a nivel de fila o múltiples escritores simultáneos.
- En aplicaciones con muchas escrituras concurrentes puede ser un cuello de botella.

- **Las que no ofrece**

1. Administración multiusuario nativa

- No maneja múltiples usuarios con permisos diferentes dentro de la base.

2. Replicación y clustering

- No ofrece replicación automática ni alta disponibilidad como otros RDBMS.

3. Procedimientos almacenados y triggers avanzados

- Soporta triggers, pero no procedimientos almacenados ni funciones definidas por el usuario en SQL puro.

Arquitectura de tres niveles ANSI-SPARC

La arquitectura ANSI-SPARC de tres niveles (también llamada modelo de tres esquemas) define como un SGBD idealmente debería organizar los datos y su acceso:

1. Nivel interno (físico) → cómo se almacenan realmente los datos en disco.
2. Nivel conceptual (lógico global) → estructura lógica de toda la base (tablas, relaciones, restricciones).
3. Nivel externo (vistas de usuario) → cómo los distintos usuarios ven partes específicas de los datos (vistas, reportes, etc.).

SQLite sí implementa los tres niveles en lo esencial (interno, conceptual, externo), pero de manera simplificada. No aprovecha al 100% la independencia de datos ni la gestión multiusuario, por lo que no se apega completamente al modelo ANSI-SPARC, aunque sigue siendo un SGBD relacional válido.