

Qlikの基礎技術

クリックテック・ジャパン株式会社

Qlik  LEAD WITH DATA™

Copyright 2012 QlikTech Japan KK. All rights reserved.

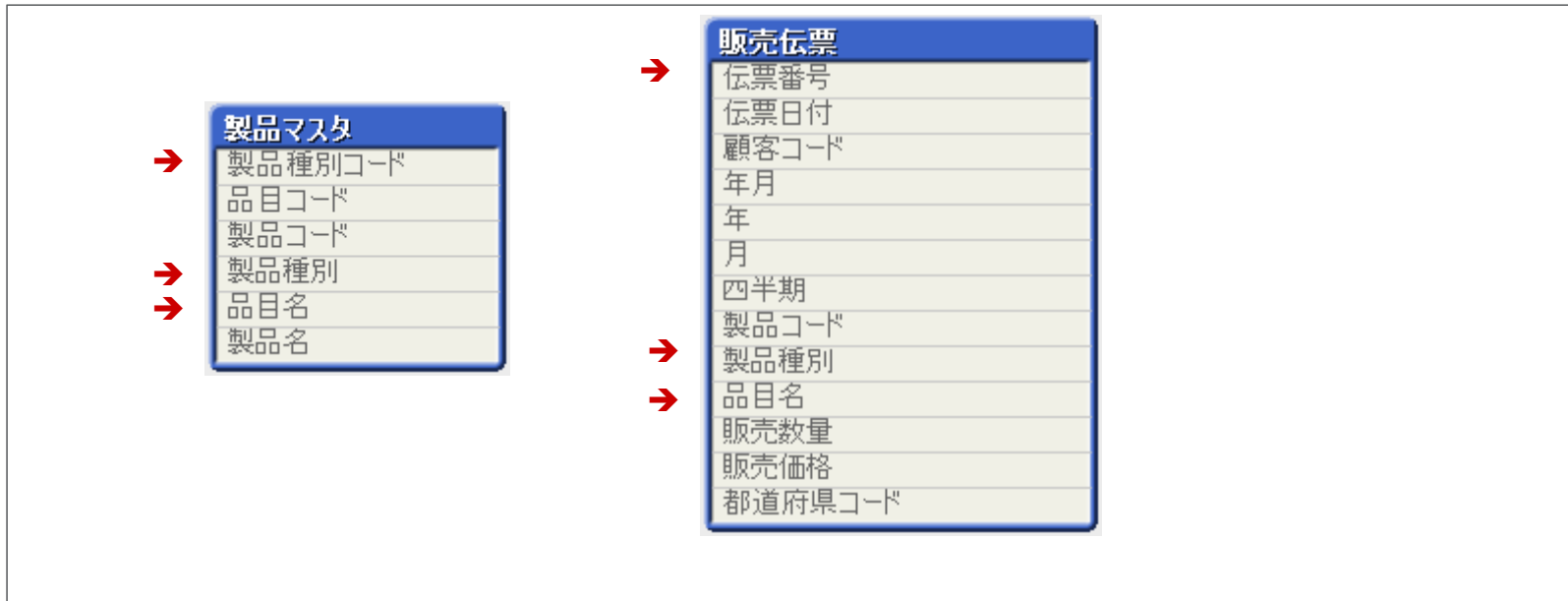


- 合成キー&合成テーブル
- 循環参照
- SET分析とよく使う関数
- QVD

合成キーと合成テーブル

合成キー

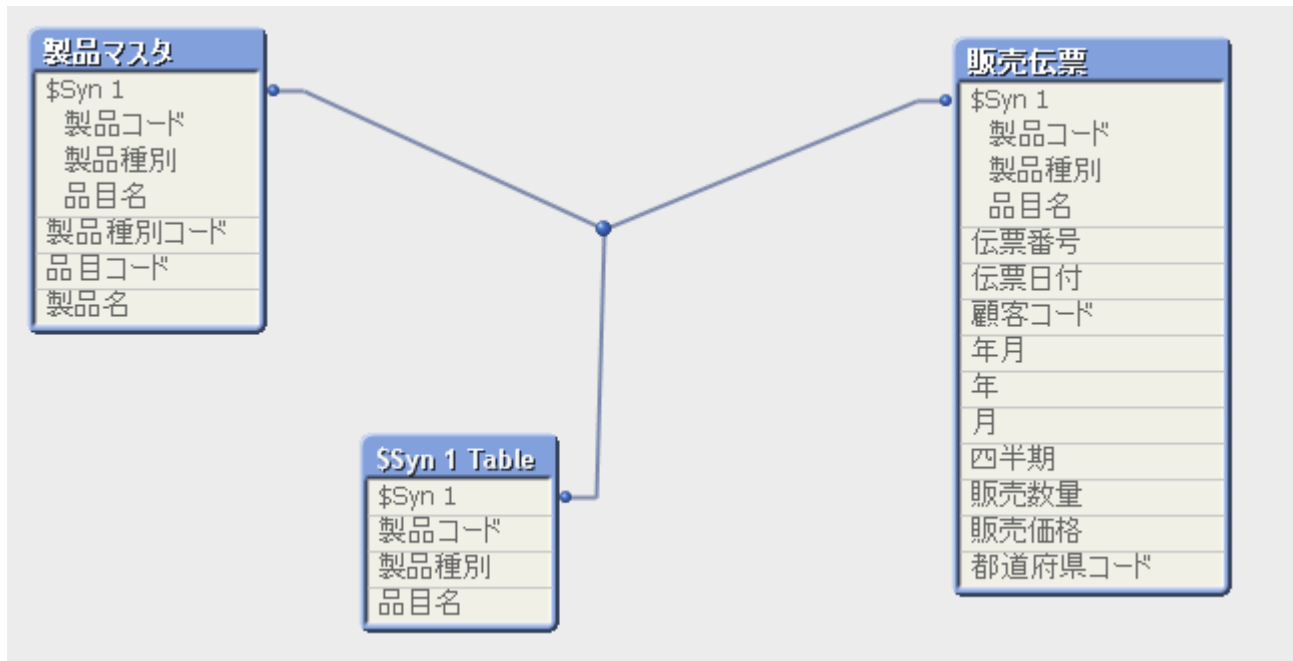
- 各テーブルに共通の項目が2つ以上ある場合



- …ロードすると、次のようになります

合成キー

- Qlik は、自動的に合成キーを生成します



合成キー

Q: 合成キーとは?

A: テーブル間で共通のフィールドのすべての組み合わせを含むフィールド

Q: 合成キーがあると良くないのでしょうか？

A: 極力避けるべきです。Qlikが自動的に作成してしまうため、データモデルの制御が難しくなります。

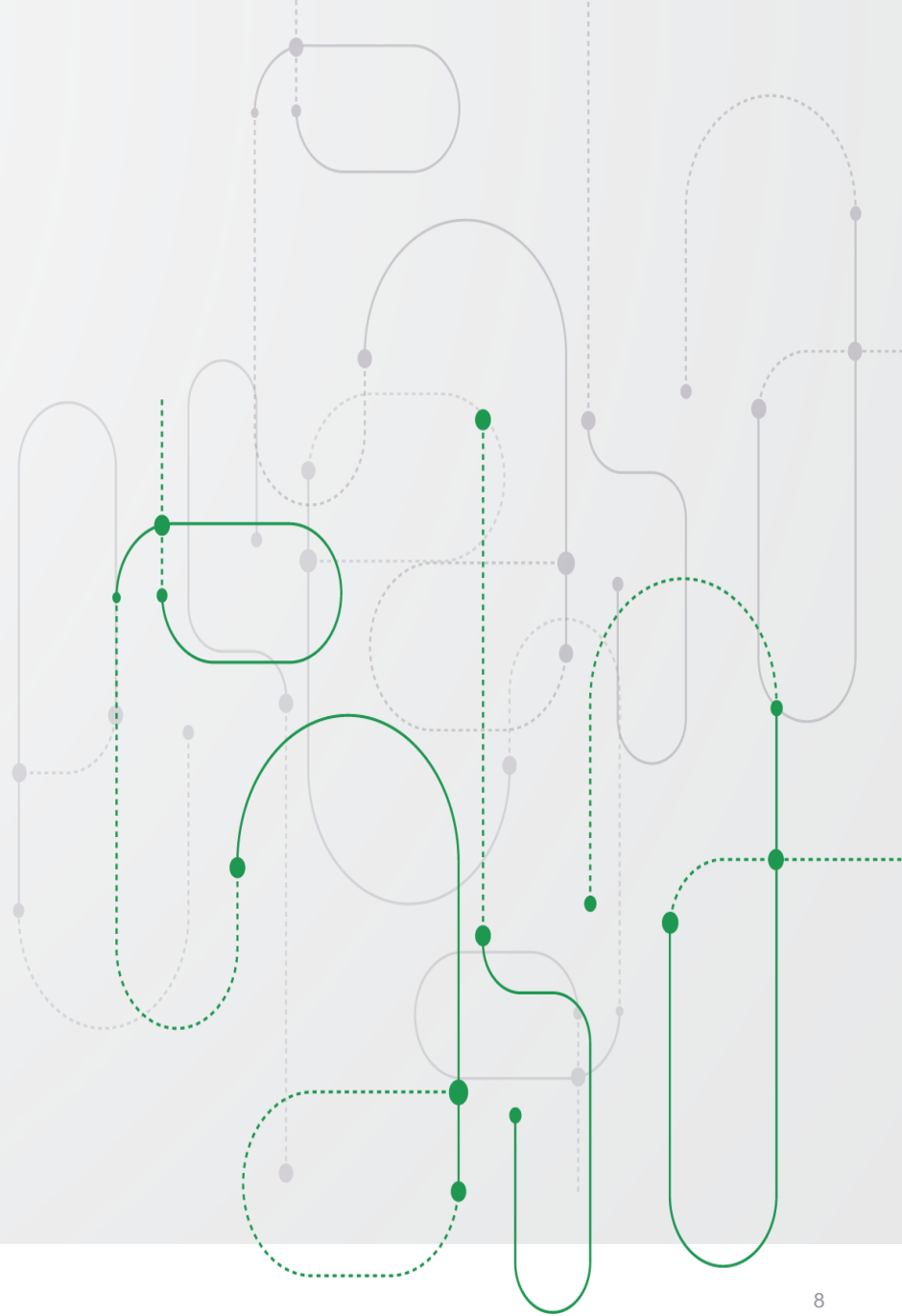
合成キーの解決

- 合成キーをなくすためには何種類の方法があるでしょうか？

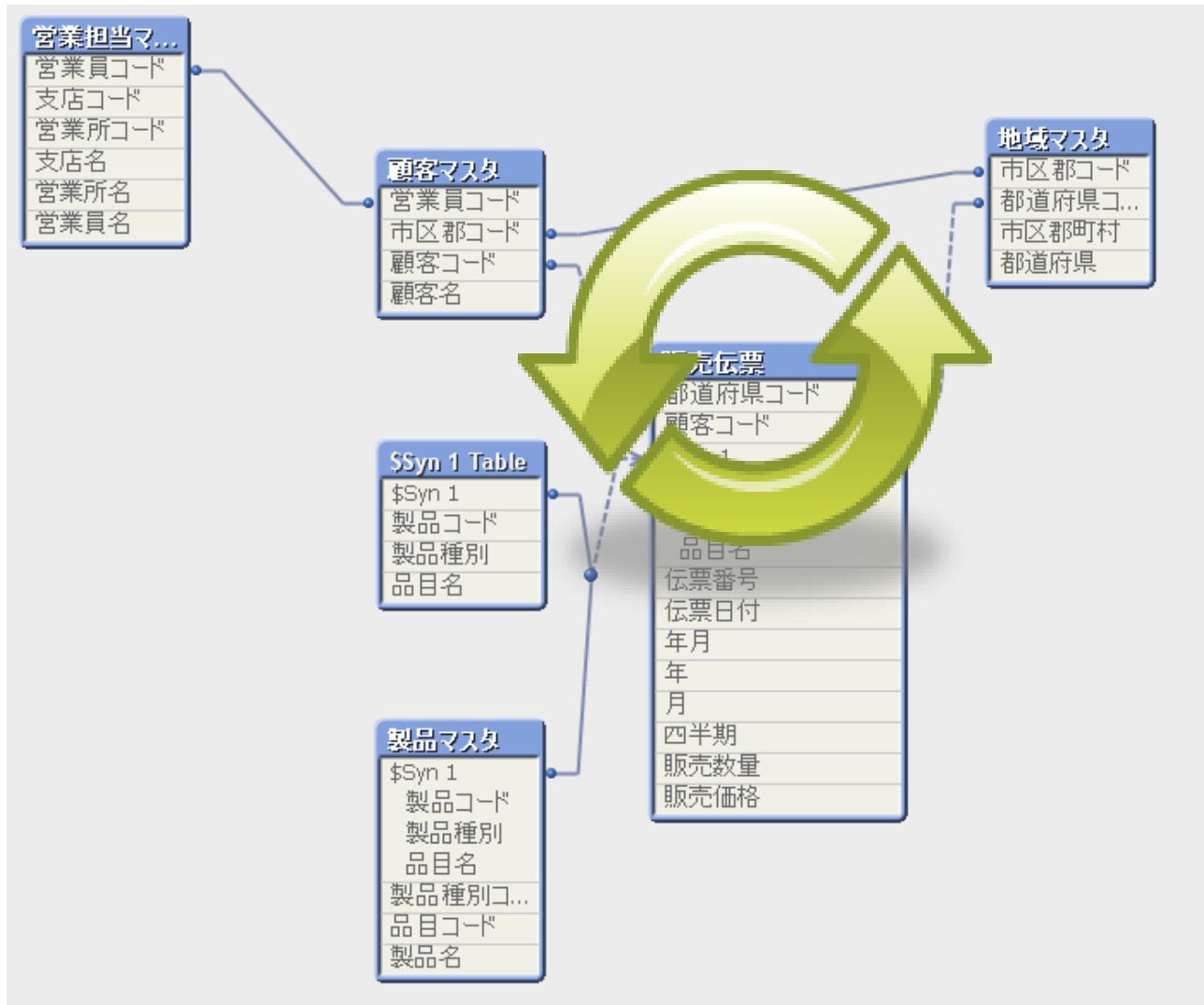
4

1. JOIN
2. キーのConcatenate
3. テーブルのConcatenate
4. リンクテーブル

循環參照



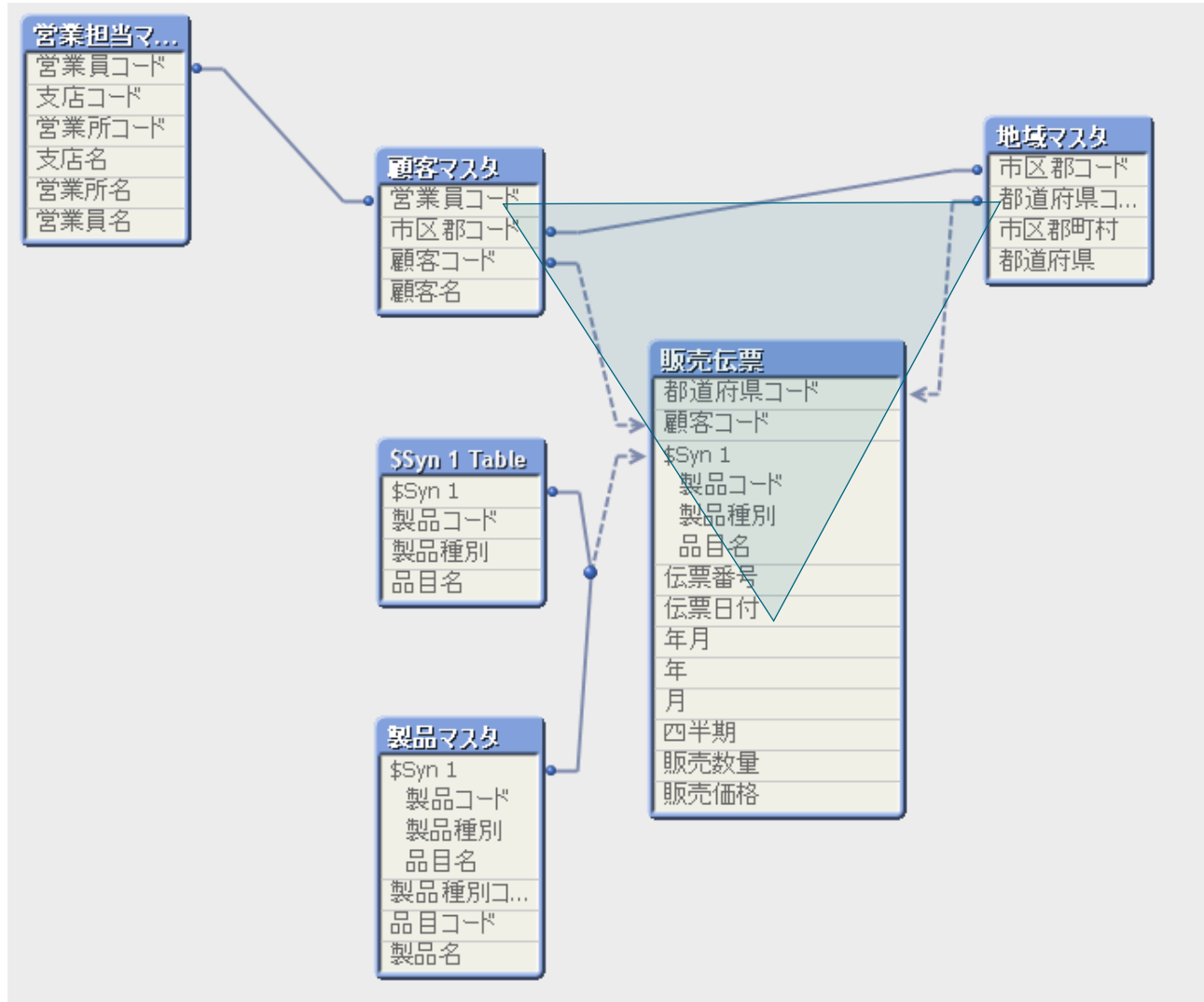
循環参照



循環参照

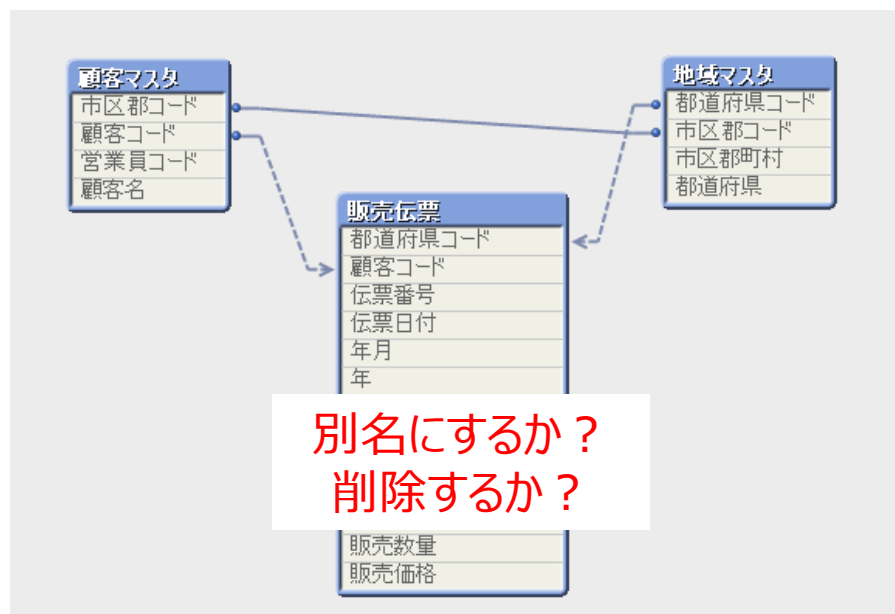
- Qlikファイルの中ではテーブル間の関連性はそれぞれ 1 通りしか持つことが出来ないため、Qlikでは**循環参照**が頻繁に発生します
- 循環参照が発生したら、重複項目等の余分な関連を引き起こす項目が本当に必要かどうか吟味する必要があります。**可能な限り別名とするか、取り除くようにしてください**
- または、**連結 (concatenate)** や**リンクテーブル** を使って循環参照を避けるようにしてください

循環参照の解決

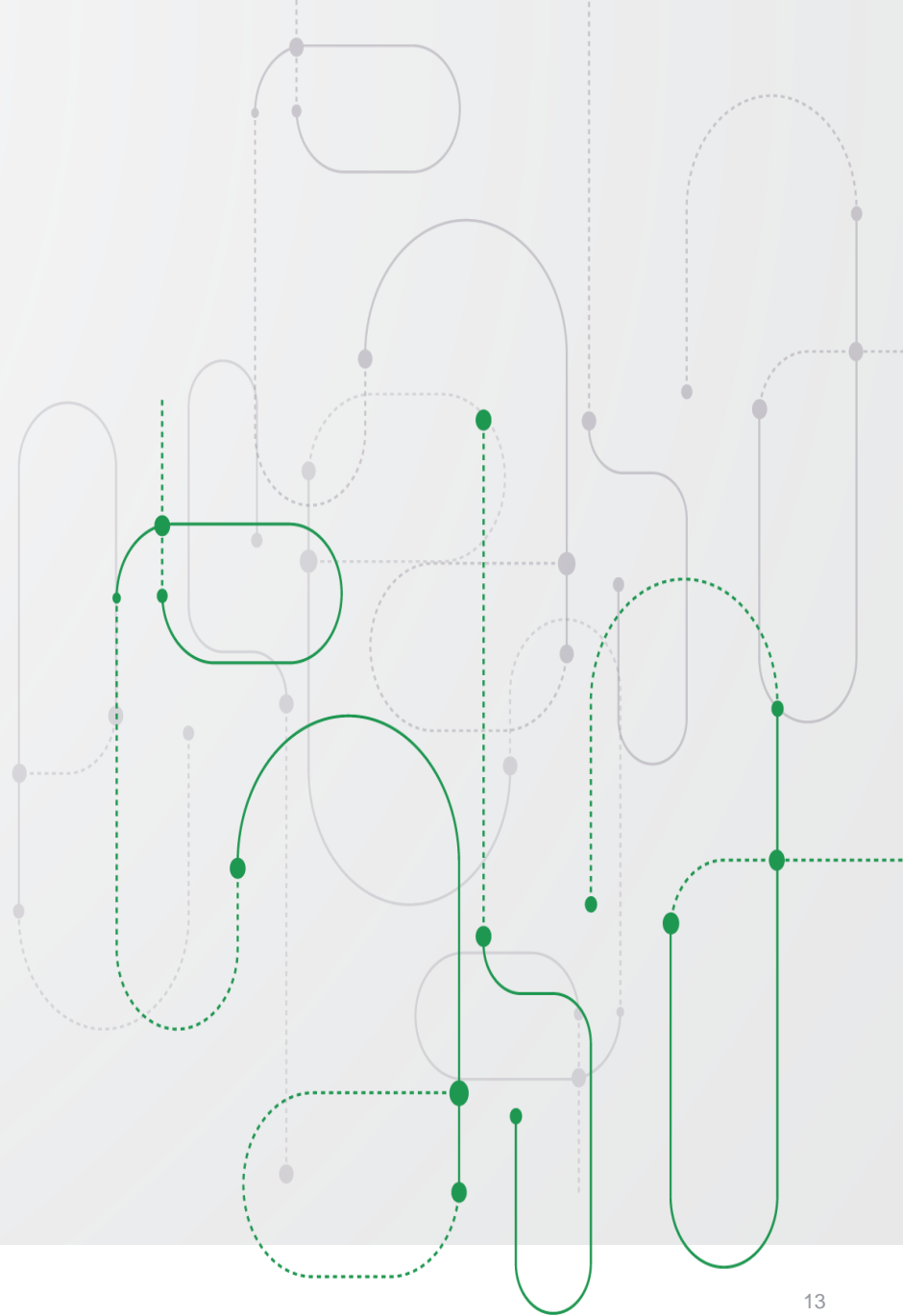


循環参照の解決方法

- ビジネス要件により異なります
- 今回の例は非常にシンプルです
 - 都道府県コードは販売伝票に必要かどうか？



SET分析



SET分析 とは

集計機能は通常、現在の選択を基に集計結果を算出します。

sum(Sales)

SET分析では、別の軸も使用して集計結果を算出します。

sum({ \$<Year={2008}> } Sales)

なぜSET分析を使うのか

- 比較分析を行うのに有効です。
 - 当年 v s 前年
 - 購入品 v s 非購入品
- アプリケーション内の複雑なコード化を避けられます。
- より融通の利く分析ができます。
 - 現在の選択でないところのデータで数式を使用することができます。

SET分析演習 ステップ 1

- リストボックスを追加
 - 年、支店名、営業所名、製品種別
- ストレートテーブルを作成
 - 軸 : 営業所名
 - 数式 : sum(販売価格)
 - 数値書式 : 数値

営業所名	sum(販売価格)
	6,771,702,000
茨城課	195,414,600
横浜課	509,669,700
岡山課	105,478,800
岐阜課	218,930,600
宮城課	112,853,600
京都課	73,169,400
群馬課	204,168,400
広島課	58,737,200
埼玉1課	300,788,100
埼玉2課	135,896,200
札幌課	173,841,800
山形課	34,491,600
四国西課	101,408,600
鹿児島課	40,021,300
新潟課	136,512,300

SET分析演習 ステップ2

- ストレートテーブルに数式を追加
 - 数式 : sum({1} 販売価格)
- リストボックスで色々選択してみる

営業所名	sum(販売価格)	sum({1}販売価格)
	6,771,702,000	6,771,702,000
茨城課	195,414,600	195,414,600
横浜課	509,669,700	509,669,700
岡山課	105,478,800	105,478,800
岐阜課	218,930,600	218,930,600
宮城課	112,853,600	112,853,600
京都課	73,169,400	73,169,400
群馬課	204,168,400	204,168,400
広島課	58,737,200	58,737,200
埼玉1課	300,788,100	300,788,100
埼玉2課	135,896,200	135,896,200
札幌課	173,841,800	173,841,800
山形課	34,491,600	34,491,600
四国西課	101,408,600	101,408,600
鹿児島課	40,021,300	40,021,300
新潟課	136,512,300	136,512,300

SET分析演習 ステップ3

- ストレートテーブルに数式を追加
 - 数式 : `sum({ < 年 = { '2010' } > } 販売価格)`
- リストボックスの年で2011を選択してみる

営業所名	sum(販売価格)	sum({1}販売価格)	sum({<年={'2...
	6,771,702,000	6,771,702,000	2,714,908,700
茨城課	195,414,600	195,414,600	79,340,700
横浜課	509,669,700	509,669,700	203,023,500
岡山課	105,478,800	105,478,800	41,788,000
岐阜課	218,930,600	218,930,600	86,432,400
宮城課	112,853,600	112,853,600	41,342,100
京都課	73,169,400	73,169,400	28,152,900
群馬課	204,168,400	204,168,400	80,748,900
広島課	58,737,200	58,737,200	21,802,200
埼玉1課	300,788,100	300,788,100	125,536,800
埼玉2課	135,896,200	135,896,200	54,847,000
札幌課	173,841,800	173,841,800	68,990,200
山形課	34,491,600	34,491,600	13,990,400
四国西課	101,408,600	101,408,600	44,065,500
鹿児島課	40,021,300	40,021,300	14,929,200
新潟課	136,512,300	136,512,300	51,802,000

SET分析演習 ステップ4

- 3つ目の数式を編集

- 数式 : $\text{sum}(\{ < \text{年} = \{ \$ (= \text{max}(\text{年})) \} > \}$ 販売価格)

- リストボックスの年を選択してみる

営業所名	sum(販売価格)	sum({1}販売価格)	sum({<年={20...
	6,771,702,000	6,771,702,000	4,056,793,300
茨城課	195,414,600	195,414,600	116,073,900
横浜課	509,669,700	509,669,700	306,646,200
岡山課	105,478,800	105,478,800	63,690,800
岐阜課	218,930,600	218,930,600	132,498,200
宮城課	112,853,600	112,853,600	71,511,500
京都課	73,169,400	73,169,400	45,016,500
群馬課	204,168,400	204,168,400	123,419,500
広島課	58,737,200	58,737,200	36,935,000
埼玉1課	300,788,100	300,788,100	175,251,300
埼玉2課	135,896,200	135,896,200	81,049,200
札幌課	173,841,800	173,841,800	104,851,600
山形課	34,491,600	34,491,600	20,501,200
四国西課	101,408,600	101,408,600	57,343,100
鹿児島課	40,021,300	40,021,300	25,092,100
新潟課	136,512,300	136,512,300	84,710,300

SET分析以外で良く使う関数

良く使う関数は・・・

- Qlikで良く使う関数はなんですか？
 - SET分析
 - aggr関数
 - チャートのレコード関数
 - 範囲関数
 - Rank関数

aggr関数

- Aggr関数は指定した軸をグループピングして集計する関数です。

例

Aggr (Sum (Sales) ,年,月)

年、月をグループピングしてSalesの合計を返します。

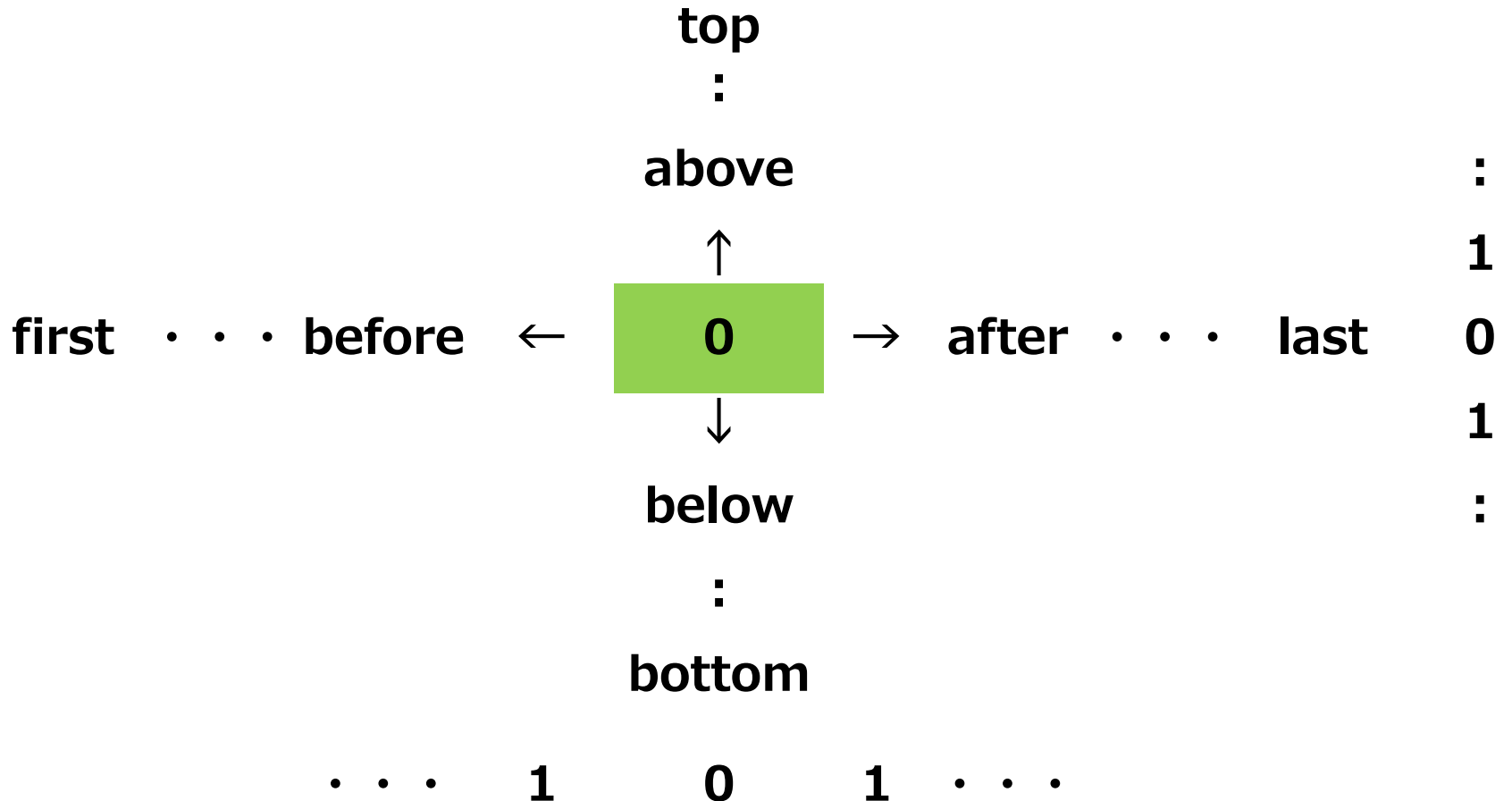
SQLであれば、

```
SELECT Sum (Sales) from Table
```

```
GROUP BY 年 , 月
```

チャートのレコード関数

- チャート内のレコード間での相対的な数値計算に使用します。



範囲関数

- 範囲関数は指定した範囲の集計値を返します。

例

```
rangesum(above(sum(Sales),0,2))
```

	Sales
a	100
b	200
c	300
d	400

rank関数

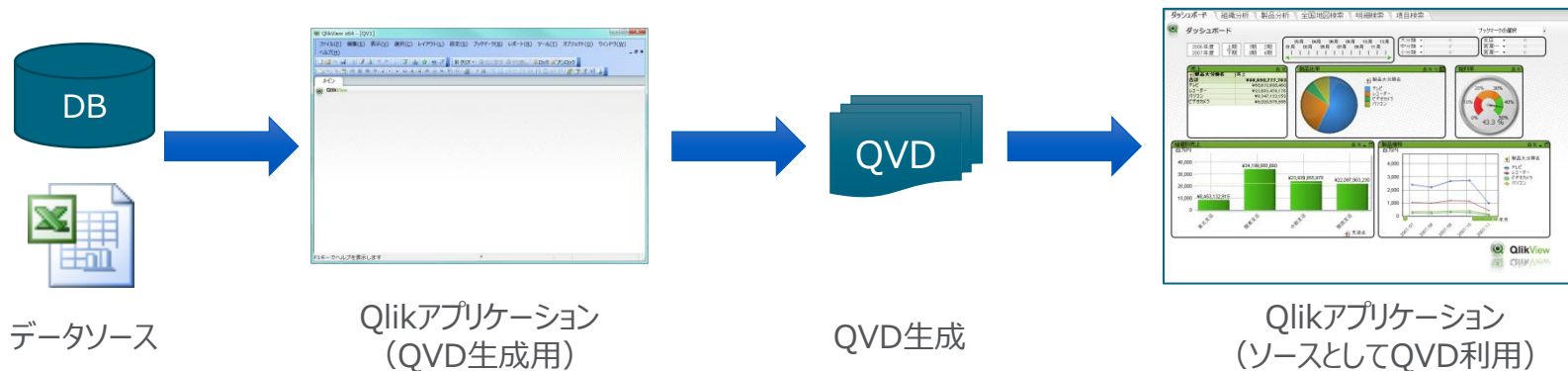
- チャート内の集計値に対して順位付けする関数です。
 - Aggr関数と一緒に使用することが多いです

例
aggr(rank(total sum(販売価格)),営業所名,年)
販売価格の合計に対するランク付けを営業所名、年毎に集計

Qlik におけるQVDファイルの利用

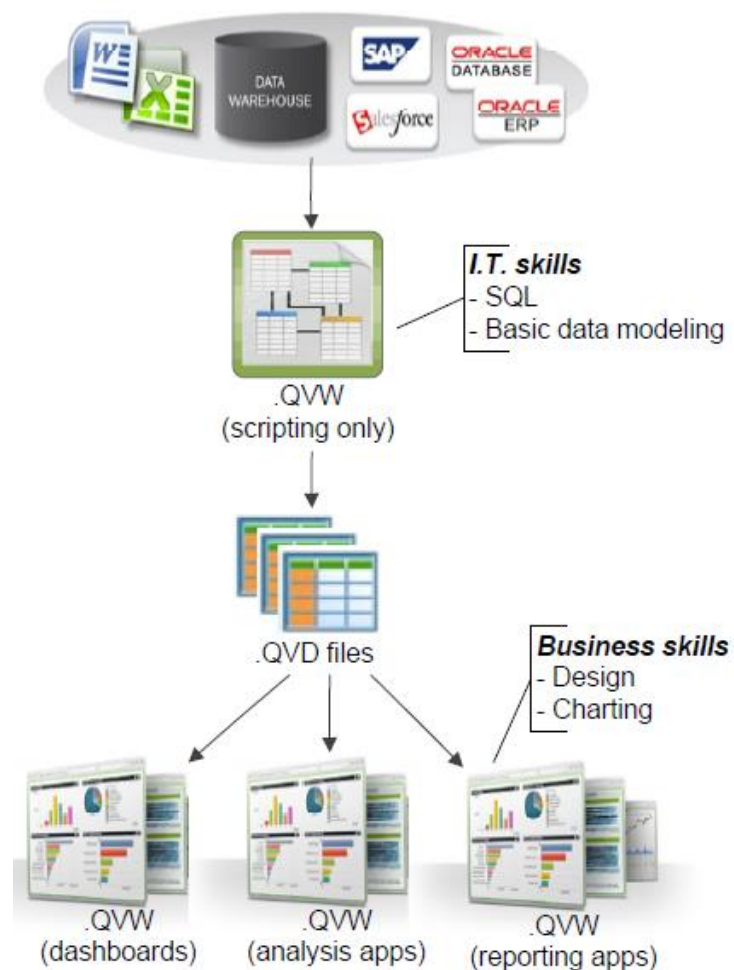
QVDの概要

- QVDファイル = Qlik Data ファイル
 - Qlikから抽出されたテーブルデータを含むファイル
 - 1 QVDファイルにつき1テーブルデータを格納
 - Qlik独自フォーマットであり、Qlikからのみ読み書き可能
- QVDファイルの作成およびロード
 - QVDファイルは、Qlikアプリケーション経由で作成され、最終的に別のQlikアプリケーションのデータソース(の一部)として利用されます



QVDを利用するメリット

- データロードの高速化
- データベースサーバーの負荷軽減
- 複数QVアプリケーションからのデータ統合
- 差分ロード

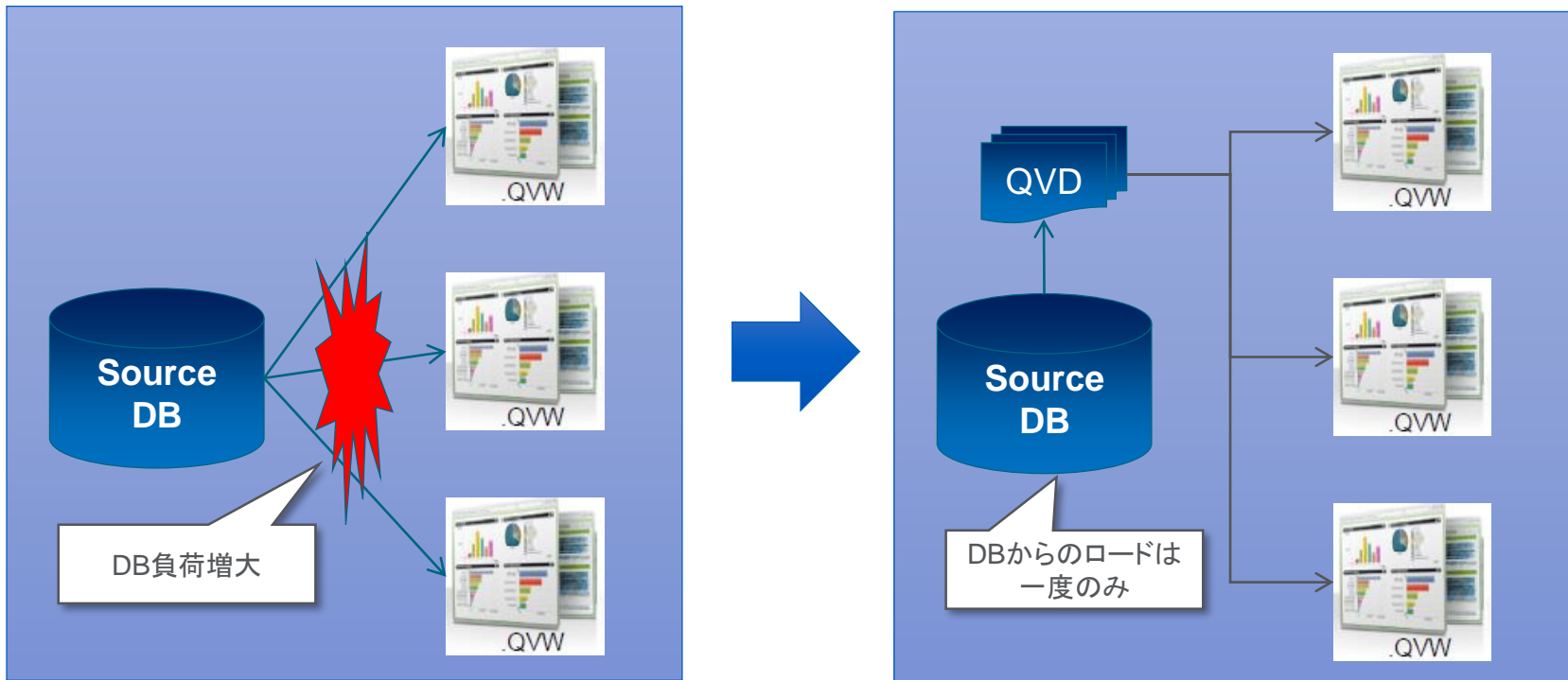


データロードの高速化

- QVDファイルの読み込みは、データベースからの読み込みに比べて高速となります。
(10～100倍)
- 多くの場合、アプリケーションの開発時には、スクリプトを繰り返し実行する必要があります。このような状況で QVD ファイルを使用すると、データセットがあまり大きくない場合でも、繰り返される待機時間を大幅に短縮できます。

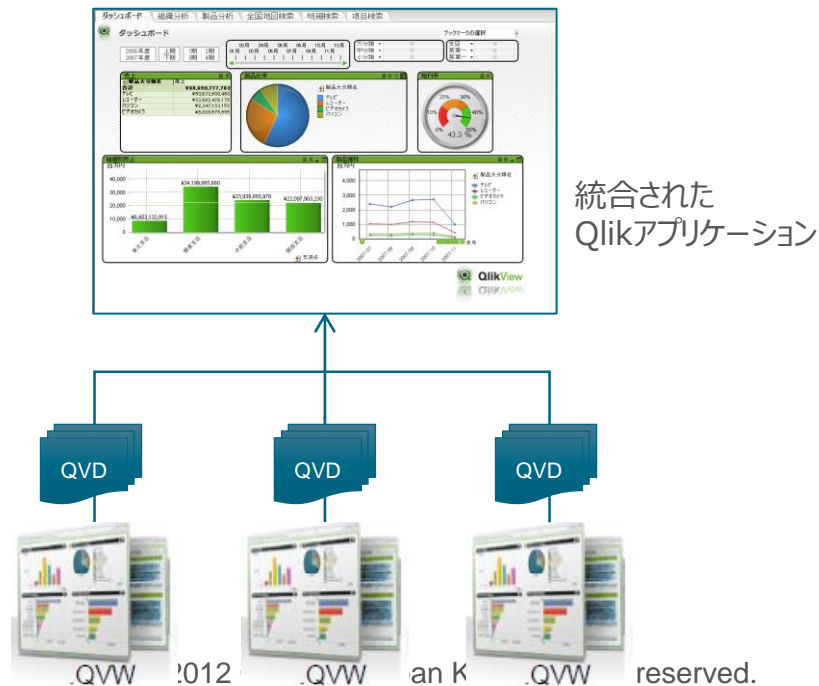
データベースサーバーの負荷軽減

- 複数の Qlik スクリプトが同じデータを共有する場合は、ソースデータベースからデータを一度ロードするだけで済みます。他のアプリケーションは、QVD ファイルからそのデータを使用できます。



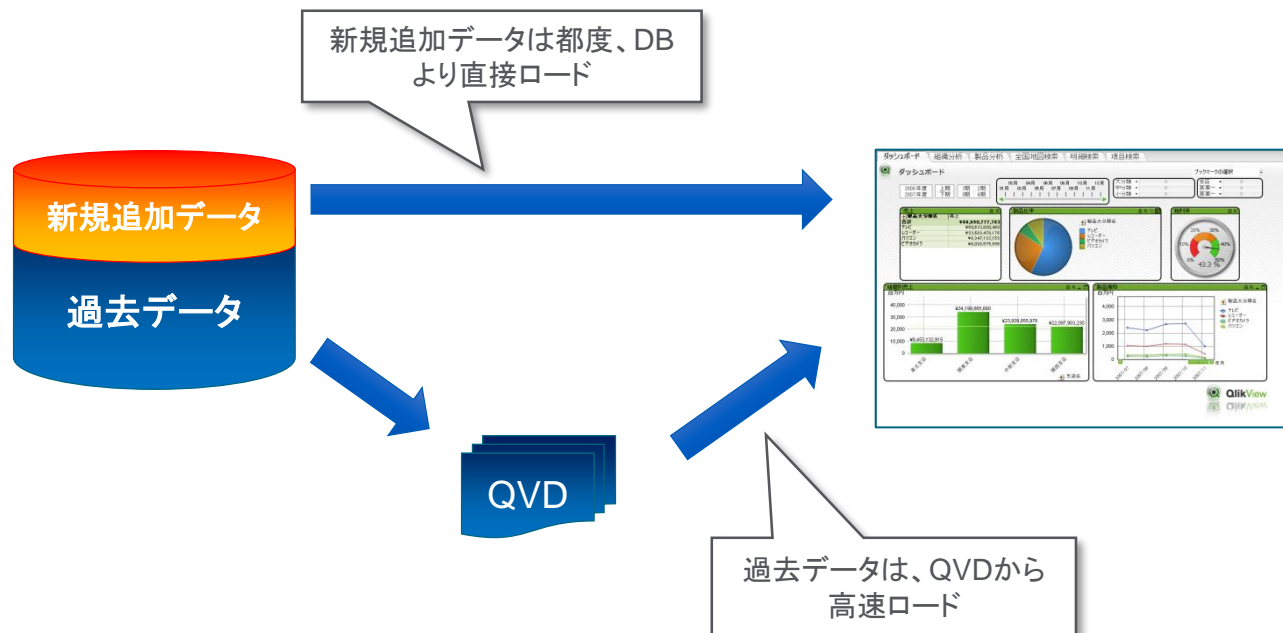
複数Qlikアプリケーションからのデータ統合

- QVWからQVWへのバイナリロードは1 QVWのみですが、QVD ファイルを使用すれば複数Qlik アプリケーションからのデータを統合できます。
- 例えば、各部署で個別に作成したデータを統合することができます。



差分ロード

- QVD ファイルを経由することにより、増大するデータベースから新しいレコードだけをロードする増分ロードを実装できます。



QVDの概要

- QVDファイル生成スクリプトの例

Data:

Load ID,

Description

FROM C:¥Data.xls (biff, embedded labels, table is
[Core Data\$]);

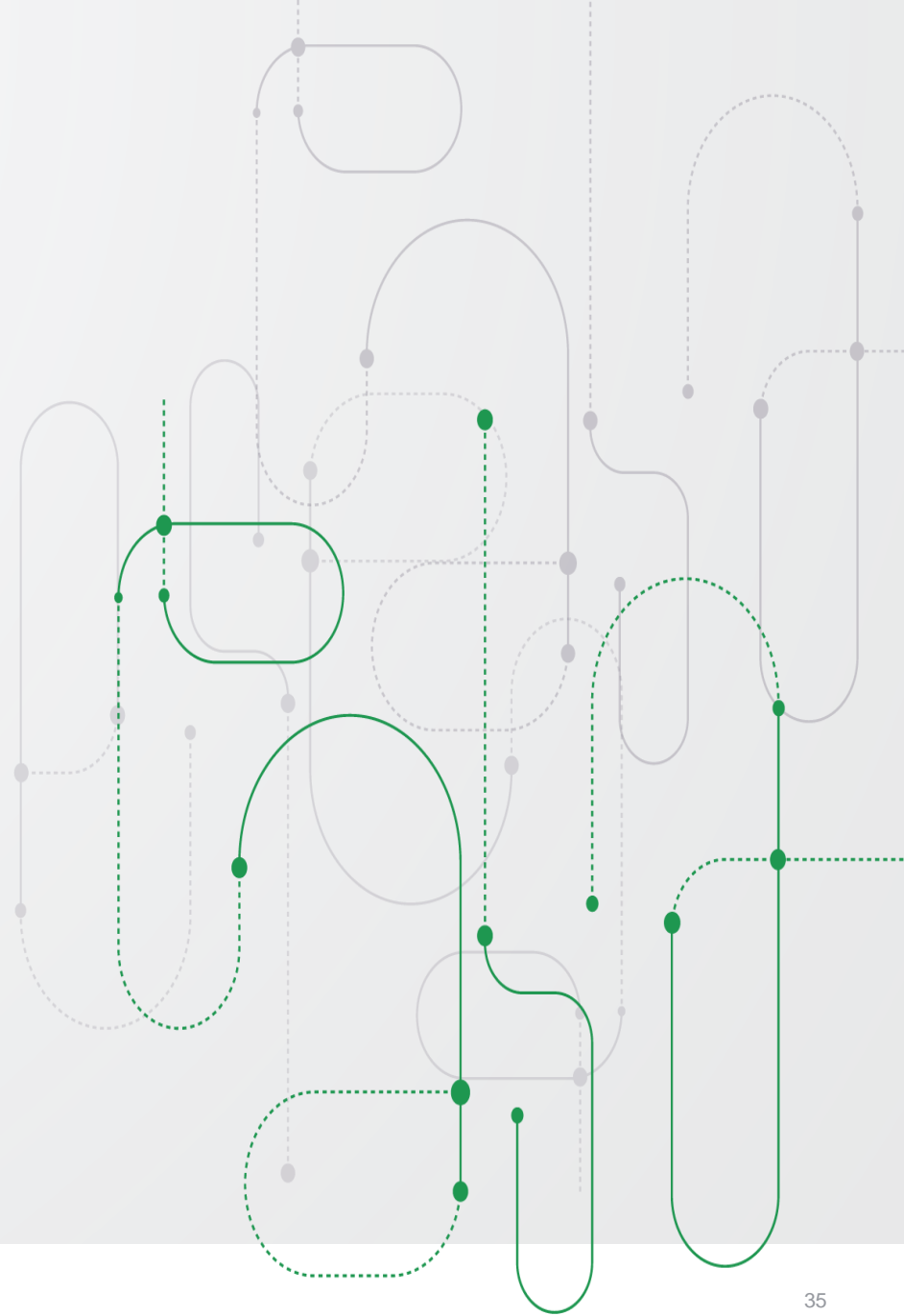
Store Data into C:¥Data.qvd;

Drop Table Data;

Thank you



付録：SET分析追記



SET分析 – どうやって使うのか

- SET分析の構文は3つのカテゴリに分けられます:
 - 識別子
 - 演算子
 - 修飾子
 - 高度なSET分析

識別子

SET分析において、選択状態やブックマークの状態を識別します。

- 1
す。 — 選択を無視して軸毎の集計を実施します。
- \$ — 現在の選択状態
- \$1 — 1つ前の選択状態
- \$_1 — やり直しボタンと同じで1つ次の選択状態
- Bookmark01 — ブックマークのIDでの選択状態

識別子— 例

sum({**1**} Sales)

選択状態を無視してSalesの合計を軸毎に表示します。

sum({**\$**} Sales)

現在の選択におけるSalesの合計を表示します。

sum({**\$1**} Sales)

現在の選択の1つ前の選択におけるSalesの合計を表示します

sum({**Bookmark01** <CategoryName={ 'Babywear' } >} Sales)

Bookmark1の選択および、CategoryNameが「Babywear」のSalesの合計を表示します

演算子

すべての SET 演算子はオペランドとして SET を使用し、結果として SET を返します。

+	Union	2 つの SET 被演算子のいずれかに属するレコードを含む SET を返します
-	除外	2 つの内、最初の SETの被演算子にのみ属する (もう一方には属さない) レコードを含む SET を返します また、単項演算子として使用する場合は、補集合を返します。
*	Intersection	2 つの SET被演算子の両方に属するレコードを含む SET を返します
/	対称差 (XOR)	2 つの SET被演算子のどちらかに属していて、両方に属していないレコードを含む SET を返します

るのは、

予測不可能な結果の原因となりまうので、避けして下さい。

例えば、*Quantity* と *Price* が別々のテーブルの項目の場合に、
`sum({$*BM01}Quantity*Price)` という数式は避けて下さい。

演算子 – 例

`sum({1 - $} Sales)`

現在の選択のすべての除外値による sales を返します。

`sum({$ * Bookmark01} Sales)`

現在の選択とブックマーク BM01 の共有部分における sales を返します。

`sum({ - ($ + Bookmark01)} Sales)`

現在の選択とブックマーク BM01 の除外値における sales を返します。

修飾子

SET は選択の追加や変更で修飾できます。

そういった修飾は、SET 数式に設定します。

修飾子は、

`<Year={2007,+2008},Region={US}>`

のように、項目名と項目の選択を定義する値を含み、“<”と“>”で囲まれます。

項目名と項目値は、

`<[Sales Region]={'West coast', 'South America'}>`

のように通常どおりに囲みます。

修飾子 - 例

sum({\$<OrderDate = DeliveryDate>} Sales)

現在の選択にOrderDate = DeliveryDateの条件式に合致する Salesを返します。

sum({1<Region = {US}>} Sales)

現在の選択を無視して、Region が US の sales を返します。

sum({\$<Region = >} Sales)

Region での選択を除いた現在の選択における sales を返します。

sum({<Region = >} Sales)

すぐ上の例と同じ値を返します。SET 修飾子は、省略されると \$ と見なされます。

sum({\$<Year={2000}, Region={"U*">} Sales)

現在の選択に Year と Region への新しい選択を追加した sales を返します。

高度なSET分析

1. \$記号展開
2. 暗黙的項目値定義を使った SET分析

SET修飾子と\$記号展開

変数と\$記号展開はSET分析で使用することができます。

例:

```
sum( {$<Year = {$(vLastYear)}>} Sales )
```

現在の選択に対する前年の Sales の合計を返します。ここでは、対応する年を含む変数 vLastYear がドル記号展開に使用されています。

```
sum( {$<Year = {$(=Only(Year)-1)}>} Sales )
```

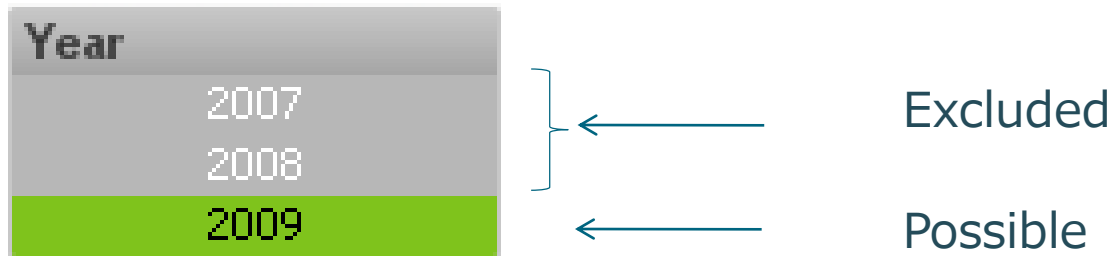
現在の選択に対する前年の Sales の合計を返します。ここでは、ドル記号展開は前年を計算するために使用されています。

暗黙的項目値定義を使ったSET分析

- 全ての項目値が明示的に定義、または検索を介して定義されています。ネストされた SET 定義を使用することによって項目値のセットを定義する方法もあります。
- その場合、関数 $P()$ と $E()$ を使用し、絞込値の要素セットと項目の除外値をそれぞれ表す必要があります。括弧の中には、 $P(\{1\} \text{ Customer})$ のように、ひとつの数式とひとつの項目を指定できます。これらの関数は、他の数式では使用できません。

暗黙的項目値定義を使った SET分析

- $p() = \text{Possible}$
- $e() = \text{Excluded}$



Customer = $p(\{<Year=\{\$(=max(Year)-1)\}>\}$ Customer)
昨年のCustomerを抽出する。

$sum(\{<Year=\{\$(=max(Year))\}\}$,
Customer = $p(\{<Year=\{\$(=max(Year)-1)\}$ Customer $>\}$) Sales)
昨年SalesのあったCustomerの今年Salesを算出する。

暗黙的項目値定義を使った SET分析

```
sum({$<CompanyName = p({$<CategoryName={'Baby Clothes'}>}  
CompanyName)>} Sales)
```

Baby Clothesを購入した会社のSalesを返します。

```
sum({$<CompanyName = p({$<CategoryName={'Baby Clothes'}>}  
CompanyName) - p({$<CategoryName={'SportsWear'}>}  
CompanyName)>} Sales)
```

Baby Clothesを購入した会社でSpotswearを購入していない会社のSalesを返します。