

ATHENA FEDERATED QUERY & AMAZON QUICKSIGHT ハンズオン

Amazon Web Services Japan

第 1 版 2021/03/26 作成, 最終更新: 2021/3/26

目次

本資料について	3
ハンズオン全体像	3
事前準備・確認	3
ハンズオン実施のために必要な環境	4
ハンズオン環境のセットアップ	5
ハンズオンの環境について	5
CloudFormation テンプレートの実行	5
作成された環境の確認	7
Transaction.CSV ファイルを DynamoDB にロード	9
Athena Federated Query で DynamoDB のデータをクエリする	12
Athena のワークグループを設定し V2 エンジンを利用可能にする	12
Federated Query を設定し DynamoDB に接続する	15
DynamoDB のデータを Athena からクエリする	20

S3 上のデータと DynamoDB の表をジョインする	23
S3 に master.csv を配置する	23
Glue クローラーでカタログ登録する	25
Athena から S3 上の master.csv にクエリする	31
QuickSight で可視化する	34
QuickSight で Athena へのアクセスを設定する	34
データセットとして Athena のクエリを登録する	37
QuickSight での可視化の操作	42
フィールドの単位を調整する	44
可視化①「地域別レベニュー」	45
可視化②「総利益」	48
可視化③「レベニュー推移」	49
可視化④「レベニュー推移」に予測を追加する	51
フィルタ①item type のフィルタコントロール	52
フィルタ②年単位のフィルタコントロール	54
後片付け	58

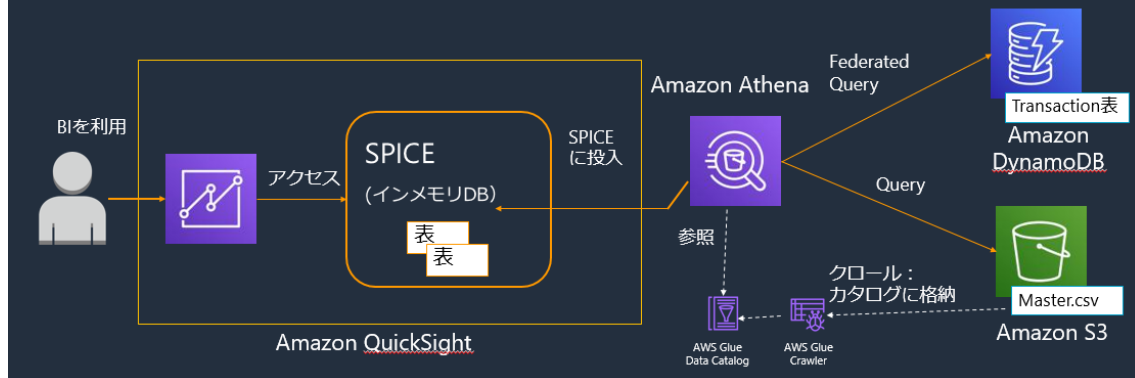
本資料について

ハンズオン全体像

本ハンズオンでは、Amazon Athena Federated Query を利用し、Amazon DynamoDB 内のデータに SQL でアクセスし、その結果を Amazon QuickSight で可視化するまでを体験します。

ハンズオン全体像

- ① Athena の Federated Query で DynamoDB のデータをクエリする
- ② S3 上の CSV データと DynamoDB のデータをジョインする
- ③ クエリ結果を SPICE に格納し、QuickSight で可視化する



- ① 最初にサンプルデータ (Transaction.csv) を DynamoDB に投入し、それを Athena からクエリするまでを体験します。Transaction.csv は販売実績のデータです。
- ② 次に S3 上に Master.csv を配置し、Glue クローラーを動作させてカタログに保存することで、Athena から Master 表として認識させます。この Master 表と Transaction 表とでジョインできる事を確認します。Master.csv は製品単価等が保存されたマスターデータです。
- ③ 最後にジョインした結果を QuickSight の SPICE に取り込み、QuickSight 上での可視化を体験します。

事前準備・確認

本ハンズオンを開始する前に、QuickSight のサインアップもしくは利用リージョンの確認が必要です。本資料を進める前に別資料「事前準備」を参照して事前に確認・準備をしておいてください。

ハンズオン実施のために必要な環境

ハンズオンを実施するには、AWS アカウント、ブラウザ（**Chrome か Firefox の最新版を推奨**）とインターネットに繋がる環境が必要です。ブラウザと QuickSight 間の通信は、HTTPS に加えて WebSockets Secure(wss://)を使用しますので、ファイアーウォール等で防がれていないかご確認ください。

また、利用ブラウザのプラグイン・アドオンに御注意ください。特に自動翻訳のプラグインが管理コンソールや QuickSight の動作を妨げる事がありますので事前の停止をおすすめします。

ハンズオン環境のセットアップ

ハンズオンの環境について

最初に AWS CloudFormation により環境を作成します。提供される CloudFormation テンプレート(JSON ファイル)を実行することで、以下の作成が行われます。

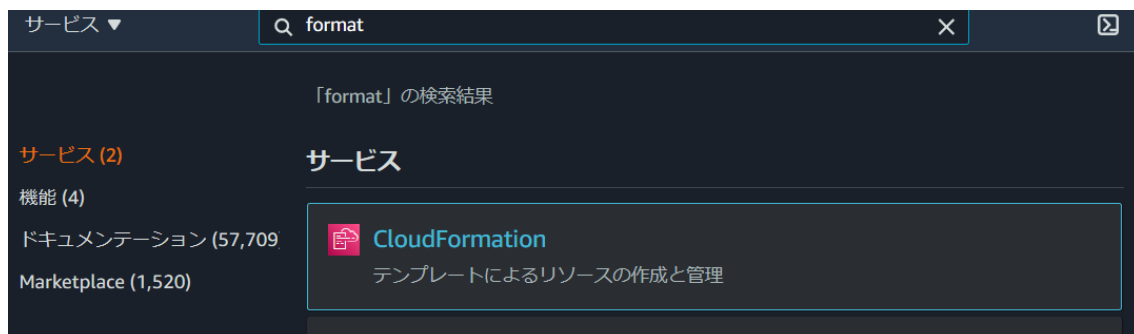
- DynamoDB の表 (**Transaction 表**)
- ハンズオン用 CSV ファイルを DynamoDB にアップロードするための Lambda 関数
- ハンズオンで必要となる、以下の S3 バケット
 - **ddb-upload-** から始まるバケット： DynamoDB にアップロードする CSV ファイル置き場
 - **athena-ws-** から始まる名前のバケット： Athena のクエリ実行結果や、Master 表等を保存する場所
 - **federated-spill-** から始まる名前のバケット： Federated Query 設定時に必要となる Spill バケット

CLOUDFORMATION テンプレートの実行

AWS マネジメントコンソールにログインします。ここで、コンソールで操作しようとしているリージョンが QuickSight のホームリージョンと同じかを必ずご確認ください。 異なっている場合は、管理コンソールの右上からリージョンを切り替えて合わせてください。（本資料では、東京リージョンを例に説明をしますが、ご利用の QuickSight ホームリージョンに読み替えてご利用ください）。



コンソールから CloudFormation を選択します。画面上部の検索枠に「formation」等とタイプすると、一覧から絞り込まれます。



CloudFormation の画面に遷移したら、「スタックの作成」をクリックします。



スタックの作成画面になりますので、「テンプレートのアップロード」にチェックを入れ、「ファイルの選択」をクリックし、配布された JSON ファイル（CSVToDynamo-transaction.json）をアップロードし、「次へ」を押します。

スタックの詳細を指定

スタックの名前

スタックの名前

myhandson

スタックの名前は任意ですが、
半角アルファベットか数字のみ利用してください

スタック名では、大文字および小文字 (A-Z~a-z)、数字 (0-9)、ダッシュ (-) を使用することができます。

パラメータ

パラメータは、テンプレートで定義されます。また、パラメータを使用すると、スタックを作成または更新する際にカスタム値を入力できます。

DynamoDBTableName

Name of the dynamoDB table you will use

transaction

FileName

Name of the S3 file (including suffix)

transaction.csv

キャンセル 戻る 次へ

スタックに名前を付けます。半角のアルファベットもしくは数字のみで名前を付けてください。ここでは myhandson としています。下段のパラメータの部分は通常変更する必要はありませんので、そのままにして「次へ」を押します。

その次の「スタックオプションの設定」はそのまま変更せず「次へ」を押します。

機能

The following resource(s) require capabilities: [AWS::IAM::Role]


このテンプレートには、ご利用の AWS アカウントに変更を加えるエンティティにアクセスを与える可能性を持つ Identity and Access Management (IAM) リソースが含まれています。これらのリソースを個別に作成し、それぞれに最小限必要な権限を与えるかどうか確認してください。 [詳細はこちら](#)

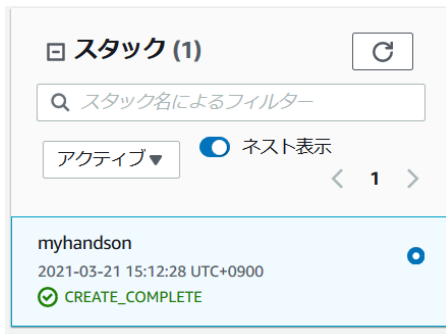
☒ AWS CloudFormation によって IAM リソースが作成される場合があることを承認します。

キャンセル 戻る 変更セットの作成 スタックの作成

最後にレビューの画面になりますので、画面最下部までスクロールし、上記のように「AWS CloudFormation によって IAM リソースが…」にチェックを入れて「スタックの作成」を押します。これで前述の環境作成が自動的行われます。(1-2 分で完了します)

作成された環境の確認

画面左上にあるスタックの作成状況が CREATE_COMPLETE になるのを待ちます。画面に変化がない場合は  ボタンを押すことで画面をリフレッシュしてください。以下のようになるとスタックが完成しています。



スタックが完成したら、リソースタブをクリックします。

スタックの情報 イベント リソース 出力 パラメータ テンプレート 変更セット							
リソース (7)							
リソースの検索							
論理 ID	物理 ID	タイプ	ステータス	状況の理由	モジュール		
AthenaBucket	athena-ws-936203727359-ap-northeast-1	AWS::S3::Bucket	CREATE_COMPLETE	-			
BucketPermission	myhandson-BucketPermission-U5S7PVVW40SF	AWS::Lambda::Permission	CREATE_COMPLETE	-			
CsvToDDBLambdaFunction	myhandson-CsvToDDBLambdaFunction-1JJZ0ANA0B5OD	AWS::Lambda::Function	CREATE_COMPLETE	-			
DynamoDBTable	transaction	AWS::DynamoDB::Table	CREATE_COMPLETE	-			
LambdaRole	myhandson-LambdaRole-1FX4OHD98RVA0	AWS::IAM::Role	CREATE_COMPLETE	-			
SpillBucket	federated-spill-936203727359-ap-northeast-1	AWS::S3::Bucket	CREATE_COMPLETE	-			
UploadBucket	ddb-upload-936203727359-ap-northeast-1	AWS::S3::Bucket	CREATE_COMPLETE	-			

このように CloudFormation で作成された AWS リソースが表示されます。

TRANSACTION.CSV ファイルを DYNAMODB にロード

今回のハンズオンでは、transaction.csv ファイルのデータを DynamoDB の transaction 表にロードします。CloudFormation のリソースタブから“ddb-upload-…”で始まる名前の S3 バケットをクリックして S3 の画面に移動します。



上記のように ddb-upload-バケットが表示されますので、画面下部の「アップロード」のところに、transaction.csv ファイルをドラッグ&ドロップします。

アップロードの確認画面になるので、画面下部の「アップロード」ボタンをクリックして、ファイルをアップロードします。少し待つとアップロードが完了しますので、右上の「閉じる」ボタンで元の画面に戻ります。



今回実行した CloudFormation ではこのフォルダに tansaction.csv という名前のファイルをアップロード（PUT）した際に Lambda 関数が呼び出され、csv ファイルの内容が DynamoDB の transaction 表に自動的にロードされるようになっています

補足

ハンズオンで利用しているアップロードの Lambda 関数や transaction.csv、master.csv は以下で公開されているものを元にハンズオン用に修正したものです。

<https://aws.amazon.com/jp/blogs/news/implementing-bulk-csv-ingestion-to-amazon-dynamodb/>

<https://github.com/aws-samples/csv-to-dynamodb>

次に DynamoDB を確認します。画面上部の検索枠に「Dynamo」等と打ち込むと DynamoDB に絞り込まれますので、クリックして DynamoDB の管理画面に遷移します。


DynamoDB の管理画面で左側の「テーブル」をクリックすると、**transaction** という表があるのが確認できます。これをクリックします。

画面右側のタブから「項目」を選択し、データを確認してください。

The screenshot shows the AWS DynamoDB console interface. On the left, the 'DynamoDB' dashboard sidebar has the 'テーブル' (Tables) option highlighted with a red box. The main area shows a list of tables with 'transaction' selected and highlighted with a red box. On the right, the 'transaction' table's '項目' (Items) tab is selected and highlighted with a red box. Below the tabs, there is a search bar and a table of items. The table has columns for 'Transaction ID' and 'Country'.

Transaction ID	Country
115096652	Haiti
207517850	Egypt
207795083	North Korea
217867314	San Marino

CSV ファイルを S3 に配置してから Lambda が起動して DynamoDB に格納されるまで、少し時間がかかります。上記のように格納されたデータが表示されなかった場合は、少し待ってから

 ボタンでリロードしてください。

これでハンズオンの環境が準備できました。

ATHENA FEDERATED QUERY で DYNAMODB のデータをクエリする

DynamoDB 上に以下の transaction 表が準備できました。これは販売記録です。

Transaction ID ⓘ	Country	Item ID	Order Date	Order Priority	Region	Sales Channel	Ship Date	Units Sold
115096652	Haiti	1	12/4/12	L	Central America and the ...	Offline	1/13/13	8512
207517850	Egypt	1	2/28/13	C	Middle East and North Afr...	Offline	3/20/13	7998
207795083	North Korea	11	7/15/17	H	Asia	Offline	7/25/17	38
217867314	San Marino	3	1/13/12	H	Europe	Offline	2/16/12	635

Transaction ID は売り上げトランザクションを一意に指す ID です。Item ID は売れたアイテムを Item の ID で記録しています（アイテムの詳細は master 表にあり transaction にはありません）。その他、販売した地域（Region, Country）や、販売チャンネル(Sales Channel)、オーダーと出荷日時、アイテムの販売個数等が読み取れます。

この Transaction 表を Athena からクエリできるように設定します。

ATHENA のワークグループを設定し V2 エンジンを利用可能にする

画面上部の検索枠に「Athena」と打つ、もしくは画面左上のサービス一覧から Athena を選択し、Athena の画面に移動します。

リージョン内で初めて Athena を利用する場合は「今すぐ始める」を押して Athena の利用を開始してください。



上記画面で、まずワークグループを設定します。ワークグループというのは、利用する Athena のエンジンや、データの出力先設定などをまとめて定義しておくものです。初期状態では”primary”というワークグループを使うようになっていますが、今回はハンズオン用の”handson”ワークグループを作成します。

画面上部の「ワークグループ」をクリックします。次のワークグループ作成画面で、「ワークグループを作成する」ボタンをクリックし、以下のように入力してください。

ワークグループを作成する

ワークグループの一意の名前を選択します。ワークグループ名を変更するには、ワークグループを削除した上で、新しい名前で作成する必要があります。 [詳細はこちら](#)

全般構成

任意のワークグループ名を入力

ワークグループ名*

handson

0-30 文字で入力してください。英数字、下線 (_)、ハイフン (-) にすることができます。)

説明

ワークグループの説明を入力します。

最大 1024 文字使用します。

クエリの結果の場所 および 暗号化

「選択する」ボタンをクリックし、athena-ws-... で始まるバケットを選択

クエリの結果の場所

s3://athena-ws-936203727359-ap-northeast-1/

選択する

S3 パスの末尾にはスラッシュが必要です (例: s3://query-results-bucket/folder/)。

クエリ結果の暗号化

☐ S3 に保存されている結果を暗号化する

クエリエンジンのバージョン

使用する Athena エンジンバージョンを指定するか、ワークグループをアップグレードするタイミングを Athena に選択させます。Athena は、 [こちら](#)

クエリエンジンを更新

☐ ワークグループをアップグレードするタイミングを Athena に選択させる。 ⓘ

☒ エンジンバージョンを手動で選択する。

エンジンバージョンを手動で選択するにし、Athena engine version 2を選択

☒ Athena engine version 2 (推奨)

このバージョンの機能には、Parquet 形式のネストされたスキーマの展開に関するサポートや、結合演算子と集計演算子でのコスト削減とパフォーマンス改善のための、ネストされたスキーマの読み込みに関するサポートなどが含まれています。 [詳細はこちら](#)

☐ Athena engine version 1

これは Athena の初期バージョンです。

ワークグループ名には handson 等任意の名前を入力しクエリ結果の場所は、CloudFormation で自動作成された athena-ws-... から始まるバケットを指定してください (バケット名の右側に

ある▲印を押すと選択できます）。またクエリエンジンには明示的に Athena engine version 2 をセットしてください。それ以外はデフォルトのままで、最下部の「ワークグループを作成する」をクリックします。



上記の画面に戻ったら、今作成したワークグループを選択して、「ワークグループを切り替える」をクリックします。

Athena のクエリエディタ画面に戻ります（再度ウェルカム画面が表示された場合は「「今すぐ始める」をクリックしてください」）。クエリエディタ右下の表記が“Athena engine version 2”になったことを確認してください。



次に画面上部の「設定」をクリックし、再度クエリの結果の場所に、先ほどと同じ S3 バケット (athnea-ws-...から始まるバケット) を設定し、保存を押します。

設定


×

デフォルトでは、すべての新しいクエリに設定が適用されます。 [詳細はこちら](#)

ワークグループ: **handson**

クエリの結果の場所

s3://athena-ws-936203727359-ap-northeast-1/

 選択する

S3 パスの末尾にはスラッシュが必要です (例: s3://query-results-bucket/folder/)

クエリ結果の暗号化 ☐ ⓘ

自動入力 ☐ ⓘ

キャンセル

保存

これで Athena engine v2 を利用してクエリを実行する準備が整いました。



FEDERATED QUERY を設定し DYNAMODB に接続する

Athena の Federated Query は、アクセス用の Lambda 関数を準備することで、多様なデータソースへのクエリを可能にします。DynamoDB 等については、GUI から選択するだけで Lambda 関数のセットアップが簡単に行われるようになっています。

画面上部の「データソース」をクリックし、次の画面で「データソースを接続する」をクリックします。











データの場所を選択する

Athena はデータがある場所でクエリを実行します。データはロードまたは移動されません。 [詳細はこちら](#)

<input type="radio"/> Amazon S3 でデータのクエリを実行する 外部データカタログを選択します。 	<input checked="" type="radio"/> データソースのクエリを実行する 一般的なデータソース用のコネクタを設定します。 
--	--

データソースを選択する

Athena でクエリを実行するデータソースを選択します。データソースを選択したら、接続を処理するように Lambda 関数を設定します。 [詳細はこちら](#)

<input type="radio"/>  Amazon CloudWatch Logs	<input type="radio"/>  Amazon CloudWatch メトリクス
<input type="radio"/>  Amazon DocumentDB	<input checked="" type="radio"/>  Amazon DynamoDB
<input type="radio"/>  Amazon Redshift	<input type="radio"/>  Apache HBase
<input type="radio"/>  MySQL	<input type="radio"/>  PostgreSQL
<input type="radio"/>  Redis	<input type="radio"/>  他のすべてのデータソース 独自のデータコネクタを作成する

[キャンセル](#)

[次へ](#)

上記のように「データソースのクエリを実行する」を選択し、下側で DynamoDB を選択し、「次へ」を押します。

データソース接続の画面になります。

接続の詳細: Amazon DynamoDB

データソースに接続するように設定された Lambda 関数を選択するか、接続を処理する Lambda 関数を作成して設定します。 [詳細はこちら](#)

Lambda 関数 データソースに接続するための新しい AWS Lambda 関数を選択または設定します。

Lambda 関数の選択

[新しい AWS Lambda 関数を設定](#)

上図で「新しい Lambda 関数を設定」ボタンを押すと、別のブラウザタブが開き、Lambda 関数のセットアップ画面に遷移します。



画面右下に設定項目がありますので、以下のように設定します。

アプリケーションの設定

アプリケーション名
AWS CloudFormation 経由で作成されたこのアプリケーションのスタック名

AthenaDynamoDBConnector

SpillBucket
The name of the bucket where this function can spill data

federated-spill-936203727359-ap-northeast-1

▼ ConnectorConfig

AthenaCatalogName
The name you will give to this catalog in Athena. It will also be used as the function name. This name must satisfy the pattern `^[a-z0-9-_.]{1,64}$`

ddbcatalog

DisableSpillEncryption
WARNING: If set to 'true' encryption for spilled data is disabled.

false

LambdaMemory
Lambda memory in MB (min 128 - 3008 max).

3008

LambdaTimeout
Maximum Lambda invocation runtime in seconds. (min 1 - 900 max)

900

SpillPrefix
The prefix within SpillBucket where this function can spill data.

athena-spill

☒ このアプリがカスタム IAM ロールを作成することを承認します。 [情報](#)

キャンセル 戻る デプロイ

SpillBucket には、CloudFormation で作成した Spill 用バケットである「federated-spill-…」で始まる名前を記入してください（CloudFormation スタックのリソース画面から物理 ID をコピーして利用してください）。

※補足：Spill バケットとは Federated Query の Lambda 関数の中のメモリやストレージにデータが入りきらなかった場合に一時的なデータ置き場として利用されるバケットです。

AthenaCatalogName には、図のように ddbcatalog と入力します（任意の名前の設定も可能ですが、**半角のアルファベット、数字、アンダースコアのみ**利用可能である事に注意してください）。設定した AthenaCatalogName は Lambda 関数の名前としても利用されます。

最後に「このアプリがカスタム IAM ロールを作成することを承認します」にチェックを入れて、「デプロイ」を押します。一分ほど待つと、以下のような画面に遷移します。これで DynamoDB コネクタの Lambda 関数が準備されました。



ブラウザの別タブにある、Athena のデータソース設定画面をひらきます。

接続の詳細: Amazon DynamoDB

データソースに接続するように設定された Lambda 関数を選択するか、接続を処理する Lambda 関数を作成して設定します。 [詳細はこちら](#)

Lambda 関数 データソースに接続するための新しい AWS Lambda 関数を選択または設定します。

ddbcatalog

新しい AWS Lambda 関数を設定

Lambda 関数の ARN [arn:aws:lambda:ap-northeast-1:936203727359:function:ddbcatalog](#)

カタログ名 SQL ステートメント内でこのデータソースを指定する一意の名前を作成します。

myddb

最大 128 文字まで使用でき、アスタリスク (*) でのみ使用できます。作成後に変更することはできません。有効な文字は、a~z、A~Z、0~9、_ (アンダースコア)、@ (アット)、- (ハイフン) です。

**説明
(オプション)**


説明を入力 (オプション)

最大 1024 文字使用します。

キャンセル

戻る

接続

「Lambda 関数の選択」の右側にある円状のボタン()を押すと、先ほど作成した ddbcatalog 関数が選択できるようになりますので、選択します。

カタログ名には myddb と入力します (任意の名前を設定可能です。本ハンズオンでは myddb という名前を前提に SQL を記述していますので、別の名前にした場合は適宜読み替えてください)。

最後に「接続」をクリックします。データソース一覧の画面に戻ってきます。myddb データソースが存在する事を確認して、画面上部の「クエリエディタ」を押して最初の画面に戻ります。

データソース

Athena が接続できるデータソースは、カタログ名で以下に記載されています。Athena を複数のデータソースに接続し、その場所のデータを確認。詳細はこちら [🔗](#)

[データソースを接続する](#) [詳細を表示する](#) [編集](#) [削除する](#)

フィルタ: データソースをフィルタリングする

カタログ名 ▲	カタログタイプ ▼
<input type="radio"/> AwsDataCatalog 🔗	AWS Glue data catalog
<input type="radio"/> myddb	Data source connector

DYNAMODB のデータを ATHENA からクエリする

クエリエディタに戻り、左側のデータソースとして myddb を選択してください。

すると、myddb 経由で DynamoDB の transaction 表が見えるようになっており、Athena からクエリできるようになったことが分かります。

データソース [データソースを接続する](#)

myddb ▼

データベース

default ▼

テーブルとビューのフィルタリング...

▼ テーブル (1)

▶ transaction [⋮](#)

Athena のクエリでオブジェクト（表やビュー）を指定する場合、以下のように記述します

“データソース名”.データベース名”.テーブル名(ビュー名)”

以下のクエリをクエリエディタに入力し、「クエリの実行」を押して実行してください。

```
SELECT COUNT(*) FROM "myddb"."default"."transaction" ;
```

10 秒ほどで応答が返ってきます。結果から transaction 表は 10 万行ある事がわかります。

データソース データソースを接続する

myddb

データベース

default

テーブルとビューのフィルタリング...

▼ テーブル (1)

▼ transaction

- Item ID (varchar)
- Units Sold (varchar)
- Ship Date (varchar)
- Sales Channel (varchar)
- Order Priority (varchar)
- Country (varchar)
- Region (varchar)
- Order Date (varchar)
- Transaction ID (varchar)

新しいクエリ 1

1 SELECT COUNT(*) FROM "myddb"."default"."transaction" ;

クエリの実行 名前を付けて保存 (実行時間: 9.64 秒, スキャン)

クエリの実行には Ctrl + Enter、オートコンプリートには Ctrl + Space を使います

結果

	_col0 ▼
1	100000

現在、画面の左側でデータソースは myddb、データベースは default を選択している状態です。これは、クエリの中でデータソースやデータベース名を明示しなかった場合のデフォルト値として利用されます。つまり、

```
SELECT COUNT(*) FROM "transaction" ;
```

としても同じ結果になります。

なおデータベース名の指定方法は、アクセス先のデータソースによって異なります。

DynamoDB の Federated Query の場合はデータベースという概念が無いため、常に default というデータベース名を指定します。

画面左の transaction 表の左側にある ▲ をクリックすると、transaction 表の列定義が表示できます。ここでいくつか自由に transaction 表にクエリを実行してみてください。もしくは以下にサンプルのクエリを記述しておきますので、コピー＆ペーストで利用してください。

クエリ例：（query-samples.sql ファイルにも同じ内容が記載されています）

```
/* Sales Channel において、オンラインとオフラインのトランザクション数比較
   以下で"Sales Channel"になっている 2 か所を"Region"等に替えるとリージョン単位の比較
   になります */
SELECT "Sales Channel", COUNT(*) "Count"
  FROM "myddb"."default"."transaction"
 GROUP BY "Sales Channel";

/* アイテム ID 別での売り上げ個数(Units Sold の合計) */
SELECT "Item ID", SUM(CAST("Units Sold" AS BIGINT)) "Total Sold"
  FROM "myddb"."default"."transaction"
 GROUP BY "Item ID";

/* アイテム ID 別、出荷にかかる平均日数 */
SELECT
  "Item ID", DAY(AVG(date_parse("Ship Date", '%c/%e/%y') - date_parse("Order Date", '%c/%e/%y')))) "number of days"
  FROM "myddb"."default"."transaction"
 GROUP BY "Item ID";
```

※補足：Athena engine v2 は OSS の Presto 0.217 をベースにしているため、利用できる関数一覧を調べる場合は以下ドキュメントを参照してください。

<https://prestodb.io/docs/0.217/index.html>

Athena でクエリを実行した場合、結果は画面に表示されるだけでなく、S3 バケットにもその結果が CSV 形式で保存されます。時間に余裕がある場合は、CloudFormation スタックのリソース一覧から athena-ws-で始まるバケット名をクリックし、その中に Unsaved/フォルダが作られて、その下に年/月/日/フォルダがあり、そこに CSV 形式で結果データが保存される事を確認してみてください。

S3 上のデータと DYNAMODB の表をジョインする

ここでは、アイテムのマスタ表である master.csv ファイルを S3 上に保存し、Glue クローラーでカタログに登録したのち、Athena からクエリできるようにします。またそのデータと、DynamoDB のデータ(transaction)ともジョイン（結合）できる事も確認します。

S3 に MASTER.CSV を配置する

master.csv ファイルは以下のように、各アイテム ID 毎に Item Type（アイテム種名）、Unit Price（販売単価）と Unit Cost（単体コスト）が格納されているマスタ表です。販売単価は Sales Channel が Online と Offline で異なります。

ID	Item Type	Unit Price	Unit Cost	Sales Channel
1	Baby Food	255.28	159.42	Online
2	Beverages	47.45	31.79	Online
3	Cereal	205.7	117.11	Online
4	Clothes	109.28	35.84	Online
5	Cosmetics	437.2	263.33	Online
6	Fruits	9.33	6.92	Online
7	Household	668.27	502.54	Online
8	Meat	421.89	364.69	Online
9	Office Supplies	651.21	524.96	Online
10	Personal Care	81.73	56.67	Online
11	Snacks	152.58	97.44	Online
12	Vegetables	154.06	90.93	Online
1	Baby Food	207.32	159.42	Offline
2	Beverages	37.95	31.79	Offline
3	Cereal	183.27	117.11	Offline
4	Clothes	97.11	35.84	Offline
5	Cosmetics	313.07	263.33	Offline
6	Fruits	7.66	6.92	Offline
7	Household	566.31	502.54	Offline
8	Meat	388.16	364.69	Offline
9	Office Supplies	466.43	524.96	Offline
10	Personal Care	70.25	56.67	Offline
11	Snacks	125.02	97.44	Offline
12	Vegetables	146.71	90.93	Offline

この master.csv ファイルをまず S3 に配置します。CloudFormation のリソースタブから、athena-ws-で始まるバケット名をクリックして、S3 バケットの画面を開いてください。そこに master.csv 格納用の master フォルダを作成します。

「フォルダの作成」ボタンをクリックし、フォルダ名に master と入れて、「フォルダの作成」をクリックします。

フォルダの作成

フォルダを使用してバケット内のオブジェクトをグループ化します。フォルダを作成すると、指定した名前の後にスラッシュ (/) が付けられて、オブジェクトが S3 によって作成されます。続いて、このオブジェクトがコンソールにフォルダとして表示されます。 [詳細](#)

ⓘ バケットポリシーによってフォルダの作成がブロックされる場合があります
バケットポリシーにより、特定のタグ、メタデータ、またはアクセスコントロールリスト (ACL) の被付与者がいないオブジェクトのアップロードが禁止されている場合、この設定を使用してフォルダを作成することはできません。代わりに、[設定のアップロード](#)を使用して空のフォルダをアップロードし、適切な設定を指定することができます。

フォルダ

フォルダ名

/

フォルダ名に「/」は使用できません。 [命名規則を参照してください](#)

サーバー側の暗号化

ⓘ 以下の設定は、新しいフォルダオブジェクトにのみ適用され、その中のオブジェクトには適用されません。

サーバー側の暗号化

☒ 無効にする

☐ 有効にする

キャンセル **フォルダの作成**

以下のようにフォルダが作成できます。



作成した master をクリックしてフォルダの中に入り、そこに master.csv をアップロードしてください。アップロードができたなら、右上の「閉じる」ボタンでアップロード画面を閉じます。



上記のように master フォルダ以下に master.csv ファイルが置けていれば OK です。

GLUE クローラーでカタログ登録する

Athena で S3 上のファイルにクエリをかける場合、その情報をカタログに登録する必要があります。Athena がデフォルトで利用している "AwsDataCatalog" は、AWS Glue のデータカタログ機能を参照しています。

カタログへの登録は、CREATE EXTERNAL TABLE を使って手動で行うこともできますが、ここでは Glue クローラーで登録します。

画面上部の検索枠に”glue”等と入力し、AWS Glue の管理画面に移動してください。Glue の画面では左側ペインから「クローラー」を選択し、右画面で「クローラーの追加」を選択します。

The screenshot shows the 'Add crawler' interface in the AWS Glue console. On the left, a sidebar lists navigation options: 'Crawler information' (highlighted with a green circle), 'Crawler source type', 'Data store', 'IAM role', 'Schedule', 'Output', and 'Verify all steps'. The main area is titled 'Add crawler information'. It contains a text input field for 'Crawler name' with the value 'master-crawler'. Below this is a section for 'Tags, description, security settings, and classification (optional)'. At the bottom right is a blue 'Next' button.

上記画面ではクローラーの名前を付けます。master-crawler と入力して「次へ」を押します。

The screenshot shows the 'Specify crawler source type' screen. The title is 'Specify crawler source type'. Below it is a paragraph: 'Choose Existing catalog tables to specify catalog tables as the crawler source. The selected tables specify the data stores to crawl. This option doesn't support JDBC data stores.' There are two sections of radio buttons. The first section, 'Crawler source type', has 'Data stores' selected. The second section, 'Repeat crawls of S3 data stores', has 'Crawl all folders' selected. At the bottom are two buttons: 'Back' and 'Next'.

上記画面ではそのまま「次へ」を押します。

データストアの追加

データストアの選択

S3

接続

接続の選択

Optionally include a Network connection to use with this S3 target. Note that each crawler is limited to one Network connection so any future S3 targets will also use the same connection (or none, if left blank).

接続の追加

クローलするデータの場所

☒ 自分のアカウントで指定されたパス
☐ 別のアカウントで指定されたパス

インクルードパス

s3://athena-ws-936203727359-ap-northeast-1/master

インクルードパスに含まれるフォルダやファイルはすべてクローलされます。たとえば、MyBucket 内の MyFolder のオブジェクトをすべてクローलするには、s3://MyBucket/MyFolder/ と入力します。

▶ エクスクルードパターン (任意)

戻る 次へ

データストア追加の画面で、上記のようにインクルードパス（クローलさせる S3 のパス）に、先ほど作成した master フォルダを指定し、「次へ」を押します。

次に別のデータストアの追加の画面がでますが、これはそのまま「次へ」を押します。

続いて IAM ロールの選択画面になります。これはクローラーに付与する IAM ロールを選択する画面です。

IAM ロールの選択

IAM ロールはクローラに実行および Amazon S3 データストアへのアクセスを許可します。詳細は[こちら](#)

☐ IAM ロールのポリシーの更新
☐ 既存の IAM ロールを選択
☒ IAM ロールを作成する

IAM ロール ⓘ

AWSGlueServiceRole-

IAM ロールを作成するには、**CreateRole**、**CreatePolicy**、および **AttachRolePolicy** アクセス権限が必要です。

"AWSGlueServiceRole-rolename" という名前の IAM ロールを作成して、AWS 管理ポリシー **AWSGlueServiceRole** と、次に対する読み取りアクセスを許可するインラインポリシーをアタッチします。

- s3://athena-ws-936203727359-ap-northeast-1/master

IAM ロールは [IAM コンソール](#) で作成することもできます。

[戻る](#) [次へ](#)

既存の IAM ロールがあればそれを選択しますが、このハンズオンではまだ作成していないので、「IAM ロールを作成する」にチェックをいれたまま、IAM ロール名のところに s3crawler と入力し、「次へ」を押します。

スケジュールの画面では「オンデマンドで実行」のまま「次へ」を押します。

クローラの出力を設定する

データベース ⓘ

[データベースの追加](#)

テーブルに追加されたプレフィックス (任意) ⓘ

▶ S3 データのグループ化動作 (任意)

▶ 設定オプション (任意)

[戻る](#) [次へ](#)

上記画面では、カタログのどのデータベース（仮想的なくくり）に格納するかを選択します。
ここでは「データベースの追加」を押して mydb と名前を付けて追加し、「次へ」を押します。

クローラの情報

名前master-crawler
タグ-

データストア

データストアS3
インクルードパスs3://athena-ws-936203727359-ap-northeast-1/master
Connection
エクスクルードパターン

IAM ロール

IAM ロールarn:aws:iam::936203727359:role/service-role/AWSGlueServiceRole-s3crawler

スケジュール

スケジュールオンデマンドで実行

出力

データベースmydb
テーブルに追加されたプレフィックス (任意)
S3 パスごとに単一のスキーマを作成するfalse
▶ 設定オプション

戻る

完了

最後に上記のような確認画面が出るので、S3 のパスが master フォルダを指しているかを確認して「完了」を押します。これでクローラーが作成されます。

クローラの追加		クローラの実行	アクション ▼	🔍 タグや属性によるフィルター	
<input checked="" type="checkbox"/>	名前	スケジュール		ステータス	ログ
<input checked="" type="checkbox"/>	master-crawler			Ready	

上記画面で、作成されたクローラーを選択し、「クローラーの実行」を押します。クローラーが起動して、クロール完了まで 1-2 分待ちます。

クローラー実行後にステータスが再度 Ready になったら、画面左から「テーブル」を選択します。



master という名前で mydb データベースに表が登録されているのが分かります。master をクリックして、詳細を確認します。

テーブル > master 最終更新日 21 3月 2021 06:59 午後 テーブル バージョン (現行バージョン) ▼

[テーブルの編集](#)
[テーブルの削除](#)
[プロパティの表示](#)
[バージョンの比較](#)
[スキーマの編集](#)

名前	master
説明	
データベース	mydb
分類	csv
場所	s3://athena-ws-936203727359-ap-northeast-1/master/
接続	
廃止	いいえ
最終更新日	Sun Mar 21 18:59:08 GMT+900 2021
入力形式	org.apache.hadoop.mapred.TextInputFormat
出力形式	org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat
Serde シリアル化ライブラリ	org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
Serde パラメータ	field.delim ,

テーブルのプロパティ

skip.header.line.count	1	sizeKey	824	objectCount	1	UPDATED_BY_CRAWLER	master-crawler
CrawlerSchemaSerializerVersion	1.0	recordCount	22	averageRecordSize	36		
CrawlerSchemaDeserializerVersion	1.0	compressionType	none	columnsOrdered	true		
areColumnsQuoted	false	delimiter	,	typeOfData	file		

スキーマ

表示中: 1 - 5 of 5 < >

	列名	データ型	パーティションキー	コメント
1	id	bigint		
2	item type	string		
3	unit price	double		
4	unit cost	double		
5	sales channel	string		

このように詳細が表示されます。ここでは画面下部のスキーマに注目します。この表は5つの列があり、それぞれ列名、データ型が認識されている事がわかります。今回はこのうち Id 型を String に修正します。（結合対象である Transaction 表の Item ID が Varchar のため）

画面右上の「スキーマの編集」をクリックしてください。

	列名	データ型	キー
1	id	string	
2	item type	string	
3	unit price	double	
4	unit cost	double	
5	sales channel	string	

スキーマの編集画面が表示されるので、Id の行で、bigint となっている部分をクリックし、string に変更してください。変更できたら右上の「保存」を押して元の画面に戻ります。

これでカタログ登録が完了しました。

ATHENA から S3 上の MASTER.CSV にクエリする

Athena のクエリエディタに戻り、画面左でデータソースを「AwsDataCatalog」（Glue クローラーで登録したカタログ）、データベースを「mydb」に切り替えてください。Athena から master 表が確認できます。

試しに以下のクエリを実行してみてください。

```
SELECT * FROM master;
```

このように CSV ファイルの内容がクエリできることが確認できます。

データソース: AwsDataCatalog

データベース: mydb

テーブル (1): master

- id (string)
- item type (string)
- unit price (double)
- unit cost (double)
- sales channel (string)

クエリの実行結果:

	id	item type	unit price	unit cost
1	1	Baby Food	255.28	159.42
2	2	Beverages	47.45	31.79
3	3	Cereal	205.7	117.11
4	4	Clothes	109.28	35.84
5	5	Cosmetics	437.2	263.33

SQL でエラーが出る場合は左のデータソースとデータベースの選択が間違っていないか確認してください。もしくは

```
SELECT * FROM "AwsDataCatalog"."mydb"."master";
```

とデータソースやデータベースを省略せずに実行すれば左側の設定に関わらずクエリできます。

最後にこの master 表と DynamoDB 上の transaction 表をジョイン（結合）してみましょう。以下の SQL を Athena から実行してみてください。（join.sql ファイルにも記載されています）

これは、2つの表をジョインし、transaction 表にある“Units Sold”（販売個数）を master 表の販売単価やコストと掛け算することで、トランザクションあたりのトータルの売り上げ（“Total Revenue”）や、かかったコスト（“Total Cost”）、その差分である粗利（“Total Profit”）を算出するクエリです。SQL の最後に LIMIT 100 とありますので、結果のうち 100 行までが返るようになっています。


```

SELECT
    t."Transaction ID",
    m."Sales Channel",
    m."Item Type",
    date(date_parse("Order Date", '%c/%e/%y')) "Order Date",
    date(date_parse("Ship Date", '%c/%e/%y')) "Ship Date",
    CAST (t."Units Sold" as INT) "Units Sold",
    t."Region",
    t."Country",
    t."Order Priority",
    m."Unit Price",
    m."Unit Cost",
    m."Unit Price" * CAST (t."Units Sold" as INT) "Total Revenue",
    m."Unit Cost" * CAST (t."Units Sold" as INT) "Total Cost",
    (m."Unit Price" - m."Unit Cost") * CAST (t."Units Sold" as INT) "Total Profit"
FROM
    "AwsDataCatalog"."mydb"."master" m,
    "myddb"."default"."transaction" t
WHERE m.id = t."Item ID" and m."Sales Channel" = t."Sales Channel"

LIMIT 100
;

```

これで Athena の Federated Query を利用して DynamoDB のデータと、S3 上のデータをジョインすることができました。

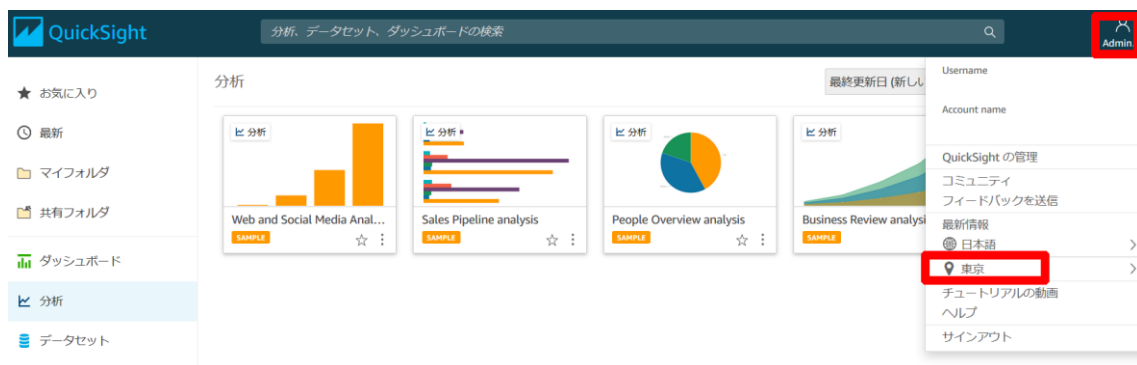
QUICKSIGHT で可視化する

Athena で DynamoDB と S3 上の表をジョイン可能になったので、ここではそのジョインの結果を QuickSight で可視化します。

QUICKSIGHT で ATHENA へのアクセスを設定する

画面上部の検索枠等から QuickSight を検索して、QuickSight の画面に遷移します。

補足： QuickSight の画面から、AWS マネジメントコンソールへのリンクは存在しません。マネジメントコンソールに戻る場合は、ブラウザのタブを開き、<https://console.aws.amazon.com/> にアクセスしてください。



QuickSight の画面ではまず右上の人のマークをクリックして、リージョンが想定したリージョンになっているかを念のため確認してください（ハンズオンの想定は東京リージョンですが、**利用環境に合わせて読み替えてください**）。もし違っている場合は切り替えてください。

次に、同じく右上の人マークメニューから「QuickSight の管理」をクリックし、管理画面左側の「セキュリティとアクセス権限」をクリックします。

ユーザーを管理

お客様のサブスクリプション

SPICE 容量

アカウント設定

セキュリティとアクセス権限

VPC 接続の管理

モバイル設定

ドメインと埋め込み

セキュリティとアクセス権限

QuickSight は、個々のユーザーおよびグループに加えて、アカウント全体の AWS リソースへのアクセスを:

QuickSight の AWS のサービスへのアクセス



Amazon Redshift



Amazon RDS



IAM



Amazon Athena

AWS のサービスへのアクセスを設定することで、QuickSight はそれらのサービスのデータにアクセスは、以下のオプションでコントロールできます。

追加または削除する

上記画面で QuickSight の AWS のサービスへのアクセスのところにある「追加または削除する」をクリックします。

QuickSight の AWS のサービスへのアクセス

QuickSightはすべてのユーザーとグループに以下の選択されたAWS製品とサービスに接続できるようにします：



Amazon Redshift

QuickSight でのクラスターの自動検出を有効にする



Amazon RDS

QuickSight でのインスタンスの自動検出を有効にする



IAM

この AWS アカウントの IAM ユーザーを招待して QuickSight にアクセスできるようにする



Amazon S3

QuickSight での Amazon S3 バケットの自動検出を有効にする [詳細](#)



Amazon Athena

QuickSight から Amazon Athena データベースへのアクセスを有効にする



上記画面で Athena にチェックを入れます。すでにチェックが入っている場合はいったんチェックを外して、再度チェックを入れてください。すると「Amazon Athena permissions」という画面が出るので、「次へ」を押します。

Amazon Athena permissions

×

QuickSight needs permissions to access Amazon S3 buckets or AWS lambda functions used by Athena

キャンセル

Skip

次へ

まず QuickSight から Athena が結果保存と Federated Query に利用している S3 バケットへのアクセスを許可します。

Amazon Athena resources

×

S3

Lambda

QuickSight アカウントにリンクされている S3 バケット

AWS 全体でアクセスできる S3 バケット

QuickSight がアクセスできるようにするバケットを選択します。

選択したバケットにはデフォルトで読み取り専用アクセス許可が付与されますが、Athena Workgroup 機能には書き込みアクセス許可を付与する必要があります。

☒ すべて選択

S3 バケット	Athena Workgroup の書き込みアクセス許可
<input checked="" type="checkbox"/> athena-ws-936203727359-ap-northeast-1	<input checked="" type="checkbox"/>
<input type="checkbox"/> cf-templates-ymgyo85o6sgh-ap-northeast-1	<input type="checkbox"/>
<input type="checkbox"/> cloudtrail-awslogs-936203727359-0sg493gy-isengard-do-not-delete	<input type="checkbox"/>
<input type="checkbox"/> ddb-upload-936203727359-ap-northeast-1	<input type="checkbox"/>
<input checked="" type="checkbox"/> federated-spill-936203727359-ap-northeast-1	<input checked="" type="checkbox"/>

今回は athena-ws-…と、federated-spill-…から始まる 2 つのバケットにチェックを入れ、それぞれ Athena Workgroup の書き込みアクセス許可にもチェックをいれます。次に右上にある「Lambda」をクリックします。

Amazon Athena resources

S3Lambda

Select Lambda functions to connect to data sources with Athena federated queries.

☒ すべて選択

Find functions or catalogs

Lambda function ARN	Catalog names
<input checked="" type="checkbox"/> arn:aws:lambda:ap-northeast-1:936203727359:function:ddbcatalog	myddb

キャンセル完了

Lambda 側の設定では、Catalog names が myddb になっている Lambda 関数にチェックをいれます。最後に「完了」を押します。

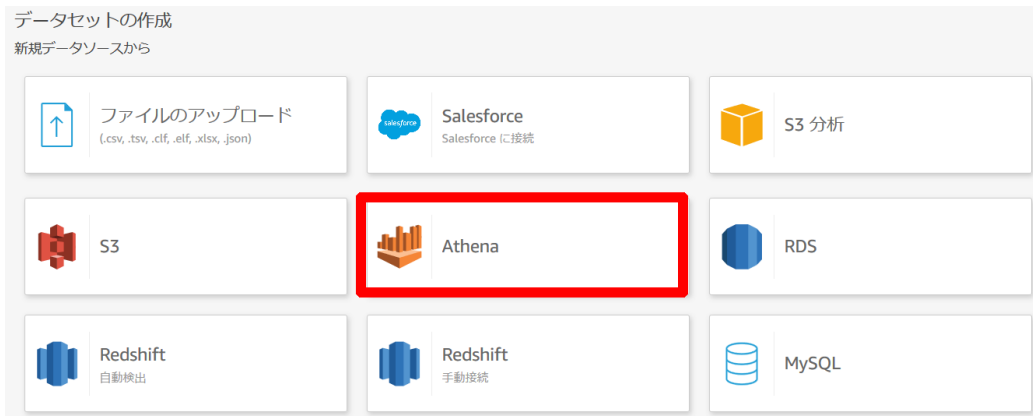
1 つ前の画面に戻るので「更新」を押します。

これで Athena へのアクセス設定は完了です。画面左上の QuickSight ログをクリックしてください。

データセットとして ATHENA のクエリを登録する

次に QuickSight が可視化に利用するデータセットを登録します。これは RDB の表を指定する場合もありますし、特定のクエリを登録することもできます。ここでは先ほど Athena で利用したジョインのクエリをデータセットとして登録し、その結果を可視化していきます。

画面左側にある「データセット」をクリックし、次に右上に出てくる「新しいデータセット」をクリックします。



上記画面で Athena を選択し、まずデータソース（Athena 自体）へのアクセス方法を登録します。

上記のようにデータソースには athena-hanson 等任意の名前を付け、ワークグループには前のステップで作成した handson ワークグループを選択、「接続を検証」を押して接続確認を行います。エラーなく「検証済み」になったら「データソースを作成」を押します。

（補足）上記の「接続を検証」でエラーが出た場合、前ステップの「セキュリティとアクセス権限」の設定がガイド通りかを見直してください。それでもエラーが解消できない場合は、なんらかの理由で QuickSight のサービスロールの権限が不足している可能性があります。その場合は下記ガイドにしたがってサービスロールの再作成を実行してください。 <https://aws.amazon.com/jp/premiumsupport/knowledge-center/quicksight-permission-errors/>

テーブルの選択 ×

athena-handson

Catalog: contain sets of databases.

AwsDataCatalog ▼

データベース: 複数のテーブルを含みます。

選択... ▼

データの準備を選択して、SQL クエリを作成するか、その他のデータ準備を実行します。それ以外の場合は **テーブルの選択** を選択します。

データの編集/プレビュー

カスタム SQL を使用

選択

上記ダイアログでは、GUI で表を個別に選択することもできますが、ここでは「カスタム SQL を使用」をクリックします。

カスタム SQL クエリの入力

SQL クエリに入力します。

transaction-master-join

9t."Country",
10t."Order Priority",
11m."Unit Price",
12m."Unit Cost",
13m."Unit Price" * CAST (t."Units Sold" as INT) "Total Revenue",
14m."Unit Cost" * CAST (t."Units Sold" as INT) "Total Cost",
15(m."Unit Price" - m."Unit Cost") * CAST (t."Units Sold" as INT) "Profit",
16FROM
17"AwsDataCatalog"."mydb"."master" m,
18"myddb"."default"."transaction" t
19WHERE m.id = t."Item ID" and m."Sales Channel" = t."Sales Channel"
20;
21

注) セミコロンの直前にあったLIMIT 100は削除してください

データの編集/プレビュー

クエリの確認

上記画面ではまずカスタム SQL の名前として transaction-master-join 等と入力します。

次に join.sql ファイルの内容をコピー&ペーストで、クエリエディタに張り付けます。この時、セミicolon直前にある LIMIT 100 を削除し忘れないようにしてください。（LIMIT 100 を付けたままだと、分析対象が 100 行だけになってしまいます）

カスタム SQL が入力できたら「クエリの確認」を押します。

データセットの作成を終了する

×

カスタム SQL: transaction-master-join
データソース: athena-handson

☒ 迅速な分析のために SPICE へインポート

✓ 1009.5MB 利用可能 SPICE

☐ データクエリを直接実行

☒ 更新が失敗したときに所有者にメールを送信

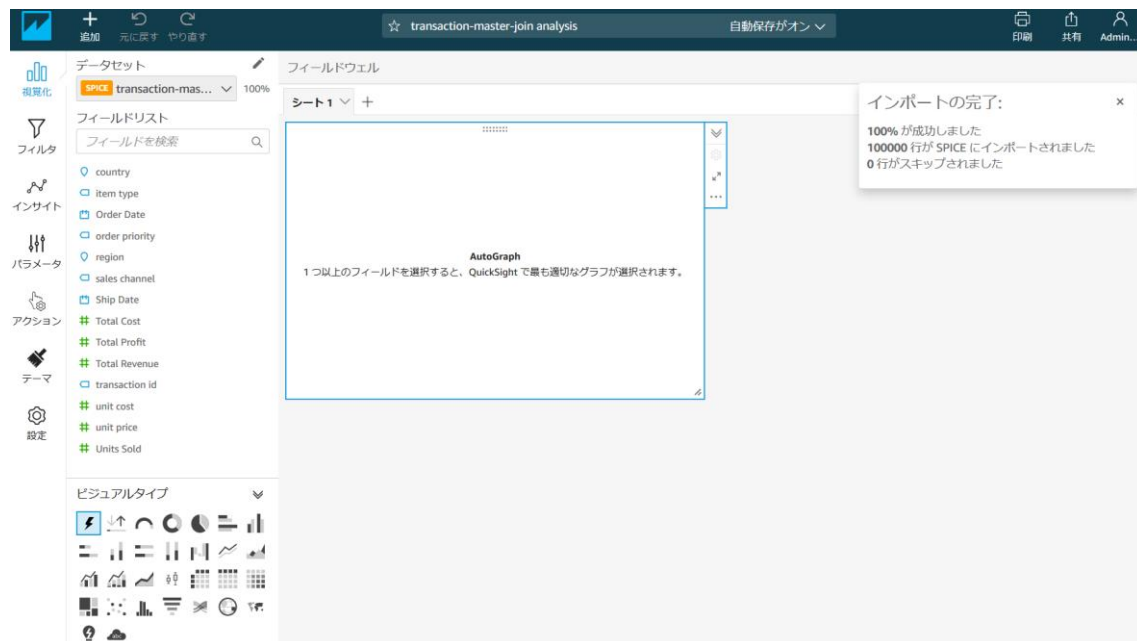
データの編集/プレビュー

SageMaker によるオーグメント

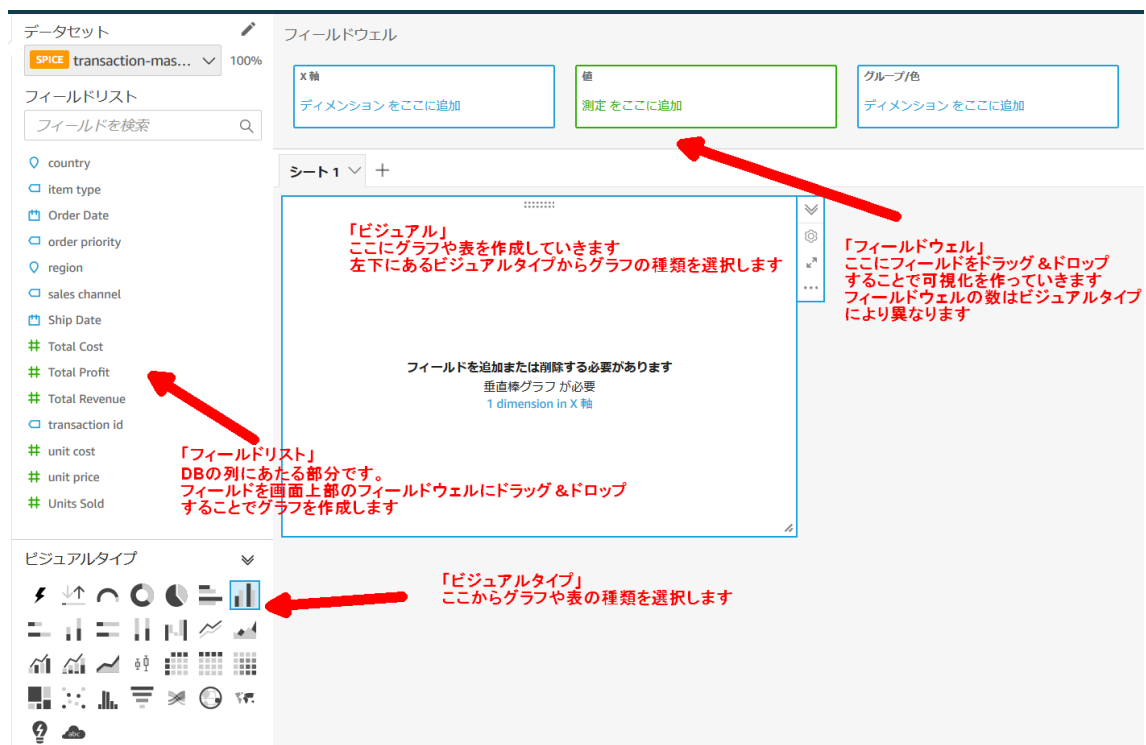
Visualize

最後にクエリ結果を SPICE（インメモリデータベース）にインポートするか、毎回クエリを発行するか（直接実行）かの画面がでますので、SPICE へのインポートが選択されている事を確認して、「Visualize」を押します。これで Athena のクエリ結果が SPICE に取り込まれ、可視化の準備が整いました。

QUICKSIGHT での可視化の操作



上記は QuickSight の可視化(Visualize)の画面です。データが Athena から SPICE にすべて読み込まれると、上記画面右上にあるように「インポートの完了」と表示されます。(1-2 分待ってもインポートが完了しない場合は、ブラウザ画面のリロードをお試しください)。インポートが完了した場合は、右上のインポートのダイアログは閉じて問題ありません。



QuickSight での可視化のおおまかな手順は以下の通りです。

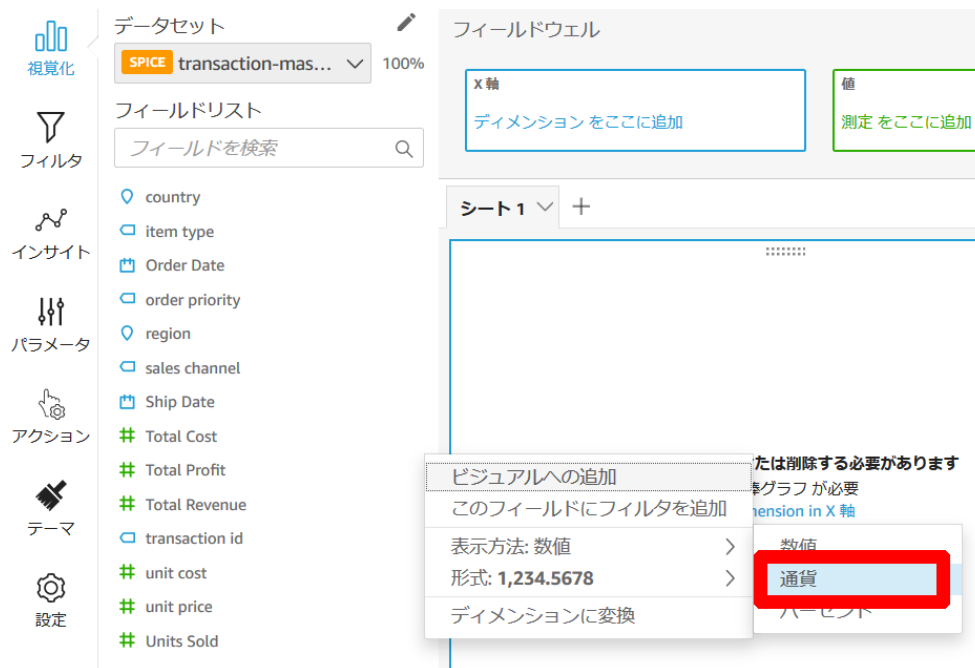
- 1) 描画したいグラフ・表を左下のビジュアルタイプから選択
- 2) ビジュアル（グラフ）に必要なフィールドウェルが上部に現れる
- 3) 左のフィールドリスト（列一覧）から分析に利用するフィールドをフィールドウェルにドラッグ&ドロップし、可視化する
- 4) 必要ならビジュアルのオプションで見栄え等を調整する

もう一つグラフを追加したい場合は画面左上の「+追加」ボタンからビジュアルを追加できます。ビジュアルは並び替えたり、サイズを変更したりすることが可能です。

フィールドの単位を調整する

すでにグラフを作り始める事が可能な状態ですが、見栄えを整えるために少しかフィールドを調整します。このステップは見栄えが変わるだけで機能面では変わりませんので、時間が無い場合は本ページをとばしても問題ありません。

画面左側のフィールドリストの”Total Profit”の上にマウスカーソルを持っていくと、[...]メニューが表示されるので、クリックし、表示方式を「通貨」に変更してください。



同様に、”Total Revenue”に対しても通貨への変更をしてください。

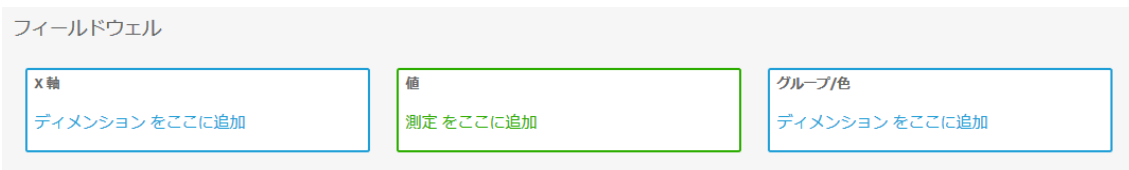
本ハンズオンではこの2つのフィールドしか使わないため、これで可視化に進みますが、時間に余裕があれば他の通貨を表しているフィールド（Cost, Revenue, Profit）も通貨に変更しておくと見栄えが良くなります。

可視化①「地域別レベニュー」

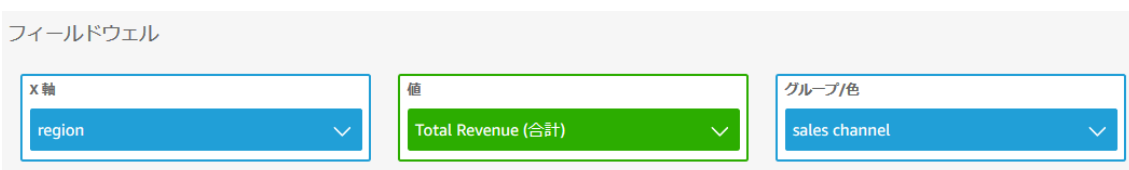
1つ目のグラフとして棒グラフを利用します。画面上にはすでに1つ目のビジュアルがありますので、ビジュアルタイプから「垂直積み上げ棒グラフ」を選択してください。（棒グラフが複数種類あるので注意してください）



すると、画面上部にフィールドウェルが出ます。



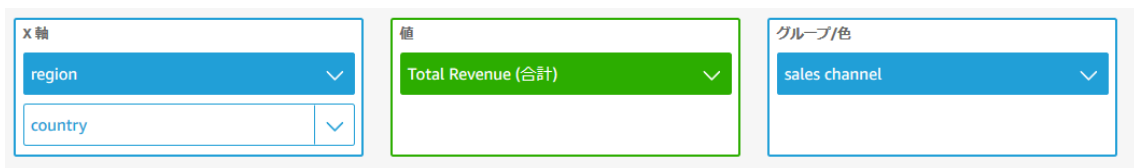
フィールドリストから、region（地域）を、「X 軸」にドラッグ&ドロップしてください。同様に、Total Revenue を「値」に、sales channel を「グループ/色」にドラッグ&ドロップしてください。以下ようになります。



次に、region の下位要素（ドリルダウン）として、country を X 軸にドラッグ&ドロップします。この時 region を置き換えるのではなく、region の下までドラッグしていき、「ドリルダウンレイヤーの追加」と表示される位置ドラッグ&ドロップしてください。



補足：間違ったフィールドを置いてしまった場合等は、画面上部の「元に戻す」ボタンでやり直すことができます。

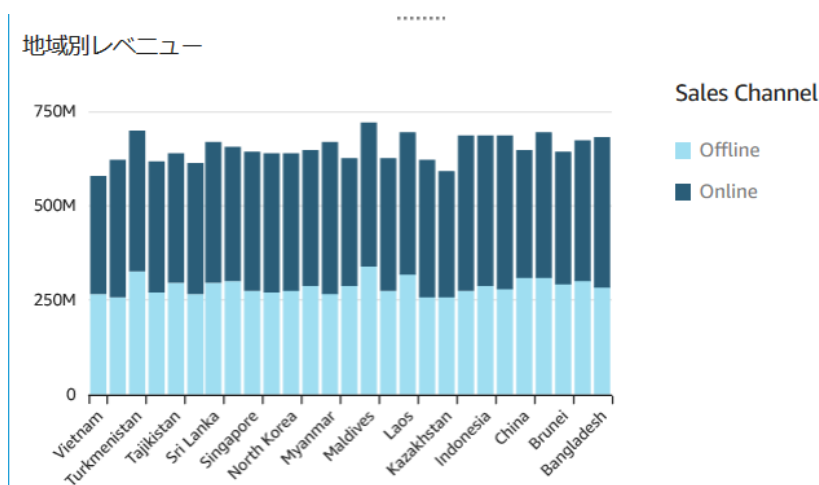


最後にビジュアルについているタイトルをクリックして、タイトルを修正します。ここでは「地域別レベニュー」としました。



上記のようになればビジュアルが完成です。各 Region での Offline/Online のトランザクション比率の違い等が読み取れます。

ドリルダウンが機能しているか確認するため、Asia の棒グラフを左クリックし、メニューから「country までドリルダウン」を選択してください。



すると ASIA の中の国単位にドリルダウンされました。

再度どれかの棒グラフをクリックして、「region までドリルアップ」を選択し、元の画面に戻しておいてください。

これで DynamoDB のデータを Athena で CSV データとジョインし、QuickSight で可視化するという作業が完了しました。この後はオプションで QuickSight の各種グラフや、フィルタ機能、予測機能の使い方を解説していますので、お時間がある範囲でお試してください。

終了する場合は、文末の「後片付け」のステップを参照してください。

可視化②「総利益」

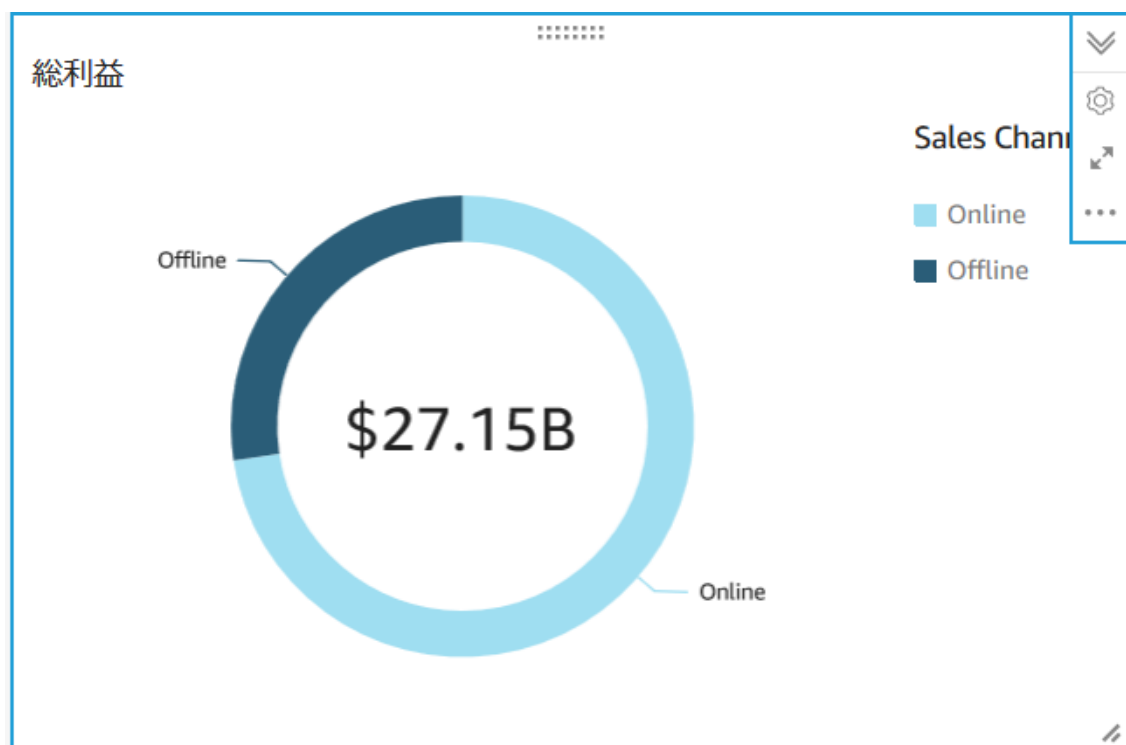
ドーナツグラフを使って Offline と Online の利益比率を確認します。

画面左上の「+追加」から「ビジュアルを追加」し、ビジュアルタイプとしてドーナツグラフを選択します。フィールドウェルには以下のようにフィールドをドラッグ&ドロップします。

フィールドウェル

グループ/色	値
sales channel	Total Profit (合計)

最後にビジュアルのタイトルを「総利益」に変更すると、以下のようなビジュアルが完成します。

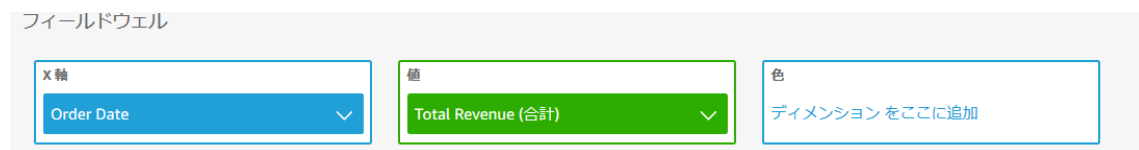



可視化③「レベニュー推移」

折れ線グラフを使ってレベニューの時系列推移を確認します。


左上の「+追加」から「ビジュアルを追加」し、ビジュアルタイプとして折れ線グラフを選択します。時系列のグラフなのでビジュアルをリサイズして横に長くしておくで見やすくなります。

フィールドウェルには以下のように設定します。



時系列で表示されますが、細かすぎてみづらいグラフになっているので X 軸の Order date の粒度を「月単位」に変更します。フィールドウェルの X 軸上にある「Order Date」の  をクリックして、集計を「月」に変更してください。



同様に「Order Date」の  を再度クリックし、形式→その他フォーマットオプションを選択します。

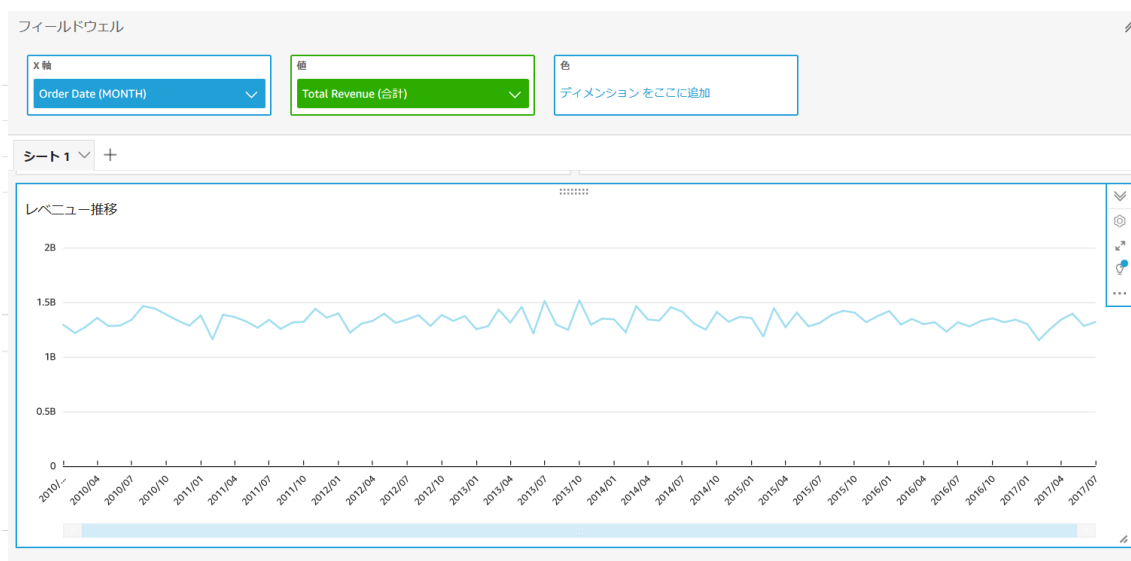
左側にフォーマットオプションが表示されるので、カスタムを選択し、カスタム文字列として「YYYY/MM」と入力してください。

データをフォーマット ×

選択したビジュアルに適用

フィールド:	Order Date
日付	⌵
時間	⌵
カスタム	⌵
YYYY/MM	
プレビュー:	2021/09
Null 値	⌵

最後にグラフのタイトルを「レベニュー推移」とします。これで以下のようなグラフが作成できました。



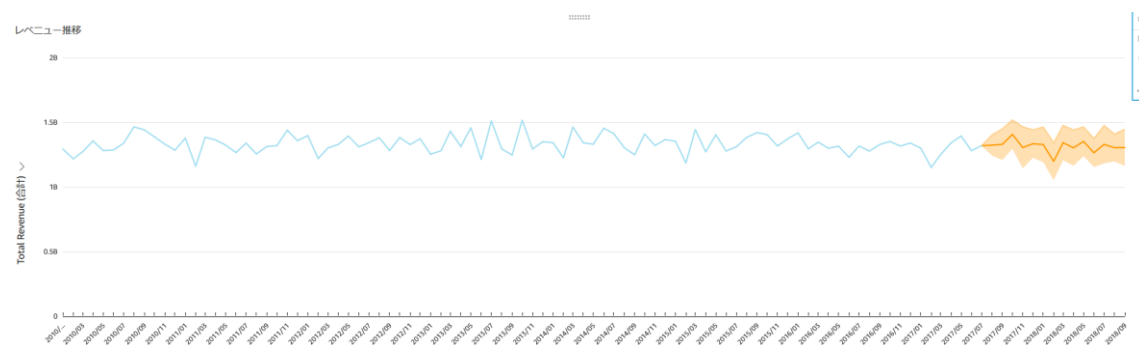
可視化④「レベニュー推移」に予測を追加する

このステップは QuickSight の Enterprise Edition でのみ作成できます。Standard Edition をご利用の方はこのステップをスキップしてください。

前ページで作成したレベニュー推移の折れ線グラフに予測を追加します。

「レベニュー推移」ビジュアルの右側にある…マークをクリックし、「予測を追加」を押します。左側に予測のプロパティが出ますので、「期間を進める」に 6 を入力して画面下部の「適用」を押してください。

これで折れ線グラフにオレンジ色のレベニュー予測が追加されました。このように簡単に時系列の予測値をグラフに挿入することが可能です。



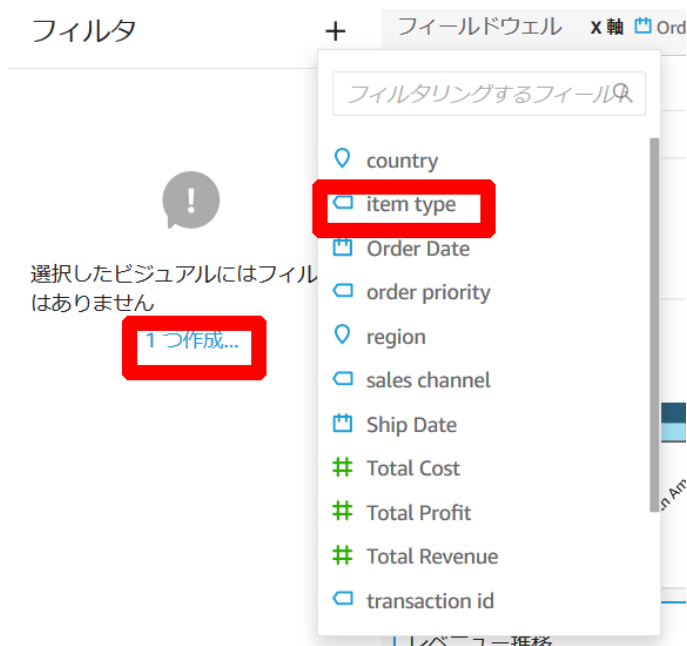
※補足：本ハンズオンで利用しているデータは 2017 年 7 月までしか入っていないため、それ以降の値は QuickSight が計算した予測値です。

フィルタ①ITEM TYPE のフィルタコントロール

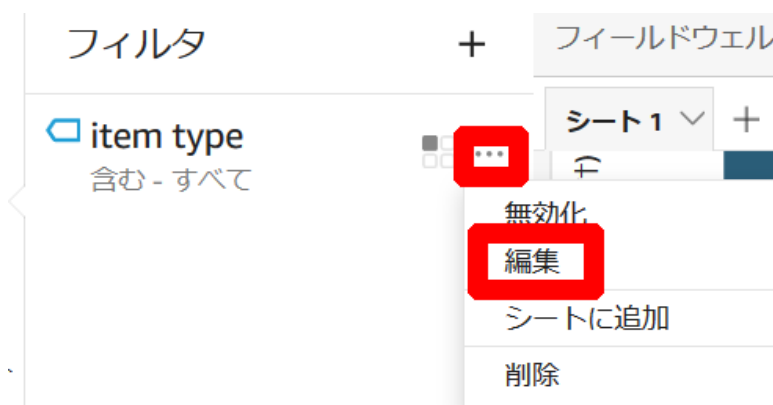
QuickSight にはデータを絞り込む「フィルタ」機能が用意されています。ここでは Item に対してフィルタを設定し、注目したい Item だけに絞って見られるようにします。

画面左側の「フィルタ」をクリックし、「1つ作成」をクリックします。

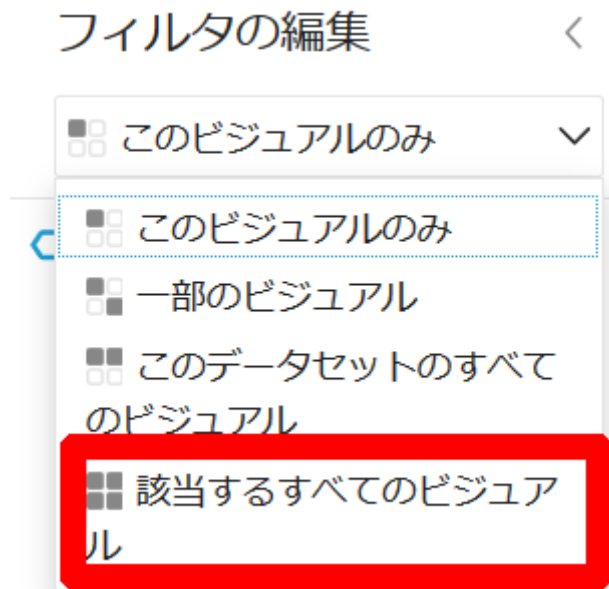
フィルタ対象のフィールドとして、「item type」を選択します。



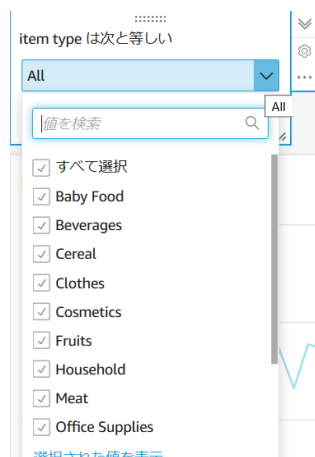
item type フィルタが作成されますので、その横にある…から編集を選択します。



最上部で「このビジュアルのみ」になっている部分をクリックし、「該当するすべてのビジュアル」に変更し、画面下部の「適用」→「閉じる」を押します。こうすることでフィルタがすべてのビジュアルにかかるようになります。



再度 item type フィルタ横の「…」をクリックし、「シートに追加」を押してください。すると画面上にこのようなフィルタコントロールが表示されます。



このフィルタコントロールでチェックボックスを付けたり外したりすることで、各ビジュアルにフィルタがかかる事が確認できます。

フィルタ②年単位のフィルタコントロール

ここでは指定した「年」でフィルタするコントロールを作成します。しかし、フィールドリストには年だけを表すフィールドが存在しません。（Order Date は年月日でデータを持っていますが、年だけはありません）

そこで、まず計算フィールドという機能を使って Order Date から疑似的に年（Year）フィールドを追加します。

画面左上の「+追加」から「計算フィールドを追加」を選択します。計算フィールドの入力画面になりますので、名前には Year、計算式として以下をコピー＆ペーストで入力し、「保存」を押してください。

```
extract('YYYY', {Order Date})
```



計算フィールドを追加

Year 

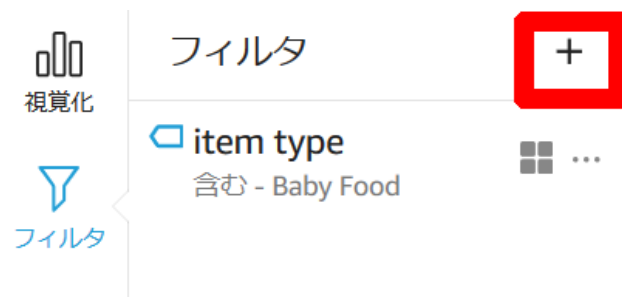
```
1 extract('YYYY', {Order Date})
```

※もし保存時に「**計算フィールドの保存中にエラーが発生しました。もう 1 度作成してください。**」というエラーが出た場合はブラウザをリロードしてから再度試してください。

Extract は、QuickSight がもつ関数の 1 つで日付から一部を抜き出します。ここでは 'YYYY' とすることで、4 桁の西暦を抜き出し Year という計算フィールドを作成しています。

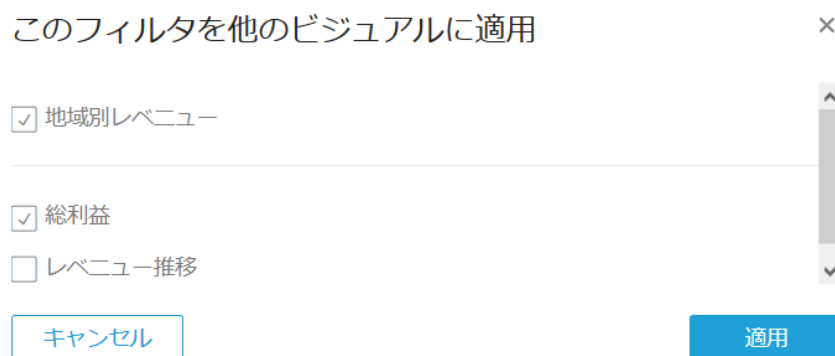
作成した計算フィールドは通常のフィールドと同様に利用できます。

Year を元にフィルタを作るため、まず一番最初に作った「地域別レベニュー」ビジュアルを選択したうえで、画面左端にある「フィルタ」をクリックし上部にある「+」を押します。フィルタするフィールドとして Year を選択してください。



作成された Year フィルタの右側にある「…」をクリックし、編集を選択します。

画面上部の「このビジュアルのみ」をクリックし、ここでは「一部のビジュアル」を選択します。すると対象とするビジュアルを選択するダイアログが出るので、「地域別レベニュー（最初に作ったビジュアル）」と「総利益」（2つ目に作ったビジュアル）にチェックをいれて「適用」を押します。



次に左側で開いているフィルタの編集を「閉じる」で閉じます。

次に Year フィルタの「…」メニューから「シートに追加」を押します。

コントロールを追加

×

フィルタ

□

表示名

Year

☒ タイトルを表示

タイトルのフォントサイズ

中

スタイル

一覧

値

☒ フィルタ

☐ 特定の値

☒ コントロールがシートにあるときに検索バーを非表示にする

☐ コントロール値から [ALL] オプションを非表示にする ①

キャンセル

追加

上記のように、表示名に Year を指定し、スタイルは「一覧」を選択して「追加」を押します。
すると以下のような一覧選択式のフィルタが表示されます。

.....

Year

☒ すべて選択

☐ 2010

☐ 2011

☐ 2012

☐ 2013

☐ 2014

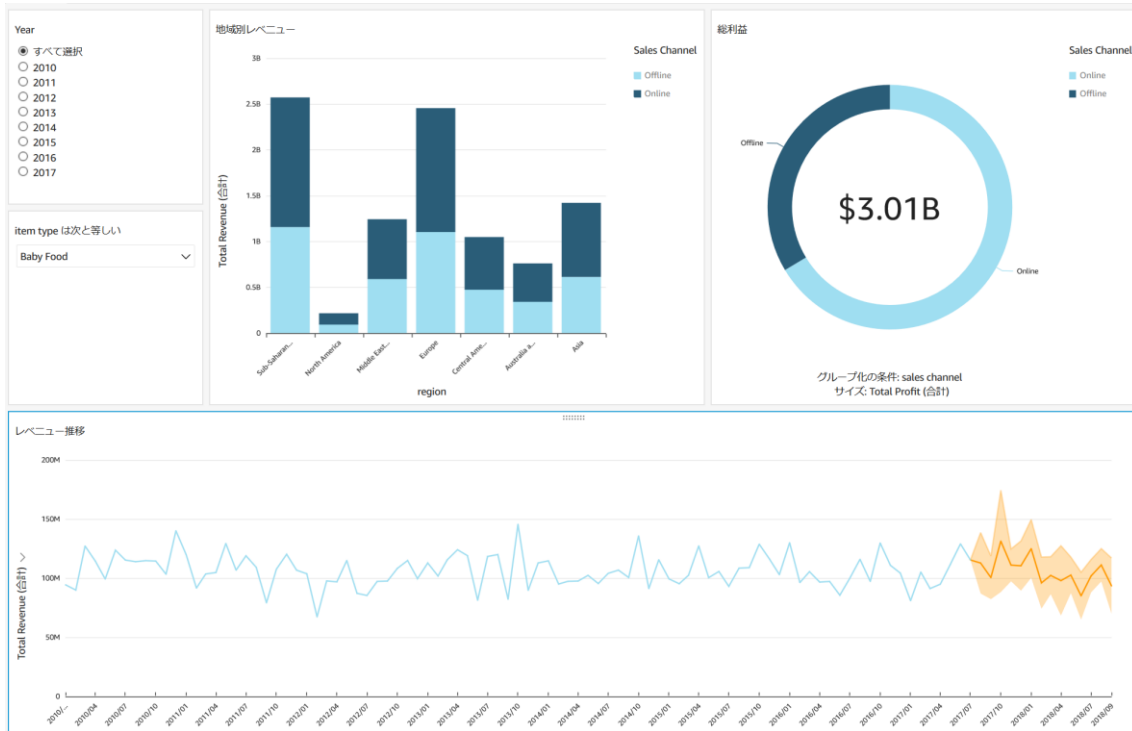
☐ 2015

☐ 2016

☐ 2017

このフィルタで、例えば 2011 を選択すると最初に作った 2 つのビジュアルは 2011 年のデータ
のみに絞り込まれます。最後にサイズや位置を調整して完成です（下図）。

これでハンズオンすべてが終了です。お疲れ様でした。



レベニュー推移

Total Revenue (合計)

後片付け

QuickSight は 1 ユーザであれば無料で利用できますし、Athena についてもクエリを動かさなければ費用は発生しません。

しかし、作成した環境を削除される場合は以下の手順で後片付けをしてください。

- QuickSight

- 右上人型マークの管理メニューから「QuickSight の管理」=>「アカウント設定」に進みます。ここで「サブスクリプション解除」を押すと、QuickSight のリソース全体を消すことが可能です。（ダッシュボード等はすべて消えます）
- サブスクライブ解除をすると、以下のような画面が表示されます。IAM の Role と Policy が削除されずに残っているという警告です。IAM Role と Policy は残しておいても費用は発生しませんが、削除したい場合は IAM の画面からロール、ポリシーを”QuickSight”等の名前で検索して削除してください

！ 他のサービスで使用している場合を除き、AWS IAM コンソールを使用して次のロールとポリシーを削除してください。



```
arn:aws:iam::aws:policy/service-role/AWSQuickSightElasticsearchPolicy
arn:aws:iam::936203727359:policy/service-role/AWSQuickSightS3ConsumersPolicy
arn:aws:iam::aws:policy/service-role/AWSQuickSightAthenaAccess
arn:aws:iam::aws:policy/service-role/AWSQuickSightSageMakerPolicy
arn:aws:iam::aws:policy/service-
role/QuickSightAccessForS3StorageManagementAnalyticsReadOnly
arn:aws:iam::936203727359:policy/service-role/AWSQuickSightRedshiftPolicy
arn:aws:iam::936203727359:policy/service-role/AWSQuickSightIAMPolicy
arn:aws:iam::936203727359:policy/service-role/AWSQuickSightRDSPolicy
arn:aws:iam::aws:policy/AWSQuickSightIoTAnalyticsAccess
arn:aws:iam::936203727359:policy/service-role/AWSQuickSightLambdaPolicy
arn:aws:iam::aws:policy/service-role/AWSQuickSightTimestreamPolicy
arn:aws:iam::936203727359:role/service-role/aws-quicksight-service-role-v0
arn:aws:iam::936203727359:role/service-role/aws-quicksight-s3-consumers-
role-v0
```



- Athena

- Athena の画面で、データソースを選択し、ハンズオンで作成した「myddb」データソースを選択して「削除する」を選択します
- 次にワークグループを選択し、作成した handson ワークグループを選択、「詳細を表示する」から削除します
- master 表を削除する場合は、master 表の横にある 3 点のボタンから削除を選択するか、DELETE TABLE で削除します（もしくは Glue data catalog 側で定義を消す事も可能です）
- S3 の画面を開き、master/以下にある master.csv ファイルを削除します
- Glue の画面を開き、master-crawler を選択して、クローラーを削除します
- CloudFormation のトップ画面に移動し、「スタックを表示」で存在するスタック一覧を表示させ、そこにある serverlessrep-AthenaDynamoDBConnector を削除します（Federated Query のための Lambda を定義したスタックです）
- ハンズオン時にセットアップした環境の削除
 - S3 の画面を開き、以下のバケットを空にします（空にしないとバケットの削除に失敗します）
 - ✧ athena-ws-…バケット
 - ✧ ddb-upload-…バケット
 - ✧ federated-spill-…バケット
 - CloudFormation の画面を開き、最初に実行した myhandson スタックを削除します

以上で後片付けは終了です。

お疲れ様でした。