Engineer

SQL

リレーショナルデータベースを操作する言語

どのようなものか

SQLは、リレーショナルデータベースのテーブルに保存されたデータを定義、変更、読み取りするための宣言的な言語です。欲しい結果を書けば、その結果を作る方法はデータベースが決めます。SQLは通常Structured Query Languageの略とされ、一文字ずつ読む場合と「シークェル」と読む場合があります。一方、SQLを定義する国際標準ISO/IEC 9075は、単にDatabase Language SQLと呼んでいます。

鍵になるのは「宣言的」という言葉です。次の文は、テーブル、条件、グループ化、並び順を指定しています。インデックスやループ、最初にどの行を読むかについては何も指定していません。

```
SELECT country, count(*) AS signups
FROM accounts
WHERE created_at >= now() - interval '30 days'
GROUP BY country
ORDER BY signups DESC;
```

PostgreSQLでは、この文字列が四つの段階を通ります。パーサーが構文を検査して木構造を作り、書き換え処理が文の参照するビューを展開し、プランナーが順次走査とインデックスの利用など、実行方法の候補を比較します。最後にエグゼキューターが行を取得します。プランナーが見いだす方法次第で、同じ文の実行に1ミリ秒かかることも、1分かかることもあります。そのため、実行計画を読むこともSQLを書く仕事の一部です。

SQLの各部分には、それぞれ名前が付いています。データ定義はテーブル、スキーマ、インデックスを作成・変更します。データ操作は行を挿入、更新、削除、マージします。クエリはSELECTで行を読み取ります。権限付与とトランザクシ

ョンは、誰に何を許すか、どの変更を一まとまりとして確定するかを決めます。これらすべてを一つの接続で文字列としてやり取りするため、SQLはアプリケーション、レポートツール、端末を操作する人の共通言語になります。

標準は何度も改訂され、最新はSQL:2023です。各エンジンは、その一部をそれぞれ実装しています。PostgreSQLの標準準拠に関する付録は、必須のCore機能177項目のうち少なくとも170項目を満たすとし、執筆時点では、Coreへの完全な準拠を主張するデータベースシステムはなかったと述べています。また、各エンジンは独自の方言も加えています。Microsoft SQL ServerはTransact-SQL、BigQueryはGoogleSQLを使い、PostgreSQLは、データベース内で実行する手続き型コード用にPL/pgSQLを追加しています。

概念を判断に生かす

「SQL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

SQLを使うチームが判断するのは、データの処理をどこで行うかです。SQLを実行するために作られたデータベースの内部で処理するか、行を取得して独自に処理するアプリケーションコードで実行するかを選びます。

集合を扱う文は、データのある場所で動きます。データベース内で1,000,000行を40行に絞り込めば、ネットワークを通るのは40行です。アプリケーションで絞り込むなら、1,000,000行を送り、その大半を捨てます。結合、集計、重複排除も同じ理屈です。こうした処理をSQLに移すことは、遅いシステムを高速化する手段の中でも、最も安価な選択肢になることがよくあります。すでにあるデータベースで実行できるためです。

その一方で、誰が保守を担うかも考える必要があります。SQLで書いたロジックはマイグレーションや関数の中にあり、ほかのコードと同じようにレビュー、テスト、バージョン管理が必要です。GoやPythonのコードほどには、長いPL/pgSQL関数をすらすら読める技術者は多くありません。実用的な分担は、多数の行を見る必要がある処理をデータベースに任せ、業務に合わせて変わる規則を、単独でテストしやすいアプリケーション側に残すことです。

二つ目のトレードオフは移植性です。結合を伴うSELECT、GROUP BY、ORDER BYといった言語の中核は、わずかな修正でエンジン間を移せます。周辺部分はそうはいきません。日付の計算、upsert、ページ分割、JSON関数、手続き型コードはすべて異なり、別のデータベースへの移行では、その部分に作業の大半を費やします。エンジンを選ぶことは方言を選ぶことでもあり、その方言に多くのロジックを置くほど、後の移行費用は増えます。

三つ目は人材の採用です。SQLは、アナリスト、バックエンド開発者、データエンジニア、データベース管理者が共通して持つ数少ないスキルの一つです。そのため、通常のSQLで書いたデータ層は、いずれの職種の人も読んでレビューできます。ORMはSQLを隠しますが、ページが遅くなれば、結局は誰かが生成されたSQLを読む必要があります。

実際の活用例

あるプラットフォームでは、PostgreSQLの関数がAPIの応答を完成したJSON文書として組み立て、Goのハンドラーがデコードも再エンコードもせず、`json.RawMessage` として転送します。JSONパススルーアーキテクチャは、応答の形を整える処理をSQLの役割とし、すでにデータがある場所で行います。小さなQueryJSONヘルパーによって、この方式を使うのが最も簡単になりました。層を重ねたAPIにたまりがちなDTO、マッパー、応答用の構造体も不要になりました。ハンドラーのコードは大幅に短くなり、関数の戻り値を変えると、ハンドラーを誰も編集せずに、新しい形がクライアントへ届きます。

同じ言語でも、データパイプラインでは姿が変わります。BigQuery、MSSQL、PostgreSQLにまたがる6本のETL/ELTパイプラインでは、bcp、psql、SSISで生のファイルをステージングスキーマへ読み込み、整形、正規化、データの拡充をPL/pgSQLとT-SQLのプロシージャで記述しました。一つの処理の中に、それぞれ独自の方言を持つ三つのエンジンがあります。上で述べた移植性の問題を、実際の作業の内側から見た例です。事例ページには、少なくとも3時間の手作業が全体20分未満に短縮され、データ更新が週次から日次になり、データを読むPython APIが30%高速化したと記録されています。

問題が起きるところ

文字列を連結して文を作る。 顧客の姓がO'Brienだと検索ページが動かなくなったり、細工した入力で他人のデータが読まれたりします。原因は、SQLの文字列に利用者の入力を埋め込み、データベースがデータと命令を区別できなくなることです。主要なドライバーがすべて対応しているように、値は文から分離し、パ

ラメーターとして送ります。PostgreSQL自身のクライアントライブラリーでは、パラメーターを使う呼び出しが一つの文字列につき一つの命令しか受け付けないため、インジェクションへの第二の防御にもなります。

たまたまの並び順に頼る。 何か月も整列して返ってきた一覧が、データの読み込みや新しい実行計画をきっかけに、ばらばらになります。ORDER BYがなければ、PostgreSQLは最も速く返せる順で行を返し、ORDER BYなしのLIMITでは、返す行の集合さえ毎回変わることがあります。結果の順序が重要なら、その順序をクエリに明記します。

NULLを通常の値として扱う。 `WHERE discount = NULL` と書いたフィルターは何も返さず、NULLを含むサブクエリに対する `NOT IN` も何も返しません。NULLとの比較は、真でも偽でもなく不明になり、WHERE節が残すのは条件が真の行だけです。`IS NULL` を使い、内側の列にNULLが入り得る場合は、`NOT EXISTS` を `NOT IN` より優先します。

方言の違いをテストで見落とす。 テストスイートはSQLiteで通るのに、本番のPostgreSQLでは同じコードが失敗したり、SQL Serverへの移行で、すべての `LIMIT` が障害になったりします。SQL Serverでは、`TOP` または `OFFSET ... FETCH` と書くためです。それぞれのエンジンに、ほかでは拒否される構文があります。デプロイするエンジンでテストし、エンジン固有の構文は、できるだけ少数の場所にまとめます。

参考資料

- 第II部：SQL言語 — 構文から性能改善のヒントまでを扱う、PostgreSQLのSQLチュートリアルと完全なリファレンスです。
- 付録D：SQLへの準拠 — ISO/IEC 9075の各部と、PostgreSQLが実装している機能を説明しています。
- Transact-SQLリファレンス — MicrosoftによるSQL Server方言のドキュメントです。
- BigQueryのSQL入門 — GoogleSQLと、それが置き換えた従来の方言を説明しています。
- SQLiteが解釈するクエリ言語 — 組み込みエンジンの構文を、文ごとに説明しています。

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

「SQL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/sql/