Engineer

BigQuery

巨大なテーブルを検索・集計するGoogleのデータウェアハウス

どのようなものか

BigQueryは、Google Cloudの管理型SQL分析サービスです。利用者がサーバーを用意する必要はなく、テーブルは列ごとに保存されます。クエリの料金は、読み取ったバイト数に応じて支払うか、事前に予約した計算能力からまかないます。Google BigQueryという名前でも提供され、その名前でも検索されます。SQLにはGoogleSQLを使います。ANSI標準に従い、独自の関数を加えた方言です。

設計の基本は、ストレージと計算を分離することです。テーブルはGoogleのストレージ層に置き、クエリは実行している間だけ計算資源を借ります。各列を個別に保存するため、五十列のテーブルから三列を指定するクエリは、その三列だけを読み、残りはディスク上に置いたままにします。したがって、調整の仕方はPostgreSQLサーバーと異なります。B-treeインデックスを設計したり、メモリー容量を決めたりする代わりに、テーブルの構成を整えます。日付・タイムスタンプの列、取り込み時刻、整数の範囲でパーティション分割でき、最大四列でクラスタリングして、関連する行を同じストレージブロックにまとめられます。

BigQueryはデータウェアハウスの一種で、通常はSnowflakeやAmazon Redshift、または会社ですでに運用しているトランザクション用データベースで分析を続ける選択肢と比較します。数百万、数十億の行を走査し、少数の集計値を返すクエリ向けに設計されています。アプリケーションが要求のたびに行う、一行ずつの検索や更新は、PostgreSQLやSQL Serverなどのデータベースが担う仕事です。両方を運用し、パイプラインで一方から他方へデータを送るのが、よくある構成です。

概念を判断に生かす

「BigQuery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。 営業、運用、経理の数値をまとめ、同じ会議に部門ごとに異なる答えを持ち寄る状況を解消します。

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。 地理情報パイプラインを50倍に高速化 — 50x 地理データパイプラインの高速化

地理情報パイプラインを50倍に高速化 お問い合わせ →

なぜ重要なのか

BigQueryを検討するチームが決めるのは、分析にどう料金を支払いたいかです。請求額は、誰かが一度選んだマシンの大きさではなく、人が書くクエリで決まります。それは利点にも、不利益にもなります。

オンデマンド料金では、各クエリが処理するバイト数で課金します。不定期の作業に適しており、週に数回、重いレポートを実行するチームは、そのレポートの分だけ支払い、実行しない間の料金はかかりません。夜間に待機するサーバーもありません。一方、不注意なクエリにも費用が生じます。一年分のセンサー測定値に対する `SELECT *` は、分析担当者が一行だけ欲しかった場合も、全行が欲しかった場合も、同じ費用がかかります。容量ベースの料金では、代わりにスロットを使います。予約を通じて購入する計算能力の単位で、請求額を事前に分かる数値にし、負荷の高い利用が安定して続く場合に、費用を抑えた計画を立てやすくします。プロジェクトは二つの料金体系を切り替えられるため、実際の利用状況が見えてから、最初の選択を見直せます。

ストレージには別のルールがあります。テーブルやパーティションを連続90日間変更せずに置くと、移動の操作をしなくても、50パーセント低い長期保存料金へ移ります。各データセットは、非圧縮の論理サイズか、圧縮済みの物理サイズのいずれかで課金されます。よく圧縮できるデータでは、後者のほうが安くなる場合があります。

契約前には、三つの点を重視すべきです。データセットの配置場所は、単一リージョン、またはUS・EUのマルチリージョンから作成時に選び、その後は固定されます。初日に行う、法令順守に関わる判断です。また、SQLの方言、`bq` コマンドラインツール、クライアントライブラリーはいずれもGoogle独自のため、クエリを別のデータウェアハウスへ移すには、そのままではなく修正が必要です。一方、テーブルをCSV、Avro、Parquetへ出力できるため、データ自体は容易に取り出せます。人材確保の難しさは、見かけほどではありません。SQLを知る分析担当者は、数日で実務に対応できます。むしろ希少なのは、クエリを行数だけでなく、バイト数でも読み解く能力です。

では、BigQueryが適切なのは、どのような場合でしょうか。イベント、ログ、メーターの測定値など、主に追記する大量のデータが入り、尋ねる内容が週ごとに変わり、チームが分析用データベースサーバーの運用を避けたい場合です。一台のPostgreSQLサーバーに余裕を持って収まるデータなら、データウェアハウスは、二つ目のシステム、二つ目の請求、その間を結ぶパイプラインを増やします。導入の意義は、データ量と、データに問いかける人の数が増えるほど大きくなります。

実際の活用例

PostgreSQL、Microsoft SQL Server、Google BigQueryに分かれた100を超えるデータベースの環境では、座標、センサー測定値、ログ、リアルタイム指標など、主に地理データと時系列データを保持していました。BigQueryは大規模な時系列データを担当し、テーブルを時間でパーティション分割し、クラスタリングしました。圧縮と列指向の保存を使い、保存費用を抑えて走査を高速にしました。BigQueryのパーティション分割・クラスタリングした走査と、環境全体でのインデックス、マテリアライズドビュー、キャッシュにより、GISと時系列クエリの応答時間を40–60%短縮しました。

遅くなっていた地理データのパイプラインでは、まずPostgreSQL、MS SQL、Google Cloud BigQueryのSQLとデータモデルを分析し、不十分なインデックス、不効率なクエリ、足りない制約を特定しました。そのうえで、各エンジンに合った手段を使いました。PostgreSQLにはB-treeとGiSTインデックス、MS SQLにはフィルター選択されたインデックス、BigQueryにはクラスタリングを適用し、モデルの正規化、パーティション分割、クエリの書き直しも行いました。創業者の実績記録では、パイプライン全体が50倍高速に動作するようになったと報告されています。ただし、ベンチマークに用いたデータセット、ハードウェア、比較手順は明記されていません。

複数の地域のスマートメーターデータでは、PostgreSQLとBigQueryに対し、500を超える分析レポートを純粋なSQLで作成しました。メーターID、タイムスタンプ、座標、環境メタデータに、CTE、ウィンドウ関数、サブクエリ、地理空間結合を使いました。各レポートには、同僚が再実行・監査できるよう注釈付きのSQLと、GPS座標の欠落やメーター値の破損など、既知のデータ品質上の問題に関する説明を添えました。レポートは部署をまたいで参照され、業務計画の効率を10–15%改善することに寄与しました。

三つの事例すべてで、PostgreSQLをBigQueryに置き換えず、両者を併用し、それぞれのエンジンに最も適した仕事を割り当てました。

問題が起きるところ

`LIMIT 10` のクエリで、テーブル全体の料金を請求されます。 クラスタリングしていないテーブルでは、`LIMIT` が減らすのは結果であり、読み取るバイト数ではありません。`SELECT *` は、存在するすべての列を読みます。少しデータを見るつもりが、全体の走査になります。問いに必要な列を指定し、行のサンプルを眺めるだけなら、無料のコンソールのテーブルプレビューや `bq head` を使います。

パーティション分割しても、テーブル全体を走査します。 パーティションを読み飛ばすには、分割に使う列のフィルターが、読み取り前にエンジンで評価できる定数である必要があります。その列をサブクエリの結果と比較したり、関数に渡したり、列に対して計算したりすると、クエリはすべてのパーティションを走査します。対策は二つあります。テーブルでパーティションフィルターを必須にすれば、使えるフィルターのないクエリを、実行せずエラーで止められます。また、試行実行は、クエリを実行せず、料金も発生させずに、処理する予定のバイト数を報告します。

```
bq query --use_legacy_sql=false --dry_run \
  'SELECT meter_id, kwh
   FROM energy.readings
   WHERE reading_date = DATE "2026-10-01"'
```

クラスタリングを加えても、読み取るバイト数がほとんど変わりません。 クラスタリングは、最大四列を使い、その順序でストレージブロックを並べます。順序は重要で、注文日、国、状態の順にクラスタリングしたテーブルは、国と状態だけのフィルターに適した配置ではありません。64 MB未満のテーブルでは効果が小さく、クラスタリングしたテーブルで実際に走査したバイト数は、クエリ終了後にしか分かりません。クエリで絞り込む順序に合わせて、クラスタリングの列を並べます。実行前に費用の上限を確定する必要がある場合は、試行実行の時点で読み飛ばす部分を見積もれるパーティション分割を使い、課金対象バイト数の上限も設定して、超過するクエリが課金前に止まるようにします。

二つのデータセットを結合するクエリが拒否されます。 各データセットは一つの場所に置かれ、クエリも一つの場所で動くため、参照するすべてのデータセットが同じ場所にある必要があります。プロジェクトごとに既定の場所を選ばせると、一方を別の場所へコピーするまで結合できないテーブルが生まれます。法令順守とデータの発生場所を考えて、一度、配置場所を決め、すべてのデータセットを意識的にそこへ作成します。

参考資料

- BigQueryの概要：サービス、分離されたストレージ層と計算層、利用者に代わって管理する範囲についてのGoogle自身の説明です。
- パーティション分割テーブルの概要：分割の種類、粒度、制限を説明します。
- パーティション分割テーブルへのクエリ：パーティションを読み飛ばせるフィルターと、すべてを走査するフィルターを説明します。
- クラスタ化テーブルの概要：列の順序とブロックの読み飛ばしの関係、それが費用見積もりに与える影響を説明します。
- 費用の見積もりと管理：試行実行、課金対象バイト数の上限、テーブルを無料でプレビューする方法を説明します。
- BigQueryのロケーション：リージョン、マルチリージョン、複数データセットを参照するクエリのルールを説明します。

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

「BigQuery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/bigquery/