Engineer

SQLite

データベース全体を一つのファイルに収めるエンジン

どのようなものか

SQLiteは、それを使うプログラムの内部でライブラリーとして動くSQLデータベースエンジンです。サーバープロセスもネットワークポートも管理者も必要なく、テーブル、インデックス、トリガー、ビューを、ディスク上の一つの通常のファイルにまとめて保存します。`sqlite3` という名前も登場しますが、これはコマンドラインシェルの名前であり、Cインターフェースの全関数に付く接頭辞です。プロジェクトが配布する一つのCソースファイルは、`sqlite3.c` という名前です。プロジェクト自身は、自己完結型、サーバーレス、設定不要、トランザクション対応と要約し、Oracleよりも `fopen()` と比較するよう求めています。置き換える対象は、自作のデータファイルであり、データベースサーバーではありません。

プロジェクトは2000年5月9日に始まり、コードはパブリックドメインにあるため、製品に組み込んで出荷する前にライセンスの許諾を得る必要はありません。ファイル形式も、この選択の利点の一つです。32ビットのマシンで作ったデータベースを64ビットのマシンへ、あるいはビッグエンディアンとリトルエンディアンのプロセッサー間でコピーし、そのまま開けます。開発者は2050年までSQLiteを支える意向を示し、米国議会図書館は、そのファイルをデータ保存用の推奨形式に挙げています。すべての機能を有効にしても、ライブラリーは900KiB未満に収まることがあります。そのため、サーバーに加え、携帯電話、自動車、時計、デスクトップアプリケーションにも組み込まれています。

内部には、結合、サブクエリ、ビュー、トリガーを備えた本格的なSQLエンジンがあります。トランザクションは、書き込みの途中で電源が切れても、原子性、一貫性、分離性、耐久性を保ちます。サーバー型データベースとの違いを生む設計上の選択は二つあります。一つ目は同時実行です。同時に読み取る接続数に制限はありませんが、どの瞬間でも書き込めるのは一つだけで、ロックはファイル全体に及びます。二つ目は型の扱いです。列に宣言した型は、SQLiteがアフィニティーと呼ぶ、優先的に使う型を示しており、実際の型は個々の値に属します。そのため、`INTEGER` と宣言した列でも、数値に見えない文字列を受け付けます。

ほかの選択肢との違いは明確です。PostgreSQL、MySQL、SQL Serverは、多くのマシンがネットワーク越しに利用する共有サービスです。SQLiteは、一台のホスト上の一つのアプリケーションが使う専用の保存領域です。CSVやJSONファイルと比べると、クエリ言語、インデックス、制約、トランザクションが加わります。その代わりに依存ライブラリーが一つ増えますが、多くの言語はすでにそれを同梱しています。

概念を判断に生かす

「SQLiteとは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 実際の業務に合った構造を設計し、将来必要になる問いにも答えられるようにします。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 ETL/ELTパイプライン6本を設計・運用 — 6 3種類のDBエンジンにまたがるETLパイプライン

ETL/ELTパイプライン6本を設計・運用 お問い合わせ →

なぜ重要なのか

多くのソフトウェアにとって、データベースを巡る難問は、どれだけ速いかではなく、誰が運用するかです。サーバーには、インストール、アップグレード、セキュリティー対策、監視、バックアップが必要で、そのすべてを行う人に費用を払わなければなりません。SQLiteは、この運用業務一式を取り除きます。データベースを開くことはファイルを開くことであり、バックアップはそのファイルのコピーです。使用中ならバックアップAPIを通じてコピーします。テストスイートでは、各テストに、それぞれ新しいデータベースを用意できます。

ファイルでできることも変わります。独自形式を考え、そのパーサーを書き、保存途中でクラッシュしないよう願う代わりに、プログラムの状態をSQLiteに保存すれば、原子的な保存を追加実装なしで得られます。プロジェクトも、この用途を明示しています。アプリケーションのファイル形式、システム間でデータを渡す入れ物、そして別の形で届いたデータを並べ替えたり集計したりする一時的なデータベースです。システム間の受け渡しでは、受け取り側はパーサーを書く代わりにSQLを実行します。

判断すべきことは限られており、コードを一行も書く前に確認できます。書き込むプロセスはいくつで、何台のマシンで動くのでしょうか。一台のホスト上の一つのプログラムが、短時間ずつ書き込むのはSQLiteの得意分野であり、アクセス量が少ない、あるいは中程度の多くのWebサイトが、SQLiteで無理なく稼働しています。複数のサーバーが同じデータに書き込む場合や、多数の利用者が絶えず

書き込む負荷には、クライアントサーバー型データベースが向いています。一つのファイルに最大281テラバイトを保存できるため、容量が判断を左右することはまれです。決め手は、書き込みを行う側の数です。

得意分野の中でも、考慮すべき制約があります。 `ALTER TABLE` が対応するのは、テーブル名や列名の変更、列の追加や削除だけです。そのため、それ以外の大半の構造変更では、新しいテーブルを作り、行をコピーし、名前を入れ替える必要があります。外部キーは、接続ごとに有効化するまで無効です。また、拒否するようテーブルを宣言しなければ、緩やかな型の扱いにより、不適切なデータが気付かれないまま入ります。

実際の活用例

記録されている開発段階では、一つのSQLiteファイルにある23のテーブルが、CV、推薦者一覧、ポートフォリオ、カバーレターを生成するシステムの正本でした。実績や企業から、プロフィールや推薦者情報までを格納していました。一つのコマンドで、空のSQLiteファイルからそのデータベースを構築し、19の手順を順番に実行していました。各手順を安全に二度実行できるため、既存のデータベースを再構築しても行が重複しませんでした。6種類の出力形式は、その一つのファイルを読む、合計10,580行の41のPythonモジュールから生成され、実行時の依存ライブラリーはちょうど二つでした。特定の読者向けの文書は、コピーを編集するのではなく、プロフィールの指定に従って実績を選ぶクエリで作っていました。代償は明確で、このシステムが文書を作る唯一の手段になっていたことです。

記録されているエクスポート段階では、このWebサイト自身のポートフォリオも同じデータベースから生成していたため、それがビルドの入力でもありました。エクスポーターはSQLiteファイルを読み取り専用で開き、495ページを書き出しました。対象は3言語で、Pythonの標準ライブラリーだけを使い、生成システムの環境がなくてもサイトをビルドできるよう、出力をコミットしていました。ページのアドレスは、実績ごと、アドレスごとに一行ずつ保存したため、文を言い換えてもページが移動しなくなりました。この作業では、次の節で説明する落とし穴も見つかりました。文字列の列に対する期間の比較が、32行すべてに一致してしまったのです。

結果を確認するだけという最小限の使い方は、最も典型的な利用法でもあります。電力の生の計測値を地域ごとの毎時消費量に変換するパイプラインは、結果をCSV、または確認用の軽量なSQLiteデータベースに書き出しました。Python版

は1日分を34ms、30日分を180msで処理しました。こうして使うデータベースには設定作業がまったく不要で、`sqlite3` シェルさえあれば、誰でもファイルを開いてクエリを実行できます。

問題が起きるところ

SQLiteの問題の多くは、小さなPostgreSQLだと考えて扱うことで生まれます。仕組みも、制約の現れ方も異なるエンジンです。

年のフィルターが、すべての行に一致する。 SQLiteが比較前に値を変換するのは、いずれかの側にアフィニティーがある場合だけです。`substr()` のような関数の結果にはアフィニティーがなく、数値リテラル単体にもありません。そのため、SQLiteは文字列 `'2019'` と整数 `2025` をそのまま比較します。そして、SQLiteの並び順では、どの数値も、どの文字列よりも前に来ます。

```
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
```

文字列と数値が混在する箇所では、明示的に型変換します。新しいテーブルでは、意図した型を宣言し、バージョン3.37.0から使える `STRICT` キーワードを加えます。型を厳格に検査する `INTEGER` 列でも `'42'` は受け付け、数値として保存しますが、`'ten'` はそのまま保存せず、エラーとして拒否します。

書き込みが「database is locked」で失敗し始める。 一つの接続が書き込みロックを持っており、二つ目の書き込み側は、待つか、直ちに断念します。一般的な原因は、長いトランザクションや、既定のロールバックジャーナル方式で読み取り側が書き込み側を待たせていることです。読み取り側と書き込み側が互いを妨げなくなる先行書き込みログ方式へ切り替え、書き込みトランザクションを短く保ち、ロック待ちのタイムアウトを設定します。書き込み側がすぐエラーになる代わりに、少し待てるようにするためです。本当に多数の書き込み側が絶えず動くなら、サーバー型へ移るべき兆候です。

共有ドライブ上でファイルが破損する。 SQLiteはOSのファイルロックに依存しますが、多くのネットワークファイルシステムでは、そのロックの実装に不具合があり、二台のマシンがファイルの同じ部分へ同時に書き込むことがあります。先行書き込みログ方式は、ネットワークファイルシステムではまったく動きません。データベースを使うすべてのプロセスが、一台のホスト上でメモリーを共有する必要があるためです。データベースは、書き込む一つのプログラムから見てローカルのディスクに置き、ほかのマシンがデータを必要とするなら、その手前にサーバーを置きます。

参照先の親が消えても、行が残る。 外部キー制約はスキーマに宣言されていても、接続ごとに `PRAGMA foreign_keys = ON` を実行するまで無効です。そのため、このプラグマを実行していない接続で削除すると、警告もなく親のない行が残ります。アプリケーションが接続を開く処理を一か所にまとめ、そこで必ずプラグマを設定し、有効になっていることをテストします。

参考資料

- SQLiteについて — プロセス内で動き、サーバーを必要とせず、パブリックドメインであり、一つのファイルを異なる環境で使えることを説明する、プロジェクト自身の概要です。
- SQLiteのデータ型 — 上の落とし穴の背景となる、ストレージクラス、型アフィニティー、比較の規則を説明しています。
- STRICTテーブル — 列に宣言した型をテーブルに強制させる方法です。
- 先行書き込みログ — 読み取り側と書き込み側を並行して動かす仕組みと、一台のホストに限定される理由です。
- SQLiteに適した用途 — 適する場面、クライアントサーバー型がより適する場面、確認項目の一覧です。
- SQLiteの外部キー対応 — 制約が既定で無効である理由と、有効化する方法です。



用語集

実際の制約に合わせて選ぶ

代表的なデータセット、そこから答えを得たい問い、現在のボトルネックをお知らせください。品質、処理量、復旧の測り方を合意できます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「SQLiteとは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

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

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/sqlite/