



# Google BigQuery & Tableau: ベストプラクティス

# はじめに

Tableau と Google BigQuery を使用すれば、使いやすいビジュアルインターフェイスを用いて大量のデータを分析し、即座に回答を得ることができます。このツールを組み合わせることで、以下のことが可能になります。

- Google BigQuery の機能を活用して、一般のユーザーが高速かつインタラクティブな分析を実現できるようにします。
- ビジュアル分析ツールを使用して、何十億ものデータ行を数秒で分析できます。1 行のコードも記述する必要はなく、サーバー側の管理も不要です。
- 魅力的なダッシュボードを数分で作成できます。ここから、自分の Google BigQuery データに接続し、最新の情報を組織に提供します。
- Tableau Server と Tableau Online を使用してレポートやインサイトを Web 上で共有できます。これにより、誰もがあらゆるデバイスからアクセスできます。
- Google BigQuery によるクラウドのアジャイル性と Tableau の驚異的なスピードの組み合わせにより、プロジェクトの価値をすばやく認識できます。

2 つのテクノロジーの最も効果的な組み合わせにより、パフォーマンスの飛躍的な向上、設計サイクルの短縮、およびユーザーと組織のさらなる成功がもたらされます。本書では、ビジュアライゼーションの即応性を最大限に高めるために、データモデリングとクエリ作成を最適化する技術について説明します。また、Tableau と BigQuery を最も費用効率よく組み合わせる方法もお伝えします。

## 執筆者

Tableau プロダクトマネージャー、Pierce Young

Tableau シニアプロダクトマネージャー、Vaidy Krishnan

Tableau シニアプロダクトマーケティングスペシャリスト、Riley Maris

Google ソリューションエンジニア、Babu Prasad Elumala

Google テクニカルプログラムマネージャー、Seth Hollyman

Google エンタープライズソリューションエンジニア、Tino Tereshko

Google ソリューションエンジニア、Mike Graboski

# 目次

テクノロジーの概要.....	4
Google BigQuery .....	4
Tableau .....	5
パフォーマンス向上のベストプラクティス: Tableau.....	6
パフォーマンスレコーダー .....	6
コンテキストフィルター .....	7
セットとグループ .....	8
最初にフィルターを追加.....	8
自動更新をオフにする.....	9
警告ダイアログ .....	9
パラレルクエリの最適化.....	10
費用とパフォーマンスのベストプラクティス: Google BigQuery .....	10
非正規化と事前結合 .....	10
日付ごとの表シャーディング .....	11
類似のクエリを多数実行する場合の配置先の指定 .....	12
Tableau を使用した、Google BigQuery ML モデルの結果の可視化 .....	12
ケーススタディ: Tableau および Google BigQuery を使って、 セルフサービス分析を実現した zulily 社からの重要なヒント .....	13
まとめ .....	13
Tableau について .....	14
その他のリソース .....	14

# テクノロジーの概要

## Google BigQuery

BigQuery はペタバイトのデータをプレーン SQL 形式で、わずか数秒で処理できます。微調整や特別なスキルは必要ありません。膨大な量のデータセットを分析する Google の画期的なテクノロジーの Dremel をベースにしており、かつて大企業が非常に大きなコストをかけて得ていたレベルのパフォーマンスを、1 GB あたり数セントで提供します。

BigQuery は、膨大な量のデータセット、構造化データセット、半構造化データセットに対して SQL クエリを実行するのに最適なデータウェアハウスです。使用事例とデータセットの例を以下に示します。

- アドホック分析
- Web ログ
- マシン/サーバーログ
- モノのインターネット (IoT) のデータセット
- E コマースでの顧客行動
- モバイルアプリデータ
- 小売分析
- ゲーミングテレメトリ
- Google アナリティクスプレミアムデータ
- 従来の RDBMS ではバッチクエリの実行に数分 (数時間) かかるデータセット

BigQuery は完全に NoOps (自動化が進み専門の管理要員を必要としない状態) かつメンテナンスフリーで、Google Cloud Platform と統合されています。他のクラウドベースの分析ソリューションと異なり、BigQuery ではサーバーのクラスタを事前にプロビジョニングする必要がありません。処理クラスタでは、実行時に BigQuery がサイズ変更とプロビジョニングを行います。

データサイズが大きくなると、BigQuery は自動的に処理能力を追加します。それでも、1 GB あたりの費用は同じです。

## レガシー SQL と標準 SQL

Google BigQuery では API のアップグレードが行われ、BigQuery SQL (現在ではレガシー SQL と呼びます) に加えて、標準 SQL も使用できるようになりました。それに伴い、Tableau も Google BigQuery コネクタをアップグレードし、標準 SQL への移行をサポートしました。標準 SQL には BigQuery ユーザーに対し、詳細レベル表現、より高速なメタデータ検証、使用中の接続で請求プロジェクトを選択できる、などのメリットがあります。本ガイドは、標準 SQL を前提にしています。

レガシー SQL から標準 SQL への移行についての詳細は、[レガシー SQL からの移行に関するオンラインヘルプガイド](#)をご覧ください。

## Tableau

Tableau Software のミッションは、お客様がデータを見て理解できるように支援することです。スタンフォード大学で開発されたテクノロジーを基盤とする Tableau のモダン分析プラットフォームを使用することで、一般のユーザーがデータの威力を活用できるようになります。これにより、幅広い層のユーザーがデータに関わり、質問をし、問題を解決し、インサイトを共有し、変革的な価値を創造することが可能になりました。従来の BI ツールのスキルレベルにかかわらず、誰もが Tableau をすばやく活用できるようになり、直感的なドラッグ & ドロップのユーザーインターフェイスにより、情報が豊富でインタラクティブなビジュアライゼーションや、効果的なダッシュボードを作成して操作できます。

Tableau プラットフォームの最近の拡張機能には、視覚的で直観的、かつスマートな **データ準備機能**、そして、**パブリッシュ済みのデータソースに対して自然言語クエリを実行**する機能が含まれています。

### Tableau のネイティブな最適化

**データソースコネクタ** — Tableau には、Google BigQuery 用に最適化されたネイティブコネクタがあり、ライブデータ接続とインメモリデータ抽出の両方をサポートしています。Tableau のデータブレンディング機能を使用すれば、BigQuery のデータを他の 67 のサポート対象データソースのデータと組み合わせることができます。Tableau Server または Tableau Online を使用してクラウドにパブリッシュされたビジュアライゼーションの場合は、Google BigQuery への直接的な接続が確保されます。

**パラレルクエリ** — Tableau は Google BigQuery およびその他のデータソースの機能を活用して、複数のクエリを同時に実行します。最大実行数は合計で 16 個に上ります。結果がキャッシュにない場合、独立したクエリと重複排除されたクエリのバッチが 1 つのグループとして、BigQuery に送信されます。BigQuery のスケールアウトアーキテクチャがもたらすパラレルクエリによって、ユーザーはパフォーマンスの大幅な向上を期待できます。

**クエリ統合** — Tableau は可能な限り、ワークブックおよびダッシュボードから複数のクエリを取得して 1 つに統合します。それにより、BigQuery に送信するクエリ数を減らすことができます。まず、Tableau は類似のクエリを特定し、戻される列で差を除外します。次に、その差が集計レベルまたはユーザーの計算のみであった場合に、クエリを組み合わせます。

**外部クエリキャッシュ** — 同じクエリを最後に実行してから参照元データソースが変化していない場合、Tableau は以前に保存されたクエリキャッシュから自動的に読み出しを行います。これにより、ほぼ瞬時のロードが可能になります。

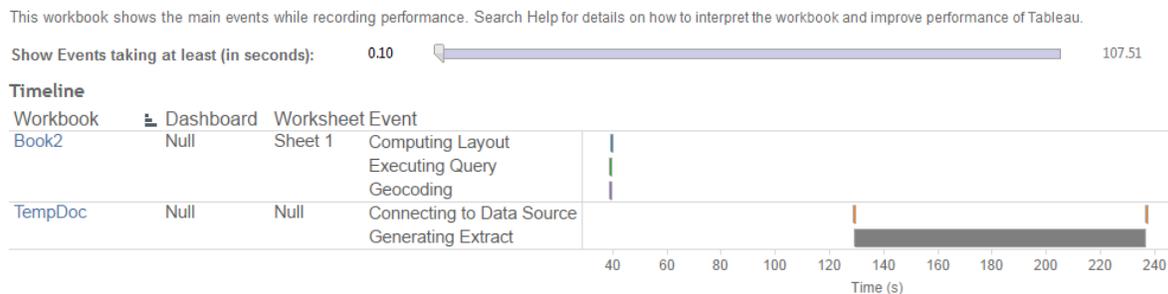
**Tableau Desktop からのオンデマンド接続** — パブリッシュされたワークブックを開く時に、Tableau Desktop は、開いているシートのデータを表示するのに必要なデータソースにのみ接続します。つまり、より高速にデータを閲覧できるということです。

# パフォーマンス向上のベストプラクティス: Tableau

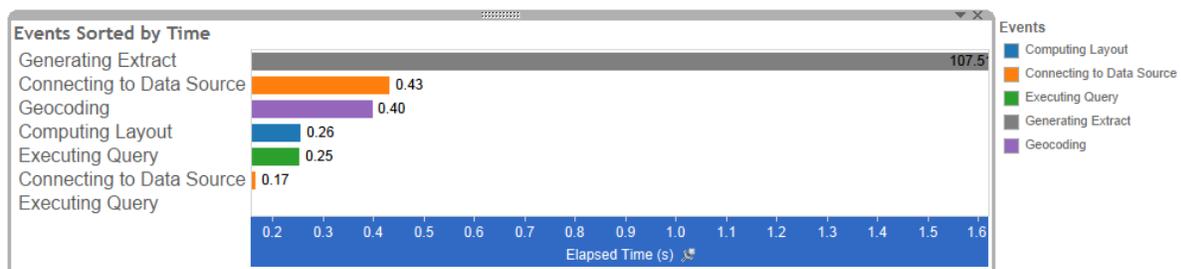
他のツールやカスタム設定の説明に入る前に、最初に推奨することは、可能なときに継続して Tableau の導入環境をアップデートすることです。そうすることで、製品の最新バージョンに常に取り入れられるパフォーマンス向上のメリットを活用することができます。

## パフォーマンスレコーダー

パフォーマンスレコーダーは、遅いクエリを正確に特定し、ワークブックを最適化して、最大のパフォーマンスを実現する強力なビルトインツールです。このために、パフォーマンスレコーダーは、個々のワークブックによるクエリ実行とレイアウト計算の所要時間を追跡します。下の緑のバー上にマウスを移動すると、BigQuery に対して生成中のクエリが表示されます。遅いクエリを特定すれば、そのデータモデルを見直すことによって、往々にしてパフォーマンスの問題を解決できます。



[タイムライン] ビューでは、[ワークブック]、[ダッシュボード]、[ワークシート] の各列でイベントのコンテキストが識別されます。



所要時間がより長いイベントは、ワークブックを高速化する場合に最初に注目すべき場所を特定するのに役立ちます。

パフォーマンス記録の作成方法または解釈方法の詳細については、下記をご覧ください。

[Tableau Desktop のパフォーマンスレコーダー \(作成\)](#)

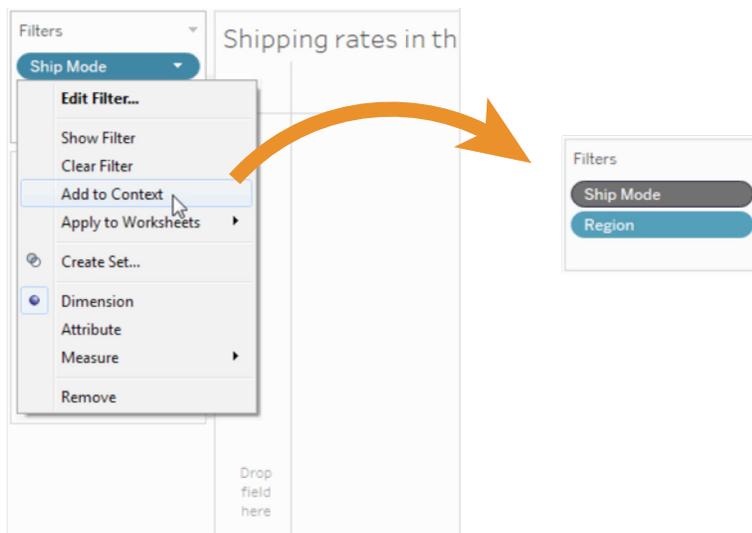
[Tableau Server のパフォーマンスレコーダー \(解釈\)](#)

## コンテキストフィルター

大規模なデータソースにフィルターを適用する場合には、コンテキストフィルターを設定することでパフォーマンスを高めることができます。コンテキストフィルターは最初にデータソースに適用されるため、追加のフィルターは結果レコードにのみ適用されます。この順序により、データソース内で各フィルターが各レコードに適用されるのを回避できます。

データセットのサイズを大幅に削減するフィルターを設定して複数のデータビューで使用する場合は、このフィルターをコンテキストフィルターとして設定する必要があります。

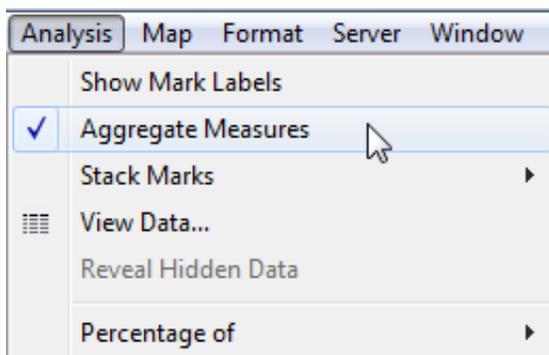
詳しくは、[コンテキストフィルターによるビューのパフォーマンス改善に関する Tableau オンラインヘルプガイド](#)をご覧ください。



1つまたは複数のコンテキストフィルターを設定することで、パフォーマンスを改善できます。

## メジャーの集計

作成するビューが遅い場合は、集計されていないメジャーではなく、集計されたメジャーで作業してください。ビューが遅いのは、行数の多いデータを一度に表示しようとするのが一般的な原因です。行数は、データを集計することで減らすことができます。



[分析] メニューでメジャーが集計されているかを確認してください。メジャーの既定集計を設定することもできます。

詳しくは、[データ集計に関する Tableau オンラインヘルプガイド](#)をご覧ください。

## セットとグループ

メジャーの値の範囲に基づきディメンションをフィルタリングしてメンバーを除外する場合は、定量的なフィルターを使用する代わりに、セットを作成します。たとえば、ディメンション内のすべてのアイテムではなく、上位 50 アイテムのみを返すセットを作成できます。

選択項目からグループを作成する場合は、必ず必要な列だけを含めるようにしてください。セットに含める列を追加すればするほどパフォーマンスは低下していきます。

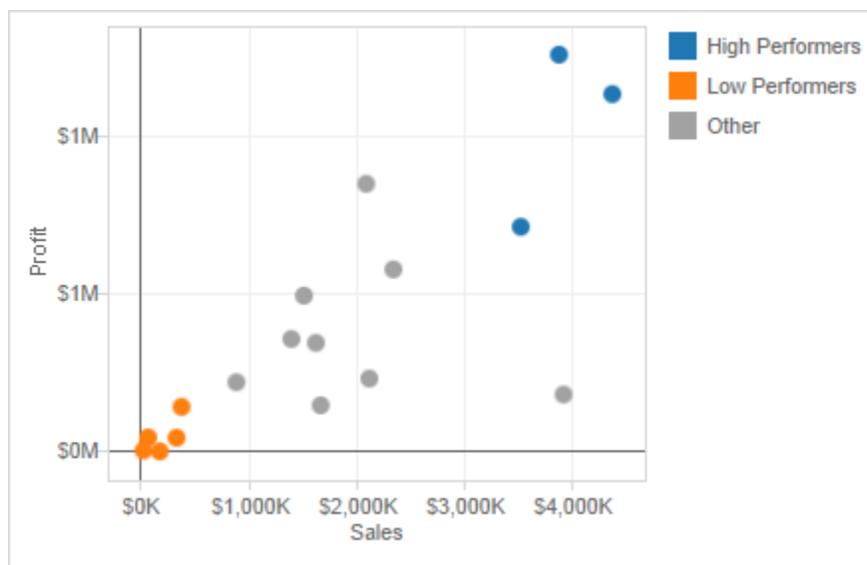


Tableau でグループを作成する際には、残りすべてのメンバーを「その他」のグループとしてグループ化するオプションがあります。

詳しくは、[セットの作成に関する Tableau オンラインヘルプガイド](#)、および、[グループの作成に関する Tableau オンラインヘルプガイド](#)をご覧ください。

## 最初にフィルターを追加

大規模なデータソースを扱う場合に自動更新がオフになっていると、ビューにフィルターを追加するときに非常に遅いクエリが作成される可能性があります。ビューを構築してからフィルターを指定するのではなく、フィルターを指定してからフィールドをビューにドラッグしてください。そうすると、更新を実行するか、自動更新をオンにしたときに、まずフィルターが評価されます。

## 自動更新をオフにする

フィールドをシェルフ上に配置すると、データソースへのクエリが自動的に実行され、Tableau によってビューが生成されます。密度の高いデータビューを作成する場合、クエリに時間がかかり、システムパフォーマンスが著しく低下する可能性があります。このような場合は、ビュー構築中はクエリをオフにするよう Tableau に指示することができます。クエリは、結果を確認する準備ができた時点で再度オンにすることができます。

詳しくは、[自動更新とパフォーマンスに関する Tableau オンラインヘルプガイド](#)をご覧ください。

## 警告ダイアログ

Tableau では、大規模なディメンション (多くのメンバーを含む) をシェルフに配置しようとする、パフォーマンスに関する警告ダイアログボックスが表示されます。このダイアログボックスでは、下図に示すような 4 つの選択肢を利用できます。[すべてのメンバーを追加] を選択すると、パフォーマンスが著しく低下する可能性があります。

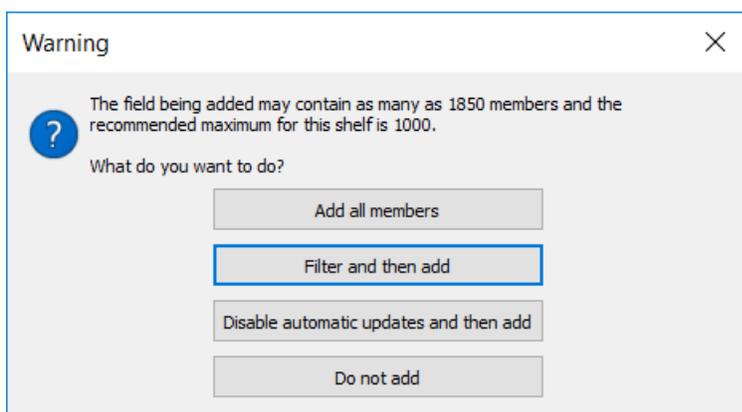


Tableau では、シェルフに大きいディメンションを配置するとパフォーマンスが低下する恐れがある場合、警告が表示されます。

## パラレルクエリの最適化

パラレルクエリを設定することで、大容量の結果セットを BigQuery から Tableau Online、Tableau Server、Tableau Desktop に返すときのパフォーマンスを向上させるために、カスタマイズ属性を使用できます。Tableau Online や Tableau Server にワークブックやデータソースをパブリッシュする前にカスタマイズ属性を指定している場合のみ、パブリッシュしたワークブックやデータソースに、その属性を含めることができます。

詳しくは、[Google BigQuery の Tableau オンラインヘルプガイド](#)内の「カスタマイズ属性を使用してクエリのパフォーマンスを向上させる」をご覧ください。

## 費用とパフォーマンスのベストプラクティス: Google BigQuery

一般的に、高パフォーマンスのクエリ実行とコストの削減を実現する最も良い方法は、データが Google Cloud Storage のような外部データソースにある、フェデレーションテーブルを使わないようにすることです。その場合、データセットに対して繰り返しクエリを実行するのであれば、(Tableau の外で) クエリ API を利用して BigQuery 内でデータを計算済みにすることにより、データセットに対する Tableau のクエリ実行パフォーマンスを高めることができます。

## 非正規化と事前結合

BigQuery は大規模な結合をサポートしており、結合のパフォーマンスにも非常に優れています。しかし同時に、BigQuery はいわゆる「カラム型データストア」であり、非正規化データセットに対して最大のパフォーマンスを発揮します。

クラウドのメリットの 1 つとして、ストレージリソースとコンピュートリソースを切り離して、ユーザーがそれぞれを独自にスケールリングおよび支払えるようになることが挙げられます。BigQuery のストレージは非常に安価でスケラブルなので、多くの場合、データセットを同質の表に非正規化し、事前結合することが賢明です。つまり、より少ないコンピュートリソースを使用する代わりに、より多くのストレージリソースを使用することになります (後者の方がパフォーマンスと費用対効果に優れています)。カラム型ストアである BigQuery はデータ圧縮機能に優れているため、コンピュートリソースの代わりにストレージリソースを使用することは選択肢として悪くはなく、費用も抑えられる可能性があります。

BigQuery は優れた ETL ツールであり、大量の変換とパイプライン処理を迅速かつ効率的に実行できます。128 MB を超えるデータセットを計算済みにするときは、必ず [Allow Large Results] (大きなクエリ結果を許可する) を有効にしてください。

読み込むデータの準備方法や、BigQuery の SQL ダイアレクトを使ったデータのクエリ方法の詳細については、下記をご覧ください。

[非正規化されたデータ、ネストされたデータ、繰り返しデータの読み込み  
サイズの大きいクエリ結果を書き込む](#)

## 日付ごとの表シャーディング

表を複数の小さなパーティションに分割する「シャーディング」を行うことで、データ管理を簡素化し、クエリのパフォーマンスおよび費用対効果を高めることができます。さらに、BigQuery では分割された表のクラスタリングもサポートしているため、データがすでに日付やタイムスタンプの列に基づいて分割されている場合や、クエリで特定の列に対してフィルターまたは集計が設定されている場合に役立ちます。

一部のデータは、もともと日付ごとの区分の指定に適しています。たとえば、ログデータや、単調に増加するタイムスタンプをレコードに含むデータなどです。この場合は、BigQuery の表を日付ごとにシャーディングし、その日付を表名に含めます。この機能を活用するには、Tableau でカスタム SQL を使用する必要があります。

詳しくは、[カスタム SQL クエリへの接続に関する Tableau オンラインヘルプガイド](#)をご覧ください。

たとえば、表に mytable\_20170501、mytable\_20170502 などの名前を付けます。

次に、日付ごとにフィルタリングを行うクエリを実行する場合は、BigQuery のワイルドカードテーブルの機能を使用します。

```
SELECT
  name
FROM
  `myProject.myDataSet.mytable_*`
WHERE
  age >= 35
```

上の例では、「mytable\_」という接頭辞が付いたすべての表が自動的に含まれます。

ワイルドカードを使用するには、以下のパターンに従って表に名前を付ける必要があります。[任意の接頭辞]YYYYMMDD。

その他のデータベースシステムは、シャーディングを使用してパフォーマンスの向上を図っています。実際には、日付ごとのシャーディングを使用しても、BigQuery でのパフォーマンスにはほとんど差はありませんが、費用が大きく異なります。処理するデータ量が少なければ、クエリ単位の費用も少なくなるためです。

ここで注意しなければならないのは、分の単位でシャーディングするとシャードが多くなりすぎることがあり、その場合はパフォーマンスに直接影響するという点です。一度にシャーディングを行いすぎないように気をつけてください。一般的には、1日に一度程度であれば問題はありません。

シャーディングの詳細については、下記をご覧ください。

[分割テーブルの概要](#)

[クラスタ化テーブルの概要](#)

[ワイルドカードテーブルを使用した複数テーブルに対するクエリ](#)

## 類似のクエリを多数実行する場合の配置先の指定

同一のクエリを大量に実行する場合はクエリキャッシュが有効です。ただし、似てはいるもののわずかに異なるクエリ (クエリの実行の間に WHERE 節の値のみを変更したものなど) の場合は役に立ちません。この場合は、ソースの表でクエリを実行し、クエリを繰り返し実行するレコードを新しい配置先の表に書き込みます。そのあとで、作成した新しい配置先の表にクエリを実行します。

たとえば、以下の 3 つの異なる WHERE 節を使用して 3 つのクエリを実行するとします。

```
WHERE col1 = "a"  
WHERE col1 = "b"  
WHERE col1 = "c"
```

ソースの表にクエリを実行し、出力レコードを配置先の表に書き込みます。

```
SELECT col1  
FROM source  
WHERE col1 = "a" OR col1 = "b" OR col1 = "c"
```

WHERE 節の論理和演算によって、関連するレコードをすべてキャプチャします。新しい配置先の表は、元のソースの表よりもかなり小さくなる可能性があります。BigQuery は 1 つのクエリ内で処理されるデータ量に基づいて課金します。このため、後続のクエリを、ソースの表に直接実行するのではなく、新しい配置先の表に実行することで、費用を節約できます。なお、こうした表によりストレージコストが上がっていくのを避けるために、いずれかの時点で表を削除してください。

## Tableau を使用した、Google BigQuery ML モデルの結果の可視化

BigQuery ML によって、ユーザーは、埋め込みの機械学習テクノロジーを使用して、BigQuery に格納されているデータに基づきモデルをトレーニングすることができます。ただし、他のいかなるデータでの作業と同じように、データベースに対して直接クエリを実行することは、モデルの結果を探索する上で、必ずしも最適な方法であるとは言えません。

Tableau には BigQuery コネクタが用意されているため、データの直観的な理解を促進させる形で、予測モデルの結果を簡単に操作することができます。さらに、Tableau では、作成したモデルおよびその結果を他のユーザーと簡単に共有できるため、すべての作業を他のユーザーが活用できるようになります。

例をご覧ください。には、[BigQuery ML と Tableau を使用した住宅価格の予測](#)についてお読みください。

この演習では、Google BigQuery ML で機械学習アルゴリズムを呼び出すために、カスタム SQL が必要となりますが、他のシナリオで、Tableau のネイティブ接続の代わりに、カスタム SQL を使用する場合には、パフォーマンス上の考慮事項に注意する必要があります。可能な限り、Tableau のネイティブデータソース接続を使用することで、最適なパフォーマンスを実現できます。

標準 SQL クエリを使用した機械学習モデルの作成および実行に関する詳細は、[Google BigQuery ML のオンラインヘルプガイド](#)をご覧ください。

# ケーススタディ: Tableau および Google BigQuery を使って、セルフサービス分析を実現した zulily 社からの重要なヒント

急成長している e コマース企業である **zulily 社**は、ビジネスデータウェアハウスとして Google BigQuery、データへのアクセスとビジュアル分析には Tableau を使用することで、ビッグデータプラットフォームを構築しました。BigQuery と Tableau を統合したことで、分析担当者が、日常業務で IT 部門に頼らずに、データの取得、取り込み、利用にすぐ取りかかり、レポートやモデルを作成できるようになりました。さらに、ビジネスユーザーが、シンプルなインサイトを得るのにアナリストの助けを借りる必要がなくなり、迅速に意思決定を行うための重要なデータをリアルタイムで利用できるようになりました。

下記に、zulily 社のベストプラクティスをいくつか記載しています。

**Google Compute Engine で Tableau Server を使用することでレイテンシーを低減** — 各リージョンが別々の VPC に分散される従来型のモデルではなく、Google のプライベートバックボーンを活用できます。インターネットを経由する必要がなく、追加のセットアップも不要です。また、過剰にプロビジョニングすることなく、規模に応じたデプロイメントも可能となります。

**フェデレーションソースを使用し、Tableau で BigQuery を指定** — Google Cloud 内のデータについては、外部データソースに対するクエリ実行を可能とする BigQuery の機能を活用し、BigQuery をデータレイクとして利用します。特定のシナリオでは、分析のためにネットワークおよび Tableau にプッシュする必要があるデータ量を減らします。

**BigQuery でライブ接続を使用して大規模なデータセットを処理** — BigQuery の機能を活用して、大規模なデータセットを処理し、結果のみをネットワーク上に取り込みます。データを抽出する特別な理由がない限り、BigQuery に対する Tableau の既定接続は「ライブ」に設定してください。

詳細および 10 個のヒントのすべてについて確認するには、下記の 2 つのパートで構成されているブログシリーズをご覧ください。

**パート 1:** zulily 社が Tableau と Google BigQuery でセルフサービス型のマーケティング分析プラットフォームを構築した理由

**パート 2:** Google BigQuery および Tableau を使って、セルフサービス分析を実現した zulily 社からの 10 の重要なヒント

## まとめ

ベストプラクティスを適用することによって、ビジネスユーザーもデータアナリストも、Google BigQuery に対して構築された Tableau ビジュアライゼーションのパフォーマンスと応答性を最大限に高めることができます。このようなテクノロジーの組み合わせにより、ユーザーは実際に数十億のデータ行をわずかな時間で視覚化できます。

## Tableau について

Tableau は、規模に応じた超高速セルフサービス分析を通じてお客様がデータを見て理解できるように支援する、全てがそろった使いやすいエンタープライズ対応のビジュアル BI プラットフォームです。オンプレミスでもクラウドでも、また Windows でも Linux でも、Tableau はテクノロジーへの既存の投資を生かし、お客様のデータ環境の変化と成長に合わせた規模の拡大が可能です。最も貴重なアセットであるデータと人の力を解き放ちます。

## その他のリソース

[Tableau 無料トライアル](#)

[Tableau オンラインヘルプガイド: Google BigQuery](#)

[Tableau Server と Google Cloud Platform: クラウド上の超高速処理ビジネスインテリジェンス](#)

[Tableau と Google のソリューション](#)

[Tableau とビッグデータ: 概要](#)

[ビジネス分析をクラウドで行う理由](#)

[効率的に作業できるワークブックの設計](#)

