

Caso de estudio de un SMDB: SQLite

¿Cuáles de las ventajas/desventajas de los sistemas de bases de datos se ofrecen en SQLite?

- Control de la redundancia de los datos:
Sí, evita duplicidad gracias a claves primarias y restricciones.
- Coherencia de los datos:
Sí, maneja restricciones (PRIMARY KEY, UNIQUE, FOREIGN KEY) y transacciones ACID que ayudan a mantener coherencia.
- Más información a partir de la misma cantidad de datos:
Sí, permite usar consultas SQL complejas (joins, subconsultas) para obtener nueva información.
- Compartición de los datos:
No, puede compartirse entre procesos, pero no es multiusuario a gran escala
- Mayor integridad de los datos:
Sí, soporta restricciones de integridad y tipos de datos, aunque de forma más flexible que otros motores
- Mayor seguridad:
No, SQLite no tiene gestión de usuarios ni permisos avanzados. La seguridad depende del sistema operativo.
- Imposición de estándares:
En parte sí, implementa buena parte de SQL, pero no el estándar completo ej. carece de ciertas funciones avanzadas.
- Economía de escala:
No, está diseñado para aplicaciones pequeñas/medianas, no para cargas enormes ni multiusuario.
- Equilibrio entre requerimientos conflictivos:
No mucho, está optimizado para simplicidad, pero no equilibra grandes volúmenes de datos ni alta concurrencia.
- Mejor accesibilidad a los datos y mayor capacidad de respuesta:
Sí, muy rápido en aplicaciones pequeñas, acceso directo al archivo.
- Mayor productividad:
Sí, fácil de usar, no requiere configuración de servidor.
- Mantenimiento más sencillo gracias a la independencia de los datos:
Sí, el archivo .db es portable y fácil de respaldar.
- Mayor nivel de concurrencia:
No mucho, soporta lecturas concurrentes, pero una sola escritura a la vez.

- Servicios mejorados de copia de respaldo y recuperación:
No mucho tampoco, tiene utilidades .backup, .dump pero no ofrece recuperación automática tras fallos graves como los motores cliente-servidor.

¿Cuáles de las funciones de un sistema manejador de bases de datos se ofrecen en SQLite?

¿Cuáles están ausentes?

- Almacenamiento, recuperación y actualización de los datos:
Si se encuentra, maneja todo esto dentro de un solo archivo.
- Un catálogo accesible por el usuario:
Se encuentra, usa la tabla especial sqlite_master para describir tablas, índices, vistas y triggers.
- Soporte de transacciones:
Si esta, es transaccional (ACID) y soporta Begin, Commit, Rollback.
- Servicios de recuperación:
Algo limitado, puede recuperar consistencia tras fallos menores, pero no tiene mecanismos avanzados de replicación o journaling distribuido como PostgreSQL.
- Servicios de control de concurrencia:
Si, pero básicos, usa bloqueo de archivos (file locking). Permite múltiples lectores, pero solo un escritor a la vez.
- Servicios de autorización:
No hay, no maneja usuarios ni roles. La seguridad depende del sistema operativo.
- Servicios de integridad:
Si tiene, soporta PRIMARY KEY, FOREIGN KEY, UNIQUE, CHECK y NOT NULL.

¿Se apega SQLite a la arquitectura de tres niveles ANSI-SPARC?

Si se apega, pero tiene algunos detalles. Si cumple con el nivel interno, maneja como se almacenan los datos en el archivo, también cumple con el nivel conceptual, porque el usuario puede definir tablas, relaciones y restricciones, y por último, para el nivel externo, soporta vistas "Create View", pero no maneja diferentes esquemas de usuario ni vistas personalizadas de seguridad.

CONCLUSIÓN

Pudimos comprobar que SQLite cumple con los requisitos indispensables para un sistema manejador de base de datos, descubrimos que esta muy bien estructurado todo y que es sencillo, accesible y ligero, por lo que es bueno trabajar con proyectos pequeños, la única desventaja es que no está pensado para proyectos demandantes o multiusuario.

Entre las cosas que pude rescatar de la interacción en equipo son las siguientes fortalezas y limitaciones:

Sus principales fortalezas serian:

- Ligero y sin necesidad de servidor.
- Transacciones ACID.
- Alta portabilidad (un solo archivo).
- Soporte de SQL estándar.
- Fácil integración y mantenimiento.

Mientras que en lo que te limita es:

- Falta de control de acceso y autenticación.
- Baja concurrencia de escritura.
- Ausencia de usuarios y roles.
- Recuperación y respaldo básicos.

La mayoría de estas ya las aclaré en este documento, pero me agradó como lo organizó el compañero.