はじめに

コース概要と目的

本コースは、Qlik Sense 入門の内容を踏まえ、より実践的なアプリケーション開発についてご覧いただくコースです。

データベースから取り込んだデータを自由に加工する方法や、チャートにさまざまな集計値を表示 する方法などについて、身に付けることを目的とします。

受講対象

「Qlik Sense 入門」を受講された方、もしくは同等の知識をお持ちの方。

製品のバージョンと形態

本テキストでは、以下のバージョンならびに利用形態の Olik Sense を使用しています。

- · Olik Sense May 2024
- · Qlik Sense Enterprise Client-Managed

※Qlik Sense Enterprise Client-Managed と Qlik Sense Enterprise SaaS は、基本的なアプリ 作成の手順は同じですが、操作画面や仕様に違いがあります。Qlik Sense Enterprise SaaS における操作画面や仕様については、研修の補足資料をご確認ください。

テキスト中の表記について

画面中に表示される文字の表記

ウィンドウ、ボタン、メニューなど、画面に表示される文字は[]で囲んで表記しています。 画面上に表示される選択肢や、入力する文字については「」で囲んで表記しています。

ディレクトリの表記

本テキストではファイルのパスを表す際、すべてデフォルトのインストールディレクトリを基準に 表記しています。お客様の環境と異なる場合は、適宜読み替えてください。

コマンドの構文規約

值	この表記は、ユーザーが適切な値に置き換えて指定することを 表します。
{ 値1 値2 値3 }	この表記は $\{\}$ の中から、いずれか 1 つを選択することを表します。
[値]	この表記はその指定が任意に指定できる、オプションの指定で あることを表します。必ずしも指定する必要はありません。

- ・Qlik、Qlik Alerting、Qlik Analytics Platform、Qlik Cloud、Qlik Connectors、Qlik Data-Transfer、Qlik Catalog、Qlik GeoAnalytics、Qlik Geocoding、Qlik NPrinting、NPrinting、Qlik Sense、Qlik Sense Cloud、QlikTech、QlikView、Qlik Replicate は、米国およびその他の国で登録された OlikTech International AB、の商標または登録商標です。
- ・Microsoft、Windows、Windows Server、SQL Server、Internet Explorer、Microsoft Edge、Excel、PowerPoint、その他本書に記載されるマイクロソフト製品は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。
- ·Linux は、Linus Torvalds 氏の米国およびその他の国における登録商標または商標です。
- ・Oracle、Java、MySQL 及び NetSuite は、Oracle Corporation、その子会社及び関連会社の 米国及びその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商 標である場合があります。
- ・IBM、ibm.com は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。
- ・Mac、macOS、Safari、iPhone、iPad は、米国およびその他の国で登録された Apple Inc.の商標です。
- ・Google、Google Chrome、Google Workspace、Google ドライブ、Google スライド、Google スプレッドシート、Google ドキュメント、Google フォーム、Android は、Google LLC の商標または登録商標です。
- ・Adobe、Adobe ロゴ、Adobe PDF ロゴ、Acrobat は、米国およびその他の国における Adobe の登録商標または商標です。
- ・Apache、Apache Tomcat および Tomcat は、Apache Software Foundation の米国および その他の国における登録商標または商標です。これらのマークの使用は、Apache Software Foundation による保証を意味するものではありません。
- ・Mozilla、Firefox および Firefox ロゴは、米国およびその他の国における Mozilla Foundation の商標です。
- ・その他記載されている会社名、製品名は、各社の商標または登録商標です。
- ・本書では、R、C、TMマークなどは特に明記していません。
- ・本書に記載された内容を使用して、いかなる損害が発生しても、株式会社アシスト、および、著者、本書製作関係者は一切責任を負いません。

目次

第1章 ロードスクリプトとデータロードエディタ	
1.1 ロードスクリプトとデータロードエディタ	1-1
1.1.1 データロード概要とデータマネージャーのおさらい	1-1
1.1.2 ロードスクリプト	
1.1.3 データロードエディタとデータマネージャー	1-5
1.1.4 データロードエディタの操作方法	1-8
1.1.5 テーブルの関連付け	1-14
1.1.6 計算項目の追加	
1.1.7 データの並べ替えとグループ化(Order By 句と Group By 句)	1-26
1.2 練習問題	1-30
1.2.1 練習問題	1-30
1.3 第1章のまとめと章末付録	1-32
1.3.1 まとめ	1-32
1.3.2 章末付録	1-32
第 2 章 Qlik Sense の関数とデータ型	
-	2 1
2.1 Qlik Sense の関数	
2.1.1 Qlik Sense の関数	
2.1.3 グループ化関数(Class 関数)	
2.1.4 日付関数	
2.2 Qlik Sense のデータ型	
2.2.1 Qlik Sense のデータ型	
2.2.2 データ型変換関数と書式設定関数	
2.2.3 デュアル値と Dual 関数	
2.2.4 NULL 値について理解する	
225 端数処理	7-39

2.3 軸単位の集計値を返す(Aggr 関数)	2-40
2.3.1 Aggr 関数とは	2-40
2.3.2 計算軸の作成	
2.3.3 集計関数のネスト	2-45
2.4 レコード関数と範囲関数	2-49
2.4.1 レコード関数(Above 関数など)	2-49
2.4.2 範囲関数(RangeSum 関数)	2-53
2.5 練習問題	2-57
2.5.1 練習問題	2-57
2.6 第2章のまとめと章末付録	2-60
2.6.1 まとめ	
2.6.2 章末付録	
第3章 データモデリング	
3.1 データモデリングの概要	3-1
3.1.1 データモデリングの概要	
3.2 データ構造の変換	
3.2.1 列から行への変換(Crosstable プレフィックス)	
3.2.2 テーブルの連結(Concatenate プレフィックス)	
3.3 データの結合とデータ値の置き換え	
3.3.1 テーブルの結合 (Join プレフィックス)	
3.3.2 データ値の置き換え(Mapping プレフィックスと ApplyMap 関数)	
3.4 集合演算	
3.4.1 集合演算(Keep プレフィックス)	
3.5 テーブルの再読み込み	
3.5.1 テーブルの再読み込み(Resident 句)	
3.5.2 Resident 句の使用例	
3.5.3 Load 文のネストと Resident 句	
3.6 合成キーの作成とその回避方法	
3.6.1 合成キーの作成	
3.6.2 合成キー(Synthetic キー)の回避	
3.6.3 キーの作成(AutoNumber 関数、AutoNumberHash 関数)	3-44

3.7 循環参照	3-55
3.7.1 循環参照	3-55
3.8 マスターカレンダーとその他の参考情報	3-57
3.8.1 マスターカレンダーとは	3-57
3.8.2 マスターカレンダーの作成方法	3-58
3.8.3 その他の参考情報	3-59
3.9 練習問題	3-60
3.9.1 練習問題	3-60
3.10 第3章のまとめと章末付録	3-64
3.10.1 まとめ	3-64
3.10.2 章末付録	3-65
第4章 QVD ファイルと差分更新	
4.1 QVD ファイル	4-1
4.1.1 QVD ファイルとは	4-1
4.1.2 QVD ファイルの作成(Store 文)	4-3
4.2 差分更新	4-6
4.2.1 Qlik Sense での差分更新	4-6
4.2.2 差分更新のシナリオ	4-15
4.2.3 差分更新の例(挿入のみの場合)	4-16
4.3 練習問題	4-26
4.3.1 練習問題	4-26
4.4 第4章のまとめと章末付録	4-28
4.4.1 まとめ	4-28
4.4.2 章末付録	4-28
第5章 比較分析	
5.1 比較分析	
5.1.1 比較分析	
5.2 SET 分析	
5.2.1 SET 分析の基本	
5.2.2 条件式の指定	
5.2.3 複数の値の指定	5-12

5.2.4 さまざまな SET 分析の指定5-	17
5.2.5 SET 数式の入力補助機能5-	18
5.2.6 動的な値の取得5-:	24
5.3 並列ステート5-2	28
5.3.1 並列ステート5-	28
5.3.2 数式でのステートの指定5-:	34
5.4 練習問題5-4	10
5.4.1 練習問題5	40
5.5 第5章のまとめと章末付録5-4	12
5.5.1 まとめ5	42
5.5.2 章末付録5	
笠く笠 コヤルのポノント	
第6章 可視化のポイント	
6.1 可視化のポイント6	
6.1.1 可視化のポイント6	
6.1.2 情報の階層を整理する6	
6.1.3 チャートの配置を検討する	
6.1.4 色を効果的に使う6	
6.1.5 アプリをシンプルにする6-	
6.1.6 検索機能とブックマークを活用する6-	
6.2 練習問題6-2	
6.2.1 練習問題6-7	
6.3 第6章のまとめと章末付録6-2	29
6.3.1 まとめ6-:	29
6.3.2 章末付録6-2	29
A 付録	
A.1 付録	_1
A.1.1 マッピングテーブルを使用した項目名の変更	
A.1.1 マッピングテーブルを使用した項目者の変更A.A.A.A.A.A.A.A.A.A.A.A.A.A.A.A	
A.1.2 Jータベース中の値でを数に指摘する(Peek 関数)	
A.1.3 Iniline C Autogenerate	
A.1.5 マルチファクトの解決(リンクテーブル)A	
A.1.6 更新や削除がある場合の差分更新(Exists 関数)	

2.1 Qlik Sense の関数

2.1.1 Qlik Sense の関数

Olik Sense には、独自の関数が数多く用意されています。

多くの関数はロードスクリプトとチャートの数式の両方で使用できますが、ロードスクリプトでの み使用できる関数、チャートの数式でのみ使用できる関数もあります。

下図は関数の一例です。例えば、If 関数はロードスクリプトにもチャートの数式にも指定できます。





この章ではよく使用する関数についてご覧いただきます。

その他の関数や、関数の構文の詳細については Qlik Sense のヘルプでご確認いただけます。

スクリプトおよびチャート関数 - Qlik Sense

https://help.qlik.com/ja-

JP/sense/May2024/Subsystems/Hub/Content/Sense_Hub/Scripting/functions-in-scripts-chart-expressions.htm

ヘルプ画面の左側で、ご利用の Olik Sense のバージョンを選択してください。



2.1.2 条件分歧関数 (If 関数)

If 関数は条件式を元に演算を行います。

条件式を指定して、条件式を満たすときの値と満たさなかったときの値を指定します。

If(条件式, 值1 [, 值2])

条件式 条件式を指定します。

値 1 条件式を満たすときの値を指定します。

値 2 条件式を満たさなかったときの値を指定します。

省略すると、条件式を満たさなかった場合は NULL 値が返されます。

例 実績が目標以上だった場合は「達成」、目標未満だった場合は「未達成」と表示します。

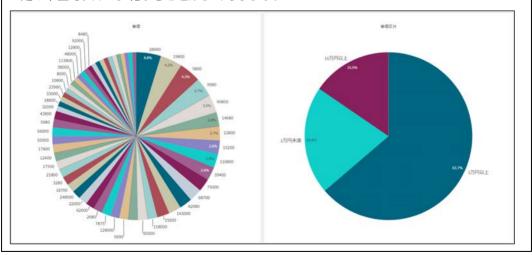
If(実績 >= 目標, '達成', '未達成')

例題1 If 関数

If 関数を使用して、単価の項目をもとに単価区分の項目を作成してみましょう。

下図は単価ごとに納品数量を集計した円グラフです。

単価のような金額のデータを軸にする場合、元のデータをそのまま使用するのではなく、ある 一定の単位でグループ化することがよくあります。

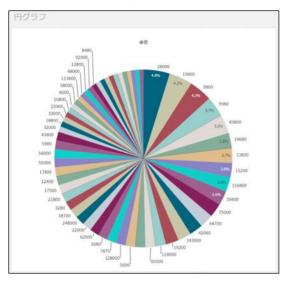


以降の例題で、データロードエディタでロードスクリプトを編集するものは、データロードエディタの「Data」タブのロードスクリプトを編集します。操作時は「Data」タブに切り替えてロードスクリプトの編集を行ってください。

1. o201 を開き、「円グラフ」シートを開きます。

単価を軸にして、納品数量を集計する円グラフがあります。

単価のような金額の項目は値の種類が多く、軸に設定しても傾向を掴むことは難しいです。これをわかりやすくするために、単価を「1万円未満」、「1万円以上」、「10万円以上」の3つの区分でグループ化します。



2. データロードエディタを起動します。

まずは、単価を「10万円以上」と「10万円未満」の2つの条件でグループ化します。 「単価」の末尾に「、」を指定し、商品テーブルの末尾に以下の指定を追加します。

If(単価 >= 100000, '10 万円以上', '10 万円未満') as 単価区分;

追加後、データをロードします。

```
| LIB CONNECT TO 'Access'; | LOAD 商品番号, 商品名, 单值 | If(単価 >= 100000, '10万円以上', '10万円未満') as 単価区分; | Sol. St.Ect | 向品番号, 章品名 | 单值 | FROM 商品番号, 和品数量, 表上実籍番号; | SOL ELECT 商品番号, 納品数量, 表上实籍番号 | FROM 売上明細; | |
```

3. 「テーブル」シートを開きます。シートの編集画面を表示し、キーボードの Shift キーを押しながら、「アセット」パネルから「単価区分」をシートトに追加します。

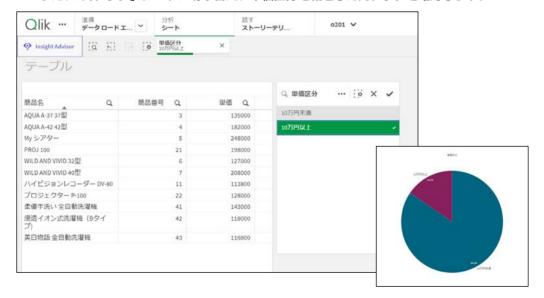


Tip

[アセット] パネルから項目をシート上にドラッグ&ドロップで追加する際、キーボードの Shift キーを押しながら追加すると、フィルターパネルが作成されます。

キーボードの Shift キーを押さずに追加すると、Qlik Sense のチャート推奨の機能により、追加した項目のデータ型などをもとに、さまざまなチャートが自動生成されます。

4. 単価がグループ化されていることを確認します。 単価区分の値を選択し、動作を確認します。 また、「円グラフ」シートに切り替え、単価区分を軸とした円グラフを確認します。



5. 次に、「1万円未満」、「1万円以上」、「10万円以上」という条件でグループ化します。 データロードエディタを起動し、以下の指定に編集します。

```
If(単価 >= 100000, '10万円以上', If(単価 >= 10000, '1万円以上', '1万円未満')) as 単価区分; 追加後、データをロードします。
```

```
LIB CONNECT TO 'Access';

LOAD 商品番号,
商品名,
[F(単価 >= 100000, '10万円以上',
IF(単価 >= 100000, '10万円以上',
'1万円未満')) as 単価区分;

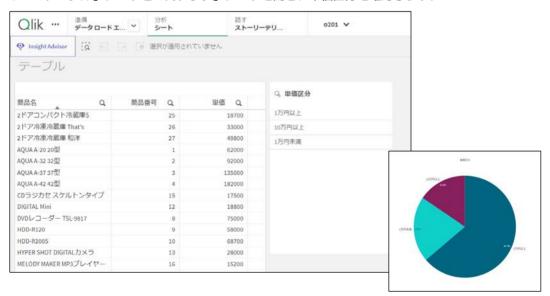
SOL SELECT 商品番号,
商品名,
单価
FROM 商品。

LOAD 商品番号,
和品数量,
表上実績番号;
SOL SELECT 商品番号,
和品数量,
表上実績番号;
FROM 売品;
FROM 売品等号,
和品数量,
表上実績番号;
FROM 売上明細;
```

Tip

この例では見やすいように、If 関数の途中で改行していますが、改行は必須ではありません。

6. 「テーブル」シートと「円グラフ」シートを開き、単価区分を確認します。



Tip

If 関数で作成した値のソートを確認すると、「1 万円以上」、「1 万円未満」、「10 万円以上」という順番でソートされています。これを「1 万円未満」、「1 万円以上」、「10 万円以上」(あるいはその逆の順番)などの任意の順番でソートすることも可能です。詳細は、『2.2.3 デュアル値と Dual 関数』でご紹介します。

5.1 比較分析

5.1.1 比較分析

データを分析する際、複数の値を並べて比較したい場合があります。例えば、今年度と前年度の値 を比較して、前年比を求めたい場合があります。

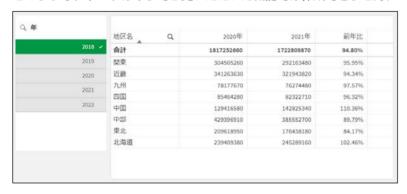
まず、Olik Sense は基本的に1つのアプリ中で、1つの選択状態を持ちます。

例えば、フィルターパネルなどから 2021 年を選択すると、アプリ中の全チャートの値が 2021 年 に絞り込まれます。

ただし、2021年と2020年の値を並べて、比較分析したいという場合があります。

以下のチャートでは、2021年と2020年の売上を並べて表示しています。また、フィルターパネルで選択された2018年という値は無視されています。

このようなチャートは今までご覧いただいた機能では作成できません。



Olik Sense では、比較分析の機能として、SET 分析と並列ステートがあります。

SET 分析

SET 分析特有の式を使用し、数式の中で選択状態を制御する機能です。例えば、値の選択にかかわらず、常に 2022 年の売上の値を表示できます。

並列ステート

1 つのアプリの中に、複数の選択状態を定義する機能です。例えば、あるチャートは東京都の値を表示し、別のチャートは大阪府の値を表示できます。

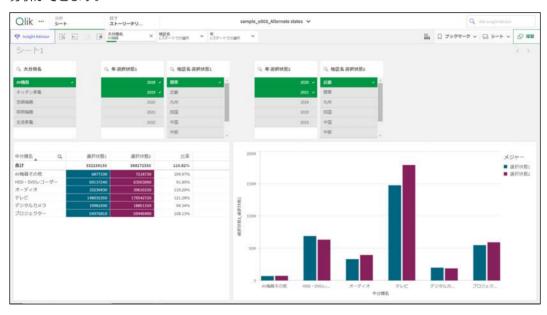
5.3 並列ステート

5.3.1 並列ステート

並列ステートは1つのアプリ中に複数の選択状態を作成する機能です。

アプリ中に選択状態を表す名前を定義し、シートやチャートに選択状態を割り当てていくことができます。

下図のアプリでは、2018 年、2019 年の関東の売上と、2020 年、2021 年の近畿の売上を比較しています。並列ステートを使用すれば、このようにユーザーが選択した任意の選択値をもとに比較分析ができます。





1. o506 を開きます。

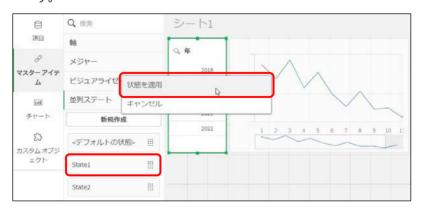
まずは並列ステートを作成します。[アセット] パネルの [マスターアイテム] タブを開きます。「並列ステート」-「新規作成」ボタンをクリックします。[新規状態名] に「State1」と指定して [作成] ボタンをクリックします。



2. 再度[新規作成]ボタンをクリックし、[新規状態名]に「State2」と指定して[作成]ボタンをクリックします。



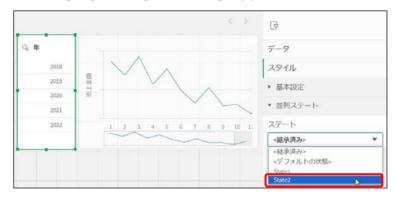
3. State1 を左側の年のフィルターパネルにドラッグ&ドロップし、「状態を適用」を選択します。



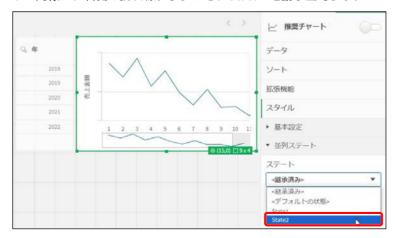
4. 同様に、左側の折れ線グラフにも、State1を割り当てます。



5. [プロパティ] パネルからもステートを割り当てられます。 右側にある年のフィルターパネルの [プロパティ] パネルを開き、[スタイル] - [並列ステート] - [ステート] を「State2」に変更します。



6. 同様に、右側の折れ線グラフにも、State2を割り当てます。



7. フィルターパネルの選択状態が、折れ線グラフに対して個別に割り当てられていることを確認します。



8. 並列ステートを設定した場合の、選択の解除を確認します。 並列ステートを設定した場合、ステートごとの値の選択をクリアできます。また、「すべてク リア」ボタンで、すべてのステートをクリアすることも可能です。

