Engineer

データセット

一まとまりとして公開されるデータの集まり

どのようなものか

データセットは、英語ではdatasetともdata setとも書き、人が一つのものとして扱う、範囲を区切ったデータの集まりです。名前、発生源、対象期間、各レコードに共通する形を持ち、この四つによって、どこからどこまでを含むかが決まります。一つのファイルに入った一か月分のメーター読み取り値も、関連する三つのテーブルを毎晩エクスポートして得るデータも、データセットです。大きさや形式を示す語ではなく、その中のデータを一緒に扱うという意味です。

このサイトのページも、この実務上の意味で使っています。あるページは、JSONL形式の履歴データセットと、性能比較に使った1日分と30日分のデータセットについて述べます。別のページは、ほかへ読み込むために出力した、整備済みのデータセットとテーブルを扱います。さらに別のページは、スマートメーターが生む膨大なデータセットを扱います。どの場合も、処理にかける前に、対象データの範囲を定めています。

W3CのData Catalog Vocabulary、DCATは、この語に正式な枠組みを与えます。バージョン3は、2024年8月22日にW3C勧告となりました。データセットを、「単一の主体が公開または整備し、一つ以上の直列化された表現または形式でアクセスやダウンロードが可能な、データの集まり」と定義します。意図的に広い概念で、数値、テキスト、画像、音声のいずれも含みます。その上で、日常の言葉では混同する、三つのものを分けます。

- **データセット**：データそのものの概念です。たとえば、2024年の地域ごとの一時間単位の消費量です。
- **配布物**：ダウンロードできるCSVファイルなど、アクセス可能な一つの形です。一つのデータセットに、複数あって構いません。
- **データセット系列**：月ごとのファイルのように、別々に公開され、まとめられるだけの共通点を持つデータセット群です。

したがって、形式の異なる二つのファイルが、一つのデータセットの場合もあります。同じ名前の下にある二つの月別ファイルが、一つの系列の二つの要素である場合もあります。

Google BigQueryでは、もっと狭い第二の意味があります。データセットは、テーブル、ビューなどのリソースを入れる、プロジェクト内の入れ物です。テーブルは、必ずデータセットに属するため、データを読み込む前に、データセットを作ります。GoogleSQLでは `project.dataset` 、bqコマンドラインツールでは `project:dataset` として指定します。どちらの意味かは、文脈で決まります。

概念を判断に生かす

「データセットとは？」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。 意思決定を左右する少数の指標を示すダッシュボードやレポートを、依頼しなくても更新される形で提供します。

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。 読み手の反応に基づくブランド発信 — 読み手の反応に基づくブランド発信

読み手の反応に基づくブランド発信 お問い合わせ →

なぜ重要なのか

範囲を区切ることで、データを比較できます。二組の読み取り値を比較できるのは、対象期間、場所、レコードの間隔が同じときだけです。データセットに名前を付けることで、その条件を固定します。DCATは、間隔に時間解像度という専用のプロパティーを用意しています。要素間の最小間隔を、一日を表す `P1D` のような期間で記します。解像度を明記したデータセットなら、最初に合計する前に、行が一時間ごとか、十五分ごとか、不規則なのかが分かります。

説明は、データと一緒に渡す必要があります。DCATは、配布物が自分について示せる情報として、メディアタイプ、形式、圧縮形式、パッケージ形式、バイト数、チェックサムを挙げます。最後のチェックサムがあれば、受け取った人は、中身が変わっていないか確かめられます。仕様自体は、圧縮形式の例にgzipを使っています。こうした情報があれば、届いたファイルを検査し、開けます。何も付いていなければ、推測しなければなりません。

配布形式によって、誰が読めるかが決まります。よく使うものは、三つあります。

- **CSV。** 2005年のRFC 4180は、多くの実装が従っていると思われる形を記録しています。一行に一レコードを置き、見出し行は任意です。カンマ、改行、二重引用符のいずれかを含むフィールド全体を二重引用符で囲み、フィールド内の二重引用符は二つ重ねて書きます。ほとんどの道具で開けます。
- **JSON Lines。** 一行に一つの有効なJSON値を置き、UTF-8を使い、行の区切りには `\n` を使います。拡張子は `.jsonl` 、gzip圧縮なら `.jsonl.gz` です。ファイル末尾の改行は強く推奨されますが、必須ではありません。プログラムは、一レコードずつ読めます。Unix系のテキストツールやシェルのパイプでも扱いやすい形式です。
- **Parquet。** 各列の値をまとめて保持する、オープンソースのデータファイル形式です。効率のよい保存と取得のために作られ、圧縮と符号化の方式によって、複雑なデータを大量に扱えます。

最初の二つの形式で、一レコードを表すと次のようになります。注記に含むカンマのため、CSVでは引用符が必要です。

```
id,read_at,kwh,note
M-17,2024-01-08T10:00:00Z,3.21,"late, manual"

{"id":"M-17","read_at":"2024-01-08T10:00:00Z","kwh":3.21,"note":"late, manual"}
```

実際の活用例

複数の発生源から一つのデータセットを作る仕事では、電力データの報告頻度が、発生源ごとに違っていました。一時間ごとから15分ごとまであり、不規則なタイムスタンプも含まれ、一貫して比較できるデータセットを作りにくい状態でした。履歴のJSONLデータセットの生イベントを、一時間ごとの消費量へ変換しました。一時間、一地域につき一行とし、各区間の発電構成を合計し、輸入を足し、輸出を引きます。Python、JQ、SQLの三つの実装を作りました。Pythonのパイプラインは、JSONLファイルを解析してデータフレームにし、時系列を一時間間隔にリサンプリングしました。結果はCSVとして出力するか、確認用の軽量なSQLiteデータベースへ読み込みました。1日分と30日分の履歴データセットで測ると、三つの中で最速で、一日分は34 ms、三十日分は180 msでした。入力から出力への作業は、粒度の変更です。各発生源固有の間隔を、一時間、一地域につき一行という固定の形へそろえます。

データセットを移すのは、別の仕事です。六つのETL/ELTパイプラインで、IPsec VPNの先にある遠隔のウェアハウスから、Google BigQueryへデータを運びました。入力は、Parquet、CSV、.bakファイルを含む圧縮アーカイブで、Bashと Pythonのスクリプトが、種類とスキーマで各ファイルを分類しました。処理で

行をクレンジングし、正規化し、情報を補った後、整備済みのデータセットとテーブルを中間形式へ出力しました。gzipで圧縮し、Google Cloud Storageのバケットへ転送し、そこからbq CLIとPythonスクリプトでBigQueryへ読み込みました。手作業では3時間以上かかっていた転送が、自動化後は20分未満となり、週一回だったテーブルの更新も、毎日になりました。

第三は、データセットを読む仕事です。複数の地域のスマートメーターが、膨大なデータセットを生みました。それを元に、PostgreSQLとBigQueryで、純粋なSQLによる500本を超える分析レポートを作成しました。データセットには、位置座標、メーターID、タイムスタンプ付きの電力使用量、環境メタデータが含まれていました。各レポートには、同僚が再現し、監査できる注記付きSQLを付けました。トラブルシューティングの説明には、GPS座標の欠落やメーター値の破損など、よくあるデータ品質の問題と、推奨する扱いを記録しました。同僚が再現し、監査できるデータセットなら、計画担当者も信頼できます。この仕事は、運用計画の効率を10–15%高めるのに貢献しました。

問題が起きるところ

データセットの問題の多くは、データについての事実が、一人の頭の中にしか残っていないことから生じます。

粒度が書かれていません。 一時間ごとの行と十五分ごとの行を足すと、一部の区間を四回数えてしまいます。UTCで一時間、一地域につき一行など、粒度を一文で書き、合計する前に、すべての発生源をそれへそろえます。

欠測をゼロと読みます。 空のフィールド、null、0は、それぞれ別の意味です。合計では、欠けた読み取り値を、黙って使用量なしとして扱いがちです。スマートメーターの仕事では、GPS座標の欠落とメーター値の破損が多く、扱いを文書化しました。欠測を捨てるか、補うか、印を付けるかを列ごとに決め、データセットの説明に明記します。

CSVの方言がずれます。 RFC 4180は共通の形を記しますが、実際のファイルは、別の区切り文字、見出しの欠落、引用符なしのカンマ入りフィールドなど、そこから外れます。RFCでは見出しは任意で、有無を示すのはMIMEパラメーターだけです。ファイルをディスクへコピーすると、その情報は失われます。新しいファイルを読み込む前に、最初の行を、期待する見出しと照合します。

コピーが、出所を忘れます。 整備済みの抽出データを第二のシステムへ読み込むと、一か月後には元のデータと区別がつかなくなります。コピーごとに、発生源、対象期間、抽出時刻、チェックサムを添え、レポートが、どの抽出データを読んだか示せるようにします。

BigQueryのデータセットを、誤った場所に置きます。 ロケーションは、作成時に固定されます。移すには、コピーするか、データセットを作り直します。一つのクエリーに含むすべてのテーブルは、同じロケーションのデータセットにある必要があるので、最初の読み込み前に選びます。

参考資料

- Data Catalog Vocabulary（DCAT）バージョン3 — W3Cによる、データセット、配布物、データセット系列の定義と、それぞれを説明するプロパティーです。
- BigQuery：データセットの概要 — 入れ物としての意味を説明します。プロジェクト、ロケーション、命名、制限を含みます。
- JSON Lines — この形式の三つの規則と、ファイル名の慣例です。
- RFC 4180：CSVファイルの共通形式とMIMEタイプ — 多くの道具が従うCSVの形、引用符の規則、text/csvタイプです。
- Apache Parquet：概要 — 列指向のファイル形式の説明と、仕様の置き場所です。

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

「データセットとは？」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/dataset/