Engineer

データベース

プログラムで問い合わせられるように整理したデータの保管場所

どのようなものか

データベースは、何を入れてよいかという規則を持つ、決まった構造で保存したデータの集まりです。ソフトウェアが、中身を探し、変更し、検査できるようにします。この語はDBと略されることも多く、その保存を担うプログラムを大まかに指す場合もあります。プログラムの正確な名前は、データベース管理システム、DBMSです。NISTの用語集には二つの定義があり、広い方では、情報やデータの保管場所について、「従来のリレーショナルデータベースシステムである場合も、そうでない場合もある」と説明しています。

この語には三つの使い道があり、二人が違うものを指すと、会話がかみ合いません。組織が保存する顧客、注文、センサー読み取り値といった、データそのものを指します。PostgreSQL、SQL Server、SQLiteといった、保存するエンジンを指します。そして、サーバー内では、複数ある入れ物の一つを指します。PostgreSQLのチュートリアルは、その階層を示しています。名前と型を持つ列のある行がテーブルを作り、テーブルがデータベースにまとまり、一つのサーバーインスタンスがまとめて管理するデータベース群が、データベースクラスターになります。SQL Serverも、インスタンスと、その中のデータベースを同じように分けます。「データベースが遅い」は、一つのテーブル、一つの入れ物、一台のマシン全体の、どれを指すこともあります。

日常的に使うエンジンの多くはリレーショナルです。データを、数学でテーブルに当たるリレーションとして保存し、SQLで問いを受け付けます。PostgreSQLのチュートリアルは、ほかの構成も挙げています。Unix系システムのファイルとディレクトリーは、階層型データベースを形成し、その後、オブジェクト指向データベースも現れました。Redisは、自らをデータ構造サーバーと呼びます。テーブルの代わりに、文字列、ハッシュ、リスト、集合などの組み込み型を提供し、キャッシュ、キュー、イベント処理に使います。

エンジンは、動く場所も異なります。クライアント／サーバー型は、長時間動作するプロセスで、多数のプログラムがネットワーク越しに接続します。組み込み型は、一つのプログラムの内部にあるライブラリーです。SQLiteは、テーブル、インデックス、トリガー、ビューを含むデータベース全体を、普通の一つのファイルに保存します。SQLiteプロジェクトは、競合相手はサーバー型のデータベースではなく、 `fopen()` だと説明しています。違いは、プログラムの接続方法に現れます。

```
# クライアント／サーバー型：起動済みのサービスに接続する
psql "host=db.internal dbname=shop user=app"

# 組み込み型：プログラムがファイルを開き、その内部でエンジンが動く
sqlite3 shop.db
```

第三の系統は、分析向けです。従来の多くのシステムが、一行のフィールドをディスク上で一緒に保存するのに対し、BigQueryは、列ごとに分けて保存します。行単位の保存は、一件のレコードを探すのに向いています。列単位の保存は、数百万行にわたって一つのフィールドを合計するのに向いています。クエリーがその列だけを読み、残りに触れずに済むためです。

概念を判断に生かす

「データベースとは？」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

ファイルもデータを保持します。データベースは、多数のプログラムが同時に使うとき、そのデータについての保証を加えます。

第一の保証は、同時利用の下でも、一貫した状態を見せることです。PostgreSQLは、各文に対し、ある時点のデータのスナップショットを与えます。多版型同時実行制御、MVCCと呼ぶ設計で、読み取りが書き込みを待つことも、書き込みが読み取りを待つこともありません。SQLiteは、読み取り側の数を

制限せずに同時利用を認めますが、書き込み側はどの瞬間も一つだけで、ほかは順番を待ちます。適した書き込み主体の数は両者で異なり、その数が、選択を大きく左右します。

第二の保証は、データの形を保つことです。スキーマは、テーブルにどの列があり、それぞれがどの型を持つかを定めます。キーと制約は、どんな行が存在してよいかを定めます。データベースが規則を適用すれば、書き込むすべてのプログラムが従います。規則がアプリケーションコード内だけにあると、最初に無視したプログラムが不適切な行を残し、後続のすべてのクエリーが、それに対処しなければなりません。

第三は、方言のある共通言語です。SQLはISO/IEC 9075という標準で、最新の改訂はSQL:2023です。それでも、PostgreSQLの適合性についての付録は、現在のどのシステムも、その中核部分であるCoreへの完全適合を主張していないと記しています。スキーマとクエリーは、それぞれのエンジンに属するものです。別のエンジンへ移す作業は、それに応じたテストを伴う移行になります。

SQLiteの文書は、エンジン選びを短い確認表にしています。SQLを発行するコードとデータがネットワークで隔てられる、多数のプロセスが同じ瞬間に書き込む、内容がテラバイト規模へ近づく、という条件は、どれもクライアント／サーバー型を選ぶ理由になります。それ以外なら、SQLiteの方が単純です。大規模なレポートには、もう一つ問いがあります。ETLで読み込む別の分析用コピーに重い読み取りを任せ、長いレポートが、顧客の待つクエリーと競合しないようにすべきか、という点です。

実際の活用例

大きな規模の例では、急成長する技術企業でのデータベースエンジニアの役割として、PostgreSQL、Microsoft SQL Server、Google BigQuery上の100以上のデータベースを作成し、管理し、最適化しました。その大半に、GISデータと、センサーの読み取り値、ログ、リアルタイムの指標などの時系列データが保存されていました。PostgreSQLとSQL Serverでは、正規化したスキーマと非正規化したスキーマを作り、空間インデックス（PostgreSQL側ではPostGIS）と時系列のパーティション分割を使いました。BigQueryでは、大量の時系列データを、時間でパーティション分割し、クラスタリングしたテーブルに保持しました。インデックス、マテリアライズドビュー、キャッシュを導入した結果、GISと時系列のクエリーにかかる時間は40–60%短くなり、自動化した監視で、停止時間を50%減らしました。

より小規模なシステムでは、記録されたある開発段階で、一つのSQLiteファイルが、三言語のCV、推薦者一覧、ポートフォリオ、カバーレターの元になる事実を保持していました。データベースには二十三のテーブルがあり、一つのコマンドが、順序の決まった十九の段階で構築していました。どの段階も、安全に再実行できました。データベースから六つの出力形式を生成するのは、合計10,580行の41のPythonモジュールでした。特定の読者向けの文書は、「コピーではなくクエリー」でした。プロファイルが、どの実績をどの順で載せるかを選んでいました。引き換えに、その時点では、文書をそのデータベース以外から作れなくなりました。これらの実装の数値は、記録されたその段階を表し、現在のリポジトリや対応言語の範囲を示すものではありません。

その二つの規模の間には、負荷の高いデータベースならどれも直面する問いがあります。シャーディングの前に最適化するというノートは、作業の順序で答えています。観察し、無駄を減らし、プライマリーから読み取りを移し、負荷の集中点に合わせてテーブルをパーティション分割し、一台が有用な仕事で埋まって初めて、シャーディングします。観察とは、PostgreSQL自身に尋ねることです。`pg_stat_statements`で総コストの大きい文を調べ、次に`EXPLAIN (ANALYZE, BUFFERS)`で、最も高コストの文が何をしているかを見ます。

問題が起きるところ

データベースの問題の多くは、データが少なく、負担がないように見えた初期の判断から始まります。

一つの事実の二つのコピーが、食い違います。 上のCV生成器があるのは、四つの文書として持つ事実を四回ずつ直すうち、二つの内容が食い違い、どちらが最新か分からなくなるためです。テーブルを表計算へ出力して編集したり、別のサービスが独自の顧客一覧を持ったりしても、同じことが起きます。各事実には、それを管理する保存場所を一つ決め、ほかのすべてのコピーに、出所と時点を明記します。ノートのキャッシュは、この規則を厳密に守ります。商品がすべて予約済みだと購入者に伝えることはできますが、購入可能だという肯定の答えは、常にデータベースから返します。

実行計画を読む前に、マシンを増やします。 データをサーバー間に分けても、遅い文を多くのハードウェアへ分散するだけで、個々の文の処理コストは下がりません。それ以外すべてに、継続的な負担が加わります。ノートが挙げるように、移行、バックアップ、復元、レポートは、それぞれのシャードで一回ずつ行い、シャードをまたぐ結合はアプリケーションコードになります。まず`pg_stat_statements`と実行計画を読み、その結果を基準として残します。

遅れているコピーから、読み取ります。 ストリーミングレプリカは、プライマリーの読み取りを肩代わりしますが、少し遅れています。書き込んだばかりの利用者が、レプリカから読み返すと、古い値が見える場合があります。利用者自身の最近の書き込みは、プライマリーから読みます。

レポートと顧客が、一つのエンジンを共有します。 二年分の履歴を走査するレポートは、店のクエリーに必要なメモリーとディスクを使い、購入者にも影響します。重い分析をウェアハウスのコピーへ移し、業務用のデータベースは、本来担当する多数の小さな読み書きに使います。

参考資料

- NIST用語集：データベース — NISTの刊行物にある、二つの短い定義です。一方は、リレーショナルでないデータベースも認めています。
- PostgreSQLチュートリアル：概念 — テーブル、行、列、データベース、データベースクラスターを、一ページにまとめています。
- PostgreSQL：MVCCの概要 — スナップショットによって、読み取りと書き込みが互いをブロックせずに動く仕組みです。
- PostgreSQL：SQLへの適合性 — ISO/IEC 9075の各部とバージョン、Coreへの適合性の意味を説明します。
- SQLiteの適切な用途 — 組み込み型が合う場合、クライアント／サーバー型の方が合う場合、選択のための確認表です。
- BigQueryストレージの概要 — 行指向と列指向の保存方法と、分析で後者が好まれる理由です。

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

「データベースとは？」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/database/