Engineer

SQLite

Una base de datos completa en un solo archivo

Qué es

SQLite es un motor de bases de datos SQL que se ejecuta dentro del programa que lo utiliza, como una biblioteca, sin proceso servidor, puerto de red ni administrador. Mantiene sus tablas, índices, disparadores y vistas juntos en un archivo corriente del disco. El nombre también aparece como `sqlite3` : así se llama la herramienta de línea de órdenes y ese es el prefijo de todas las funciones de su interfaz C. El único archivo fuente C que distribuye el proyecto se llama `sqlite3.c` . El propio proyecto lo resume como autónomo, sin servidor, sin configuración y transaccional, y pide que se compare con `fopen()` en lugar de con Oracle. Lo que sustituye es el archivo de datos creado a medida, no el servidor de base de datos.

El proyecto comenzó el 9 de mayo de 2000 y el código es de dominio público, por lo que no hay que gestionar ninguna licencia antes de incluirlo en un producto. El formato de archivo también es una de sus ventajas. Una base de datos escrita en una máquina de 32 bits se puede copiar a una de 64 bits, o entre procesadores big-endian y little-endian, y abrirla sin cambios. Sus desarrolladores dicen que pretenden mantener SQLite hasta el año 2050, y la Biblioteca del Congreso de Estados Unidos incluye sus archivos entre los formatos recomendados para conservar datos. Con todas las funciones activadas, la biblioteca puede ocupar menos de 900 KiB. Por eso se encuentra en teléfonos, coches, relojes y aplicaciones de escritorio, además de en servidores.

Por dentro es un motor SQL completo: operaciones JOIN, subconsultas, vistas, disparadores y transacciones que siguen siendo atómicas, coherentes, aisladas y duraderas incluso si se corta la corriente en mitad de una escritura. Dos decisiones de diseño lo distinguen de las bases de datos con servidor. La primera es la concurrencia. Cualquier número de conexiones puede leer a la vez, pero solo una puede escribir en cada instante, y el bloqueo afecta al archivo entero. La

segunda es el tipado. El tipo declarado de una columna es una preferencia, que SQLite llama afinidad, y el tipo pertenece realmente a cada valor. Por tanto, una columna declarada `INTEGER` acepta texto que no parece un número.

La distinción frente a las alternativas es clara. PostgreSQL, MySQL y SQL Server son servicios compartidos con los que muchas máquinas se comunican por red. SQLite es un almacén privado de una aplicación en un equipo. Frente a los archivos CSV o JSON, añade un lenguaje de consulta, índices, restricciones y transacciones, a cambio de una dependencia que la mayoría de los lenguajes ya incluyen.

Utilice el concepto en una decisión

Aplique «¿Qué es SQLite?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Una estructura que refleja cómo funciona el negocio para poder responder también a las preguntas que surjan más adelante.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. Diseño y operación de 6 flujos ETL/ELT — 6 flujos ETL en tres motores de bases de datos

Diseño y operación de 6 flujos ETL/ELT Póngase en contacto →

Por qué importa

En buena parte del software, la pregunta difícil sobre una base de datos no es lo rápida que resulta, sino quién la opera. Un servidor necesita instalación, actualizaciones, protección, supervisión y copias de seguridad, y hay que pagar a alguien para que se encargue de todo ello. SQLite elimina toda esa categoría de trabajo. Abrir una base de datos consiste en abrir un archivo; una copia de seguridad es una copia de ese archivo, realizada mediante la API de copias mientras está en uso; y una batería de pruebas puede dar a cada prueba su propia base de datos nueva.

También cambia lo que puede ser un archivo. Un programa que, de otro modo, inventaría un formato, escribiría un analizador y confiaría en que nunca se produzca una caída a mitad de guardado puede conservar su estado en SQLite y obtener guardados atómicos sin trabajo adicional. El proyecto menciona expresamente estos usos: formato de archivo de una aplicación; contenedor para trasladar datos entre sistemas, donde el receptor ejecuta SQL en lugar de escribir un analizador; y base de datos temporal para ordenar y resumir datos que llegaron con otra estructura.

La decisión importante es concreta y se puede comprobar antes de escribir una línea de código: ¿cuántos procesos escriben y en cuántas máquinas? Un programa en un equipo, con ráfagas breves de escritura, es el terreno natural de SQLite, y muchos sitios de tráfico bajo o medio lo utilizan sin dificultad. Varios servidores que escriben sobre los mismos datos, o una carga en la que muchos usuarios escriben en todo momento, corresponden a una base de datos cliente-servidor. El tamaño rara vez es lo decisivo, ya que un archivo puede contener hasta 281 terabytes. Lo decisivo es el número de escritores.

Incluso en ese terreno hay costes que valorar. `ALTER TABLE` solo permite renombrar una tabla o una columna y añadir o eliminar una columna. La mayoría de los demás cambios de estructura exigen crear una tabla nueva, copiar las filas e intercambiar los nombres. Las claves foráneas están desactivadas hasta que cada conexión las activa. Y el tipado flexible deja entrar datos incorrectos sin avisar, salvo que la tabla se declare de modo que los rechace.

Cómo se aplica en la práctica

En la fase de desarrollo documentada, veintitrés tablas de un archivo SQLite, desde logros y empresas hasta perfiles y referencias, eran la fuente de verdad de un generador de currículums, hojas de referencias, portafolios y cartas de presentación. Una orden convertía un archivo SQLite vacío en esa base de datos mediante diecinueve pasos ordenados. Como cada paso podía ejecutarse dos veces de forma segura, reconstruir sobre una base ya existente no añadía filas duplicadas. Los seis formatos de salida procedían de 41 módulos Python con 10.580 líneas que leían ese único archivo, con exactamente dos dependencias en tiempo de ejecución. Un documento dirigido a un tipo de lector era una consulta sobre los logros, elegida por un perfil, en lugar de una copia editada. El precio era claro: el sistema se había convertido en la única manera de producir un documento.

En la fase de exportación documentada, la misma base de datos también alimentaba la generación de este sitio: de ella se obtenía su portafolio. Un exportador abría el archivo SQLite en modo de solo lectura y escribía 495 páginas en tres idiomas usando únicamente la biblioteca estándar de Python, y la salida se guardaba en el repositorio para que el sitio pudiera generarse sin el entorno del generador. Las direcciones de las páginas se almacenaban como filas, una por dirección y por logro, de modo que reformular una frase ya no cambiaba de lugar una página. Ese trabajo también descubrió la trampa de la sección siguiente: una comparación de periodos contra una columna de texto coincidía con las treinta y dos filas.

El uso más modesto, examinar resultados, también es el más habitual. Una canalización que convertía lecturas eléctricas en bruto en consumo horario por zona escribía sus resultados como CSV o en una base de datos SQLite ligera para su inspección, y la versión Python procesaba los datos de un día en 34 ms y los de treinta días en 180 ms. Una base de datos usada así no necesita configuración: cualquiera que tenga la herramienta `sqlite3` puede abrir el archivo y consultarlo.

Dónde surgen los problemas

La mayoría de los problemas con SQLite proceden de tratarlo como un PostgreSQL pequeño. Es una máquina distinta, con límites diferentes.

Un filtro por año coincide con todas las filas. SQLite convierte un valor antes de compararlo solo cuando uno de los lados tiene afinidad. El resultado de una función como `substr()` no la tiene, ni tampoco un número sin más. Por eso SQLite compara el texto `'2019'` con el entero `2025` tal como están, y en su ordenación todos los números preceden a todos los textos:

```
CREATE TABLE stint (period TEXT, hours TEXT);
INSERT INTO stint VALUES
  ('2019-03', '9'), ('2024-11', '10'), ('2026-01', '42');

SELECT count(*) FROM stint
WHERE substr(period, 1, 4) >= 2025;                  -- 3

SELECT count(*) FROM stint
WHERE CAST(substr(period, 1, 4) AS INTEGER) >= 2025;  -- 1
```

Haga conversiones explícitas siempre que el texto se encuentre con un número. Para tablas nuevas, declare tipos de datos concretos y añada la palabra clave `STRICT`, disponible desde la versión 3.37.0: una columna `INTEGER` estricta sigue aceptando `'42'` y lo almacena como número, pero rechaza `'ten'` con un error en lugar de guardarlo.

Las escrituras empiezan a fallar con «database is locked». Una conexión mantiene el bloqueo de escritura y un segundo escritor espera o abandona de inmediato. La causa suele ser una transacción larga o un lector que retiene a un escritor en el modo predeterminado de registro de reversión. Cambie la base de datos al registro de escritura anticipada, donde los lectores y el escritor dejan de

bloquearse entre sí, mantenga cortas las transacciones de escritura y configure un tiempo de espera ante bloqueos para que el escritor espere brevemente en lugar de producir un error. Si de verdad hay muchos escritores constantemente, es la señal para pasar a un servidor.

El archivo se corrompe en una unidad compartida. SQLite depende de los bloqueos de archivos del sistema operativo, y muchos sistemas de archivos en red los implementan con errores, de modo que dos máquinas pueden acabar escribiendo a la vez en la misma parte del archivo. El registro de escritura anticipada no funciona en absoluto sobre un sistema de archivos en red, porque todos los procesos que usan la base de datos tienen que compartir memoria en un equipo. Mantenga la base de datos en un disco local del equipo donde se ejecuta el único programa que escribe en ella. Si otras máquinas necesitan los datos, coloque un servidor delante.

Una fila sobrevive al padre al que apunta. Las restricciones de clave foránea se declaran en el esquema, pero siguen inactivas hasta que cada conexión ejecuta `PRAGMA foreign_keys = ON`. Por tanto, una eliminación en una conexión que omitió ese pragma deja registros huérfanos sin que se avise. Establezca el pragma en el único punto donde la aplicación abre conexiones y compruebe que esté activado.

Lecturas adicionales

- Acerca de SQLite — La descripción del propio proyecto: ejecución dentro del proceso, sin servidor, dominio público y un archivo multiplataforma.
- Tipos de datos en SQLite — Clases de almacenamiento, afinidad de tipos y las reglas de comparación que explican la trampa anterior.
- Tablas STRICT — Cómo hacer que una tabla aplique los tipos declarados de sus columnas.
- Registro de escritura anticipada — Lectores y un escritor en paralelo, y por qué se mantiene en un equipo.
- Usos adecuados de SQLite — Dónde encaja, dónde conviene más una base de datos cliente-servidor y una lista de comprobación.
- Compatibilidad de SQLite con claves foráneas — Por qué las restricciones están desactivadas de forma predeterminada y cómo activarlas.

Glosario

Elija según sus restricciones reales

Traiga un conjunto de datos representativo, las preguntas que debe responder y el cuello de botella actual. Podemos acordar cómo medir la calidad, la capacidad de procesamiento y la recuperación.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es SQLite?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/sqlite/