Engineer

SQLite

ціла база даних в одному файлі

Що це таке

SQLite — це рушій SQL-бази даних, який працює як бібліотека всередині програми, що ним користується: без серверного процесу, мережевого порту й адміністратора, а таблиці, індекси, тригери та представлення лежать разом в одному звичайному файлі на диску. Назва трапляється й у формі `sqlite3`: так звуть консольну оболонку, так названо єдиний файл із сирцями на C, `sqlite3.c`, і з `sqlite3_` починається кожна функція її C-інтерфейсу. Сам проєкт описує рушій як самодостатній, безсерверний, без конфігурації та транзакційний і радить порівнювати його з `fopen()`, а не з Oracle. Він заміняє саморобний файл даних, а не сервер бази даних.

Проєкт стартував 9 травня 2000 року, а код перебуває в суспільному надбанні (public domain), тож перед вбудовуванням у продукт не треба розбиратися з ліцензією. Формат файлу — частина домовленості: базу, записану на 32-бітній машині, можна скопіювати на 64-бітну або між процесорами big-endian і little-endian та відкрити без змін. Розробники мають намір підтримувати SQLite до 2050 року, а Бібліотека Конгресу США рекомендує цей формат для зберігання даних. З усіма увімкненими можливостями бібліотека може займати менше ніж 900KiB, тому вона живе в телефонах, автомобілях, годинниках і настільних програмах так само, як і на серверах.

Усередині це повноцінний SQL: з’єднання, підзапити, представлення, тригери й транзакції з властивостями ACID, які зберігаються навіть тоді, коли живлення зникає посеред запису. Від серверних баз SQLite відрізняють два рішення. Перше стосується паралельної роботи: читати одночасно може скільки завгодно з’єднань, а писати в кожну мить — лише одне, і блокування охоплює весь файл. Друге стосується типів. Оголошений тип стовпчика —

лише побажання, яке SQLite називає спорідненістю (affinity), а справжній тип належить кожному окремому значенню. Тож стовпчик, оголошений як `INTEGER`, охоче прийме текст, геть не схожий на число.

Межа з альтернативами чітка. PostgreSQL, MySQL і SQL Server — спільні сервіси, до яких багато машин звертаються мережею. SQLite — приватне сховище однієї програми на одній машині. Порівняно з файлами CSV чи JSON SQLite додає мову запитів, індекси, обмеження й транзакції ціною однієї залежності, яку більшість мов програмування вже мають у комплекті.

Застосуйте поняття до рішення

Застосуйте «Що таке SQLite?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Структура, що відповідає реальній роботі бізнесу, тож потрібні згодом запитання до даних усе ще можна буде поставити.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Спроєктувала й підтримувала 6 пайплайнів ETL/ELT — 6 пайплайнів ETL на трьох рушіях баз даних

Спроєктувала й підтримувала 6 пайплайнів ETL/ELT Зв'язатися з нами →

Чому це важливо

Найважче питання про базу даних часто стосується не швидкості, а того, хто її обслуговуватиме. Сервер треба встановити, оновлювати, захищати, моніторити й резервувати, і комусь за це платять. SQLite прибирає цю статтю витрат цілком. Відкрити базу — означає відкрити файл, резервна копія — це копія файлу, знята через backup API навіть під час роботи, а набір тестів може дати кожному тесту власну свіжу базу.

SQLite змінює й уявлення про те, чим може бути файл. Програма, яка інакше вигадувала б власний формат, писала б до нього парсер і сподівалася, що збій не трапиться посеред збереження, може тримати свій стан у SQLite й отримати атомарне збереження задарма. Проєкт сам називає такі сценарії: формат файлу застосунку, контейнер для перенесення даних між системами, де отримувач пише SQL замість парсера, і тимчасова база для сортування та підсумовування даних, що прийшли в іншій формі.

Вибір зводиться до одного питання, на яке можна відповісти ще до першого рядка коду: скільки процесів пишуть і на скількох машинах? Одна програма на одній машині, що пише короткими сплесками, — рідна територія SQLite, і більшість сайтів із низьким чи помірним трафіком почуваються на ній добре.

Кілька серверів, що пишуть в одні й ті самі дані, або багато користувачів, які пишуть щомиті, — це справа клієнт-серверної бази. Розмір рідко щось вирішує, адже один файл уміщує до 281 терабайта. Вирішує кількість тих, хто пише.

Навіть на рідній території є ціна. `ALTER TABLE` уміє лише перейменувати таблицю чи стовпчик і додати або видалити стовпчик, тож більшість інших змін структури означає нову таблицю, копіювання рядків і обмін назвами. Зовнішні ключі вимкнено, доки кожне з'єднання не ввімкне їх саме. А вільна типізація тихо пропускає хибні дані, якщо таблицю не оголошено так, щоб вона їх відкидала.

Як це виглядає на практиці

На задокументованому етапі розробки SQLite був джерелом істини для генератора резюме, аркушів із рекомендаціями, портфоліо та супровідних листів: усе бралося з бази SQLite на 23 таблиці, від досягнень і компаній до профілів і рекомендацій, яку збирав конвеєр із 19 кроків: від порожнього файлу до повної бази однією командою. Кожен крок можна було безпечно повторити, тож конвеєр проганяли й по наявній базі, не дублюючи жодного рядка. З цього одного файлу 41 модуль Python загальним обсягом 10 580 рядків видавав шість форматів, і залежностей під час виконання було рівно дві. Документ для певного читача був запитом до досягнень, заданим профілем, а не відредагованою копією. Ціну сторінка теж називала: система стала єдиним способом отримати документ.

На задокументованому етапі експорту портфоліо цього сайту генерувалося з тієї ж бази, і тут SQLite був вхідними даними для збирання. Експортер відкривав файл SQLite лише для читання й писав 495 сторінок трьома мовами, користуючись тільки стандартною бібліотекою Python, а результат зберігався в репозиторії, тож сайт збирався без середовища генератора. Адреси сторінок зберігалися як рядки, по одному на адресу на досягнення, тож переформульоване речення більше не пересувало сторінку. Під час тієї ж роботи зафіксували пастку з наступного розділу: порівняння періоду з текстовим стовпчиком зачепило всі тридцять два рядки.

Найменший і найтиповіший випадок — SQLite як місце, де переглядають результати. Конвеєр, що перетворював сирі показники електроенергії на погодинне споживання по зонах, записував результати в CSV або у легку базу SQLite для перегляду, і версія на Python обробляла дані за один день

за 34 мс, а за тридцять днів — за 180 мс. Така база не потребує жодного налаштування: будь-хто з оболонкою `sqlite3` відкриє файл і поставить йому запит.

Де виникають проблеми

Більшість клопотів із SQLite починається тоді, коли з нею поводяться як із маленьким PostgreSQL. Це інша машина з іншими гострими кутами.

Фільтр за роком повертає всі рядки. SQLite перетворює значення перед порівнянням лише тоді, коли один із боків має спорідненість. Результат функції на кшталт `substr()` її не має, як і голе число, тому текст `'2019'` порівнюється з цілим `2025` як є. А в порядку сортування SQLite будь-яке число стоїть перед будь-яким текстом:

```
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
```

Явно приводьте типи там, де текст зустрічається з числом. Новим таблицям давайте справжні типи й ключове слово `STRICT`, доступне з версії 3.37.0: суворий стовпчик `INTEGER` і далі приймає `'42'` та зберігає його як число, а `'ten'` відкидає з помилкою.

Запис раптом завершується помилкою `database is locked`**.** Одне з’єднання тримає блокування запису, а наступний записувач чекає або здається одразу. Причина зазвичай у довгій транзакції або в читачеві, який у типовому режимі rollback-журналу затримує записувача. Увімкніть write-ahead logging, щоб читачі й записувач перестали блокувати одне одного, тримайте транзакції запису короткими і задайте busy timeout, щоб записувач трохи чекав, а не повертав помилку. Якщо записувачів справді багато й вони пишуть постійно, це сигнал переходити на сервер.

Файл псується на мережевому диску. SQLite покладається на файлові блокування операційної системи, а в багатьох мережевих файлових системах саме вони реалізовані з помилками, тож дві машини можуть водночас писати в ту саму частину файлу. Write-ahead logging над мережевою файловою системою не працює взагалі, бо всі процеси, що користуються базою, мають ділити пам'ять на одній машині. Тримайте базу на локальному диску тієї єдиної програми, яка в неї пише, а якщо даних потребують інші машини, поставте перед ними сервер.

Рядок переживає батьківський запис, на який посилається. Зовнішні ключі записано в схемі, але вони неактивні, доки кожне з'єднання не виконає `PRAGMA foreign_keys = ON` . Видалення через з'єднання, яке це пропустило, тихо залишає рядки-сироти. Вмикайте прагму там, де програма відкриває з'єднання, в одному місці коду, і перевіряйте тестом, що вона справді ввімкнена.

Що почитати далі

- About SQLite — опис від самого проєкту: у процесі програми, без сервера, у суспільному надбанні, один файл для всіх платформ.
- Datatypes In SQLite — класи зберігання, спорідненість типів і правила порівняння, що стоять за пасткою вище.
- STRICT Tables — як змусити таблицю дотримуватися оголошених типів стовпчиків.
- Write-Ahead Logging — читачі й записувач паралельно і чому для цього потрібна одна машина.
- Appropriate Uses For SQLite — де SQLite доречна, де краще підходить клієнт-серверна база, і контрольний список.
- SQLite Foreign Key Support — чому обмеження за замовчуванням вимкнені і як їх увімкнути.

Словник

Обирайте за фактичними умовами

Надайте репрезентативний набір даних, запитання до нього та поточне вузьке місце. Узгодимо вимірювання якості, пропускної здатності та відновлення.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке SQLite?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

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

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/sqlite/